risks

Machine Learning in Insurance

Edited by

Jens Perch Nielsen, Vali Asimit and Ioannis Kyriakou

Printed Edition of the Special Issue Published in *Risks*

www.mdpi.com/journal/risks

MDPI

Machine Learning in Insurance

Machine Learning in Insurance

Special Issue Editors

Jens Perch Nielsen
Vali Asimit
Ioannis Kyriakou

MDPI • Basel • Beijing • Wuhan • Barcelona • Belgrade • Manchester • Tokyo • Cluj • Tianjin

Special Issue Editors

Jens Perch Nielsen
Cass Business School,
City, University of London
UK

Vali Asimit
Cass Business School, City,
University of London
UK

Ioannis Kyriakou
Cass Business School, City,
University of London
UK

Editorial Office
MDPI
St. Alban-Anlage 66
4052 Basel, Switzerland

This is a reprint of articles from the Special Issue published online in the open access journal *Risks* (ISSN 2227-9091) (available at: https://www.mdpi.com/journal/risks/special_issues/Machine_Learning_Insurance).

For citation purposes, cite each article independently as indicated on the article page online and as indicated below:

LastName, A.A.; LastName, B.B.; LastName, C.C. Article Title. *Journal Name* **Year**, *Article Number*, Page Range.

ISBN 978-3-03936-447-3 (Hbk)
ISBN 978-3-03936-448-0 (PDF)

© 2020 by the authors. Articles in this book are Open Access and distributed under the Creative Commons Attribution (CC BY) license, which allows users to download, copy and build upon published articles, as long as the author and publisher are properly credited, which ensures maximum dissemination and a wider impact of our publications.
The book as a whole is distributed by MDPI under the terms and conditions of the Creative Commons license CC BY-NC-ND.

Contents

About the Special Issue Editors

Jens Perch Nielsen is an actuary from Copenhagen, but also a statistician from UC-Berkeley. He worked as an appointed actuary in his youth and led various product development departments, before specializing in research and development. In 1999, he became the research director of RSA, with responsibilities in life, as well as non-life, insurance. From 2006 to 2012, he worked as an entrepreneur, and he is still the co-owner and board member of Copenhagen-based ScienceFirst, London-based Operational Science and Cyprus-based Emergent. He is co-author of more than 100 scientific papers in peer-reviewed journals of actuarial science, economics, econometrics and statistics, and a book on Quantitative Operational Risk Models. He is an Associate Editor for several journals.

Vali Asimit joined Cass Business School in January 2011, as a Lecturer in Actuarial Science. Previously, he was a Lecturer in Actuarial Science at the University of Manchester, for two years. Prof. Asimit studied Economics at the Academy of Economic Studies, Bucharest, Romania. He has an MSc in Statistics from the University of Western Ontario, Canada, where he also pursued his doctoral research on Dependence Modelling with Applications in Finance and Insurance. As part of his academic work, he has published and acted as a referee for international, statistical and actuarial journals. Prof. Asimit received the 2010 Fortis Award for the best Insurance: Mathematics and Economics (IME) journal paper, which was presented at the 14th International Congress of IME.

Ioannis Kyriakou obtained his PhD in Finance from City, following his MSc in Risk and Stochastics from LSE, and his BSc in Actuarial Science from City. He completed his Diploma in Actuarial Techniques at the Institute and Faculty of Actuaries, UK. He works in the area of quantitative methods, on both the development of numerical techniques and applications in the fields of operations research and management science, finance, actuarial science and sector studies, including derivatives, risk management, shipping, commodities, pension product design and communication, stock returns forecasting, and machine learning. He is the Director of the world-renowned Cass MSc in Actuarial Science and MSc in Actuarial Management. Previously, he worked for Lloyd’s Treasury and Investment Management.

risks

Editorial

Special Issue “Machine Learning in Insurance”

Vali Asimit, Ioannis Kyriakou and Jens Perch Nielsen *

Faculty of Actuarial Science and Insurance, Cass Business School, City, University of London, 106 Bunhill Row, London EC1Y 8TZ, UK; alexandru.asimit.1@city.ac.uk (V.A.); ioannis.kyriakou@city.ac.uk (I.K.)
* Correspondence: Jens.Nielsen.1@city.ac.uk; Tel.: +44-(0)20-7040-0990

Received: 2 May 2020; Accepted: 5 May 2020; Published: 25 May 2020

It is our pleasure to prologue the special issue on “Machine Learning in Insurance”, which represents a compilation of ten high-quality articles discussing avant-garde developments or introducing new theoretical or practical advances in this field.

Two articles deal with reserving in non-life insurance. In the first one, Bischofberger (2020) provides an innovative approach to understanding operational time in this context: reverting the time scale enables a very complex correlation structure to be modelled via one-dimensional models only. Validation is performed appropriately based on state-of-the-art machine learning principles. The second paper on reserving by Elpidorou et al. (2019) shows that prior knowledge can be incorporated in the reserving process without violating standard mathematical statistics. The paper does provide a likelihood principle to incorporate prior knowledge.

There are two articles on telematics in insurance by Qazvini (2019) and Pesantez-Narvaez et al. (2019), where the authors present complicated mathematical statistical methodologies. Within the spirit of machine learning, both use model selection and validation to choose the best-predicting model out of a complex array of possibilities. The paper by Bermúdez et al. (2020) also considers claim count models based on new actuarial techniques.

The remaining papers in this collection pertain also to finance. Assa et al. (2019) study deposit insurance pricing, whereas Bärtl and Krummaker (2020) the accurate prediction of export credit insurance claims. With a focus on deriving solvency capital requirements, Krah et al. (2020) analyze adaptive machine learning approaches to proxy modelling of life insurance companies. The paper by Sarabia et al. (2020) revisits the ideas of the so-called semiparametric methods which are very useful when applying machine learning in insurance. For the modelling of prior knowledge, the authors introduce classes of distributions for financial data. They then illustrate the proposed procedures with data on stock returns. Finally, Mammen et al. (2019) apply machine learning to forecast the conditional variance of long-term stock returns measured in excess of different benchmarks, considering the short and long-term interest rate, the earnings-by-price ratio, and the inflation rate.

We are indebted to all the reviewers who collaborated and thankful to all the authors for their contributions. It is our hope that the research articles that were assembled for this Special Issue will cast light on the field and prove a fruitful reading for our audience.

References

Assa, Hirbod, Mostafa Pouralizadeh, and Abdolrahim Badamchizadeh. 2019. Sound deposit insurance pricing using a machine learning approach. *Risks* 7: 45. [CrossRef]

Bärtl, Mathias, and Simone Krummaker. 2020. Prediction of claims in export credit finance: A comparison of four machine learning techniques. *Risks* 8: 22. [CrossRef]

Bermúdez, Lluís, Dimitris Karlis, and Isabel Morillo. 2020. Modelling unobserved heterogeneity in claim counts using finite mixture models. *Risks* 8: 10. [CrossRef]

Bischofberger, Stephan M. 2020. In-sample hazard forecasting based on survival models with operational time. *Risks* 8: 3. [CrossRef]

Elpidorou, Valandis, Carolin Margraf, María Dolores Martínez-Miranda, and Bent Nielsen. 2019. A likelihood approach to Bornhuetter–Ferguson analysis. *Risks* 7: 119. [CrossRef]

Krah, Anne-Sophie, Zoran Nikolić, and Ralf Korn. 2020. Machine learning in least-squares Monte Carlo proxy modeling of life insurance companies. *Risks* 8: 21. [CrossRef]

Mammen, Enno, Jens Perch Nielsen, Michael Scholz, and Stefan Sperlich. 2019. Conditional variance forecasts for long-term stock returns. *Risks* 7: 113. [CrossRef]

Pesantez-Narvaez, Jessica, Montserrat Guillen, and Manuela Alcañiz. 2019. Predicting motor insurance claims using telematics data—XGBoost versus logistic regression. *Risks* 7: 70. [CrossRef]

Qazvini, Marjan. 2019. On the validation of claims with excess zeros in liability insurance: A comparative study. *Risks* 7: 71. [CrossRef]

Sarabia, José María, Faustino Prieto, Vanesa Jordá, and Stefan Sperlich. 2020. A note on combining machine learning with statistical modeling for financial data analysis. *Risks* 8: 32. [CrossRef]

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).

risks

MDPI

Article

Sound Deposit Insurance Pricing Using a Machine Learning Approach

Hirbod Assa [1], Mostafa Pouralizadeh [2,*] and Abdolrahim Badamchizadeh [2]

[1] Mathematical Sciences Building, University of Liverpool, Liverpool L69 7ZL, UK; assa@liverpool.ac.uk

[2] Department of Statistics, Faculty of Mathematical Science and Computer, Allameh Tabataba'i University, Tehran 1489684511, Iran; badamchi@atu.ac.ir

* Correspondence: m.pouralizadeh@gmail.com

Received: 30 December 2018; Accepted: 13 April 2019; Published: 19 April 2019

Abstract: While the main conceptual issue related to deposit insurances is the moral hazard risk, the main technical issue is inaccurate calibration of the implied volatility. This issue can raise the risk of generating an arbitrage. In this paper, first, we discuss that by imposing the no-moral-hazard risk, the removal of arbitrage is equivalent to removing the static arbitrage. Then, we propose a simple quadratic model to parameterize implied volatility and remove the static arbitrage. The process of removing the static risk is as follows: Using a machine learning approach with a regularized cost function, we update the parameters in such a way that butterfly arbitrage is ruled out and also implementing a calibration method, we make some conditions on the parameters of each time slice to rule out calendar spread arbitrage. Therefore, eliminating the effects of both butterfly and calendar spread arbitrage make the implied volatility surface free of static arbitrage.

Keywords: deposit insurance; implied volatility; static arbitrage; parameterization; machine learning; calibration

1. Introduction

Banks can lend or invest most of their money deposits. However, if bank's borrowers default, the bank's creditors, particularly depositors, risk loss. In order to protect depositors from this risk, policy makers have promoted deposit insurance schemes that are majorly issued by government run institutions. In the global scale, International Association of Deposit Insurers (IADI) was formed in 2002 "to enhance the effectiveness of deposit insurance systems by promoting guidance and international cooperation". Even though experiences from bank runs during the 1929 Great Depression led to the introduction of the first deposit insurances in the US, they have been identified as one of the contributors to the 2008 financial crisis. The major issue due to these type of insurances is that they encourage the risk of moral hazard. While this problem has been studied to some extent in the literature (see Assa (2015) and Assa and Okhrati (2018)), there is another issue relevant to the incorrect contract design and miss-pricing which needs further attention. More precisely, in addition to the moral hazard risk, arbitrage also needs to be removed in designing a sound deposit insurance. In this paper, we first show that the removal of the arbitrage for the policies with no risk of moral hazard is tantamount to the removal of static arbitrage. This fact lead us to naturally use machine learning methods to improve the precision of estimation for implied volatility.

As it is discussed in Assa and Okhrati (2018), in a very general framework a sound deposit insurance that rules out the risk of moral hazard is a two layer policy. A two layer policy can be considered as the subtract of two European options. This helps us to use the financial engineering formalism on derivative pricing in our setting. There are some existing models for predicting the price of an option, most of which spin around the Black-Scholes model. The Black-Scholes formula is one of the most famous and frequently used methods of option pricing. However, it is derived under some constraining assumptions including variability due to the randomness of the underlying Brownian motion, no transaction costs, and fixed volatility and interest rate (Black and Scholes (1973)). In the Black-Scholes formula, all parameters are given in the market except the the stock price volatility. However, this parameter can be estimated by the past stock price data; it usually gives different Black-Scholes option prices than the market option prices because the assumption of fixed volatility does not hold in real markets. To overcome this drawback, option traders use implied volatility to adapt the market prices for options with the Black-Sholes formula. In fact, they consider an option price in terms of the Black-Sholes implied volatility.

Volatility is a measure of the variability of returns for a given security and it can be measured by the standard deviation of returns for a particular period of time usually for one year. However, implied volatility is the estimated volatility of a security's price and it can be obtained by options trading prices based on the Black-Scholes framework. While historical volatility has only some information about underlying price fluctuation for a period of time in the past, implied volatility contains more information about option price future behavior.

The market volatility can be considered as a proxy of the bank portfolio riskiness, as proved in Zhang (2015). Volatility modeling proven to be a challenging task and there are only a few popular models for stochastic implied volatility. For instance, one can consider the stochastic alpha, beta, rho (SABR) parameterization Avellaneda (2005), Vana-Volga (VV) model Castagno (2007), a parametric model of implied volatility Zhao (2013) and Stochastic Volatility Inspired (SVI) of Gatheral (2014). Furthermore, some other studies like Malliaris (1996), Cont (2002), Alentorn (2004) and Roux (2007) tried to parameterize implied volatility using neural network, regression and other machine learning tools. However, none of these models could eliminate arbitrage opportunity.

In this study, a machine learning approach is proposed to model implied volatility and also to remove static arbitrage. Since the price of a European call option depends on the price movement of the underlying asset, we implement a quadratic machine learning approach to parametrize total implied variance for the European Black-Scholes call options with less than one year to maturity. That, how much the model is qualified to fit the implied volatility data, is verified both theoretically and empirically. We also use a regularized cost function for each volatility slice to rule out both underfitting and overfitting Hastie (2002). The main observation of this study is to explore how a regularized cost function can help eliminate static arbitrage, whereas this idea has not been successfully studied in the literature.

This paper is organized as follows: In Section 2, first we design a risk management framework, then provide some basic materials of implied volatility, static arbitrage and machine learning which are necessary for the rest of the paper. We propose a quadratic model for implied volatility and then some necessary conditions are provided on the parameters of the model to get rid of static arbitrage in Section 3. In Section 4, we implement a numerical example to illustrate the validity of the proposed model. Eventually, the paper is finished by a suggestion for future possible works in Section 5.

2. Sound Deposit Insurance

In Assa and Okhrati (2018), a deposit insurance where the risk of moral hazard is ruled out is discussed. In their paper they have shown a sound insurance contract in many cases, including when using VaR and CVaR to model the risk aversion behavior of the investors, has a two layer structure. As we want to address another caveat, that is to rule out the arbitrage, in a similar setting we use their framework. Adopting notations in Assa and Okhrati (2018), let $(\Omega, \Im, F = (\Im_t)_{0 \leq t \leq T}, \mathbb{P})$ be a completed probability space, where Ω is the set of all scenarios, $\mathbb{P}$ is the physical probability measure and $(\Im_t)_{0 \leq t \leq T}$ is a filtration with usual conditions and $\Im = \Im_T$ is a σ-field of measurable subsets of Ω. Furthermore, $\mathbb{E}$ denotes the mathematical expectation with respect to $\mathbb{P}$. Policies are issued at $t = 0$, and liabilities are settled at $t = T$. Random variables represent losses for different scenarios at time T. The cumulative distribution function associated with a random variable X is denoted by F_X. The market risk free interest rate is a non-negative number $r \geq 0$. Let us consider a bank with an initial capital[1] $\exp(-rT)\, b$, and a non-negative loss variable associated with the deposit insurance denoted by $\mathcal{L} \geq 0$. The bank wants to hedge its global position by transferring part of its losses to another party (usually an insurance company). The insurance policy is denoted by a non-negative random variable I and it has to satisfy $0 \leq I \leq \mathcal{L}$. The price of the policy is given by a premium function $\pi : \mathcal{D} \to \mathbb{R}$ at time 0, where $\mathcal{D}$ is the domain of π. Therefore, the bank's position is composed of four parts:

1. The initial capital at time 0 i.e., $\exp(-rT)\, b$;
2. The global loss, $\mathcal{L}$;
3. The insurance policy, $-I$;
4. The premium payed for the insurance policies, at time T, $\exp(rT)\,\pi(I)$.

Therefore, the total loss is

$$\text{Total loss} = \exp(rT)\,\pi(I) + \mathcal{L} - b - I.$$

The bank wants its global position to be solvent. We use a risk measure to measure the solvency; particularly in this paper we consider Value at Risk (VaR) or Conditional Value at Risk (CVaR) recommended in the Basel II accord for the banking system (also in the Solvency II for the insurance industry). In this paper, ϱ denotes the risk measure recommended by regulator. The bank is solvent if its capital b is adequate for the solvency i.e., $\varrho(\exp(rT)\,\pi(I) + \mathcal{L} - b - I) \leq 0$. Then, an optimal decision for the bank is to buy the cheapest insurance contract i.e.,

$$\begin{cases} \min \pi(I) \\ \varrho(\exp(rT)\,\pi(I) + \mathcal{L} - b - I) \leq 0 \\ 0 \leq I \leq \mathcal{L} \end{cases} \tag{1}$$

Now, we move one step forward to use a more specific model for the bank's asset. We use an approach similar to Merton (1997), by considering that the bank's asset follows a geometric Brownian motion. This choice is very crucial, since one can use the risk neutral valuation in order to find the "market (consistent) value" of an insurance contract which is a necessary practice by Solvency II. Denoting the underlying by S_t, we assume it follows the following stochastic differential equation:

$$\begin{cases} dS_t = \mu S_t dt + \sigma S_t dW_t \\ S_0 > 0 \end{cases}$$

Here W_t, μ and σ are respectively a standard Wiener process, drift, and volatility (constant numbers). It is also known that:

$$S_t = S_0 \exp\left(\left(\mu - \frac{\sigma^2}{2}\right) t + \sigma W_t\right)$$

We assume that the bank's loss is a non-negative and non-increasing function of its assets value. In mathematical terms, $\mathcal{L} = L(S_T)$, where $L : \mathbb{R} \to \mathbb{R}_+ \cup \{0\}$ is a non-increasing function:

$$L(x) = \begin{cases} \exp(rT)\, S_0 - x & \text{if } x \leq \exp(rT)\, S_0 \\ 0 & \text{if } x > \exp(rT)\, S_0 \end{cases} \tag{2}$$

It is clear that L is equal to $(\exp(rT)\, S_0 - x)_+$.

In Assa and Okhrati (2018) it is assumed that there is no risk of moral hazard, meaning that both bank and insurance feel risk of an adverse event. For that, Assa and Okhrati (2018) assume that

[1] For technical reasons we assume the value of b at time T and discount it to make it comparable to today's value.

both the bank and insurance loss variables are non-decreasing functions of the global loss variable. This assumption rules out the risk of moral hazard, as both sides have to feel any increase in the global loss (see for example Heimer (1989) and Bernard and Tian (2009)) Therefore, we assume that $I = f(\mathcal{L})$ where both f and $\mathrm{id} - f$ are non-negative and non-decreasing functions (here id denotes the identity function).

Using the no-moral-hazard assumption, Assa and Okhrati (2018) have managed to find the sound deposit insurances where the risk of insolvency is measured by a distortion risk measure. However, in this paper we only restrain ourselves to the one mentioned by regulator (and also the most popular ones), VaR and CVaR:

$$\mathrm{VaR}_\alpha(X) = \inf\{x \in \mathbb{R} | P(X > x) \le 1-\alpha\}, \alpha \in [0,1],$$

and

$$\mathrm{CVaR}_\alpha(X) = \frac{1}{1-\alpha}\int_\alpha^1 \mathrm{VaR}_t(X)dt. \tag{3}$$

For these particular risk measures, Assa and Okhrati (2018) have shown that the contract has a two-layer structure. By combining Corollary 1, Theorem 3 and Theorem 4 in Assa and Okhrati (2018) we get the following theorem:

Theorem 1. *If $\varrho = \mathrm{VaR}_\alpha$ or $\varrho = \mathrm{CVaR}_\alpha$, and $\mu - r \ge 0$ hold, then the optimal deposit insurance is a two layer policy on loss $\mathcal{L}$ i.e.,*

$$I = f(\mathcal{L}),$$

where f is defined as

$$f(x) = \begin{cases} 0 & \text{if } x \le l \\ x - l & \text{if } l \le x \le u \\ u - l & \text{if } u \le x \end{cases}, \tag{4}$$

for upper and lower retention levels u and l, respectively.

Now it is important to observe that such a contract can be written as the difference of two call option policies. To see this we have to take the following steps:

$$f \circ L(x) = \begin{cases} 0 & \text{if } L(x) \le l \\ L(x) - l & \text{if } l \le L(x) \le u \\ u - l & \text{if } u \le L(x) \end{cases}.$$

First, observe that if $\exp(rT) \le l$ then $L(x) = (\exp(rT)S_0 - x)_+ \le l$ always holds and as a result $I = 0$. Otherwise, if $\exp(rT) > l$, then $L(x) = (\exp(rT)S_0 - x)_+ \le l$ is equivalent to $\exp(rT)S_0 - l \le x$. On the other hand, $u \le \mathcal{L} = (\exp(rT)S_0 - x)_+$ is always equivalent to $x \le \exp(rT)S_0 - u$. So we have the following policies:

1. If $\exp(rT) \le l$ then $I = 0$
2. If $\exp(rT) > l$

$$f \circ L(x) = \begin{cases} 0 & \text{if } \exp(rT)S_0 - l \le x \\ \exp(rT)S_0 - x - l & \text{if } \exp(rT)S_0 - u \le x \le \exp(rT)S_0 - l \\ u - l & \text{if } x \le \exp(rT)S_0 - u \end{cases}.$$

or

$$f \circ L(x) = (x - \exp(rT)S_0 + l)_+ - (x - \exp(rT)S_0 + u)_+ + u - l.$$

This indicates that I can be written as the difference of two call options

$$I = \left(S_T - \exp\left(-rT\right)S_0 + l\right)_+ - \left(S_T - \exp\left(-rT\right)S_0 + u\right)_+ + u - l. \tag{5}$$

Now, we want to introduce the risk premium. An important implication of what we have done above is that all insurance contracts are in the form of a contingent claim i.e., for $f \in \mathbb{C}$, $f\left(\mathcal{L}\right) = f\left(L\left(S_T\right)\right) = \left(f \circ L\right)\left(S_T\right)$. To find the market value of a contingent claim we use the no-arbitrage valuation, so we have:

$$\pi\left(I\right) = \exp\left(-rT\right)\mathbb{E}\left(\frac{d\mathbb{Q}}{d\mathbb{P}}I\right) = \exp\left(-rT\right)\mathbb{E}^*\left(I\right),$$

where $\frac{d\mathbb{Q}}{d\mathbb{P}}$ is the Radon-Nikodym derivative of the risk neutral probability measure $\mathbb{Q}$ with respect to $\mathbb{P}$ and $E^*$is the expectation with respect to this measure. However, as we have seen in (5), this contract can be written as the difference of two call options plus a constant value. So we can then use the following valuation of the contract in our setup

$$\begin{aligned}\pi\left(I\right) &= e^{-rT}E^*\left(I\right)\\ &= C_{BS}\left(S_0, \exp\left(rT\right)S_0 - l, T, \sigma, r\right) - C_{BS}\left(S_0, \exp\left(rT\right)S_0 - u, T, \sigma, r\right)\\ &\qquad + \exp\left(-rT\right)\left(u - l\right),\end{aligned} \tag{6}$$

where in general $C_{BS}\left(S_0, K, \tau, \sigma, r\right)$ denotes the value of a call option with maturity τ, strike price K, volatility σ, interest rate r and initial underlying value S_0, in a Black-Scholes model. So we have the following corollary:

Corollary 1. *If $\varrho = \mathrm{VaR}_\alpha$ or $\varrho = \mathrm{CVaR}_\alpha$, and $\mu - r \geq 0$ hold, then the optimal deposit insurance is the difference of two call options plus a constant value. As a result, for a no-arbitrage valuation, the no-arbitrage assumption needs only to hold for the call options.*

2.1. Black-Scholes Model

The price of a European style call option Black and Scholes (1973) is calculated as follows:

$$\begin{aligned}C_{BS}\left(S_0, K, \tau, \sigma, r\right) &= \exp\left(-r\tau\right)\ E\left(S_T - K\right)_+\\ &= S_0N(d_1) - \exp\left(-r\tau\right)KN(d_2)\end{aligned} \tag{7}$$

$$d_1 = \frac{\ln\left(\frac{S_0}{K}\right) + \left(r + \frac{\sigma^2}{2}\right)\tau}{\sigma\sqrt{\tau}}, \quad d_2 = d_1 - \sigma\sqrt{\tau}$$

where S_0 denotes the risky asset price at time 0, K is the exercise price, τ is the time to expiration, σ is the standard deviation of the security's return, N is the distribution function for the standard normal distribution, and r is the rate of interest.

2.2. Implied Volatility

The implied volatility of a risky asset S is the unique value of σ_{imp} that solves the following equation

$$C = C_{BS}\left(\tau, K, \tau\sigma_{imp}^2, S, r, t\right) \tag{8}$$

where C is the market price for the call option written at time t with strike price K and T is the expiration time.

Another version of implied volatility is calculated by the underlying price process being replaced by the forward price in the Black-Scholes model. This version of implied volatility has some nice properties that facilitate application of mathematical techniques. The Black formula is as follows:

$$C_B\left(\tau, K, \tau\sigma^2_{imp}, S, r, t\right) = F_{[t,t+\tau]}\, N\left(d_1\right) - KN\left(d_2\right) \tag{9}$$

$$d_1 = \frac{\log\left(\frac{F_{[t,t+\tau]}}{k}\right) + \frac{1}{2}\tau\sigma^2_{imp}}{\sqrt{\tau\sigma^2_{imp}}} \quad , \quad d_2 = \frac{\log\left(\frac{F_{[t,t+\tau]}}{k}\right) - \frac{1}{2}\tau\sigma^2_{imp}}{\sqrt{\tau\sigma^2_{imp}}}$$

where $F_{[t,t+\tau]} = \exp\left(-r\tau\right) S_t$ is the forward price.

2.3. Static Arbitrage

Now, we provide mathematical definition Roper (2009) of static arbitrage and then present an equivalent definition which connects it to the two other types of arbitrage called calendar spread and butterfly.

Definition 1. *A surface of call option C is said to be free of static arbitrage if there exists a non-negative martingale X on* $(\Omega,\ \Im,\ \mathcal{F} = (\Im_t)_{t\geq 0},\ \mathbb{P})$ *which the call price formula can be reached by*

$$C(K,\tau) = E\left((X_\tau - k)_+\right) \quad , \quad \forall (k,\tau) \in [0,\infty) \times [0,\infty) \tag{10}$$

In other words, there exists a non-negative martingale which is associated with the security price process in distribution, in fact both the security price and the equivalent martingale follow the same probabilistic rules. The next two theorems by Kellerer (1972) provide some conditions on call surface and some equivalent conditions on volatility surfaces to make them free of static arbitrage.

Theorem 2. *A call option surface written on underlying S, with expiration time T*

$$C : (0,\infty) \times R \rightarrow (0,\infty)$$

$$(\tau,\ k) \rightarrow E\left((S_T - k)_+\right)$$

is said to be free from static arbitrage if the following conditions are satisfied:

1. $\partial_\tau C > 0$
2. $\lim\limits_{k\rightarrow\infty} C(\tau,\ k) = 0$
3. $\lim\limits_{k\rightarrow-\infty} C(\tau,\ k) + k = a\ ,\ a \in R$
4. $C(\tau,\ k)$ *is convex in* k
5. $C(\tau,\ k) \geq 0$

Theorem 3. *On the surface of total implied variance* $w_{imp} = \tau\sigma^2_{imp}$ *where*

$$w_{imp} : (0,\infty) \times R \rightarrow (0,\infty)\,,$$

$$(\tau,\ K) \rightarrow w_{imp}\left(\tau,\ K\right),$$

The conditions in Theorem 2 are derived by the following arguments

1. $\partial_\tau w_{imp} > 0;$
2. $\lim\limits_{k\rightarrow\infty} d_1(k) = -\infty;$
3. $\tau\sigma_{imp} \geq 0;$
4. $\left(1 - \frac{x}{2w_{imp}}\partial_x(w_{imp})\right)^2 - \frac{1}{4}\left(\frac{1}{w_{imp}} - \frac{1}{4}\right)\left(\partial_x(w_{imp})\right)^2 + \frac{1}{2}\partial_{xx}(w_{imp}) \geq 0.$

The first condition in Theorem 3 which implies the first one in Theorem 2 means that total implied variance is increasing with respect to time to maturity. Moreover, if this condition holds, there is no calendar spread arbitrage Fengler (2009), otherwise the opportunity of calendar spread arbitrage emerges in the market, so one can do a risk-free trading strategy at a given moment. As a matter of fact, the existence of calendar spread arbitrage addresses a trader to buy a nearby option and sell the farther in the case of the large time spread between the two options and sell the nearby and buy the farther if the spread is narrow Carr and Madan (2005). Conditions 2 and 3 in Theorem 3 imply condition 2 of Theorem 2 which reveals that the price of an option for large exercise prices, tends to zero. The third argument in Theorem 2 is derived by conditions 2, 3 and 4 in Theorem 3. Finally, the inequality 4, known as Durrleman's condition Durrleman (2003), is a part of the second derivative of call surface with respect to strike price.

Conditions 2 and 4 in Theorem 3 provide a volatility surface free of butterfly arbitrage. For example, let C_1 and C_2 are two call options with expiration time T and exercise prices K_i that $K_1 < K_2$, and suppose an option with the same maturity time T and the strike price K, where $K_1 < K < K_2$, exists in the market. If the call surface is non-convex with respect to exercise price, there is an opportunity to sell two options at the middle strike price K and buy one at the strike price K_1 and one at the strike price K_2 and by this strategy a trader can gain a risk-free profit. So, condition 4 of Theorem 3 assigns a non-negative value for the second derivative of a call surface to get rid of butterfly arbitrage.

Now it is time to provide another definition for a volatility call surface Gatheral (2011) to make it free of static arbitrage based on materials related to both types of arbitrage, calendar spread and butterfly.

Definition 2. *There is no static arbitrage on a volatility surface if and only if*

1. *It is free of calendar spread arbitrage;*
2. *The volatility slice is free of butterfly arbitrage for any fixed time to maturity.*

Particularly, no butterfly arbitrage is equivalent to the existence of a positive probability density Breeden and Breeden and Litzenberger (1978), and no calendar spread arbitrage implies that the option price is increasing with respect to time to expiration.

2.4. Parameterization of the Implied Volatility

For a fixed time to expiration, the SVI model Gatheral (2004) is given by

$$w_{imp}^{SVI}(x) = a + b(\rho(x-m) + \sqrt{(x-m)^2 + \sigma^2}) \tag{11}$$

$$a \in \mathbb{R} \;\;,\;\; b \geq 0 \;\;,\;\; |\rho| < 1 \;\;,\;\; m \in \mathbb{R} \;\;,\;\; \sigma > 0 \;\;,\;\; x = \log \frac{K}{F_{[t,\, t+\tau]}}$$

in this parametrization, x is moneyness, $w_{imp}^{SVI}(x) = \tau\sigma_{imp}^2$ is total implied variance and $\{a, b, \sigma, \rho, m\}$ is the set of parameters that are supposed to be estimated. The behavior of volatility smile is highly affected by variations in these five parameters; moreover, the reason to use total implied variance instead of implied volatility is that in Equation (9) the volatility parameter σ is always accompanied with a $\sqrt{\tau}$ Zhu (2013).

2.5. Machine Learning Approach

Machine learning is a branch of artificial intelligence (AI) that has many applications used to model the behavior of natural phenomena and predict their future outcomes. The basic intuition behind this methodology is that there is a training set that consists of empirical data $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})$, where m is the number of training examples; moreover, a learning algorithm (learning hypothesis) fits the data to determine how to learn from the training set and how well the result can be generalized to the unseen data. The vector of parameters θ is reached by the following strategy:

$$\hat{\theta} = \arg\min_{\theta} J(\theta) = \arg\min_{\theta} \frac{1}{2m} \sum_{i=1}^{m} V\left(h_\theta(x^{(i)}), y^{(i)}\right) \tag{12}$$

V is the cost of predicting $y^{(i)}$ based on hypothesis $h_\theta(x^{(i)})$ for the *i*-th training example. The cost V for the *i*-th training example is a function of the difference between the target value $y^{(i)}$ and the estimated values $h_\theta(x^{(i)})$. Usually this function is considered to be L-1 norm or L-2 norm loss function that the L-1 norm is absolute difference and the L-2 norm is the square difference. A learning hypothesis is a predetermined function, usually chosen by experts, that is considered to fit the data to describe its behavior inside and outside the training set.

However, sometimes choosing an adequate learning algorithm which best describes the trend of data outside the training set is the area of difficulty and a wrong learning algorithm takes a lot of time investigating without coming up to a real conclusion. So, we should know what is the best promising avenue to spend time pursuing. If our selected hypothesis does an excellent job predicting y from x for observations in the training set but not for those outside the training set, we face overfitting, on the other hand, if the hypothesis does not do well, predicting y in both the training set and outside the training set, we encounter underfitting. Most of the time the algorithm is faced with overfitting since a learning algorithm usually does a good job for data that builds the model and the problem is how well it fits to the unseen data. Conquering these obstacles, we add a regularization term to the cost function and estimate parameters as follows:

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2m} \left(V\left(h_\theta\left(x^{(i)}\right), y^{(i)}\right) + \lambda R\left(h_\theta\left(x^{(i)}\right)\right)\right) \tag{13}$$

The penalty term is used when there is model complexity, in other words, as long as the algorithm encounters underfitting or overfitting the penalty term keeps the parameters small to preclude these types of complexity. To give a break down explanation of regularization, the parameter λ is called the regularization parameter assigned to control the trade-off between underfitting and overfitting. R is the regularization function which provides a penalty for the hypothesis complexity to impose some certain restrictions on parameters space. Furthermore, the regularization function improves the hypothesis to generalize well to the data beyond the training set Nilsson (2005).

There are some methods to debug a learning algorithm to rule out underfitting and overfitting. To fix overfitting, we can get more training examples try smaller sets of features and try increasing λ; moreover, to rule out underfitting, some adjustments like getting additional features, adding polynomial features, and trying to decrease λ are helpful according to Hastie (2002).

3. The Quadratic Parametrization

Different types of quadratic models have been proposed for implied volatility parameterization in recent years, but none of them are qualified enough to be free of static arbitrage. For instance, Avellaneda (2005) proposed a quadratic model to parameterize implied volatility, however, as mentioned in Roper (2010), this model does not guarantee the Durrleman's function to be everywhere non-negative around ATM, so the absence of butterfly arbitrage is not satisfied. There are some other types of quadratic models, like Roux (2007), but there is no condition on the parameters to remove static arbitrage, hence it is seemingly impossible to be encountered with this inadequacy in the area of quadratic parametrization of implied volatility. Now, we introduce our proposed quadratic model to parameterize implied volatility for call options with less than one year time to expiration, then provide some special conditions on the model parameters, we preclude static arbitrage.

3.1. The Raw Quadratic Model

The quadratic parameterization of total implied variance with respect to moneyness x is given by:

$$w_{imp}^{Q2}(x,\,\eta) = \theta_0 + \theta_1 x + \theta_2 x^2 \tag{14}$$

where $\theta_0 > 0,\ \theta_1 \in \mathbb{R}$. The condition of $\theta_2 > 0$ along with the condition of $\theta_1^2 - 4\theta_0\theta_2 < 0$ make the function $x \ \rightarrow\ w_{imp}^{Q2}\,(x,\,\eta)$ positive and strictly convex for all $x \in \mathbb{R}$.

3.2. Elimination of Static Arbitrage

In this section, we present some conditions on the parameters of the quadratic model (14) to make it free of static arbitrage. However, since (14) is a model with fixed time to maturity, we introduce an equivalent parameterization for implied variance with respect to ATM variance, ATM volatility skew and the lower bound of variance. Then, we make some conditions on the parameters of the equivalent model to guarantee the absence of calendar spread arbitrage. These parameters are more familiar for market traders than the raw parameters in (14) since they reveal some characteristics of market data which are known for investors. The idea begins with the following definition.

Definition 3. *For a fixed time to maturity and a parameter set $\chi = \{v_\tau, \psi_\tau, \mu_\tau\}$, the equivalent quadratic parameterization of implied variance is*

$$\sigma_{imp}^2 = v_\tau + (2\sqrt{v_\tau}\psi_\tau)\,x + \left(\frac{v_\tau\psi^2}{v_\tau - \mu_\tau}\right)x^2 \tag{15}$$

$$v_\tau > 0\ \ ,\ \ \psi_\tau \in \mathbb{R}\ \ ,\ \ \mu_\tau > 0,$$

where v_τ is ATM variance, ψ_τ is ATM volatility skew, and μ_τ is the minimum level of variance. Therefore, this is a calibration to three given quantities which are more understandable for market traders than the raw parameters. For a fixed time to maturity, the following relations hold between the raw parameters and the equivalent quadratic parameters:

$$v_\tau = \frac{\theta_0}{\tau} \quad , \quad \psi_\tau = \frac{1}{\sqrt{\tau}}\frac{\theta_1}{2\sqrt{\theta_0}} \quad , \quad \mu_\tau = \frac{1}{\tau}\left(\theta_0 - \frac{\theta_1^2}{4\theta_2}\right)$$

Proposition 1. *The equivalent parameterization of implied variance is not affected by calendar spread arbitrage if the following arguments are held*

1. $\psi_\tau\,(\partial_\tau\psi_\tau) > 0$
2. $\partial_\tau\left[\ln\left(\frac{v_\tau}{v_\tau-\mu_\tau}\right)\right] > \frac{(v_\tau-\mu_\tau)}{4v_\tau^3}$
3. $\partial_\tau\,[\ln\psi_\tau] < \frac{2v_\tau}{v_\tau-\mu_\tau} - \frac{1}{v_\tau}$

Proof. We are supposed to show that the following expression, which is the first derivative of the surface with respect to time to maturity, always takes positive values

$$\begin{aligned}\partial_\tau\sigma_{imp}^2 =&\partial_\tau v_\tau + 2\left\{\frac{\psi_\tau}{2\sqrt{v_\tau}} + \sqrt{v_\tau}(\partial_\tau\psi_\tau)\right\}x \\ &+\left\{\frac{\{2\psi_\tau(\partial_\tau\psi_\tau)v_\tau + \psi_\tau^2(\partial_\tau v_\tau)\}\,(v_\tau-\mu_\tau) - \{v_\tau\psi_\tau^2\partial_\tau(v_\tau-\mu_\tau)\}}{(v_\tau-\mu_\tau)^2}\right\}x^2.\end{aligned}$$

Since this is a quadratic function of x, we just need to show that the coefficient of the highest degree is positive and the discriminant is negative. So, doing some rearrangement of the numerator of the coefficient in the highest degree, we should proof the following inequality:

$$\{2\psi_\tau(\partial_\tau\psi_\tau)v_\tau\}\,(v_\tau-\mu_\tau)+\psi_\tau^2\,\{(\partial_\tau v_\tau)(v_\tau-\mu_\tau)-v_\tau\partial_\tau(v_\tau-\mu_\tau)\}>0$$

The above inequality is satisfied based on conditions 1 and 2 since v_τ and $(v_\tau-\mu_\tau)$ are positive due to the initial conditions on raw quadratic parameters of Section 3.1. Another step to make the quadratic function everywhere non-negative is to make the discriminant everywhere negative since a strictly positive quadratic function should not cross the x axis. Therefore, by some simple rewriting of the discriminant we come up with the following inequality:

$$a=\frac{v_\tau\psi_\tau^2}{v_\tau-\mu_\tau}\quad,\quad b=2\sqrt{v_\tau}\psi_\tau\quad,\quad c=v_\tau$$

$$\begin{aligned}4ac-b^2=&4\psi_\tau^2v_\tau^2\left\{\{(\partial_\tau v_\tau)(v_\tau-\mu_\tau)-v_\tau\partial_\tau(v_\tau-\mu_\tau)\}-\frac{(v_\tau-\mu_\tau)^2}{4v_\tau^2}\right\}\\&+4v_\tau\psi_\tau(\partial_\tau\psi_\tau)(v_\tau-\mu_\tau)\left\{2\,v_\tau^2-\left\{v_\tau\frac{(\partial_\tau\psi_\tau)}{\psi_\tau}+1\right\}(v_\tau-\mu_\tau)\right\}>0.\end{aligned}$$

So, we are supposed to make the above function strictly positive by providing some conditions on the three introduced parameters. The first part of the function above is positive due to the condition 2, and the second part is non-negative based on conditions 1 and 3. So, our convex quadratic model never crosses the x axis. Therefore, the proof is complete. □

Note that, in the previous proposition we provided some conditions on the parameters which are familiar for market traders and each of them is a function of time to maturity. So, to implement this strategy to market data all these parameters should be available in terms of expiry time. In the next proposition, we provide some conditions on the raw parameters to rule out static arbitrage. We will discuss ways and means of implementing this strategy to market data in Section 4.

Proposition 2. *The quadratic surface 14 is not influenced by calendar spread arbitrage if for any two times to maturity $\tau_1<\tau_2$ corresponding to $w(.,\tau_1)$ and $w(.,\tau_2)$ by the parameters sets $\eta_1=\{\theta_{01},\theta_{11},\theta_{21}\}$ and $\eta_2=\{\theta_{02},\theta_{12},\theta_{22}\}$ the following conditions satisfy:*

1. $\theta_{22}-\theta_{21}>0$;
2. $\theta_{22}\theta_{01}+\theta_{21}\theta_{02}<\frac{\theta_{12}\theta_{11}}{2}$.

Proof. To show that the two volatility slices never cross each other we should prove that the following quadratic function takes positive values everywhere. Hence, it should be a convex function with no real root

$$w(.,\tau_2)-w(.,\tau_1)=(\theta_{22}-\theta_{21})x^2+(\theta_{12}-\theta_{11})x+(\theta_{02}-\theta_{01}).\tag{16}$$

Condition 1 guarantees the quadratic Function (16) to be convex. In addition, we need to show that it does not have a real root, so the discriminant should take a negative value

$$\begin{aligned}\Delta=&(\theta_{12}-\theta_{11})^2-4(\theta_{22}-\theta_{21})(\theta_{02}-\theta_{01})\\=&(\theta_{12}^2-4\theta_{22}\theta_{02})+(\theta_{11}^2-4\theta_{21}\theta_{01})+4(\theta_{22}\theta_{01}+\theta_{21}\theta_{02})-2\theta_{12}\theta_{11}\end{aligned}$$

The first two terms are negative based on the initial conditions in Section 3.1, and also condition 2 makes the other two expressions negative. Therefore, $\Delta<0$ and the proof is complete. □

So, we use these conditions to parametrize total implied variance slice by slice. This means they play the role of optimization constraints for each fixed time to maturity to preclude calendar spread arbitrage. A common approach is a forward strategy which performs these conditions separately for the shortest time to expiration up to the longest one. Now, we set some conditions on the parameters to make a volatility slice free of butterfly arbitrage.

Proposition 3. *The quadratic volatility model in Section 3.1, for options with less than one year to maturity ($\tau < 1$), is free of butterfly arbitrage if*

1. $\theta_1^2 - 4\theta_0\theta_2 + \theta_2 < 0$;
2. $\frac{1}{4} < \theta_0 < 1$.

Proof. First of all, we show that the minimum value of the proposed model belongs to the interval $[0, 1]$ since we assumed options with less than one year expiry time which makes $w_{imp}^{Q^2}$ bounded between 0 and 1. So, the inequality $0 < w_{imp}^{Q^2}(-\frac{\theta_1}{2\theta_2}, \eta) < 1$ and equivalently the inequality $4(\theta_0 - 1)\theta_2 < \theta_1^2 < 4\theta_0\theta_2$ must be held. It is easily satisfied because of conditions 2 and also the initial conditions of Section 3.1. Moreover, another intuition behind the condition $\theta_0 < 1$ is to guarantee the model to be less than one in case of ATM. Now, we do some rearrangement to make the Durrleman's function take positive values everywhere.

$$
\begin{aligned}
g(x) &= \left(1 - \frac{x}{2w}\partial_x(w)\right)^2 - \frac{1}{4}\left(\frac{1}{w} - \frac{1}{4}\right)(\partial_x(w))^2 + \frac{1}{2}\partial_{xx}(w) \\
&= \left(1 - \frac{x(\theta_1 + 2\theta_2 x)}{2(\theta_0 + \theta_1 x + \theta_2 x^2)}\right)^2 + \left(-\frac{(\theta_1 + 2\theta_2 x)^2}{4(\theta_0 + \theta_1 x + \theta_2 x^2)} - \frac{(\theta_1 + 2\theta_2 x)^2}{16} + \theta_2\right) \\
&= f(x) + h(x)
\end{aligned}
$$

For the Durrleman's function g, we begin with the first expression as follows:

$$
f(x) = \left(1 - \frac{x(\theta_1 + 2\theta_2 x)}{2(\theta_0 + \theta_1 x + \theta_2 x^2)}\right)^2 = 1 + \frac{x^2(\theta_1 + 2\theta_2 x)^2}{4(\theta_0 + \theta_1 x + \theta_2 x^2)^2} - \frac{x\theta_1 + 2\theta_2 x^2}{\theta_0 + \theta_1 x + \theta_2 x^2}
$$

Rearranging the third term of function f, we get the following function:

$$
\begin{aligned}
\frac{x(\theta_1 + 2\theta_2 x)}{\theta_0 + \theta_1 x + \theta_2 x^2} &= \frac{\theta_1 x + \theta_2 x^2 + \theta_2 x^2 + \theta_0 - \theta_0}{\theta_0 + \theta_1 x + \theta_2 x^2} = \frac{\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_2 x^2 - \theta_0}{\theta_0 + \theta_1 x + \theta_2 x^2} \\
&= 1 + \frac{\theta_2 x^2 - \theta_0}{\theta_0 + \theta_1 x + \theta_2 x^2}
\end{aligned}
$$

Therefore, we have

$$
\begin{aligned}
f(x) &= \left(1 - \frac{x(\theta_1 + 2\theta_2 x)}{2w}\right)^2 = 1 + \frac{x^2(\theta_1 + 2\theta_2 x)^2}{4w^2} - 1 - \frac{\theta_2 x^2 - \theta_0}{w} \\
&= \frac{\theta_1^2 x^2 + 4\theta_1\theta_2 x^3 + 4\theta_2^2 x^4 - 4\theta_0\theta_2 x^2 + 4\theta_0^2 - 4\theta_1\theta_2 x^3 + 4\theta_0\theta_1 x - 4\theta_2^2 x^4 + 4\theta_0\theta_2 x^2}{4w^2} \\
&= \frac{\theta_1^2 x^2 + 4\theta_0\theta_1 x + 4\theta_0^2}{4w^2}
\end{aligned}
$$

Since $\theta_1^2 > 0$ and $\Delta = 0$, the numerator of f is a convex and strictly positive quadratic function which takes its minimum value at $x = 0$

$$
f\left(\frac{-4\theta_0\theta_1}{2\theta_1^2}\right) = f\left(\frac{-2\theta_0}{\theta_1}\right) = 4\theta_0^2 - 8\theta_0^2 + 4\theta_0^2 = 0
$$

So, regardless of the value of the parameters, the convex function f takes its minimum at 0, so we are not supposed to subtract any positive value from function f because we desire to make the Durrleman's function g everywhere positive. Now we have to work on other parts of g, working toward making some conditions on the parameters to rule out butterfly arbitrage. Based on condition 1 we have

$$\begin{aligned}(\theta_1+2\theta_2x)^2 =&\theta_1^2+4\theta_1\theta_2x+4\theta_2^2x^2\\ \leq&4\theta_0\theta_2+4\theta_1\theta_2x+4\theta_2^2x^2=4\theta_2w\end{aligned}$$

So, the following inequality is satisfied for the function h

$$\begin{aligned}h(x)=&-\frac{(\theta_1+2\theta_2x)^2}{4w}-\frac{(\theta_1+2\theta_2x)^2}{16}+\theta_2\\ \geq&\frac{4w\theta_2-(\theta_1+2\theta_2)^2}{4w}-\frac{4\theta_2w}{16}\\ =&\frac{1}{4}\left(\frac{4\theta_0\theta_2-\theta_1^2}{w}-\theta_2w\right)\end{aligned}$$

Since we assume this parameterization for options with less than one year to expiration ($\tau<1$), we have $w=\tau\sigma_{imp}^2<1$; thus, the fact that $-w\geq-\frac{1}{w}$ lets us make the function h everywhere positive

$$\begin{aligned}h(x)\geq&\frac{1}{4}\left(\frac{4\theta_0\theta_2-\theta_1^2}{w}-\frac{\theta_2}{w}\right)=\frac{1}{4}\left(\frac{4\theta_0\theta_2-\theta_1^2-\theta_2}{w}\right)\\ =&\frac{1}{4}\left(\frac{\theta_2(4\theta_0-1)-\theta_1^2}{w}\right)\geq 0.\end{aligned}$$

The last inequality is satisfied because of the first and second conditions we assumed for the model, so $g(\theta)\geq 0$. Note that we limit our work on options with less than one year to maturity, hence the data we use as w is between 0 and 1. Now we show that the second condition in Theorem 3 is satisfied

$$\begin{aligned}\lim_{k\to\infty}d_1\leq\lim_{k\to\infty}\sup d_1=&\lim_{k\to\infty}\sup\frac{\log\left(\frac{F_{[t,t+\tau]}}{k}\right)+\frac{1}{2}\tau\sigma_{imp}^2}{\sqrt{\tau\sigma_{imp}^2}}\\ =&\lim_{x\to-\infty}\sup\frac{x+\frac{1}{2}(\theta_0+\theta_1x+\theta_2x^2)}{\sqrt{\theta_0+\theta_1x+\theta_2x^2}}\\ =&\lim_{u\to\infty}\sup\frac{-u+\frac{1}{2}(\theta_0-\theta_1u+\theta_2u^2)}{\sqrt{\theta_0-\theta_1u+\theta_2u^2}}\\ =&\lim_{u\to\infty}\sup\frac{-\sqrt{u}}{\sqrt{2}}\left(\frac{\sqrt{2u}}{\sqrt{\theta_0-\theta_1u+\theta_2u^2}}-\frac{\sqrt{\theta_0-\theta_1u+\theta_2u^2}}{\sqrt{2u}}\right)\end{aligned}$$

Roper (2010) proved that if the superior limit of the second term in parenthesis tends to a constant in the interval [0, 1), then the last limit above goes to minus infinity

$$\lim_{u\to\infty}\sup\frac{\sqrt{\theta_0-\theta_1u+\theta_2u^2}}{\sqrt{2u}}<\lim_{u\to\infty}\sup\frac{1}{\sqrt{2u}}=0\in[0,1)$$

The inequality above is satisfied because we set $\theta_1\in\mathbb{R}$, therefore $\lim_{k\to\infty}d_1=-\infty$ and the proposed model is free of butterfly arbitrage. □

Now, due to the Propositions 1 and 3, we come up with the following conclusion that provides some conditions to rule out static arbitrage when we parametrize implied variance with respect to ATM variance, ATM volatility skew and the minimum level of variance.

Theorem 4. *The equivalent parameterization of implied variance for options with less than one year to maturity, is not faced with static arbitrage if*

1. $\psi_\tau \left(\partial_\tau \psi_\tau\right) > 0$
2. $\partial_\tau \left[\ln\left(\frac{v_\tau}{v_\tau - \mu_\tau}\right)\right] > \frac{(v_\tau - \mu_\tau)}{4v_\tau^3}$
3. $\partial_\tau \left[\ln \psi_\tau\right] < \frac{2v_\tau}{v_\tau - \mu_\tau} - \frac{1}{v_\tau}$
4. $0 < \tau v_\tau < \frac{1}{4}$
5. $\frac{v_\tau \psi_\tau^2}{v_\tau - \mu_\tau} \left(4\tau\mu_\tau - 1\right) > 0$

So far, we have provided some conditions that guarantee the absence of static arbitrage; thus, we have everything to fit the proposed quadratic model to implied volatility data.

4. Numerical Implementation

In this section, we provide a learning algorithm to modeling implied volatility data which is earned by S&P 500 European call options written on 15 December 2014. In other words, we consider bank asset to be S&P 500 index fund and we implement the proposed strategy to price call options written on this asset. The reason to choose S&P 500 as underlying asset is the simplicity and availability of this important data to make the numerical part move straightforward upon a well-defined path; whereas, underlying price process S_t, can be replaced by any type of risky asset.

The idea behind our strategy is that since the total implied variance of a security price is a smile-shaped function of log-moneyness, we fit the quadratic model 14 to the data. In other words, instead of just learning from input data x, we learn based on a mapping from x to its second degree polynomial. The training set of this investigation includes x as log-moneyness and w as total implied variance. To improve the robustness of the algorithm, training set data is randomly divided into two portions: 70% for the training set and 30% for the cross-validation set. The cost function consists of a penalty to control the trade-off between underfitting and overfitting. Finally, to illustrate the efficiency of the proposed approach, we perform it for six different times to maturity.

4.1. The Cost Function

The cost function we use to estimate the parameters of each volatility slice (for a fixed time to maturity) is a machine learning regularized cost function and the parameters are estimated by the following strategy:

$$\hat{\theta} = \arg\min_{\theta} \frac{1}{2m} \left(\sum_{i=1}^{m} \left(w_\theta^{Q^2}(x^{(i)}) - w^{(i)} \right)^2 + \lambda \sum_{j=1}^{2} \theta_j^2 \right) \tag{17}$$

$w^{(i)}$ is the corresponding total implied variance for the i-th training example and $w_\theta^{Q^2}(x^{(i)})$ is the quadratic model proposed in Section 3.1 and in this case, it plays the role of learning hypothesis. The cost function is a L-2 norm loss function plus a penalty term. L-2 function is chosen because it is the most common cost function; furthermore, it has one stable solution whereas the L-1 loss function has unstable and possibly multiple solutions. Since the goal is to estimate the parameters of a quadratic model, a L-2 regularization term is reasonable, and it encourages parameter values toward zero, but not exactly zero; moreover, the distribution of parameters is approximately a zero mean normal distribution. In case of model complexity (High test error), the penalty term keeps the parameters small to make the hypothesis relatively simple to avoid overfitting. λ is the regularization parameter that controls the trade-off between underfitting and overfitting.

When we choose a lambda value, the goal is to provide the right balance between simplicity and training-data fit. If lambda is too high, the model will be simple, but we may face the risk of underfitting and the model will not learn enough from the training set to make useful predictions. On the other hand, if lambda is too low, there is more model complexity, and we encounter the risk of overfitting; in addition, the model will learn too much from the training set and will not be able to generalize to unseen data. The ideal value of lambda provides a model that generalizes well to the data outside the training set, but it depends on data and we need to do some tuning. Therefore, based on a trial and error strategy, we check model complexity and change the value of λ, then the algorithm runs again to update parameters based on the new value for λ. Finally, the value of λ with the lowest complexity will be chosen as the ideal one. The way we choose the value of λ is clearly explained by a pseudo code in the next section.

To perform the algorithm, we learn the parameters from the training set, then the training error and the cross-validation error are computed based on the learned hypothesis in the training set, and learning curve which is the plot of the cross-validation error and the training error versus the size of the training set helps us diagnose if the model is affected by underfitting or overfitting. The training error and the cross-validation error are computed as follows:

$$J_{train}(\theta) = \frac{1}{2m},\sum_{i=1}^{m}\left(w_{\theta}^{Q^2}(x_{train}^{(i)}) - w_{train}^{(i)}\right)^2$$

$$J_{cv}(\theta) = \frac{1}{2m},\sum_{i=1}^{m}\left(w_{\theta}^{Q^2}(x_{cv}^{(i)}) - w_{cv}^{(i)}\right)^2$$

To overcome the effects of underfitting and overfitting for each volatility slice, the validation curve which is the cross-validation error plotted versus the regularization parameter λ helps us select the value of λ which minimizes the cross-validation error.

4.2. The Algorithm, Step by Step

In this section, to provide a better understanding of the proposed algorithm, we itemize a simple pseudo code to show how to plot the Durrleman's function and also choose the optimum value of λ that rule out both underfitting and overfitting. The algorithm runs as follows:

1. Start by a volatility data $(x^{(i)}, w^{(i)})$ for any fixed time to maturity.
2. Using the training set data and the conditions in Propositions 2 and 3, estimate parameters by minimizing the cost function for a fixed value of λ (For the first implementation let $\lambda = 0$).
3. Using the estimated parameters, compute training error and cross-validation error for different values of m.
4. Plot learning curve which is the training error and the cross-validation error versus m.
5. (a) If the learning curve shows no drawback of overfitting and underfitting, plot Durrleman's function based on the estimated parameters.

   (b) Otherwise, plot the validation curve which is the cross-validation error versus the regularization parameter λ, and choose the value of λ which minimizes the cross-validation error, then move on to step 2.

4.3. Ruling Out Calendar Spread Arbitrage

A forward approach is implemented to fit the proposed quadratic model 3.1 to the total implied variance data calculated by the Black-Scholes implied volatility in 9. Considering the initial conditions in Section 3.1 and others in Remark 3, the parameterization is not encountered with butterfly arbitrage for each volatility slice, but we need to determine some relations among parameters of different slices to organize them to be an increasing function of τ. First of all, we implement the optimization for the shortest time to maturity and simultaneously we implement conditions in Section 3 and Remark 2 to estimate the parameters, then we assign the conditions of Remark 2 for the second shortest expiry time due to the values of the estimated parameters for the first slice. For example, if the estimated parameters for the shortest expiry time are:

$$\theta_{2(1)} = a \quad , \quad \theta_{1(1)} = b \quad , \quad \theta_{0(1)} = c \quad , \quad a, b, c \in \mathbb{R}$$

where $\theta_{i(j)}$ is the i-th estimated parameter in the optimization for the j-th slice, we add some extra constraints for optimization in the second shortest expiry time as follows:

1. $\theta_{2(2)} > a$
2. $c\theta_{2(2)} + a\theta_{0(2)} < \frac{b\theta_{1(2)}}{2}$

So, in this way it is guaranteed for the two slices not to cross each other and also the second slice is everywhere greater than the first one. In the next step, doing the optimization forward, the same strategy is performed to the third shortest expiry time by some additional constraints due to the values of the parameters for the second slice. Therefore, by implementing the forward method from the slice with the shortest expiry time up to the one with the longest time to maturity, we ensure that the calibration provides a volatility surface with no calendar spread arbitrage for the volatility surface, and also no butterfly arbitrage for each slice. In general, for the optimization of the *n-th* slice we have the following calibration rules:

1. $\theta_{2(n)} > \theta_{2(n-1)}$
2. $\theta_{2(n)}\theta_{0(n-1)} + \theta_{2(n-1)}\theta_{0(n)} < \frac{\theta_{1(n)}\theta_{1(n-1)}}{2}$

Therefore, based on Definition 2, we have everything to rule out static arbitrage.

4.4. Discussion

Numerical implementation of the quadratic approach is done over six different times to maturity for S&P 500 call option data traded on December 15, 2014. Table 1 represents the optimal values of λ for each of the six different times to maturity. Figure 1 illustrates the plots of total implied variance for all six volatility slices and it shows that total implied variance is an increasing function of time to expiration since the volatility slices never cross each other, so the calibration method eliminates calendar spread arbitrage. Plots for all six Durrleman's functions are shown separately for each volatility slice in Figure 2. The plots of Durrleman's function for all six times to maturity are strictly positive around at-the-money, implying the absence of butterfly arbitrage for each volatility slice. Therefore, due to the conditions of Definition 2, we parameterized total implied variance for S&P 500 call option data in such a way that there is no static arbitrage.

Table 1. Times to maturity and the optimum values of the regularization parameter for each volatility slice.

Expiry Date	Time to Maturity	λ
20 December 2014	0.0136	0.3
2 January 2015	0.0465	3
17 January 2015	0.0876	1.2
23 January 2015	0.1041	2.8
20 February 2015	0.178	0.9
20 March 2015	0.232	1.3

To sum up, modeling implied volatility with respect to time to expiration and strike price, and precluding static arbitrage simultaneously, we can be aware of the upcoming price fluctuation of the risky asset and use it to price the options in Equation (6). Therefore, the risk management contract (6) can be priced more precisely based on the behavior of implied volatility. It is necessary to note that we did not implement an algorithm to price the contract since the main focus of this paper is to parametrize implied volatility to improve the precision of contract pricing and the rest is just related to option pricing that is widely studied in the literature.

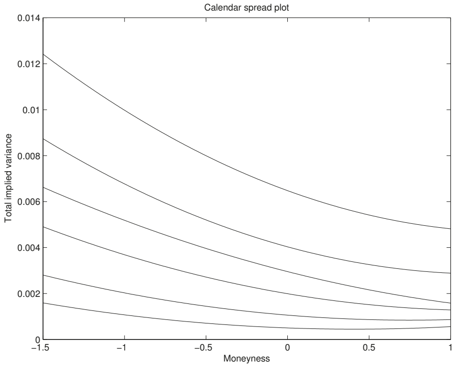


Figure 1. Plots of the total implied variance for six different times to maturity following the forward slice-by-slice method of Section 4.3.

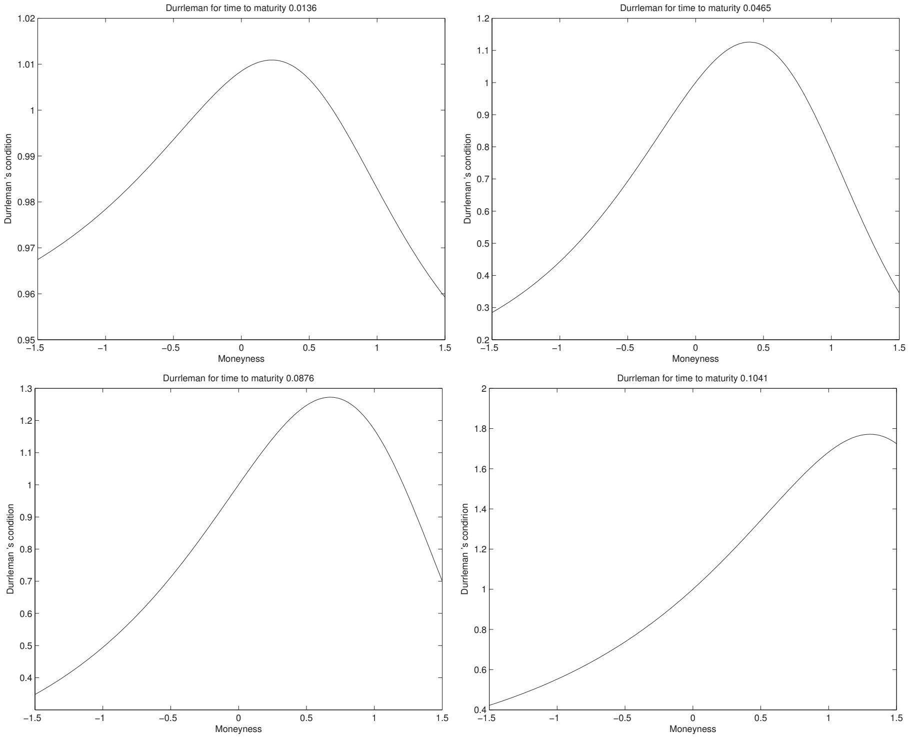


Figure 2. *Cont.*

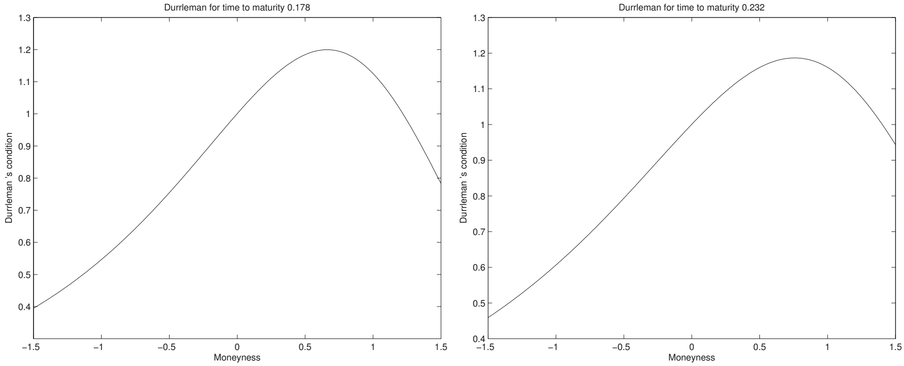


Figure 2. Plots of the Durrleman's function implemented for six different times to maturity.

5. Conclusions

Deposit insurances are introduced after the 1929 Great Depression as a tool to reduce the risk of depositors' loss. There are two major issues related to deposit insurances: the risk of moral hazard on the one hand, and the risk of miss-pricing and arbitrage on the other hand. The main objective of this study is to focus on the second issue by correctly pricing deposit insurances via improving the implied volatility calibration. As the deposit insurances have been blamed for generating the moral hazard risk, we considered a framework where the risk of moral hazard is ruled out (Assa and Okhrati (2018)) and we focused our attention on arbitrage. In the first step, we showed that in this framework no-arbitrage assumption can be reduced to no-static-arbitrage assumption. This paves the way towards parametrization of the implied volatility. After introducing a quadratic approach to parameterized implied volatility, we mathematically proved that for options with less than one year to maturity and under some special conditions on parameters of the model, there is no opportunity for static arbitrage. The results of the numerical implementation have shown that the proposed quadratic model can be a helpful strategy for modeling implied volatility. Furthermore, our approach improved other quadratic approaches which have already been proposed, since none of them could take care of arbitrage opportunity. Another interesting property of the model is the simplicity of the quadratic function which is understandable by a basic knowledge of mathematics. However, we believe this area of volatility modeling still has some room to improve based on additional market features like the underlying price, time to expiration and strike price, which we leave for future works.

Author Contributions: Supervision, project administration, resources, A.H.; conceptualization of the material, methodology and designing the framework, A.H. and P.M.; investigation, data curation, visualization, formal analysis, software, validation, writing–original draft preparation, writing–review and editing, A.H., P.M. and B.A.

Funding: This research received no external funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

Alentorn, Amadeo. 2004. Modelling the Implied Volatility Surface, an Empirical Study for FTSE Options. Available online: www.theponytail.net/CCFEA (accessed on 5 May 2004).

Assa, Hirbod. 2015. Risk Management under a Prudential Policy. *Decisions in Economics and Finance* 38: 217–230. [CrossRef]

Assa, Hirbod, and Ramin Okhrati. 2018. Designing sound deposit insurances. *Journal of Computational and Applied Mathematics* 327: 226–242. [CrossRef]

Avellaneda, Marco. 2005. From SABR to Geodesics. In *Conference Presentation at Courant Institute*; New York: New York University.

Bernard, Carole, and Weidong Tian. 2009. Optimal reinsurance arrangements under tail risk measures. *Journal of Risk and Insurance* 76: 709–725. [CrossRef]

Black, Fischer, and Myron Scholes. 1973. The pricing of options and corporate liabilities. *Journal of Political Economy* 81: 637–654. [CrossRef]

Breeden, Douglas, and Robert Litzenberger. 1978. Prices of state-contingent claims implicit in option prices. *Journal of Business* 51: 621–651. [CrossRef]

Carr, Peter, and Dilip Madan. 2005. A note on sufficient conditions for no arbitrage. *Finance Research Letters* 30: 125–30. [CrossRef]

Castagna, Antonio, and Fabio Mercurio. 2007. OPTION PRICING: The vanna-volga method for implied volatilities. *Risk* 20: 106.

Cont, Rama, and José Da Fonseca. 2002. Dynamics of implied volatility surfaces. *Quantitative Finance* 2: 45–60. [CrossRef]

Durrleman, Valdo. 2003. A Note on Initial Volatility Surface. Unpublished manuscript.

Fengler, Matthias. 2009. Arbitrage-free smoothing of the implied volatility surface. *Quantitative Finance* 9: 417–428. [CrossRef]

Gatheral, Jim. 2004. A parsimonious arbitrage-free implied volatility parameterization with application to the valuation of volatility derivatives. Paper presented at Global Derivatives & Risk Management, Madrid, Spain, May 26.

Gatheral, Jim. 2011. *The Volatility Surface: A Practitioner's Guide*. Hoboken: John Wiley & Sons, Inc., vol. 357.

Gatheral, Jim, and Antoine Jacquier. 2014. Arbitrage-free SVI volatility surfaces. *Quantitative Finance* 14: 59–71. [CrossRef]

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2002. *The Elements of Statistical Learning, Data Mining, Inference, and Prediction*. Berlin: Springer.

Heimer, Carol. 1989. *Reactive Risk and Rational Action: Managing Moral Hazard in Insurance Contracts*. Berkeley: University of California Press, vol. 6.

Kellerer, Hans. 1972. Markov-komposition und eine anwendung auf martingale. *Mathematische Annalen* 198: 99–122. [CrossRef]

Malliaris, Mary, and Linda Salchenberger. 1996. Using neural networks to forecast the S&P 100 implied volatility. *Neurocomputing* 10: 95–183.

Merton, Robert. 1977. An analytic derivation of the cost of deposit insurance and loan guarantees an application of modern option pricing theory. *Journal of Banking & Finance* 1: 3–11.

Nilsson, Nils. 2005. *Introduction to Machine Learning*. Stanford: Department of Computer Science, Stanford University

Roper, Michael. 2009. *Implied Volatility: General Properties and Asymptotics*. Kensington: The University of New South Wales, pp. 2–3.

Roper, Michael. 2010. Arbitrage-Free Implied Volatility Surfaces. Available online: www.maths.usyd.edu.au/u/pubs/publist/preprints/2010/roper-9.pdf (accessed on 18 April 2019).

Roux, Martin. 2007. A long-term model of the dynamics of the S&P500 implied volatility surface. *North American Actuarial Journal* 119: 61–75.

Zhang, Tao, and Li Liu. 2015. Economic policy uncertainty and stock market volatility. *Finance Research Letters* 15: 99–105.

Zhao, Bo, and Stewart, Hodges. 2013. Parametric modeling of implied smile functions: A generalized SVI model. *Review of Derivatives Research* 16: 53–77. [CrossRef]

Zhu, Anyi. 2013. Implied Volatility Modeling. Master's dissertation, University of Waterloo, Waterloo, ON, Canada.

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).

risks

MDPI

Article

Predicting Motor Insurance Claims Using Telematics Data—XGBoost versus Logistic Regression

Jessica Pesantez-Narvaez, Montserrat Guillen * and Manuela Alcañiz

Department of Econometrics, Riskcenter-IREA, Universitat de Barcelona, 08034 Barcelona, Spain; jessica.pesantez@ub.edu (J.P.-N.); malcaniz@ub.edu (M.A.)

* Correspondence: mguillen@ub.edu; Tel.: +34-934-037-039

Received: 9 May 2019; Accepted: 12 June 2019; Published: 20 June 2019

Abstract: XGBoost is recognized as an algorithm with exceptional predictive capacity. Models for a binary response indicating the existence of accident claims versus no claims can be used to identify the determinants of traffic accidents. This study compared the relative performances of logistic regression and XGBoost approaches for predicting the existence of accident claims using telematics data. The dataset contained information from an insurance company about the individuals' driving patterns—including total annual distance driven and percentage of total distance driven in urban areas. Our findings showed that logistic regression is a suitable model given its interpretability and good predictive capacity. XGBoost requires numerous model-tuning procedures to match the predictive performance of the logistic regression model and greater effort as regards to interpretation.

Keywords: dichotomous response; predictive model; tree boosting; GLM; machine learning

1. Introduction

Predicting the occurrence of accident claims in motor insurance lies at the heart of premium calculation, but with the development of new artificial intelligence methods, the question of choosing a suitable model has yet to be completely solved. In this article, the recently proposed methods of XGBoost (Chen and Guestrin 2016) and logistic regression are considered and compared regarding their predictive performance in a sample of insured drivers, along with their telematic information.

The advantages and disadvantages of XGBoost compared to logistic regression are discussed and this study showed that a slightly improved predictive power is only obtained with the XGBoost method, but this has complicated the interpretation of the impact of covariates on the expected response. In the case of automobile insurance, where the premium calculation is regulated and has to be fully specified, the weight of each risk factor in the final price needs to be disclosed and the connection between the observed covariate value and the estimated probability of a claim needs to be shown. If these conditions are not met, the regulating authority may deny the insurance company the right to commercialize that product. This study discussed, nevertheless, why the use of an XGBoost algorithm remains interesting for actuaries and how methods both old and new might be combined for optimum results. This study does not examine any other boosting methods. However, excellent descriptions can be found in Lee and Lin (2018), while extensions to high dimensional datasets are presented in Lee and Antonio (2015), both of which presented cases studies of insurance applications. Many of those alternatives placed their emphasis on algorithm speed, but in terms of their essential setups they do not differ greatly from XGBoost.

To compare the two competing methods, a real dataset comprising of motor insurance policy holders and their telematics measurements were used, that is, real-time driving information collected and stored via telecommunication devices. More specifically, GPS-based technology captures an insured's driving behavior patterns, including distance travelled, driving schedules, and driving speed, among many others. Here, pay-as-you-drive (PAYD) insurance schemes represent an alternative method for pricing premiums based on personal mileage travelled and driving behaviors. Guillen et al. (2019), Verbelen et al. (2018), and Pérez-Marín and Guillén (2019) showed the potential benefits of analyzing telematics information when calculating motor insurance premiums. Gao and Wüthrich (2019) analyzed high-frequency GPS location data (second per second) of individual car drivers and trips. Gao and Wüthrich (2018) and Gao et al. (2019) investigated the predictive power of covariates extracted from telematics car driving data using the speed-acceleration heatmaps proposed by Wüthrich (2017). Further, Hultkrantz et al. (2012) highlighted the importance of PAYD insurance plans insofar as they allow insurance companies to personalize premium calculation and, so, charge fairer rates.

The rest of this paper is organized as follows. First, the notation is introduced and the logistic regression and XGBoost methods are outlined. Second, our dataset is described and some descriptive statistics are provided. Third, the results of our comparisons in both a training and a testing sample are reported. Finally, following the conclusion, some practical suggestions are offered about the feasibility of applying new machine learning methods to the field of insurance.

2. Methodology Description

In a data set of n individuals and P covariates, there is a binary response variable Y_i, $i = 1, \ldots,$ n taking values 0, 1; and a set of covariates denoted as X_{ip}, $p = 1, \ldots, P$. The conditional probability density function of $Y_i = t$ ($t = 0, 1$) given X_i ($X_{i1}, \ldots, X_{iP}$), is denoted as $h_t(X_i)$. Equivalently, it can be said that $\text{Prob}(Y_i = t) = h_t(X_i)$, and that $\text{E}(Y_i) = \text{Prob}(Y_i = 1) = h_1(X_i)$.

2.1. Logistic Regression

Logistic regression, a widely recognized regression method for predicting the expected outcome of a binary dependent variable, is specified by a given set of predictor variables. McCullagh and Nelder (1989) presented the logistic regression model as part of a wider class of generalized linear models. A logistic regression is distinguished from a classical linear regression model primarily because the response variable is binary rather than continuous in nature.

The logistic regression uses the logit function as a canonical link function, in other words, the log ratio of the probability functions $h_t(X_i)$ is a linear function of X; that is:

$$\ln \frac{h_1(X_i)}{h_0(X_i)} = \ln \frac{\text{Prob}(Y_i = 1)}{\text{Prob}(Y_i = 0)} = \beta_0 + \sum_{p=1}^{P} X_{ip}\beta_p, \tag{1}$$

where $\beta_0, \beta_1, \ldots, \beta_P$ are the model coefficients[1], and $\text{Prob}(Y_i = 1)$ is the probability of observing the event in the response (response equal to 1), and $\text{Prob}(Y_i = 0)$ is the probability of not observing the event in the response (response equal to 0).

The link function provides the relationship between the linear predictor $\eta = \beta_0 + \sum_{p=1}^{P} X_{ip}\beta_p$ and the mean of the response given certain covariates. In a logistic regression model, the expected response is:

$$E(Y_i) = \text{Prob}(Y_i = 1) = \frac{e^{\beta_0 + \sum_{p=1}^{P} X_{ip}\beta_p}}{1 + e^{\beta_0 + \sum_{p=1}^{P} X_{ip}\beta_p}}. \tag{2}$$

A logistic regression can be estimated by the maximum likelihood (for further details see, for example, Greene 2002). Therefore, the idea underlying a logistic regression model is that there must be a linear combination of risk factors that is related to the probability of observing an event. The data analyst's task is to find the fitted coefficients that best estimate the linear combination in (2) and to interpret the relationship between the covariates and the expected response. In a logistic regression

[1] Note we have opted to refer here to coefficients as opposed to parameters to avoid confusion with the values defined below when describing the XGBoost method.

model, a positive estimated coefficient indicates a positive association. Thus, when the corresponding covariate increases, the probability of the event response also increases. If the estimated coefficient is negative, then the association is negative and, therefore, the probability of the event decreases when the observed value of the corresponding covariate increases. Odds-ratios can be calculated as the exponential values of the fitted coefficients and they can also be directly interpreted as the change in odds when the corresponding factor increases by one unit.

Apart from their interpretability, the popularity of logistic regression models is based on two characteristics: (i) The maximum likelihood estimates are easily found; and (ii) the analytical form of the link function in (2) always provides predictions between 0 and 1 that can be directly interpreted as the event probability estimate. For these motives, logistic regression has become one of the most popular classifiers, their results providing a straightforward method for predicting scores or propensity values which, in turn, allow new observations to be classified to one of the two classes in the response. For R users, the glm function is the most widely used procedure for obtaining coefficient estimates and their standard errors, but alternatively, a simple optimization routine can easily be implemented.

2.2. XGBoost

Chen and Guestrin (2016) proposed XGBoost as an alternative method for predicting a response variable given certain covariates. The main idea underpinning this algorithm is that it builds D classification and regression trees (or CARTs) one by one, so that each subsequent model (tree) is trained using the residuals of the previous tree. In other words, the new model corrects the errors made by the previously trained tree and then predicts the outcome.

In the XGBoost, each ensemble model[2] uses the sum of D functions to predict the output:

$$\hat{Y}_i = \text{₣}(X_i)= \sum\nolimits_{d=1}^{D} f_d(X_i),\ f_d \in \mathrm{F},\ i = 1,\ldots,n \tag{3}$$

where F is the function space[3] of the CART models, and each f_d corresponds to an independent CART structure which is denoted as q. In other words, q is the set of rules of an independent CART that classifies each individual i into one leaf. The training phase involves classifying n observations so that, given the covariates X, each leaf has a score that corresponds to the proportion of cases which are classified into the response event for that combination of X_i. This score is denoted as $w_{q(X)}$.

Thus, q can be written as a function q: $\mathbb{R}^P \rightarrow T$, where T is the total number of leaves of a tree and j is later used to denote a particular leaf, $j = 1, \ldots, T$. To calculate the final prediction for each individual, the score of the leaves are summed as in (3), where F = $\{f(\mathrm{X}) = w_{q(X)}\}$, with q: $\mathbb{R}^P \rightarrow T$, and $w \in \mathbb{R}^T$.

In general, boosting methods fit D models in D iterations (each iteration denoted by d, $d = 1, \ldots, D$) in reweighted versions. Weighting is a mechanism that penalizes the incorrect predictions of past models, in order to improve the new models. The weighting structures are generally optimal values, which are adjusted once a loss function is minimized. Then, new learners incorporate the new weighting structure in each iteration, and predict new outcomes.

In particular, the XGBoost method minimizes a regularized objective function, i.e., the loss function plus the regularization term:

$$\mathcal{L} = \sum\nolimits_{i=1}^{n} \ell(Y_i, \hat{Y}_i) + \sum\nolimits_{d=1}^{D} \acute{\eta}(f_d), \tag{4}$$

[2] Natekin and Knoll (2013) explain that the ensemble model can be understood as a committee formed by a group of base learners or weak learners. Thus, any weak learner can be introduced as a boosting framework. Various boosting methods have been proposed, including: (B/P-) splines (Huang and Yang 2004); linear and penalized models (Hastie et al. 2009); decision trees (James et al. 2013); radial basis functions (Gomez-Verdejo et al. 2005); and Markov random fields (Dietterich et al. 2008). Although Chen and Guestrin (2016) state f_k as a CART model, the R package xgboost currently performs three boosters: linear, tree and dart.

[3] The XGBoost works in a function space rather than in a parameter space. This framework allows the objective function to be customized accordingly.

where ℓ is a convex loss function that measures the difference between the observed response Y_i and predicted response $\hat{Y}_i$ and $\acute{\eta} = \mu T + \frac{1}{2}\lambda\|w\|_2^2$, $\acute{\eta}$ is the regularization term also known as the shrinkage penalty which penalizes the complexity of the model and avoids the problem of overfitting. The tree pruning parameter μ regulates the depth of the tree and λ is the regularization parameter that is associated with *l*2-norm of the scores vector, which is a way of evaluating the magnitude of scores. Including this norm, or any other similar expression, penalizes excessive sizes in the components of w.

It is noted that pruning is a machine learning technique which reduces the size of a decision tree by removing decision nodes whose corresponding features have little influence on the final prediction of the target variable. This procedure reduces the complexity of the model and, thus, corrects overfitting.

The *l*2-norm is used in the L2 or Ridge regularization method, while the *l*1-norm is used in the L1 or Lasso regularization method. Both methods can take the Tikhonov or the Ivanov form (see Tikhonov and Arsenin 1977; Ivanov et al. 2013).

2.2.1. A Closer Look at the XGBoost Minimization Algorithm

A loss function or a cost function like (4) measures how well a predictive algorithm fits the observed responses in a data set (for further details, see Friedman et al. 2001). For instance, in a binary classification problem, the logistic loss function is suitable because the probability score is bounded between 0 and 1. Then, by selecting a suitable threshold, a binary outcome prediction can be found. Various loss functions have been proposed in the literature, including: The square loss; the hinge loss (Steinwart and Christmann 2008); the logistic loss (Schapire and Freund 2012); the cross entropy loss (De Boer et al. 2005); and the exponential loss (Elliott and Timmermann 2003).

The intuition underpinning the regularization proposed in (4) involves reducing the magnitude of w, so that the procedure can avoid the problem of overfitting. The larger the e, the smaller the variability of the scores (Goodfellow et al. 2016).

The objective function at the d-th iteration is:

$$\mathcal{L}^{(d)} = \sum\nolimits_{i=1}^{n} \ell\left(Y_i,\ \hat{Y}_i^{(d-1)} + f_d(X_i)\right) + \acute{\eta}(f_d), \tag{5}$$

where $\hat{Y}_i^{(d-1)}$ is the prediction of the i-th observation at the $(d-1)$-th iteration. It is noted that $\ell(\cdot,\cdot)$ is generally a distance so its components can be swapped, i.e., $\ell(Y_i, \hat{Y}_i) = \ell(\hat{Y}_i,\ Y_i)$. Following Chen and Guestrin (2016), it is assumed that the loss function is a symmetric function.

Due to the non-linearities in the objective function to be minimized, the XGBoost is an algorithm that uses a second-order Taylor approximation of the objective function $\mathcal{L}$ in (5) as follows:

$$\mathcal{L}^{(d)} \cong \sum\nolimits_{i=1}^{n} \left[\ell\left(Y_i,\ \hat{Y}_i^{(d-1)}\right) + g_i f_d(X_i) + \frac{1}{2} h_i {f_d}^2(X_i)\right] + \acute{\eta}(f_d), \tag{6}$$

where $g_i = \partial_{\hat{y}_i^{(d-1)}}\ \ell\left(Y_i,\ \hat{Y}_i^{(d-1)}\right)$ and $h_i = \partial^2_{\hat{Y}_i^{(d-1)}}\ \ell\left(Y_i,\ \hat{Y}_i^{(d-1)}\right)$ denote the first and second derivatives of the loss function ℓ with respect to the component corresponding to the predicted classifier.

Since the authors minimized (6) with respect to f_d, this expression can be simplified by removing the constant terms as follows:

$$\mathcal{L}^{(d)} = \sum\nolimits_{i=1}^{n} \left[g_i f_d(X_i) + \frac{1}{2} h_i {f_d}^2(X_i)\right] + \acute{\eta}(f_d). \tag{7}$$

Substituting the shrinkage penalty $\acute{\eta}$ of (4) in (7), the authors obtained:

$$\mathcal{L}^{(d)} = \sum\nolimits_{i=1}^{n} \left[g_i f_d(X_i) + \frac{1}{2} h_i {f_d}^2(X_i)\right] + uT + \frac{1}{2}\lambda\|w\|_2^2. \tag{8}$$

The *l2*-norm shown in (8) is equivalent to the sum of the squared weights of all *T* leafs. Therefore (8) is expressed as:

$$\mathcal{L}^{(d)} = \sum\nolimits_{i=1}^{n}[g_i f_d(X_i) + \frac{1}{2} h_i f_d{}^2(X_i)] + u\mathrm{T} + \frac{1}{2}\lambda \sum\nolimits_{j=1}^{T} w_j^2. \tag{9}$$

The definition of I_j = {$i|q(X_i)$}, I_j is the set of observations that are classified into one leaf *j*, *j* = 1, … , *T*. Each I_j receives the same leaf weight w_j. Therefore, $\mathcal{L}^{(d)}$ in (9) can also be seen as an objective function that corresponds to each set I_j. In this sense, the $f_d(X_i)$, which is assigned to the observations, corresponds to the weight w_j that is assigned to each set I_j. Therefore (9) is expressed as:

$$\mathcal{L}^{(d)} = \sum\nolimits_{j=1}^{T}\left[\left(\sum\nolimits_{i\in I_j} g_i\right)w_j + \frac{1}{2}\left(\sum\nolimits_{i\in I_j} h_i + \lambda\right)w_j^2\right] + u\mathrm{T}. \tag{10}$$

In order to find the optimal leaf weight w_j^*, the authors derived (10) with respect to w_j, let the new equation be equal to zero, and cleared the value of w_j^*. Then the authors obtained:

$$w_j^* = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda}. \tag{11}$$

The (10) was updated by replacing the new w_j^*. The next boosting iteration minimized the following objective function:

$$\begin{aligned}\hat{\mathcal{L}}^{(d)} &= \textstyle\sum_{j=1}^{T}\left[\left(\sum_{i\in I_j} g_i\right)\left(-\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i+\lambda}\right) + \frac{1}{2}\left(\sum_{i\in I_j} h_i + \lambda\right)\left(-\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i+\lambda}\right)^2\right] + u\mathrm{T} \\ &= -\tfrac{1}{2}\textstyle\sum_{i=1}^{n} \frac{\left(\sum_{i\in I_j} g_i\right)^2}{\left(\sum_{i\in I_j} h_i+\lambda\right)} + u\mathrm{T}.\end{aligned} \tag{12}$$

Once the best objective function has been defined and the optimal leaf weights assigned to I_j, the best split procedure is considered. As (12) is derived for a wide range of functions, the authors were not able to identify all possible tree structures *q* in each boosting iteration. This algorithm starts by building a single leaf and continues by adding new branches. Consider the following example:

Here, I_L and I_R are the sets of observations that are in the left and right parts of a node following a split. Therefore, $I = I_L + I_R$.

$$\hat{\mathcal{L}}^{(d)} = \frac{1}{2}\left[-\sum\nolimits_{i=1}^{n} \frac{\left(\sum_{i\in I_j} g_i\right)^2}{\left(\sum_{i\in I_j} h_i + \lambda\right)} + \sum\nolimits_{i=1}^{n} \frac{\left(\sum_{i\in I_L} g_i\right)^2}{\left(\sum_{i\in I_L} h_i + \lambda\right)} + \sum\nolimits_{i=1}^{n} \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\left(\sum_{i\in I_R} h_i + \lambda\right)}\right] - u, \tag{13}$$

$\hat{\mathcal{L}}^{(d)}$ of (13) is the node impurity measure, which is calculated for the *P* covariates. The split is determined by the maximum value of (13). For example, in the case of CART algorithms, the impurity measure for categorical target variables can be information gain, Gini impurity or chi-square, while for continuous target variables it can be the Gini impurity.

Once the tree f_d is completely built (i.e., its branches and leaf weights are established), observations are mapped on the tree (from the root to one corresponding leaf). Thus, the algorithms will update from (5) to (14) as many times as D boosting iterations are established and the final classification is the sum of the *D* obtained functions which are shown in (3). Consequently, the XGBoost corrects the mistaken predictions in each iteration, as far as this is possible, and tends to overfit the data. Thus, to prevent overfitting, the regularization parameter value in the objective function is highly recommended.

2.2.2. Implementation

An example of R code is given in the Appendix A.

The implementation of XGBoost has proved to be quite effective for fitting real binary response data and a good method for providing a confusion matrix, i.e., a table in which observations and predictions are compared, with very few false positives and false negatives. However, since the final prediction of an XGBoost algorithm is the result of a sum of D trees, the graphical representation and the interpretation of the impact of each covariate on the final estimated probability of occurrence may be less direct than in the linear or logistic regression models. For instance, if the final predictor is a combination of several trees, but each tree has a different structure (in the sense that each time the order of segmentation differs from that of the previous tree), the role of each covariate will depend on understanding how the covariate impacts the result in the previous trees and what the path of each observation is in each of the previous trees. Thus, in the XGBoost approach, it is difficult to isolate the effect on the expected response of one particular covariate compared to all the others.

Under certain circumstances, the XGBoost method can be interpreted directly. This happens when f_d has analytical expressions that can easily be manipulated to compute $\sum_{d=1}^{D} f_d(X_i)$. One example is the linear booster, which means that each f_d is a linear combination of the covariates rather than a tree-based classifier. In this case of a linear function, the final prediction is also a linear combination of the covariates, resulting from the sum of the weights associated with each covariate in each f_d.

The results for the true XGBoost predictive model classifier can easily be obtained in R with the xgboost package.

3. Data and Descriptive Statistics

Our case-study database comprised of 2767 drivers under 30 years of age who underwrote a pay-as-you-drive (PAYD) policy with a Spanish insurance company. Their driving activity was recorded using a telematics system. This information was collected from 1 January through 31 December 2011. The data set contained the following information about each driver: The insured's age (*age*), the age of the vehicle (*ageveh*) in years; the insured's gender (*male*); the driving experience (*drivexp*) in years; the percentage of total kilometers travelled in urban areas (*pkmurb*); the percentage of total kilometers travelled at night—that is, between midnight and 6 am (*pkmnig*); the percentage of kilometers above the mandatory speed limits (*pkmexc*); the total kilometers (*kmtotal*); and, finally, the presence of an accident claim with fault (Y) which was coded as 1 when, at least, one claim where the fault occurred in the observational period and was reported to the insurance company, and 0 otherwise. This study is interested in predicting Y using the aforementioned covariates. This data set has been extensively studied in Ayuso et al. (2014, 2016a, 2016b) and Boucher et al. (2017).

Table 1 shows the descriptive statistics for the accident claims data set. This highlighted that a substantial part of the sample did not suffer an accident in 2011, with just 7.05% of drivers reporting at least one accident claim. The insureds with no accident claim seemed to have travelled fewer kilometers than those presenting a claim. The non-occurrence of accident claims was also linked to a lower percentage of driving in urban areas and a lower percentage of kilometers driven above mandatory speed limits. In this dataset, 7.29% of men and 6.79% of women had an accident during the observation year.

The data set was divided randomly into a training data set of 1937 observations (75% of the total sample) and a testing data set of 830 observations (25% of the total sample). The function CreateDataPartion of R was used to maintain the same proportion of events (coded as 1) of the total sample in both the training and testing data sets.

Table 1. The description of the variables in the accident claims data set [1].

Variables		Non-Occurrence of Accident Claims (Y = 0)	Occurrence of Accident Claims (Y = 1)	Total
Age (years)		25.10	24.55	25.06
Gender	Female	1263 (93.21%)	92 (6.79%)	1355
	Male	1309 (92.71%)	103 (7.29%)	1412
Driving experience (years)		4.98	4.46	4.94
Age of vehicle (years)		6.37	6.17	6.35
Total kilometers travelled		7094.63	7634.97	7132.71
Percentage of total kilometers travelled in urban areas		24.60	26.34	24.72
Percentage of total kilometers above the mandatory speed limit		6.72	7.24	6.75
Percentage of total kilometers travelled at night		6.88	6.66	6.86
Total number of cases		2572 (92.95%)	195 (7.05%)	2767

[1] The mean of the variables according to the occurrence and non-occurrence of accident claims. The absolute frequency and row percentage is shown for the variable gender.

4. Results

In this section, the results obtained in the training and testing samples were compared when employing the methods described in Section 2.

4.1. Coefficient Estimates

Table 2 presents the estimates obtained using the two methods. It is noted, however, that the values are not comparable in magnitude as they correspond to different specifications. The logistic regression uses its classical standard method to compute the coefficients of the variables and their standard errors. However, the boosting process of the XGBoost builds D models in reweighted versions and, therefore, a historical record of the D times $P + 1$ coefficient estimates was obtained. XGBoost can only obtain a magnitude of those coefficients if the base learner allows it, and this is not the case when f_d are CART models.

The signs obtained by the logistic regression point estimate and the mean of the XGBoost coefficients are the same. The inspection of the results in Table 2 shows that older insureds are less likely to suffer a motor accident than younger policy holders[4]. In addition, individuals who travel more kilometers in urban areas are more likely to have an accident than those that travel fewer kilometers in urban areas. The authors were not able to interpret the coefficients of the XGBoost, but by inspecting the maximum and minimum values of the linear booster case, an idea of how the estimates fluctuate until iteration D was obtained.

Only the coefficients of age and percentage of kilometers travelled in urban areas were significantly different from zero in the logistic regression model, but the authors preferred to keep all the coefficients of the covariates in the estimation results to show the general effect of the telematics covariates on the occurrence of accident at-fault claims in this dataset, and to evaluate the performance of the different methods in this situation.

[4] In general, this is only partially true. The relation of the variable age is typically non-linear, U-shaped, as (very) young drivers also cause a lot of accidents. The maximum age in this sample is 30 and so, even if models with age and age^2 were estimated, the results did not change substantially.

Table 2. The parameter estimates of the logistic regression and XGBoost with linear booster.

Training Data Set							
Parameter Estimates	**Logistic Regression**				**XGBoost (Linear Booster)**		
	Lower Bound	**Estimate**	**Upper Bound**	***p*-Value**	**Minimum**	**Mean**	**Maximum**
Constant	−2.8891	−0.5442	1.8583	0.6526	−2.6760	−2.6690	−1.7270
* age	−0.2059	−0.0994	0.0011	0.0591	−0.2573	−0.2416	−0.0757
drivexp	−0.1285	−0.0210	0.0906	0.7060	−0.0523	−0.0517	−0.0069
ageveh	−0.0786	−0.0249	0.0257	0.3481	−0.0897	−0.0885	−0.0220
male	−0.3672	0.0039	0.3751	0.9837	0.0019	0.0020	0.0070
kmtotal	−0.0203	0.0266	0.2505	0.0137	0.1164	0.1176	
pkmnig	−0.0354	−0.0046	0.0239	0.7625	−0.0292	−0.0290	−0.0061
pkmexc	−0.0122	0.0144	0.0385	0.2650	0.0180	0.1007	0.1016
* pkmurb	0.0002	0.0146	0.0286	0.0425	0.0436	0.2008	0.2023

In the logistic regression columns, the point estimates are presented with the lower and upper bound of a 95% confidence interval. In the XGBoost columns, the means of the coefficient estimates with a linear boosting of the *D* iterations are presented. Similarly, bounds are presented with the minimum and maximum values in the iterations. There are no regularization parameter values. * Indicates that the coefficient is significant at the 90% confidence level in the logistic regression estimation. The calculations were performed in R and scripts are available from the authors.

Figure 1 shows the magnitude of all the estimates of the XGBoost in 200 iterations. From approximately the tenth iteration, the coefficient estimates tend to become stabilized. Thus, no extreme changes were present during the boosting.

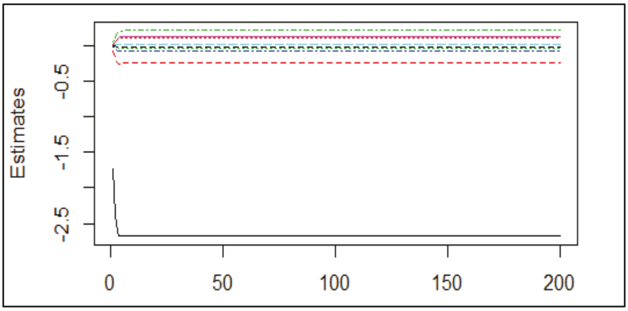


Figure 1. The magnitude of all the estimates in the *D* = 200 iterations. The different colors indicate each of the coefficients in the XGBoost iteration.

4.2. Prediction Performance

The performance of the two methods was evaluated using the confusion matrix, which compares the number of observed events and non-events with their corresponding predictions. Usually, the larger the number of correctly classified responses, the better the model. However, the out-of-sample performance was even more important than in-sample results. This means that the classifier must be able to predict the observed events and non-events in the testing sample and not just in the training sample.

The predictive measures used to compare the predictions of the models were sensitivity, specificity, accuracy and the root mean square error (RMSE). Sensitivity measures the proportion of actual positives that are classified correctly as such, i.e., True positive/(True positive + False negative). Specificity measures the proportion of actual negatives that are classified correctly as such, i.e., True negative/(True negative + False positive). Accuracy measures the proportion of total cases classified correctly (True positive + True negative)/Total cases. RMSE measures the distance between the observed and predicted values of the response. It is calculated as follows:

$$\sqrt{\sum\nolimits_{i=1}^{n} \frac{\left(Y_i - \hat{Y}_i\right)^2}{n}}, \tag{14}$$

The higher the sensitivity, the specificity and the accuracy, the better the models predict the outcome variable. The lower the value of RMSE, the better the predictive performance of the model.

Table 3 presents the confusion matrix and the predictive measures of the methods (the logistic regression, XGBoost with a tree booster and XGBoost with a linear booster) for the training and testing samples. The results in Table 3 indicated that the performance of the XGBoost with the linear booster (last column) was similar to that of the logistic regression both in the training and testing samples[5]. XGBoost using the tree approach provided good accuracy and a good RMSE value in the training sample, but did not perform as well as the other methods in the case of the testing sample. More importantly, XGBoost failed to provide good sensitivity. In fact, the XGBoost with the tree booster clearly overfitted the data, because while it performed very well in the training sample, it failed to do so in the testing sample. For instance, sensitivity was equal to 100% in the training sample for the XGBoost tree booster methods, but it was equal to only 7.9% in the testing sample.

Table 3. The confusion matrix and predictive measures of the logistic regression, XGBoost with a tree booster and XGBoost with a linear booster for the testing and training data sets.

Testing Data Set			
Predictive Measures	**Logistic Regression**	**XGBoost (Tree Booster)**	**XGBoost (Linear Booster)**
$Y_i = 0, \hat{Y}_i = 0$	524	692	516
$Y_i = 1, \hat{Y}_i = 0$	38	58	38
$Y_i = 0, \hat{Y}_i = 1$	243	75	251
$Y_i = 1, \hat{Y}_i = 1$	25	5	25
Sensitivity	0.3968	0.0790	0.3968
Specificity	0.6831	0.9022	0.6728
Accuracy	0.6614	0.8397	0.6518
RMSE	0.2651	0.2825	0.2651
Training Data Set			
Predictive Measures	**Logistic Regression**	**XGBoost (Tree Booster)**	**XGBoost (Linear Booster)**
$Y_i = 0, \hat{Y}_i = 0$	1030	1794	1030
$Y_i = 1, \hat{Y}_i = 0$	55	0	55
$Y_i = 0, \hat{Y}_i = 1$	775	11	775
$Y_i = 1, \hat{Y}_i = 1$	77	132	77
Sensitivity	0.5833	1.0000	0.5833
Specificity	0.5706	0.9939	0.5706
Accuracy	0.5715	0.9943	0.5715
RMSE	0.2508	0.0373	0.2508

The threshold used to convert the continuous response into a binary response is the mean of the outcome variable. The authors performed the calculations.

It cannot be concluded from the foregoing, however, that XGBoost has a poor relative predictive capacity. Model-tuning procedures have not been incorporated in Table 3. However, tuning offers the possibility of improving the predictive capacity by modifying some specific parameter estimates.

[5] This is not surprising because XGBoost (linear) is a combination of linear probability models.

The following are some of the possible tuning actions that could be taken: Fixing a maximum for the number of branches of the tree (maximum depth); establishing a limited number of iterations of the boosting; or fixing a number of subsamples in the training sample. The `xgboost` package in R denotes these tuning options as general parameters, booster parameters, learning task parameters, and command line parameters, all of which can be adjusted to obtain different results in the prediction.

Figure 2 shows the ROC curve obtained using the three methods on the training and testing samples. This study confirmed that the logistic regression and XGBoost (linear) have a similar predictive performance. The XGBoost (tree) presented an outstanding AUC in the case of the training sample, and the same value as the logistic regression in the testing sample. However, as discussed in Table 3, it failed to maintain this degree of sensitivity when this algorithm is used with new samples.

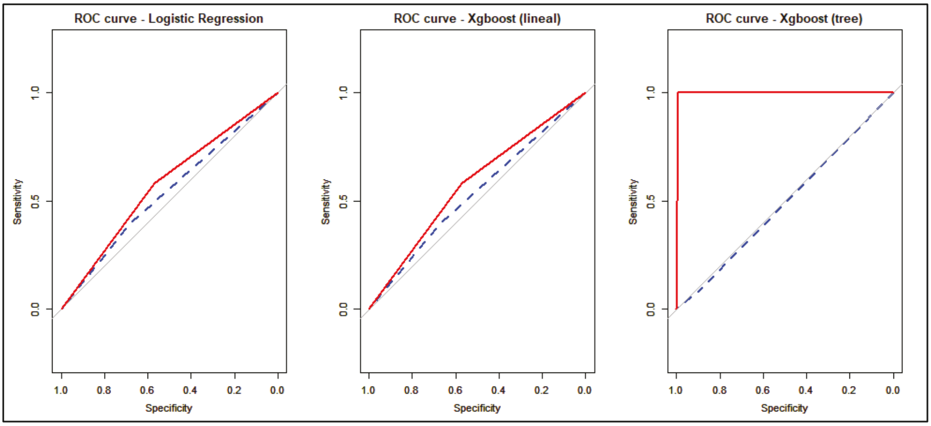


Figure 2. The receiver operating characteristics (ROC) curve obtained using the three methods on the training and testing samples. The red solid line represents the ROC curve obtained by each method in the training sample, and the blue dotted line represents the ROC curve obtained by each method in the testing sample. The area under the curve (AUC) is 0.58 for the training sample (T.S) and 0.49 for the testing sample (Te.S) when logistic regression is used; 0.58 for the T.S and 0.53 for the Te.S when XGBoost (linear booster) is used; and, 0.997 for the T.S and 0.49 for the Te.S when the XGBoost (tree booster) is used.

4.3. Correcting the Overfitting

One of the most frequently employed techniques for addressing the overfitting problem is regularization. This method shrinks the magnitude of the coefficients of the covariates in the modelling as the value of the regularization parameter increases.

In order to determine whether the XGBoost (tree booster) can perform better than the logistic regression model, a simple sensitivity analysis of the regularization parameters was proposed. In so doing, the evolution of the following confusion matrix measures was evaluated: Accuracy, sensitivity and specificity—according to some given regularization parameter values for the training and the testing sample—and, finally, the regularization parameter was chosen that gives the highest predictive measures in the training and testing samples.

Two regularization methods were considered. First, the L2 (Ridge) was considered, which is Chen and Guestrin (2016) original proposal and takes the $l2$-norm of the leaf weights. It has a parameter λ that is multiplied to the $l2$-norm. Second, the L1 (Lasso) method was considered, which is an additional implementation possibility of the `xgboost` package in R that takes the $l1$-norm of the leaf weights. It has a parameter α that is multiplied to the $l1$-norm. Consequently, λ and calibrated the regularization term in (4). For simplicity, no tree pruning was implemented, so $\mu = 0$ in (4).

The values of α and λ should be as small as possible, because they add bias to the estimates, and the models tend to become underfitted as the values of the regularization parameters become larger. For this reason, their changes were evaluated in a small interval. Figure 3 shows the predictive measures for the testing and training samples according to the values of α when the L1 regularization method was implemented. When $\alpha = 0$, exactly the same predictive measure values as in Table 3 (column 3) were obtained because the objective function had not been regularized. As the value of α increased, the models' accuracy and sensitivity values fell sharply—to at least $\alpha \simeq 0.06$ in the training sample. In the testing sample, the fall in these values was not as pronounced. However, when α was lower than 0.06, the specificity performance was the lowest of the three measures. Moreover, selecting an α value lower than 0.05 resulted in higher accuracy and sensitivity measures, but lower specificity. In contrast, when α equaled 0.06 in the testing sample, the highest specificity level of 0.5079 was obtained, with corresponding accuracy and sensitivity values of 0.5892 and 0.5988, respectively. In the training sample, when $\alpha = 0.06$ the specificity, accuracy and sensitivity were: 0.7227, 0.6086, and 0.6000, respectively. As a result, when was fixed at 0.06, the model performed similarly in both the testing and training samples.

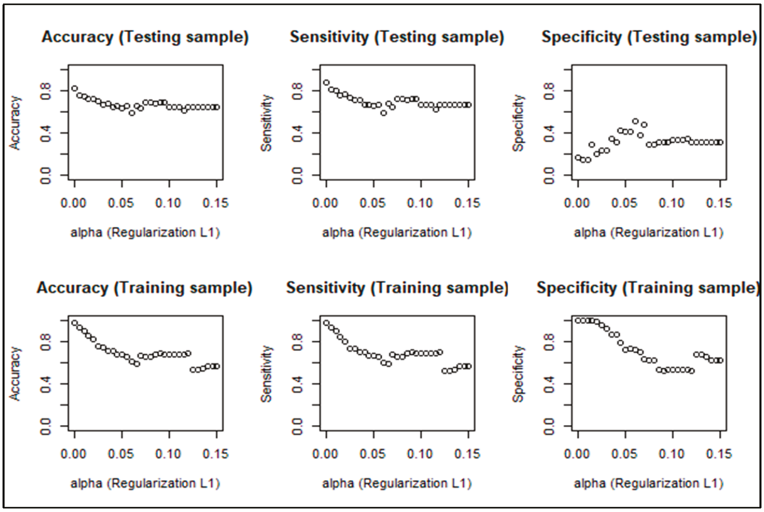


Figure 3. The predictive measures according to α. L1 method applied to the training and testing samples.

Thus, with the L1 regularization method ($\alpha = 0.06$), the new model recovered specificity, but lost some sensitivity when compared with the performance of the first model in Table 3, for which no regularization was undertaken. Thus, the authors concluded that $\alpha = 0.06$ which can be considered as providing the best trade-off between correcting for overfitting while only slightly reducing the predictive capacity.

Figure 4 shows the predictive measures for the testing and training samples according to the values of λ when the L2 regularization method is implemented. From $\lambda = 0$ to $\lambda = 0.30$. all predictive measures were approximately 100% in the training sample. However, very different results were recorded in the testing sample. Specifically, accuracy and sensitivity fell slowly, but specificity was low—there being no single λ that made this parameter exceed at least 20%. Therefore, no λ could help improve specificity in the testing sample. The L2 regularization method did not seem to be an effective solution to correct the problem of overfitting in our case study data set.

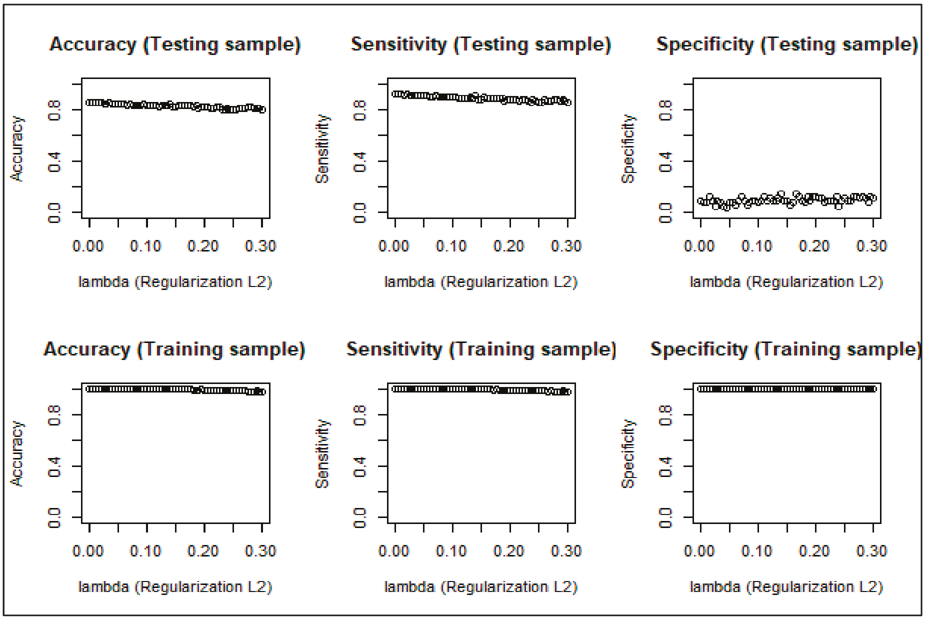


Figure 4. The predictive measures according to λ. L2 method applied to the training and testing samples.

The difference in outcomes recorded between the L1 and L2 regularization approaches might also be influenced by the characteristics of each regularization method. Goodfellow et al. (2016) and Bishop (2007) explained that L1 penalizes the sum of the absolute value of the weights, and that it seems to be robust to outliers, has feature selection, provides a sparse solution, and is able to give simpler but interpretable models. In contrast, L2 penalizes the sum of the square weights, has no feature selection, is not robust to outliers, is more able to provide better predictions when the response variable is a function of all input variables, and is better able to learn more complex models than L1.

4.4. Variable Importance

Variable importance or feature selection is a technique that measures the contribution of each variable or feature to the final outcome prediction based on the Gini impurity. This method is of great relevance in tree models because it helps identify the order in which the leaves appear in the tree. The tree branches (downwards) begin with the variables that have the greatest effect and end with those that have the smallest effect (for further details see, for example, Kuhn and Johnson 2013).

Table 4 shows the three most important variables for each method. The two agree on the importance of the percentage of total kilometers travelled in urban areas as a key factor in predicting the response variable. Total kilometers driven and age only appeared among the top three variables in the case of logistic regression, while the percentage of kilometers travelled over the speed limits and the percentage of kilometers driven at night appeared among the most important variables in the case of the XGBoost method.

Table 4. Variable Importance. The most relevant variables of the different methods.

Level of Importance	Logistic Regression	XGBoost (Tree Booster)
First	percentage of total kilometers travelled in urban areas	percentage of kilometers above the mandatory speed limits
Second	age	percentage of total kilometers travelled in urban areas
Third	total kilometers	percentage of total kilometers travelled at night

5. Conclusions

XGBoost, and other boosting models, are dominant methods today among machine-learning algorithms and are widely used because of their reputation for providing accurate predictions. This novel algorithm is capable of building an ensemble model characterized by an efficient learning method that seems to outperform other boosting-based predictive algorithms. Unlike the majority of machine learning methods, XGBoost is able to compute coefficient estimates under certain circumstances and, therefore, the magnitude of the effects can be studied. The method allows the analyst to measure not only the final prediction, but also the effect of the covariates on a target variable at each iteration of the boosting process, which is something that traditional econometric models (e.g., generalized linear models) do in one single estimation step.

When a logistic regression and XGBoost compete to predict the occurrence of accident claims without model-tuning procedures, the predictive performance of the XGBoost (tree booster) was much higher than the logistic regression in the training sample, but considerably poorer in the testing sample. Thus, a simple regularization analysis has been proposed here to correct this problem of overfitting. However, the improvement in predictive performance of the XGBoost following this regularization was similar to that obtained by the logistic regression. This means additional efforts have to be taken to tune the XGBoost model to obtain a higher predictive performance without overfitting the data. This might be considered as the trade-off between obtaining a better performance, and the simplicity it provides for interpreting the effect of the covariates.

Based on our results, the classical logistic regression model can predict accident claims using telematics data and provided a straightforward interpretation of the coefficient estimates. Moreover, the method offered a relatively high predictive performance considering that only two coefficients were significant at the 90% confidence level. These results are not bettered by the XGBoost method.

When the boosting framework of XGBoost is not based on a linear booster, interpretability becomes difficult, as a model's coefficient estimates cannot be calculated. In this case, variable importance can be used to evaluate the weight of the individual covariates in the final prediction. Here, different conclusions were obtained for the two methods employed. Thus, given that the predictive performance of XGBoost was not much better than the logistic regression, even after careful regularization, the authors concluded that the new methodology needs to be adopted carefully, especially in a context where the number of event responses (accident) is low compared to the opposite response (no accident). Indeed, this phenomenon of unbalanced response is attracting more and more attention in the field of machine learning, because it is known that machine learning algorithms do not work well in datasets with imbalanced responses (He and Garcia 2008). XGBoost might perform better in other problems, especially when the number of events and no events are balanced. The reputation of XGBoost (tree booster) may be due to its capacity of accuracy. In our case study, XGBoost has proven the highest accuracy in the testing and training data sets, but it does not seem to be effective for sensitivity.

For future work, it would be interesting to see bigger datasets with thousands of explanatory variables to conclude whether or not XGBoost has better predictive performance than a regularized version of logistic regression. Similarly, it would also be interesting to see comparative studies for other machine learning approaches using this dataset, including but not limited to neural network approaches.

Author Contributions: All authors contributed equally to the conceptualization, methodology, software, validation, data curation, writing—review and editing, visualization, and supervision of this paper.

Acknowledgments: We thank the Spanish Ministry of Economy, FEDER grant ECO2016-76203-C2-2-P. The second author gratefully acknowledges financial support received from ICREA under the ICREA Academia Program.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

R Code

```
# Loading data
    load(“data.Rdata”)
    x<-data
# Training and test data sets
# We divide 70% of the data set as training, and 30% as testing
    library(caret)
    part<-createDataPartition(x$Y,p = 0.70, list = F)
    train.set<-x[part,] # training data set
    train.set<-train.set()[−1]
    testing.set<- x[-part,] # testing data set
    testing.set<-testing.set()[−1]
## First Method: Logistic Regression
    logistic1 <- glm(factor(train.set$Y) ~ x2+I(x2^2)+x3+x4+factor(x1)+x5+x6+x7+x8,
    data = train.set,family = binomial(link = ‘logit’))
    summary(logistic1)
# Predicting the output with the testing data set
    predicted.log.test <- predict(logistic1,testing.set, type = ‘response’)
# Predicting the output with the training data set
    predicted.log1.train<- predict(logistic1,train.set, type = ‘response’)
# Variable Importance
    varImp(logistic1)
## Second Method: XGBoost (tree booster)
    library(xgboost)
    library(Matrix)
# Function xgboost requires sparsing data first
    sparse_xx.tr<- sparse.model.matrix(Y ~ x2+I(x2^2)+x3+x4+factor(x1)+x5+x6+x7+x8, data = train.set)
    sparse_xx.te<- sparse.model.matrix(Y ~ x2+I(x2^2)+x3+x4+factor(x1)+x5+x6+x7+x8,
    data = testing.set)
    xgboost_reg <- xgboost(data = sparse_xx.tr, label = train.set$Y, objective = “binary:logistic”,
    nrounds = 100, verbose = 1)
# Predicting the output with testing data set
    pred.xgboost.test<- predict(xgboost_reg,sparse_xx.te, outputmargin = F)
# Predicting the output with training data set
    pred.xgboost.train<-predict(xgboost_reg,sparse_xx.tr, outputmargin = F)
# Variable Importance
    importance <- xgb.importance(feature_names = sparse_xx.tr@Dimnames[(2)],
    model = xgboost_reg)
```

References

Ayuso, Mercedes, Montserrat Guillén, and Ana María Pérez-Marín. 2014. Time and distance to first accident and driving patterns of young drivers with pay-as-you-drive insurance. *Accident Analysis and Prevention* 73: 125–31. [CrossRef] [PubMed]

Ayuso, Mercedes, Montserrat Guillén, and Ana María Pérez-Marín. 2016a. Using GPS data to analyse the distance travelled to the first accident at fault in pay-as-you-drive insurance. *Transportation Research Part C* 68: 160–67. [CrossRef]

Ayuso, Mercedes, Montserrat Guillén, and Ana María Pérez-Marín. 2016b. Telematics and gender discrimination: some usage-based evidence on whether men's risk of accident differs from women's. *Risks* 4: 10. [CrossRef]

Bishop, Christopher M. 2007. Pattern recognition and machine learning. *Journal of Electronic Imaging* 16: 049901. [CrossRef]

Boucher, Jean-Philippe, Steven Côté, and Montserrat Guillen. 2017. Exposure as duration and distance in telematics motor insurance using generalized additive models. *Risks* 5: 54. [CrossRef]

Chen, Tianqi, and Carlos Guestrin. 2016. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York: ACM, pp. 785–94. [CrossRef]

De Boer, Pieter-Tjerk, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. 2005. A tutorial on the Cross Entropy Method. *Annals of Operations Research* 134: 19–67. [CrossRef]

Dietterich, Thomas G., Pedro Domingos, Lise Getoor, Stephen Muggleton, and Prasad Tadepalli. 2008. Structured machine learning: The next ten years. *Machine Learning* 73: 3–23. [CrossRef]

Elliott, Graham, and Allan Timmermann. 2003. *Handbook of Economic Forecasting*. Amsterdam: Elsevier.

Friedman, Jerome, Trevor Hastie, and Robert Tibshirani. 2001. *The Elements of Statistical Learning*. New York: Springer.

Gao, Guangyuan, and Mario V. Wüthrich. 2018. Feature extraction from telematics car driving heatmaps. *European Actuarial Journal* 8: 383–406. [CrossRef]

Gao, Guangyuan, and Mario V. Wüthrich. 2019. Convolutional neural network classification of telematics car driving data. *Risks* 7: 6. [CrossRef]

Gao, Guangyuan, Shengwang Meng, and Mario V. Wüthrich. 2019. Claims frequency modeling using telematics car driving data. *Scandinavian Actuarial Journal* 2019: 143–62. [CrossRef]

Gomez-Verdejo, Vanessa, Jeronimo Arenas-Garcia, Manuel Ortega-Moral, and Anıbal R. Figueiras-Vidal. 2005. Designing RBF classifiers for weighted boosting. *IEEE International Joint Conference on Neural Networks* 2: 1057–62. [CrossRef]

Goodfellow, Ian, Bengio Yoshua, and Courville Aaron. 2016. *Deep Learning*. Chenai: MIT Press.

Greene, William. 2002. *Econometric Analysis*, 2nd ed. New York: Chapman and Hall.

Guillen, Montserrat, Jens Perch Nielsen, Mercedes Ayuso, and Ana M. Pérez-Marín. 2019. The use of telematics devices to improve automobile insurance rates. *Risk Analysis* 39: 662–72. [CrossRef] [PubMed]

Hastie, Trevor, Rob Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Prediction, Inference and Data Mining*. New York: Springer.

He, Haibo, and Edwardo A. Garcia. 2008. Learning from imbalanced data. *IEEE Transactions on Knowledge & Data Engineering* 9: 1263–84. [CrossRef]

Huang, Jianhua Z., and Lijian Yang. 2004. Identification of non-linear additive autoregressive models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 66: 463–77. [CrossRef]

Hultkrantz, Lars, Jan-Eric Nilsson, and Sara Arvidsson. 2012. Voluntary internalization of speeding externalities with vehicle insurance. *Transportation Research Part A: Policy and Practice* 46: 926–37. [CrossRef]

Ivanov, Valentin K., Vladimir V. Vasin, and Vitalii P. Tanana. 2013. *Theory of Linear Ill-Posed Problems and Its Applications*. Zeist: VSP.

James, Gareth, Daniela Witten, Trevor Hastie, and Robert Tibshirani. 2013. *An Introduction to Statistical Learning*. New York: Springer, vol. 112, p. 18.

Kuhn, Max, and Kjell Johnson. 2013. *Applied Predictive Modeling*. New York: Springer, vol. 26.

Lee, Simon, and Katrien Antonio. 2015. Why High Dimensional Modeling in Actuarial Science? Paper presented at Actuaries Institute ASTIN, AFIR/ERM and IACA Colloquia, Sydney, Australia, August 23–27. Available online: https://pdfs.semanticscholar.org/ad42/c5a42642e75d1a02b48c6eb84bab87874a1b.pdf (accessed on 8 May 2019).

Lee, Simon CK, and Sheldon Lin. 2018. Delta boosting machine with application to general insurance. *North American Actuarial Journal* 22: 405–25. [CrossRef]

Natekin, Alexey, and Alois Knoll. 2013. Gradient boosting machines, a tutorial. *Frontiers in Neurorobotics* 7: 21. [CrossRef] [PubMed]

McCullagh, Peter, and John Nelder. 1989. *Generalized Linear Models*, 2nd ed. New York: Chapman and Hall.

Pérez-Marín, Ana M., and Montserrat Guillén. 2019. Semi-autonomous vehicles: Usage-based data evidences of what could be expected from eliminating speed limit violations. *Accident Analysis and Prevention* 123: 99–106. [CrossRef] [PubMed]

Schapire, Robert E., and Yoav Freund. 2012. *Boosting: Foundations and Algorithms*. Cambridge: MIT Press.

Steinwart, Ingo, and Andreas Christmann. 2008. *Support Vector Machines*. New York: Springer Science & Business Media.

Tikhonov, Andrej-Nikolaevich, and Vasiliy-Yakovlevich Arsenin. 1977. *Solutions of Ill-Posed Problems*. New York: Wiley.

Verbelen, Roel, Katrien Antonio, and Gerda Claeskens. 2018. Unraveling the predictive power of telematics data in car insurance pricing. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 67: 1275–304. [CrossRef]

Wüthrich, Mario V. 2017. Covariate selection from telematics car driving data. *European Actuarial Journal* 7: 89–108. [CrossRef]

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).

risks

MDPI

Article

On the Validation of Claims with Excess Zeros in Liability Insurance: A Comparative Study

Marjan Qazvini

Department of Actuarial Mathematics and Statistics, School of Mathematical and Computer Sciences, Heriot-Watt University Malaysia, 62200 Putrajaya, Wilayah Persekutuan Putrajaya, Malaysia; m.qazvini@hw.ac.uk

Received: 27 May 2019; Accepted: 19 June 2019; Published: 30 June 2019

Abstract: In this study, we consider the problem of zero claims in a liability insurance portfolio and compare the predictability of three models. We use French motor third party liability (MTPL) insurance data, which has been used for a pricing game, and show that how the type of coverage and policyholders' willingness to subscribe to insurance pricing, based on telematics data, affects their driving behaviour and hence their claims. Using our validation set, we then predict the number of zero claims. Our results show that although a zero-inflated Poisson (ZIP) model performs better than a Poisson regression, it can even be outperformed by logistic regression.

Keywords: validation; generalised linear modelling; zero-inflated poisson model; telematics

1. Introduction

There are two main types of machine learning: (i) predictive or supervised learning in which the machine trains data and learns the relationship between inputs and outputs and (ii) descriptive and unsupervised learning in which machine uses the inputs and discovers the outputs (Murphy 2012). Classification and regression are two supervised learning approaches which are well-known in general insurance. One of the objectives of the insurance companies is to charge premiums which is commensurate with the risk characteristics of their policyholders and for this, they classify the policyholders into homogeneous groups according to, say, age, sex, type of policy, subscription to telematics-based insurance pricing (see, Section 3), etc. Regression analysis and its extensions such as generalised linear modelling (GLM) are strong tools in insurance pricing. Unlike regression models, GLM is not constrained to a normal distribution and can be applied to any distribution from an exponential family. For example, a logistic regression model handles binary responses and thus is suitable for a Bernoulli distribution and a Poisson regression model applies to count data and deals with discrete random variables. GLM has long been used in actuarial practice to model claims amounts and claims frequency in the insurance portfolio (Haberman and Renshaw 1996; McCullagh and Nelder 1998).

In this study, we consider motor third party liability (MTPL) insurance. One of the problems in modelling claims frequency in this class of insurance is the number of zero claims and building a model that can capture all these zero claims. Zero claims in MTPL does not necessarily mean that there has been no accident during the term of a policy, rather it means that there has been no reported accident to the insurance company. This particularly happens under a no claim discount (NCD) system as some policyholders, known as *bonus hunger*, prefer to benefit from a discount by not reporting a claim. Another problem which is related to the previous one is the problem of *over-dispersion*. In a Poisson regression model, claims are distributed according to a Poisson distribution with equal mean and variance. Therefore, to build an appropriate model we need to test our dataset for the presence of over-dispersion (Peruman-Chaney et al. 2013; Wilson and Einbeck 2018). Binomial regression, negative binomial (NB) regression and zero-inflated Poisson (ZIP) model are techniques that can handle over and under dispersed data with the latter being able to distinguish between structured and unstructured zeros. Lambert (1992) considers a ZIP model where the probability of only possible observation, i.e., 0 and the parameter of a Poisson distribution depend on some covariates. Lambert (1992) applies this technique to model the number of defects in manufacturing. Since then, this model has been applied in different settings including insurance pricing. For example, Lee et al. (2002) use this model to analyse the impact of lifestyle and motivations on car crashes involving young drivers in Australia. Yip and Yau (2005) use ZIP to model claims frequency in car insurance. They compare different types of zero-inflated count models and conclude that a zero-inflated double Poisson regression model is a good fit for their dataset. Boucher et al. (2007) compare zero-inflated, hurdle and compound frequency models and conclude that the bonus rate is an important factor for policyholders to report the claim. In another study, Boucher et al. (2009) consider the problem of bonus hunger and construct a ZIP model to distinguish between the distribution of the number of claims and the number of accidents.

Model fitting and the selection of risk factors can be challenging in some cases. There are some papers that consider these problems. For example, Tang et al. (2014) propose a method to determine the variables in a ZIP model. They combine EM algorithm and adaptive LASSO and find that their technique performs better for the non-inflated part of the ZIP regression. Liu and Pitt (2017) also apply LASSO and ridge regression to address this issue in a bivariate negative binomial regression model. See, also, Cantoni and Auda (2018), Chowdhury et al. (2019) and Chen et al. (2019) among others.

The impact of mileage as a risk factor is considered by Lemaire et al. (2015). They conclude that annual mileage is a powerful predictor of the number of claims at-fault. Tselentis et al. (2017) provide a review of some Usage-based motor insurance (UBI) including Pay-as-you-drive (PAYD), Pay-how-you-drive (PHYD) and Pay-at-the-pump (PATP). PATP is a pricing method that considers fuel consumption as a rating factor but did not get enough attention from researchers. These new pricing methods require telematics data. In recent years, there is much research on telematics data and mileage based (MB) insurance. Boucher et al. (2017) apply generalised additive models and consider both time and mileage in insurance pricing. See the following papers on the relevance of including the mileage as a risk factor (Ayuso et al. 2019; Guillen et al. 2019; Verbelen et al. 2018).

In addition to regression analysis, neural network, decision tree, random forest and boosting algorithms such as XGBoost, etc., are other machine learning techniques that can be applied to model claims frequency and insurance pricing. However, although these models have good predictive power, unlike regression models, it is difficult to interpret their parameters and their computation time is long. Weerasinghe and Wijegunasekara (2016) study neural network, decision tree and multinomial logistic regression models. Their results show that the neural network has the best predictive performance among the three models. However, they state that to understand the relationship between independent and dependent variables, the logistic regression is the best model. Fauzan and Murfi (2018) compare XGBoost, neural network and random forest models and find that in terms of the Gini index, XGBoost is a more accurate algorithm. See, also, Spedicato et al. (2018) and Gao et al. (2019) and the references therein.

In this study, we consider the classical Poisson and logistic regression and compare our findings with a ZIP model. We divide our dataset into training and validation (hold-out) set to predict the number of zero claims. This paper is organised as follows. In the next section, we present models and notation. Section 3 discusses our dataset. In Section 4, we build our models and in Section 5 we test their validation. Finally, Section 6 concludes.

2. Methodology and Notation

Risk classification is an important concept in general insurance pricing. An insurance company tries to determine the insurance premium according to risk characteristics of policyholders such as age, sex, type of policy and car model, etc. Regression analysis is a well-known technique to incorporate such risk (rating) factors. In this section, we review Poisson regression, Logistic regression and ZIP model.

Let $y_i \in \{0, 1, 2, \dots\}$ be a dependent or response variable such as number of claims, for $i = 1, \dots, n$ that follows a Poisson distribution with parameter λ_i. Assuming a log link function and that λ_i is a linear combination of rating factors $\beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik}$ we have

$$E[y_i|x_i] = \lambda_i = \exp\{\beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik}\}, \quad y_i \sim \text{Pois}(\lambda_i) \text{ for } i = 1, 2, \dots, n. \tag{1}$$

When we consider the average number of claims for each policyholder, we need to specify a unit measure or exposure. We cannot expect two policyholders with the same risk characteristics, but different terms, to be equally risky. Normally, the length of coverage is considered as an exposure. However, in recent years, it is argued that even if policyholders join at different times, some may drive fewer distances than others. Therefore, when such information is available as in telematics data, mileage travelled is considered as a more appropriate exposure (Guillen et al. 2019). In our study, all policyholders are under observation for one year and thus the exposure for each policyholder is 1.

We use logistic regression when $y_i \in \{0, 1\}$ is a binary, also called dichotomous variable. In that case,

$$E[y_i|x_i] = \pi_i(x) = g\left(\beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik}\right)$$

where g is a logistic link function to ensure that π_i is between 0 and 1. Hence

$$\pi_i(x) = \frac{\exp\{\beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik}\}}{1 + \exp\{\beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik}\}} \tag{2}$$

or, more commonly

$$\log\left(\frac{\pi_i(x)}{1 - \pi_i(x)}\right) = \beta_0 + \beta_1 x_{i1} + \cdots + \beta_k x_{ik}.$$

In this paper, we use logistic regression to answer the question: "What is the probability of a claim $(y_i = 1)$ and zero claims $(y_i = 0)$ for a given policyholder with particular risk characteristics?"

When the mean and variance of the underlying population is not equal, the assumption of a Poisson distribution is not suitable and a better candidate is a distribution that can allow for over/under dispersion such as a binomial or NB distribution. However, sometimes we deal with a large number of zeros in our dataset. For example, we see in the next section that many policyholders have zero claims, which does not necessarily mean that they were involved in no accidents, but they are low risk. In such cases, we can apply a ZIP model which is a mixture of a point mass at zero, also called structural zeros, and another claims frequency distribution, such as a Poisson or NB, which can be written as

$$\Pr(y_i = j) = \begin{cases} \pi_i + (1 - \pi_i)\Pr(y_i = 0) & j = 0 \\ (1 - \pi_i)\Pr(y_i = j) & j = 1, 2, \dots \end{cases} \tag{3}$$

where π_i is given by Equation (2) and denotes the probability of zeros when zero is the only possible observation. In a ZIP model, y_i follows a Poisson distribution with parameters being given by Equation (1).

We can easily implement these models in R and the codes are provided in Appendix A (Frees et al. 2014, 2016).

3. Data

We use datasets provided by the French Institute of Actuaries for the 2017 pricing game, which is based on French MTPL insurance. The dataset is available in Package 'CASdatasets' by Dutang and Dutang and Charpentier (2019) and to the best of the author's knowledge, this is the first time it is used in a study. The dataset contains some information about the new pricing strategy of the company. The policyholders were given a choice whether they would like to join a new mileage-based

(MB) pricing system or not. We would like to see how policyholders' perception regarding this new system affects their driving behaviour and hence their number of claims. There are two types of datasets: (i) underwriting and (ii) claims dataset. Underwriting datasets are available for three years, whereas claims dataset is only publicly available for year 0. Therefore, we only use data from year 0. After merging claims and underwriting datasets, we randomly split our data into training and validation sets with 60% being in training and 40% in the validation set. As some policyholders have more than one car, we assume that each policy covers only one car and therefore consider the number of policies and claims per policy rather than claims per policyholder. We have 100,000 policies (rows in underwriting dataset) and 12,654 policies with claims (rows in claims dataset after consolidation). Table 1 shows the variables we use in our study. In addition to these variables, information about *Insee town code*, *make and model*, *marketing duration* and *age of driving license* are also provided. However, we do not take into account these variables as, for example, there is a considerable number of policies with 113 years for driving license age which is not reasonable.

In Table 1 policy ID refers to the combination of the vehicle ID and policyholder ID. In this study, we have 100,000 policy ID. Bonus coefficient is the percentage of the full premium that policyholders pay allowing for their claims experience and the allocated discount. There are four types of coverage available: Maxi, Median 2, Median 1 and Mini. The time from the last policy alteration, such as the inclusion of a new driver, is represented by situation duration. Payments can be made annually, semi-annually, quarterly and monthly. As it is usual for the liability insurance, some of the claims amounts are negative[1]. Therefore, we set all claims amounts of less than 30 equal to zero (Ferreira and Minikel 2012; Frees et al. 2014).

Table 1. Variables in our datasets.

Control	Policy	Driver (1 and 2)	Vehicle	Response
policy ID	bonus coefficient type of coverage duration situation duration payment frequency subscription to MB usage	driver 2? age gender	age cylinder din power fuel type max speed type value weight	number of claims

Subscription to mileage-based (MB) policy refers to a new scheme in which one of the main risk factors is the travel distance and policyholders are charged based on their mileage, also known as PAYD scheme. Policy Usage includes WorkPrivate, Retired, Professional and AllTrips. If a policy covers two drivers, age and gender are provided for both drivers. Different features of the car including age, engine power (represented by Din), fuel type, max speed (provided by manufacturing company), type—Tourism and Commercial, value and weight are provided and will be used as rating factors. In this study, we only consider the number of claims as a dependent variable.

We now provide some explanatory analysis based on the training set. The minimum policy term in our dataset is one year, which means all these policies have been under observation for at least one year. Since claims have occurred in Year 0, we consider car years or earned exposure of one year for all policies. The maximum claims number is 6, the oldest policyholder is 103 years old and the oldest car is 66 years old. Table 2 presents mean and standard deviation of our numerical explanatory variables for all policies, policies without claims and policies with at least one claim based on the training set. In order to examine which variables are considerably different in the group of policies with claims and the group of policies without claims and hence are effective on the frequency of claims, we can apply

[1] This happens due to subrogation rights of the insurer.

Mann-Whitney test. The Mann-Whitney test is a nonparametric test of the null hypothesis that it is equally likely that a randomly selected value from one sample is less than or greater than a randomly selected value from a second sample. The Mann-Whitney test shows that the difference in the mean for all these variables is statistically significant with p-value < 0.0001, except for policy duration and driver age 2 with p-values 0.001232 and 0.004252, respectively.

Table 2. The mean and standard deviation of numerical variables in the training set.

Variables	All Policies		Policies without Claims		Policies with Claims	
	Mean	SD	Mean	SD	Mean	SD
Policy duration	11.09	8.56	11.13	8.57	10.78	8.56
Policy duration since the last change	2.74	2.36	2.78	2.41	2.40	2.36
Driver age 1	54.65	14.86	54.75	14.86	53.84	14.86
Driver age 2	46.93	16.21	47.06	16.19	46.04	16.21
Vehicle value	18,086	8677.92	17,858	8618.47	19,894	8677.92
Vehicle age	9.56	7.03	9.84	7.19	7.30	7.03
Engine cylinder	1645	460.59	1,639	464.05	1,696	460.59
Speed	170.71	23.48	170.13	23.69	175.31	23.48
Weight	1171.59	288.39	1164.36	288.68	1228.89	288.39
Motor power (din)	91.43	34.41	90.58	34.35	98.23	34.41

Figure 1 shows the distribution of the number of claims. We can observe that zero claims form a large part of our portfolio.

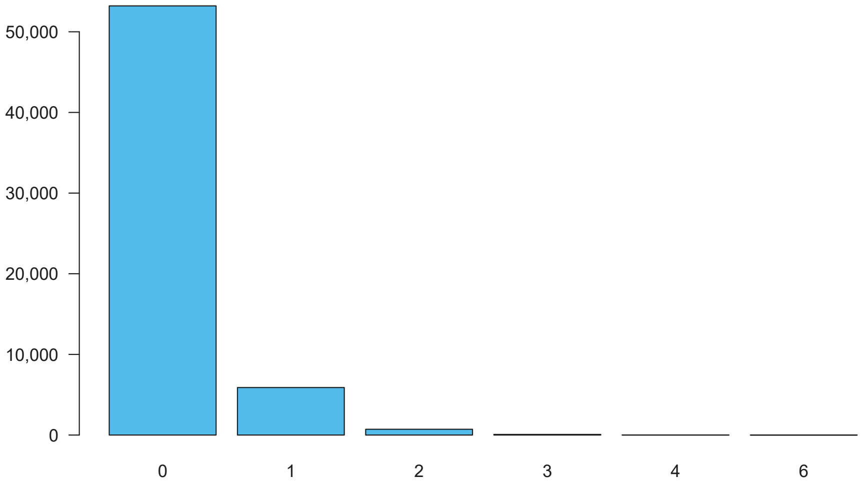


Figure 1. Distribution of claims frequency.

Figure 2 illustrates how policies are distributed across categorical variables. As we can see, most of our policies cover one driver and most of the drivers are men aged between 51 and 70. Our policyholders prefer Maxi and drive tourism cars for work and private purposes. Most of them pay annually and are distributed almost evenly across monthly and biannual payment categories. They have not registered for MB scheme and they use diesel with very few of them using a hybrid car. Next, we see how claims are distributed across categorical variables.

Table 3 presents the distribution of the number of claims across different categories. For the variable *policy usage*, although *professional* usage forms a small portion of our portfolio, claims under *professional* group is more than *private* and *retired* groups. However, from Figure 3 *professional* and *retired* groups have almost the same median loss and except for *all trips* we can see little difference among policies in this group. Under this insurance, the most comprehensive protection is provided by *maxis* and as can be expected this may lead to moral hazard. We can see there are more claims under *maxis* than under other types of coverage. The order of coverage is *maxis*, *median 2*, *median 1* and *mini* and unsurprisingly, the percentage of claims reduces in the same order. Under *mini*, 97.39% of the policies have made zero claims. Perhaps lower coverage is a motivation for policyholders to take more precautious measures. Figure 3 shows the effect of policy coverage on the amounts of claims and as we can see this will be an effective covariate in our model. From Table 3 those policyholders who were willing to subscribe to *MB plan* are less likely to have an accident. Figure 3 shows that the subscribers are less dispersed than those who have not subscribed. From the regulatory point of view, *gender* cannot be used as a discriminatory factor. In fact, we can see there is no considerable difference between *male*'s and *female*'s number of claims. In Figure 2 the least favourable *payment frequency* is *quarterly* payment, but we do not see considerable differences in claim numbers and amounts for different categories of payments. A large number of policies provide coverage only for one driver, but policies with two drivers have a slightly greater chance of making claims. The *age* of the first driver ranges from 19 to 103. We classify the policyholders in different age groups as 18–30, 31–50, 51–70, 71–85 and 85+. Most of the policyholders are in the range 51–70 and the next largest group is between 31 and 50. Both Table 3 and Figure 3 do not show a significant difference in claims frequency and claim amounts for different age categories and it seems that some categories can be combined together. In fact, in the next section we see that instead of these categories, we use *age* as a numerical covariate in our models as some categories are not statistically significant.

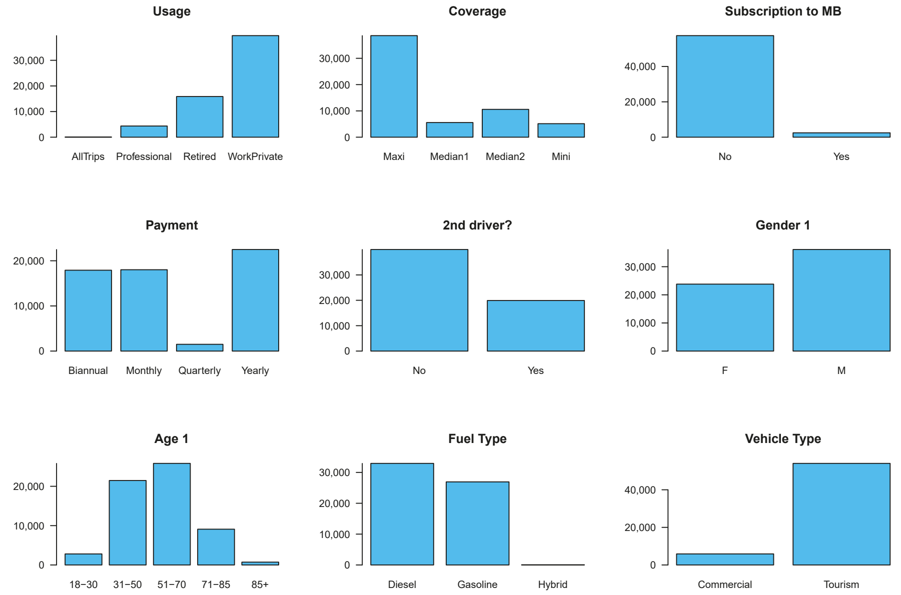


Figure 2. Distribution of policies according to categorical variables.

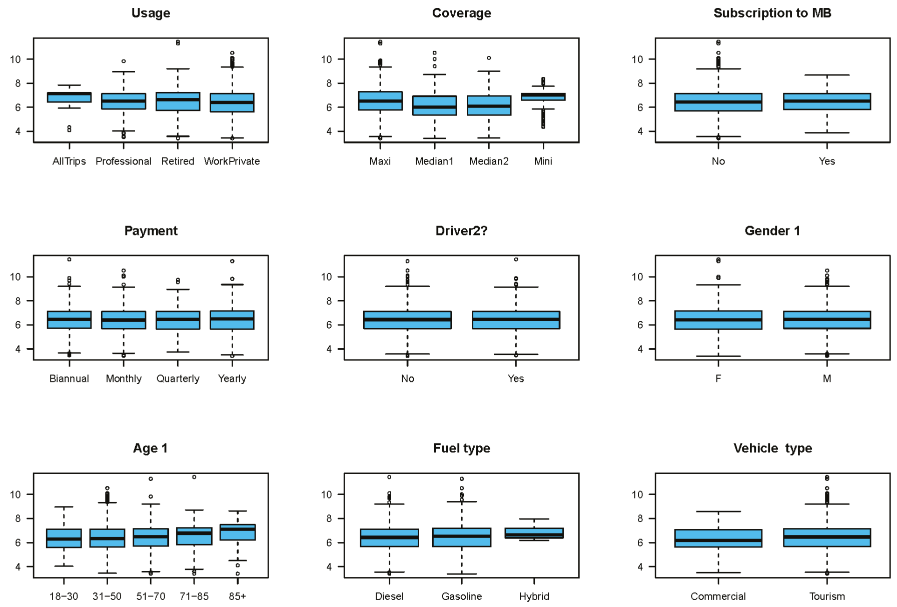


Figure 3. Distribution of log of claims amounts according to categorical variables.

Most of our policyholders drive *gasoline* cars and very few of them have *hybrid* cars.[2] According to Table 3, hybrid cars make more claims than gasoline and *diesel* cars. Most policies cover *tourism* cars and claims percentage made by this type of cars is more than *commercial* cars. Our initial analysis suggests that *payment frequency* and *gender* are not significant variables and therefore can be removed from our study. In the next section, we will see that they are indeed insignificant and are not included in our final models.

Table 3. Frequency of claims per categorical variables in the training set.

Variables	**Categories**	**Claim frequency**							**Total**
		0	**1**	**2**	**3**	**4**	**5**	**6**	
Policy usage	WorkPrivate	35,248	3, 877	450	49	7	0	1	39,632
		88.94%	9.78%	1.14%					
	Retired	14,193	1, 462	191	20	3	0	0	15,869
		89.44%	9.21%	1.20%					
	Professional	3,729	544	76	10	0	0	0	4, 359
		85.55%	12.48%	1.74%					
	All trips	41	10	1	0	0	0	0	52
		78.85%	19.23%	1.92%					
Policy coverage	Maxis	33,459	4, 489	600	70	9	0	1	38,628
		86.62%	11.62%	1.55%					
	Median 2	9,628	862	82	7	1	0	0	10,580
		91.00%	8.15%	0.78%					
	Median 1	5, 122	412	32	2	0	0	0	5, 568
		91.99%	7.04%	0.57%					
	Mini	5, 002	130	4	0	0	0	0	5, 136
		97.39%	2.53%	0.08%					

[2] According to the game document, hybrid cars were not popular at the time of collecting this dataset.

Table 3. *Cont.*

Variables	Categories	Claim frequency							Total
		0	1	2	3	4	5	6	
Subscription to MB	No	50,946	5, 714	693	76	10	0	1	57,440
		88.69%	9.95%	1.21%					
	Yes	2, 265	179	25	3	0	0	0	2, 472
		91.63%	7.24%	1.01%					
Payment	Yearly	20,094	2, 106	263	25	3	0	1	22,492
		89.34%	9.36%	1.17%					
	Biannual	15,930	1,746	199	29	3	0	0	17,907
		88.96%	9.75%	1.11%					
	Monthly	15,880	1, 875	234	23	4	0	0	18,016
		88.14%	10.41%	1.30%					
	Quarterly	1, 307	166	22	2	0	0	0	1, 497
		87.31%	11.09%	1.47%					
Policy with 2 drivers	No	35,675	3, 814	457	47	6	0	0	39,999
		89.19%	9.54%	1.14%					
	Yes	17,536	2,079	261	32	4	0	1	19,913
		88.06%	10.44%	1.31%					
Gender 1	Male	32,118	3, 501	433	52	4	0	0	36,108
		88.95%	9.70%	1.20%					
	Female	21,093	2, 392	285	27	6	0	1	23,804
		88.61%	10.05%	1.20%					
Age 1	18–30	2, 471	299	29	4	0	0	1	2, 804
		88.12%	10.66%	1.03%					
	31–50	18,961	2, 228	256	24	4	0	0	21,473
		88.30%	10.38%	1.19%					
	51–70	22,978	2, 479	322	40	3	0	0	25,822
		89.99%	9.60%	1.25%					
	71–85	8,154	822	105	9	3	0	0	9, 093
		89.67%	9.04%	1.15%					
	85+	647	65	6	2	0	0	0	720
		89.86%	9.03%	0.83%					
Vehicle fuel	Diesel	28,605	3, 783	475	54	7	0	1	32,925
		86.88%	11.49%	1.44%					
	Gasoline	24,565	2, 104	241	25	3	0	0	26,938
		91.19%	7.81%	0.89%					
	Hybrid	41	6	2	0	0	0	0	49
		83.67%	12.24%	4.08%					
Vehicle type	Tourism	47,891	5, 387	668	73	10	0	1	54,030
		88.64%	9.97%	1.24%					
	Commercial	5, 320	506	50	6	0	0	0	5, 882
		90.45%	8.60%	0.85%					
Total		53,211	5,893	718	79	10	0	1	59,912

4. Results

In this section, we use statistical software R and package “pscl” to build Poisson, logistic and ZIP models (Zeileis et al. 2008). Our purpose is to estimate the frequency and the probability of claims and compare our results with a ZIP model using Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC).

Table 4 presents three Poisson regression models with their estimated coefficients and their corresponding *p*-values. Model 1 is the full model where we consider all the variables from Section 3. However, according to pricing game document, there is a correlation between vehicle cylinder, weight, value, speed and power and in our dataset, some of the entries for weight, value and cylinder are missing. Therefore, we only incorporate speed and power into our models. We build Model 2 using the stepwise selection of variables that can be implemented in R. In Model 3 we only consider variables which are statistically significant at 0.05.

Table 4. Regression coefficient of Poisson models.

Coefficients	Model 1: All Variables		Model 2: Stepwise Selection		Model 3: Only Significant	
	Estimate	***p*-Value**	**Estimate**	***p*-Value**	**Estimate**	***p*-Value**
Intercept	−2.6883	<0.0001	−2.5645	<0.0001	−2.4729	<0.0001
Age 1	0.0048	0.0001	0.0048	0.0002	0.0036	0.0026
Age 2	−0.0034	0.0084	−0.0034	0.0087	−0.0033	0.0111
Female 1	0.0380	0.1251	0.0374	0.1285		
Driver2?	0.1790	0.0057	0.1781	0.0060	0.1717	0.0080
Situation duration	−0.0185	0.0078	−0.0185	0.0080	−0.0220	0.0013
Bonus	0.8677	<0.0001	0.8683	<0.0001	0.9125	<0.0001
Coverage(Med2)	−0.1854	<0.0001	−0.1847	<0.0001	−0.1877	<0.0001
Coverage(Med1)	−0.2282	<0.0001	−0.2280	<0.0001	−0.2306	<0.0001
Coverage(Mini)	−1.2611	<0.0001	−1.2631	<0.0001	−1.2723	<0.0001
Payment(biannual)	0.0485	0.0919	0.0487	0.0908		
Payment(quarterly)	0.1676	0.0184	0.1681	0.0181		
Payment(monthly)	0.0911	0.0018	0.0912	0.0018		
Subscription to MB	−0.1586	0.0198	−0.1587	0.0197	−0.1675	0.01370
Usage(retired)	−0.0304	0.4331	−0.0297	0.4433	−0.0315	0.4146
Usage(professional)	0.1536	0.0003	0.1535	0.0002	0.1481	0.0002
Usage(all trips)	0.3451	0.2328	0.3456	0.2321	0.3448	0.2332
Duration	−0.0025	0.0969	−0.0025	0.0978		
Fuel(gasoline)	−0.2621	<0.0001	−0.2630	<0.0001	−0.2607	<0.0001
Fuel(hybrid)	0.1265	0.6896	0.1225	0.6988	0.1196	0.70588
Type(commercial)	0.0318	0.5466				
Din(power)	0.0022	0.0004	0.0026	<0.0001	0.0024	<0.0001
Vehicle age	−0.0316	<0.0001	−0.0318	<0.0001	−0.0332	<0.0001
Vehicle speed	0.0009	0.3967				
Log-likelihood	−23,207		−23,207		−23,216	
Degrees of freedom	24		22		17	
AIC	46,462		46,458		46,466	
BIC	46,678		46,656		46,619	
Running time (s)	0.761		7.336		0.601	

As we can see, some of the coefficients are statistically significant at 0.0001. For example, the coefficient associated with the Bonus is significant and positive as expected. The bonus represents the percentage of the full premium and a large percentage shows an adverse claims history of a policyholder. The positive sign indicates that as the percentage of the full premium increases, the mean of claims frequency will increase. The coefficients associated with Coverage are negative for all categories and significant. The coefficient of Median 2 shows that the policyholders with this type of coverage have fewer claims than policyholders with Maxi coverage (the reference level). For example, in Model 1, a policyholder with a Median 2 has fewer claims than a policyholder with a Maxi coverage by $\exp(-0.1854) = 0.83$ and a policyholder with a Mini coverage has fewer claims by $\exp(-1.2611) = 0.28$. The coefficient of the car's power, represented by Din, is positive and significant, which indicates that powerful cars are more likely to be involved in an accident and therefore the mean of claims frequency for the owners of the powerful cars is higher. Unlike Ayuso et al. (2019) and Guillen et al. (2019), we found that Vehicle age has a negative impact on the number of claims. In our study, most of the policyholders are middle-aged and more likely to have old cars. In Section 3 we saw that the mean of the vehicle age is 9.56 for all policies and 7.30 for policies with at least one claim. Our portfolio of middle-aged policyholders also affects the sign of the coefficient associated with Age 1. Our dataset includes drivers as old as 103. Therefore, it seems reasonable to find a positive impact of age on the mean of the number of claims. In Model 1 the coefficients which are not significantly different from zero include Female 1, car usage for Retired and All trips, Hybrid fuel, Type and Speed. The coefficient of Professional usage indicates that Professional usage increases the mean of claims frequency compared to Work and private usage (the reference level) by $\exp(0.1536) = 1.17$. This is in line with Table 3 that policies for professional purposes make more claims. We obtain similar results for gasoline cars as in Table 3. Owners of Gasoline cars have fewer claims than owners of Diesel cars by $\exp(-0.2621) = 0.77$. We can see that the coefficient associated with Driver2 is positive. This seems reasonable as a policy that covers two drivers is more likely to make claims. The coefficient of Age 2 is negative. One interpretation can be that the average age of the second drivers is lower than the average age of the first drivers. However, in Section 3, we saw that driver age 2 is not significantly different for policies with claims and policies without claims. The coefficients associated with Duration and Policy duration are both negative. This implies that more experienced policyholders make fewer claims and also the more stable a policy is, the lower the mean of the number of claims. The coefficient of subscription to MB is negative and therefore this variable reduces the mean of claims frequency. Perhaps those who are willing to be monitored by telematics technology are more confident about their driving behaviour. We saw in the previous section that payment frequency is not a significant variable. As we can see, their corresponding p-values for some categories in models 1 and 2 are not significant at 0.05 and therefore we have removed them from Model 3. However, we decided to keep the variable Usage, although not all categories are significant at 0.05, as we found in the previous section that it is effective on the number of claims. Among our three models, Model 2 has the lowest AIC and Model 3 has the lowest BIC. As we can see, the computation time for Model 2 is longer than the other two models. The reason is that the stepwise algorithm examines different models to find the one with the smallest AIC.

Table 5 presents three logistic models with their coefficients and the corresponding p-values. Similar to Table 4, Model 1 includes all variables, Model 2 is based on the stepwise algorithm and Model 3 only includes significant variables. The interpretation of the coefficients in logistic regression is similar to Poisson regression and as we can see, the signs of the coefficients are the same. The only difference is that in logistic regression we look at the impact of variables on the odds of the occurrence of claims. So, for example, the interpretation of the coefficient associated with Bonus is that, the greater the percentage of the full premium (adverse claims history) is, the higher the odds of the occurrence of the claims for the coefficient associated with professional usage; we can say that the odds of the occurrence of claims for policyholders with professional usage increases by $\exp(0.1691) = 1.18$ as opposed to policyholders with work and private usage. For the negative coefficient associated with subscription to MB, we can say that the odds of the occurrence of claims fall for a policyholder who joins this scheme. Other variables can be similarly interpreted. Model 2 is built by examining different models and finding the one with the lowest AIC. All variables in this model are the same as the variables in stepwise Poisson regression except for duration which is not included in stepwise logistic regression. For Model 3 we again remove all variables with a p-value greater than 0.05. In addition, we do not include payment frequency as this has been proved to be insignificant in Section 3. As we can see, Model 2 has the smallest AIC and Model 3 has the smallest BIC. Further, the computation time for the stepwise algorithm is longer than the stepwise Poisson regression model.

When building a model, it is important to consider the underlying assumptions. For example, to fit a ZIP model to our data, we first need to test for the presence of over-dispersion. One approach is to fit a quasi-Poisson and to determine the dispersion parameter, i.e., θ in $\mathrm{Var}(y) = \theta\ \mathrm{E}[y]$. In our case, using only significant variables from Tables 4 and 5, the dispersion parameter is 1.1. Alternatively, we can fit NB regression and compare our new model with Poisson regression. In our case, AIC and BIC for NB regression are 46,184 and 46,346, respectively, which are lower than AIC and BIC for the Poisson regression. Now, since we have the problem of over-dispersion and excess zeros, we can fit a ZIP model to our data. Table 6 shows the estimated coefficients and their p-values for the Poisson (count) part and zero-inflated part of three ZIP models. Model 1 is the full model where we consider the variables of the full model in Table 4 for the count part and the variables of the full model in Table 5 for the zero-inflated part. As we can see, most variables are not significantly different from zero. If we consider the significant level of 0.1, the coefficient associated with Age 1 is positive as in Table 4 and statistically significant in the count part. In addition, the coefficient associated with Age 2 is positive and significant in the zero-inflated part, but not in the count part. From Section 3, we know that the second divers are younger than the first drivers. Therefore, we can claim that in this group older drivers are more likely to have zero claims. The coefficient of situation duration in the count part is negative and significant as in Table 4 with the same interpretation. The coefficients associated with coverage are significant at 0.01 in the count part with the identical signs as in Table 4, but they are not significant in the zero-inflated part. The interpretation is that the mean frequency of claims for policyholders covered under, for example, Mini coverage is less than the policyholders covered under Maxi coverage by $\exp(-1.0487) = 0.35$.

The coefficient of fuel (gasoline) is positive and significant which indicates that the odds of zero claims for drivers of gasoline cars increases by $\exp(0.5066) = 1.66$ as opposed to drivers of diesel cars. Further, in the zero-inflated part, the coefficient of Driver2? is negative and significant. Therefore, a policy with two drivers is less likely to have zero claims, in other words, a policy with the 2nd driver is more likely to be involved in an accident and to make a claim. The associated coefficient of vehicle age is positive and significantly different from zero in the zero-inflated part, which is in line with our findings for Poisson and logistic models that it is more likely for the owners of older cars to have zero claims. All other variables including subscription to MB are not significantly different from zero. The variables of Model 2 in the count and zero-inflated part come from the variables of stepwise models in Tables 4 and 5, respectively. The coefficients have the same sign and therefore similar interpretation as in Model 1. Again the coefficient of subscription to MB is not significantly different from zero. Model 3 can be built using the variables of the models that contain only significant variables in Tables 4 and 5. Coverage in the count part and Age 2, Driver2?, fuel and vehicle age in the zero-inflated part are all significantly different from zero. In Table 6 the signs of some of the coefficients do not conform to Tables 4 and 5. For example, subscription to MB is positive both in the count part and in the zero-inflated part. Since such coefficients are not statistically significant, we can conclude that they are not significantly different from zero. Comparing AIC and BIC of these three models, we can see that the smallest AIC can be obtained by Model 2 where the variables come from stepwise models in Tables 4 and 5 and the smallest BIC by Model 3. In addition, AIC has considerably improved for ZIP models compared to Poisson models in Table 4. In the next section, we show that the prediction of zero claims by ZIP is considerably better than Poisson regression.

Table 5. Regression coefficients of logistic models.

Coefficients	Model 1: All variables		Model 2: Stepwise Selection		Model 3: Only Significant	
	Estimate	*p*-Value	Estimate	*p*-Value	Estimate	*p*-Value
Intercept	−2.6571	<0.0001	−2.5321	<0.0001	−2.4255	<0.0001
Age 1	0.0043	0.0036	0.0038	0.0066	0.0032	0.0226
Age 2	−0.0048	0.0011	−0.0047	0.0013	−0.0047	0.0013
Female 1	0.0441	0.1198	0.0428	0.1275		
Driver2?	0.2410	0.0012	0.2363	0.0014	0.2340	0.0016
Situation duration	−0.0234	0.0027	−0.0248	0.0013	−0.0265	0.0006
Bonus	0.9017	<0.0001	0.9151	<0.0001	0.9447	<0.0001
Coverage(Med2)	−0.1814	<0.0001	−0.1786	<0.0001	−0.1832	<0.0001
Coverage(Med1)	−0.2111	0.0005	−0.2061	0.0007	−0.2132	0.0004
Coverage(Mini)	−1.2481	<0.0001	−1.2438	<0.0001	−1.2589	<0.0001
Payment(biannual)	0.0522	0.1121	0.0490	0.1335		
Payment(quarterly)	0.1852	0.0240	0.1883	0.0217		
Payment(monthly)	0.0939	0.0049	0.0936	0.0051		
Subscription to MB	−0.2014	0.0088	−0.2038	0.0080	−0.2098	0.0063
Usage(retired)	−0.0180	0.6847	−0.0149	0.7364	−0.0203	0.6455
Usage(professional)	0.1691	0.0007	0.1733	0.0002	0.1664	0.0004
Usage(all trips)	0.4841	0.1577	0.4856	0.1563	0.4835	0.1577
Duration	−0.0019	0.2708				
Fuel(gasoline)	−0.2885	<0.0001	−0.2914	<0.0001	−0.2875	<0.0001
Fuel(hybrid)	0.0707	0.8560	0.0691	0.8592	0.0598	0.8778
Type(commercial)	0.0462	0.4402				
Din(power)	0.0022	0.0019	0.0027	<0.0001	0.0025	<0.0001
Vehicle age	−0.0336	<0.0001	−0.0336	<0.0001	−0.0332	<0.0001
Vehicle speed	0.0010	0.4404				
Log-likelihood	−20,292		−20,293		−20,299	
Degrees of freedom	24		21		17	
AIC	40,632		40,628		40,633	
BIC	40,848		40,817		40,785	
Running time (s)	0.634		31.611		0.431	

Table 6. Regression coefficients of zero-inflated Poisson (ZIP) models.

Coefficients	Model 1 *		Model 2 *		Model 3 *	
	Estimate	*p*-Value	Estimate	*p*-Value	Estimate	*p*-Value
Poisson (count) part						
Intercept	−2.2750	<0.0001	−2.1404	<0.0001	−2.0736	<0.0001
Age 1	0.0066	0.0513	0.0063	0.0542	0.0046	0.1347
Age 2	0.0012	0.6647	0.0013	0.6290	0.0019	0.4998
Female 1	−0.0448	0.4746	−0.0479	0.4324		
Driver2?	−0.0473	0.7366	−0.0536	0.6972	−0.0809	0.5584
Situation duration	−0.0009	0.0194	−0.0017	0.9282	−0.0011	0.9558
Bonus	0.5787	0.2021	0.5953	0.2068	0.5433	0.2026
Coverage(Med2)	−0.3670	0.0006	−0.3734	0.0004	−0.3646	0.0006
Coverage(Med1)	−0.4294	0.0022	−0.4231	0.0022	−0.4269	0.0024
Coverage(Mini)	−1.0487	0.0017	−1.0599	0.0016	−1.1031	0.0007
Payment(biannual)	−0.0691	0.3318	−0.0668	0.3472		
Payment(quarterly)	-0.0569	0.7525	−0.0452	0.8002		
Payment(monthly)	0.0091	0.8961	0.0137	0.8427		
Subscription to MB	0.0596	0.7580	0.0462	0.8099	0.0126	0.9486
Usage(retired)	−0.0623	0.5523	−0.0623	0.5455	−0.0538	0.6006
Usage(professional)	0.0723	0.5491	0.0710	0.5063	0.0952	0.3725
Usage(all trips)	0.2543	0.6523	0.2375	0.6765	−0.0332	0.9544
Duration	−0.0043	0.2948	−0.0025	0.1145		
Fuel(gasoline)	−0.0346	0.6549	−0.0387	0.6228	−0.0759	0.3854
Fuel(hybrid)	0.6976	0.2553	0.7050	0.2500	0.6366	0.3302
Type(commercial)	0.0284	0.8457				
Din(power)	0.0023	0.3055	0.0026	0.1207	0.0024	0.3063
Vehicle age	0.0024	0.7925	0.0032	0.1207	0.0063	0.4799
Vehicle speed	0.0010	0.7557				
Zero-inflation part						
Intercept	−0.5544	0.6449	−0.4634	0.6883	−0.4819	0.6826
Age 1	0.0040	0.5922	0.0032	0.6558	0.0020	0.7736
Age 2	0.0108	0.0653	0.0111	0.0546	0.0121	0.0356
Female 1	−0.1962	0.1572	−0.2020	0.1358		
Driver2?	−0.5491	0.0892	−0.5633	0.0748	−0.6165	0.0522
Situation duration	0.0331	0.3289	0.0309	0.3575	0.0387	0.2442
Bonus	−0.8651	0.4958	−0.8200	0.5308	−1.0637	0.3802
Coverage(Med2)	−0.3798	0.1014	−0.3932	0.0848	−0.3595	0.1139
Coverage(Med1)	−0.4030	0.1541	−0.3871	0.1601	−0.3847	0.1645
Coverage(Mini)	0.2803	0.5949	0.2610	0.6222	0.2052	0.6923
Payment(biannual)	−0.2596	0.0910	−0.2547	0.0963		
Payment(quarterly)	−0.5661	0.2637	−0.5354	0.2798		
Payment(monthly)	−0.1721	0.2518	−0.1615	0.2751		
Subscription to MB	0.4226	0.2041	0.3978	0.2310	0.3560	0.3030
Usage(retired)	−0.0775	0.7250	−0.0783	0.7177	−0.0533	0.8043
Usage(professional)	−0.2215	0.4672	−0.2273	0.4063	−0.1510	0.5703
Usage(all trips)	−0.3050	0.8550	−0.3668	0.8345	−1.8681	0.7405
Duration	−0.0040	0.6448				
Fuel(gasoline)	0.5066	0.0017	0.5002	0.0021	0.4090	0.0236
Fuel(hybrid)	114.80	0.1984	116.60	0.1890	1.0632	0.2829
Type(commercial)	0.0041	0.9901				
Din(power)	0.0000	0.9944	-0.0000	0.9959	−0.0002	0.9627
Vehicle age	0.0696	<0.0001	0.0712	<0.0001	0.0756	<0.0001
Vehicle speed	0.0006	0.9258				
Log-likelihood	−23,044		−23,045		−23,054	
Degrees of freedom	48		43		34	
AIC	46,184		46,175		46,177	
BIC	46,616		46,563		46,482	
Running time (s)	16.719		18.81		13.951	

* Model 1: full model; Model 2: based on the variables of stepwise models in Tables 4 and 5; Model 3: based on the variables of only significant models in Tables 4 and 5.

5. Validation

In this section, we use our validation set to compare the predictability of the models discussed in Section 4. Table 7 presents the predicted number of zero and non-zero claims by our models in Section 4. In this table, individual 1 refers to an 85-year-old male policyholder with a maxi policy that pays biannually with the bonus (percentage of the full premium) of 0.5. He holds this policy for retired usage, for 29 years and has not signed to MB scheme. The policy was modified nine years ago. He owns a 10-year-old tourism car with gasoline, the din of 98 and max speed of 182. In year 0, this policyholder has not made any claim and the probability of zero claims predicted by Poisson regression according to full model is $\exp(-0.1036)$ where 0.1036 is the estimated parameter λ and the probability of zero claims predicted by logistic regression is $1 - 0.0903$ where 0.0903 is the estimated $\pi = \Pr(y = 1)$. Prediction of zero claims by ZIP is 0.9104. Individual 2 is a male policyholder with a maxi policy. This policy covers two drivers aged 54 and 56 and has been held for six years and been modified two years ago with a bonus of 0.5 and monthly premium payment. The policyholder owns a two-year old tourism car with diesel for work and private purposes with the din of 75 and max speed of 163. The estimated parameter by Poisson regression is $\lambda = 0.1794$ and by logistic regression is $\pi = 0.1525$. As we can see, the count part of ZIP for the two policyholders is very close to the estimated value of Poisson regression. If we add the probability of zero claims in all these models, we can approximate the number of zero claims. Results show that ZIP models considerably outperform Poisson regression and logistic regression performs better than ZIP models in predicting zero claims. Further, we can see that there is a slight difference between predictions made by full models, stepwise models and the models with only significant variables.

Table 7. Prediction of the probability and the number of zero claims by our models.

			Prob Zero Claims: Individual 1 **	**Prob Zero Claims: Individual 2 *****	**Total No. of Zero Claims**	**Total No. of Non-Zero Claims**
Observed value			0	1	35,772	4316
Poisson	Full model		0.9016	0.8358	35,361.44	4726.56
	Stepwise		0.9019	0.8350	35,361.27	4726.73
	Significant		0.9019	0.8410	35,360.50	4727.50
Logistic	Full model		0.9097	0.8475	35,606.88	4481.12
	Stepwise		0.9091	0.8480	35,606.95	4481.05
	Significant		0.9102	0.8512	35,606.62	4481.38
ZIP	Model 1 *	Count	0.1024	0.1778	35,602	4486
		Zero	0.9104	0.8446		
	Model 2 *	Count	0.1016	0.1785	35,601.27	4486.74
		Zero	0.9113	0.8440		
	Model 3 *	Count	0.1020	0.1710	35,601.06	4486.94
		Zero	0.9113	0.8495		

* Model 1: full model; Model 2: based on the variables of stepwise models in Tables 4 and 5; Model 3: based on the variables of only significant models in Tables 4 and 5. ** An 85-year-old male policyholder with biannual maxi coverage and bonus of 0.5 for retired usage. He had this policy for 29 years and changed it nine years ago. He has not registered for MB and owns a 10-year old tourism car with gasoline, the din of 98 and max speed of 182. *** A 54-year-old male with monthly maxi coverage and bonus of 0.5 for private usage. The 2nd driver is a 56-year-old female. The policy was written six years ago and was modified two years ago. It covers a two-year-old tourism car with diesel and din of 75 and max speed of 163. It is not part of MB scheme.

6. Conclusions

We have divided our dataset into training and validation sets. Using our training set, we have developed three models and compared our models according to their AIC and BIC values. We found that type of coverage, vehicle age and fuel are statistically significant in most of our models. We then validated our models and showed that a ZIP model can predict the frequency of claims better than a Poisson regression. Further, we have shown that if we are just concerned about the number of zero and non-zero claims, logistic regression can even outperform a ZIP model. In fact, logistic regression is a one layer neural network and there is a scope to extend our study to a more generalised form of logistic regression for future research. We saw that the policyholders who were willing to be monitored by telematics devices are less likely to make a claim. A thorough study of the policyholders' behaviour before and after being monitored by telematics devices can be another area of future research. Given the current concern regarding climate change and sustainability, the possibility of the inclusion of fuel consumption into a pricing model may be considered in the future (Tselentis et al. 2017).

Funding: This research received no external funding.

Acknowledgments: I am very grateful to the reviewers' comments and suggestions which were valuable to improve this paper.

Conflicts of Interest: The author declares no conflict of interest.

Appendix A

```
# Loading the prepared data
Data <- read.csv("Year0.csv", header = TRUE)
# Creating training and validation datasets
set.seed(123567)
random <- runif(dim(Data)[1])
# Training set is our data <<0.6
train <- random < 0.6
DataTrain <- cbind(Data, random, train)
# Validation set is everything not included in training set
valid <- !(train) ; DataValid <- cbind(Data, random, valid)
# Exporting our sets
write.csv(DataTrain[train == TRUE,], "DataTrain.csv")
write.csv(DataValid[valid == TRUE,], "DataValid.csv")
# Codes to produce Table~\ref{tab.2}:
DataTrain <- read.csv("DataTrain.csv", header = TRUE)
# Remove negative claim amounts
DataTrain$claim_amount[DataTrain$claim_amount < 30] <- 0
# Adjusting claim numbers
DataTrain$claim_nb <- DataTrain$claim_nb * (DataTrain$claim_amount > 0)
# Removing zeros
DataTrain$drv_age2[DataTrain$drv_age2==0]         <- NA
DataTrain$vh_value[DataTrain$vh_value==0]         <- NA
DataTrain$vh_cyl[DataTrain$vh_cyl==0]             <- NA
DataTrain$vh_weight[DataTrain$vh_weight==0]       <- NA
DataTrain$drv_drv2[DataTrain$drv_drv2==0]         <- NA
# Separating the training set into two sets of policies with and without claims
NClaim <- subset(DataTrain, DataTrain$claim_nb == 0)
Claim  <- subset(DataTrain, DataTrain$claim_nb > 0)
# Calculations for all policies
Mydata <- data.frame(cbind(DataTrain$claim_nb, DataTrain$pol_duration,
DataTrain$pol_sit_duration,  DataTrain$drv_age1, DataTrain$drv_age2,
DataTrain$vh_value, DataTrain$vh_age, DataTrain$vh_cyl,
DataTrain$vh_speed, DataTrain$vh_weight, DataTrain$vh_din))
Mean <- sapply(Mydata, mean, na.rm = TRUE)
SD   <- sapply(Mydata, sd,   na.rm = TRUE)
# Calculations for policies without claims
NMydata <- data.frame(cbind(NClaim$claim_nb, NClaim$pol_duration,
NClaim$pol_sit_duration, NClaim$drv_age1, NClaim$drv_age2,
NClaim$vh_value, NClaim$vh_age, NClaim$vh_cyl, NClaim$vh_speed,
NClaim$vh_weight,NClaim$vh_din))
NMean <- with(NClaim, sapply(NMydata, mean, na.rm = TRUE))
NSD   <- with(NClaim, sapply(NMydata, sd, na.rm = TRUE))
# Calculations for policies with claims
CMydata <- data.frame(cbind(Claim$claim_nb, Claim$pol_duration,
Claim$pol_sit_duration, Claim$drv_age1, Claim$drv_age2,
Claim$vh_value, Claim$vh_age, Claim$vh_cyl, Claim$vh_speed,
Claim$vh_weight, Claim$vh_din))
```

```
CMean <- with(Claim, sapply(CMydata, mean, na.rm = TRUE))
CSD   <- with(Claim, sapply(CMydata, sd, na.rm = TRUE))
# Modelling
DataTrain <- read.csv("DataTrain.csv", header = TRUE)
DataTrain$claim_amount[DataTrain$claim_amount < 30] <- 0
DataTrain$claim_nb <- DataTrain$claim_nb * (DataTrain$claim_amount > 0)
# Re-leveling categorical variables:
DataTrain$drv_sex1_r     <- relevel(factor(DataTrain$drv_sex1), ref = "M")
DataTrain$pol_coverage_r <- relevel(factor(DataTrain$pol_coverage), ref = "Maxi")
DataTrain$pol_pay_freq_r <- relevel(factor(DataTrain$pol_pay_freq), ref = "Yearly")
DataTrain$pol_payd_r     <- relevel(factor(DataTrain$pol_payd), ref = "No")
DataTrain$pol_usage_r    <- relevel(factor(DataTrain$pol_usage), ref = "WorkPrivate")
DataTrain$vh_fuel_r      <- relevel(factor(DataTrain$vh_fuel), ref = "Diesel")
DataTrain$vh_type_r      <- relevel(factor(DataTrain$vh_type), ref = "Tourism")
DataTrain$drv_drv2_r     <- relevel(factor(DataTrain$drv_drv2), ref = "No")# Poisson regression
Model.poi <- glm(claim_nb ~ drv_age1 + drv_age2 + drv_sex1_r + drv_drv2_r + pol_sit_duration
+ pol_bonus + pol_coverage_r + pol_pay_freq_r + pol_payd_r + pol_usage_r
+ pol_duration + vh_fuel_r + vh_type_r + vh_din + vh_age + vh_speed,
data = DataTrain,
family = poisson(link = "log"), offset = log(Exposures), na.action = na.omit)
# Logistic regression
# y=1 represents claim and y=0 no claim
DataTrain$y[DataTrain$claim_nb==0]  <- 0
DataTrain$y[DataTrain$claim_nb > 0] <- 1
# Model:
Model.log <- glm(y ~ drv_age1 + drv_age2 + drv_sex1_r + drv_drv2_r + pol_sit_duration
+ pol_bonus + pol_coverage_r + pol_pay_freq_r + pol_payd_r + pol_usage_r
+ pol_duration + vh_fuel_r + vh_type_r + vh_din + vh_age + vh_speed,
data = DataTrain,
family = binomial(link = "logit"), na.action = na.omit)# ZIP regression
library("pscl")
Model.zeropoi <- zeroinfl(claim_nb ~ drv_age1 + drv_age2 + drv_sex1_r + drv_drv2_r
+ pol_sit_duration + pol_bonus + pol_coverage_r + pol_pay_freq_r
+ pol_payd_r + pol_usage_r
+ pol_duration + vh_fuel_r + vh_type_r + vh_din + vh_age + vh_speed,
data = DataTrain, na.action = na.omit,
dist = "poisson", link = "logit")
# Validation:
# loading validation set
DataValid <- read.csv("DataValid.csv", header = TRUE)
DataValid$claim_amount[DataValid$claim_amount < 30] <- 0
DataValid$claim_nb <- DataValid$claim_nb * (DataValid$claim_amount > 0)
#
DataValid$pol_coverage_r <- DataValid$pol_coverage
DataValid$vh_fuel_r      <- DataValid$vh_fuel
DataValid$vh_type_r      <- DataValid$vh_type
DataValid$pol_pay_freq_r <- DataValid$pol_pay_freq
DataValid$pol_payd_r     <- DataValid$pol_payd
DataValid$drv_drv2_r     <- DataValid$drv_drv2
DataValid$pol_usage_r    <- DataValid$pol_usage
DataValid$drv_sex1_r  <- DataValid$drv_sex1
DataValid$y[DataValid$claim_nb==0]  <- 0
DataValid$y[DataValid$claim_nb > 0] <- 1
# Prediction:
predict.poi <- predict(Model.poi, DataValid, type = "response")
#
predict.log <- predict(Model.log, DataValid, type = "response")
#
predict.zeropoi <- cbind( DataValid, Mean = predict(Model.zeropoi,
DataValid, type = "response"),Probab = predict(Model.zeropoi,
DataValid, type = "prob"))
# Test for dispersion
library("AER")
```

```
dispersiontest(Model.poi,trafo=1)
Model.neg <- MASS::glm.nb(claim_nb ~ drv_age1 + drv_age2 + drv_drv2_r + pol_sit_duration
+ pol_bonus + pol_coverage_r + pol_payd_r + pol_usage_r
+ vh_fuel_r + vh_din + vh_age , data = DataTrain,
link = "log", na.action = na.omit)
odTest(Model.neg)
# Codes to predict zero claims:
sum(exp(-predict(Model.poi, DataValid, type = "response")))
sum(1-predict(Model.log, DataValid, type = "response"))
sum(predict(Model.zeropoi, DataValid, type = "prob")[,1])
```

References

Ayuso, Mercedes, Montserrat Guillen, and Jens Perch Nielsen. 2019. Improving automobile insurance ratemaking using telematics: incorporating mileage and driver behaviour data. *Transportation* 46: 735–52. [CrossRef]

Boucher, Jean-Philippe, Michel Denuit, and Montserrat Guillén. 2007. Risk classification for claim counts. *North American Actuarial Journal* 11: 110–31. [CrossRef]

Boucher, Jean-Philippe, Michel Denuit, and Montserrat Guillen. 2009. Number of accidents or number of claims? An approach with zero-inflated Poisson models for panel data. *The Journal of Risk and Insurance* 76: 821–46. [CrossRef]

Boucher, Jean-Philippe, Steven Côté, and Montserrat Guillen. 2017. Exposure as duration and distance in telematics motor insurance using generalized additive models. *Risks* 5: 54. [CrossRef]

Cantoni, Eva, and Marie Auda. 2018. Stochastic variable selection strategies for zero-inflated models. *Statistical Modelling* 18: 3–23. [CrossRef]

Chen, Kun, Rui Huang, Ngai Hang Chan, and Chun Yip Yau. 2019. Subgroup analysis of zero-inflated Poisson regression model with applications to insurance data. *Insurance: Mathematics and Economics* 86: 8–18. [CrossRef]

Chowdhury, Shrabanti, Saptarshi Chatterjee, Himel Mallick, Prithish Banerjee, and Broti Garai. 2019. Group regularization for zero-inflated poisson regression models with an application to insurance ratemaking. *Journal of Applied Statistics* 46: 1567–81. [CrossRef]

Dutang, Christophe, and Arthur Charpentier. 2019. CASdatasets: Insurance Datasets. Available online: http://dutangc.free.fr/pub/RRepos/web/CASdatasets-index.html accsed on 15 March, 2019

Fauzan, Muhammad Arief, and Hendri Murfi. 2018. The accuracy of XGBoost for insurance claim prediction. *International Journal of Advances in Soft Computing and Its Applications* 10: 159–71.

Ferreira, Joseph, and Eric Minikel. 2012. Measuring per mile risk for pay-as-you-drive automobile insurance. *Transportation Research Record: Journal of the Transportation Research Board* 2297: 97–103. [CrossRef]

Frees, Edward W., Richard A. Derrig, and Glenn Meyers. 2014. *Predictive Modeling Applications in Actuarial Science, Volume I: Predictive Modeling Techniques*. New York: Cambridge University Press.

Frees, Edward W., Richard A. Derrig, and Glenn Meyers. 2016. *Predictive Modeling Applications in Actuarial Science, Volume II: Case Studies in Insurance*. New York: Cambridge University Press.

Gao, Guangyuan, Shengwang Meng, and Mario V. Wüthrich. 2019. Claims frequency modeling using telematics car driving data. *Scandinavian Actuarial Journal* 2019: 143–62. [CrossRef]

Guillen, Montserrat, Jens Perch Nielsen, Mercedes Ayuso, and Ana M. Pérez-Marín. 2019. The use of telematics devices to improve automobile insurance rates. *Risk Analysis* 39: 662–762. [CrossRef] [PubMed]

Haberman, Steven, and Arthur E. Renshaw. 1996. Generalized linear models and actuarial science. *The Statistician* 45: 407–36. [CrossRef]

Lambert, Diane. 1992. Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics* 34: 1–14. [CrossRef]

Lee, Andy H., Mark R. Stevenson, Kui Wang, and Kelvin K. W. Yau. 2002. Modeling young driver motor vehicle crashes: Data with extra zeros. *Accident Analysis and Prevention* 34: 515–21. [CrossRef]

Lemaire, Jean, Sojung Carol Park, and Kili C. Wang. 2015. The use of annual mileage as a rating variables. *Astin Bulletin* 46: 39–69. [CrossRef]

Liu, Feng, and David Pitt. 2017. Application of bivariate negative binomial regression model in analysing insurance count data. *Annals of Actuarial Science* 11: 390–411. [CrossRef]

McCullagh, Peter, and John A. Nelder. 1998. *Generalized Linear Models*. London: Chapman and Hall.

Murphy, Kevin P. 2012. *Machine Learning—A Probabilistic Perspective*. Cambridge: The MIT Press.

Perumean-Chaney, Suzanne E., Charity Morgan, David McDowall, and Inmaculada Aban. 2013. Zero-inflated and overdispersed: What's one to do? *Journal of Statistical Computation and Simulation* 83: 1671–83. [CrossRef]

Spedicato, Giorgio Alfredo, Christophe Dutang, and Leonardo Petrini. 2018. Machine learning methods to perform pricing optimization. A comparison with standard GLMs. *Variance, Casualty Actuarial Society* 12: 69–89.

Tang, Yanlin, Liya Xiang, and Zhongyi Zhu. 2014. Risk factor selection in rate making EM adaptive LASSO for zero-inflated Poisson regression models. *Risk Analysis* 34: 1112–27. [CrossRef]

Tselentis, Dimitrios I., George Yannis, and Eleni I. Vlahogianni. 2017. Innovative motor insurance schemes: A review of current practices and emerging challenges. *Accident Analysis and Prevention* 98: 139–48. [CrossRef]

Verbelen, Roel, Katrien Antonio, and Gerda Claeskens. 2018. Unraveling the predictive power of telematics data in car insurance pricing. *Journal of the Royal Statistical Society: Series C (Applied Statistics)* 67: 1275–304. [CrossRef]

Weerasinghe, K. P. M. L. P., and M. C. Wijegunasekara. 2016. A comparative study of data mining algorithms in the prediction of auto insurance claims. *European International Journal of Science and Technology* 5: 47–54.

Wilson, Paul, and Jochen Einbeck. A new and intuitive test for zero modification. *Statistical Modelling* doi:10.1177/1471082X18762277. [CrossRef]

Yip, Karen C. H., and Kelvin K. W. Yau. 2005. On modeling claim frequency data in general insurance with extra zeros. *Insurance: Mathematics and Economics* 36: 153–63. [CrossRef]

Zeileis, Achim, Christian Kleiber, and Simon Jackman. 2008. Regression models for count data in R. *Journal of Statistical Software* 27: 1–25.

© 2019 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).

risks

Article

Conditional Variance Forecasts for Long-Term Stock Returns

Enno Mammen [1], Jens Perch Nielsen [2], Michael Scholz [3,]* and Stefan Sperlich [4]

[1] Institute for Applied Mathematics, Heidelberg University, Im Neuenheimer Feld 205, 69120 Heidelberg, Germany; mammen@math.uni-heidelberg.de
[2] Faculty of Actuarial Science and Insurance, Cass Business School, 106 Bunhill Row, London EC1Y 8TZ, UK; jens.nielsen.1@city.ac.uk
[3] Department of Economics, University of Graz, Universitätsstraße 15/F4, 8010 Graz, Austria
[4] Geneva School of Economics and Management, Université de Genève, Bd du Pont d'Arve 40, 1211 Genève, Switzerland; stefan.sperlich@unige.ch
* Correspondence: michael.scholz@uni-graz.at; Tel.: +43-316-380-7112

Received: 20 August 2019; Accepted: 29 October 2019; Published: 5 November 2019

Abstract: In this paper, we apply machine learning to forecast the conditional variance of long-term stock returns measured in excess of different benchmarks, considering the short- and long-term interest rate, the earnings-by-price ratio, and the inflation rate. In particular, we apply in a two-step procedure a fully nonparametric local-linear smoother and choose the set of covariates as well as the smoothing parameters via cross-validation. We find that volatility forecastability is much less important at longer horizons regardless of the chosen model and that the homoscedastic historical average of the squared return prediction errors gives an adequate approximation of the unobserved realised conditional variance for both the one-year and five-year horizon.

Keywords: benchmark; cross-validation; prediction; stock return volatility; long-term forecasts; overlapping returns; autocorrelation

JEL Classification: C14; C53; C58; G17; G22

1. Introduction

The volatility of financial assets has important implications for the theory and practice of asset pricing, portfolio selection, risk management, and market-timing strategies. Therefore, it is of fundamental interest to measure ex ante, or forecast successfully, the conditional variance of returns. Of course, the evaluation of the latter and the forecasting itself have been complicated by the unobservability of the realised conditional variance (Galbraith and Kisinbay 2005). An extensive amount of research is engaged in analysing the distributional and dynamic properties of stock market volatility; see, for example, Andersen et al. (2001) and citations therein. The standard approaches applied include parametric (G)ARCH-type or stochastic volatility models and estimate the underlying returns based on specific distributional assumptions. Alternatives, especially for data of higher frequency, are based on constructing model-free estimates of ex-post *realized* volatilities by adding up the squares and cross-products of intraday high-frequency returns (Andersen et al. 2001).

The present paper instead uses annual U.S. stock market data to construct excess stock returns at the one-year and five-year horizon and to examine their model-based variance forecasts. Note that the risk depends on the investment horizon considered and that different horizons are relevant for different applications (Christoffersen and Diebold 2000). Little is known about the forecastability of variance at horizons beyond a year. Here, we take the long-term actuarial view and extend the work of Kyriakou et al. (2019a, 2019b). In a two-step procedure, we first apply machine learning (ML) to predict stock returns in excess of different benchmarks, considering the short- and long-term interest rate, the earnings-by-price ratio, and the inflation rate. Second, the squared residuals are used to analyse model-based volatility forecastability. Here, we compare these forecasts with the forecast implicit in the unconditional residual variance, as proposed, for example, by Galbraith and Kisinbay (2005). We find that volatility forecastability is much less important at longer horizons regardless of the chosen model and that the homoscedastic historical average of the squared return prediction errors gives an adequate approximation of the unobserved realised conditional variance for both the one-year and five-year horizon.

Our preferred ML technique applied in this paper is local-linear smoothing in combination with a leave-*k*-out cross-validation for the following reasons.[1] First, we are interested in longer-horizon stock returns based on annual observations and their volatility. Thus, we are not in the high-frequency context where the number of observations is huge and the set of possible predictive variable combinations is enormous (and, thus, dimension reduction or shrinkage are indispensable). Our data set is, instead, sparse and a careful imposition of structure to the statistical modelling process is much more promising, as shown, for example, by Nielsen and Sperlich (2003) and Scholz et al. (2015, 2016). Second, the evidence of stock return predictability is much stronger once one allows for nonlinear functions as documented, for example, in Lettau and Van Nieuwerburgh (2008), Chen and Hong (2010), or Cheng et al. (2019). Thus, the local-linear smoother is ideally suited as it can estimate a linear function—the classical benchmark in this context—without any bias. Finally, our procedures are analytically well studied, i.e., sound and rigorous, statistical tools which let us operate in a glasshouse, not in a black box—in contrast to other fancier but less clear ML methods.[2]

Note further that longer horizons are important to long-term investors, such as pension funds or market participants saving for distant payoffs. These investors are generally willing to take on more risk for higher rewards and, thus, volatility forecastability is for them of fundamental interest. Rapach and Zhou (2013) show that longer horizons tend to produce better estimates than shorter horizons, while Munk and Rangvid (2018) point out that major finance houses today use longer horizons—up to ten years—to stabilise and improve future predictions. In our paper, we exemplarily concentrate on the one-year and five-year view.[3] However, shorter horizons based on monthly, weekly, or even daily data do not seem to provide the pension saver with good information about future income as a pensioner. Therefore, these type of short-term predictions—sometimes called investment robots—are not suitable when a pensioner should define his or her risk appetite.

The remaining of this paper is organized as follows. Section 2 presents our framework for the purpose of conditional variance prediction. We define the underlying financial model, introduce our two-step procedure, and present our validation criterion for model selection. In addition, we review different ways of estimating the conditional variance and discuss bootstrap-tests for the null hypothesis of no predictability. In Section 3, we provide a description of our data set and of our empirical findings from different validated scenarios: (i) a single benchmarking approach that uses the dependent variable transformed with the benchmark, and (ii) the case where both the independent and dependent variables are transformed with the benchmark (full benchmarking approach). Finally, we take the long-term view and comment on real income pension prediction. Section 4 summarizes the key points of our analysis and concludes the paper.

[1] Our methodology of validating a fully nonparametric structure can be viewed as one of the simplest and therefore also most transparent version of machine learning; see Section 2 of Kyriakou et al. (2019a) for more details justifying the label machine learning for our approach.

[2] Note that the use of a different ML method would come with the cost of losing interpretability, smoothness, or flexibility due to restrictions on the functional form. A comparison of different ML techniques in finding that one which gives the best predictions, wins an investment horse-race out-of-sample, or being the most robust method over different periods is out of the scope of our work.

[3] The choice of the one-year horizon is related to the frequency of the data. In contrast, the five-year horizon is arbitrary but is intended to be a starting point for actuarial long-term models for real-income savings. Other horizons and related questions remain for future research.

2. A Framework for Conditional Variance Prediction

In this section, we focus on nonlinear predictive relationships between squared residuals of model-based predicted stock returns over the next T years in excess of a benchmark and a set of explanatory variables. Our aim is the investigation of different benchmark models and their volatility predictability over return horizons of one year and five years. We consider four different benchmarks: the short- and the long-term interest rate, the earnings-by-price ratio, and the inflation rate.

2.1. One-Year Predictions

Let P_t denote the (nominal) stock price at the end of year t and D_t the (nominal) dividends paid during year t. We investigate stock returns $S_t = (P_t + D_t)/P_{t-1}$ in excess (log-scale) of a given benchmark $B_{t-1}^{(A)}$:

$$Y_t^{(A)} = \ln \frac{S_t}{B_{t-1}^{(A)}}, \tag{1}$$

where $A \in \{R, L, E, C\}$ with, respectively,

$$B_t^{(R)} = 1 + \frac{R_t}{100}, \quad B_t^{(L)} = 1 + \frac{L_t}{100}, \quad B_t^{(E)} = 1 + \frac{E_t}{P_t}, \quad B_t^{(C)} = \frac{CPI_t}{CPI_{t-1}},$$

using the short-term interest rate, R_t, the long-term interest rate, L_t, the earnings accruing to the index in year t, E_t, and the consumer price index for year t, CPI_t. The predictive and fully nonparametric regression model for a one-year horizon is then given by the location-scale model

$$Y_t^{(A)} = m(X_{t-1}) + \nu(X_{t-1})^{1/2}\zeta_t, \tag{2}$$

where

$$m(x) = \mathbb{E}(Y^{(A)}|X = x) \text{ and } \nu(x) = Var(Y^{(A)}|X = x), \; x \in \mathbb{R}^q \tag{3}$$

are unknown smooth functions for the conditional mean and variance, resp., ζ_t are serially uncorrelated zero-conditional-mean random error terms, given the past, with the conditional variance of one, and X_{t-1} is a q-dimensional vector of available explanatory variables.[4]

Our aim is to forecast the conditional variance of excess stock returns $Y_t^{(A)}$ based on model (2) and popular explanatory variables with predictive power reported in the literature, for example, the dividend-by-price ratio, $d_{t-1} = D_{t-1}/P_{t-1}$, the earnings-by-price ratio, $e_{t-1} = E_{t-1}/P_{t-1}$, the short-term interest rate, $r_{t-1} = R_{t-1}/100$, the long-term interest rate, $l_{t-1} = L_{t-1}/100$, inflation, $\pi_{t-1} = (CPI_{t-1} - CPI_{t-2})/CPI_{t-2}$, the term spread, $s_{t-1} = l_{t-1} - r_{t-1}$, and lagged excess stock return, $Y_{t-1}^{(A)}$.

Based on (2), in a two-step procedure, we first estimate $\hat{Y}_t^{(A)} = \hat{m}(X_{t-1})$ as in Kyriakou et al. (2019b), and, in a second step, we estimate $\hat{\nu}(X_{t-1})$ from

$$\nu(x) = \mathbb{E}((Y^{(A)} - m(X))^2|X = x), \; x \in \mathbb{R}^q, \tag{4}$$

using the squared residuals $\hat{\varepsilon}_t^2 := (Y_t^{(A)} - \hat{m}(X_{t-1}))^2$ as the dependent variable and a local-linear smoother in both steps. The estimates $\hat{m}$ and $\hat{\nu}$ depend on smoothing parameters (bandwidths) h and g, respectively. As we are interested in predictions, we take the values which minimize the out-of-sample prediction error using cross-validation. More details are provided in Section 2.4.[5]

[4] Note that the set of explanatory variables in (2) could be different or overlapping for the mean and variance function.

[5] For a description and statistical properties of the local-linear smoother, see, for example, Section 2.3 in Kyriakou et al. (2019b). Note further that the smoothing parameters h and g are separately chosen in each step.

2.2. Longer-Horizon Predictions

For longer horizons T, we consider the sum of annual continuously compounded returns:

$$Z_t^{(A)} = \sum_{i=0}^{T-1} Y_{t+i}^{(A)}.$$

Note that we use here overlapping returns $Z_t^{(A)}$, which require a careful econometric modelling. For illustrative purposes, assume a linear relationship in (2) between $Y_t^{(A)}$ and X_{t-1}, as well as the persistence of the forecasting variable (treating the variables as deviations from their means):

$$Y_t^{(A)} = \beta X_{t-1} + \xi_t \quad \text{and} \quad X_t = \gamma X_{t-1} + \eta_t,$$

with $\xi_t := \nu_\theta(X_{t-1})^{1/2}\zeta_t$ similar to the error term in (2) and a parametric specification for the conditional variance $\nu_\theta(\cdot)$, and η_t being white noise. The T-year regression problem that is implied by this pair of one-year regressions is now

$$\begin{aligned} Z_t^{(A)} &= Y_t^{(A)} + \ldots + Y_{t+T-1}^{(A)} = (\beta X_{t-1} + \xi_t) + \ldots + (\beta X_{t+T-2} + \xi_{t+T-1}) \\ &= \beta \sum_{i=0}^{T-1} \gamma^i X_{t-1} + \beta \sum_{i=0}^{T-1} \sum_{j=0}^{T-1-i} \gamma^j \eta_{t+i} + \sum_{i=0}^{T-1} \xi_{t+i} = \phi X_{t-1} + \psi_t, \end{aligned}$$

i.e., the excess stock return for the year t over the next T years can be decomposed in a predictive part depending on the variable X_{t-1} and an unpredictable error term ψ_t. In estimating the conditional mean and variance functions for the T-year returns $Z_t^{(A)}$, we use nonparametric models because they can capture possible misspecification due to violation of the linear models assumed above. Thus, we set up our predictive nonparametric regression model in the same fashion as in (2)

$$Z_t^{(A)} = m(X_{t-1}) + \nu(X_{t-1})^{1/2}\omega_t, \tag{5}$$

where

$$m(x) = \mathbb{E}(Z^{(A)}|X = x) \text{ and } \nu(x) = Var(Z^{(A)}|X = x),\ x \in \mathbb{R}^q \tag{6}$$

are the unknown smooth conditional mean- and variance-function. The predictive variables X under consideration are the same as for the one-year horizon. The important difference between Equations (2) and (5) is now that the error process $\psi_t := \nu(X_{t-1})^{1/2}\omega_t$ in Equation (5) will be serially correlated by construction.[6,7] For a discussion on asymptotic properties of our nonparametric estimators of model (5) and (6), see Section 2.3 in Kyriakou et al. (2019b).

[6] Our flexible location-scale model in (5), could be easily extended to time-lags of higher order. However, in the empirical application in Section 3, we see that, for example, for real-earnings—the main driver of real-returns—an AR1-type model is ideally suited. This is in line with findings from Kothari et al. (2006). Note further that one might expect risk and return to be somehow related (see, for example, Merton 1973). The parametric GARCH-in-Mean process captures this idea (Linton and Yan 2011). However, the inclusion of an interaction of mean and variance in a fully nonparametric fashion is out of the scope of this paper. To our knowledge, only semiparametric versions where either the mean or variance function is modeled parametrically can be found in the literature, see, for example, Linton and Perron (2003); Pagan and Hong (1991); Pagan and Ullah (1988).

[7] For possible solutions to the problem of autocorrelation, see, for example, Xiao et al. (2003), Su and Ullah (2006), Linton and Mammen (2008), or more recently Geller and Neumann (2018). The implementation and analysis of these techniques remain for future research. In our approach, we account for autocorrelation in the validation criterion with a leave-k-out strategy, where $k = 2T - 1$; see Section 2.4.

Based on (5), our two-step procedure consists now of, first, estimating $\hat{Z}_t^{(A)} = \hat{m}(X_{t-1})$, and second, estimating $\hat{v}(X_{t-1})$ from

$$\nu(x) = \mathbb{E}((Z^{(A)} - m(X))^2 | X = x), \; x \in \mathbb{R}^q, \tag{7}$$

using the squared residuals $\hat{\varepsilon}_t^2 := (Z_t^{(A)} - \hat{m}(X_{t-1}))^2$ as the dependent variable and a local-linear smoother again in both steps.

2.3. Alternative Ways in Estimating the Conditional Variance Function

For the estimation of the conditional variance or volatility function of a response variable Y in a location-scale model similar to (2) or (5), four different approaches are mainly proposed in the literature: the direct, the residual-based, the likelihood-based, and the difference-sequence method.

(i) The direct method uses the variance expressed as the difference of the first two conditional moments (see, for example, Härdle and Tsybakov 1997): $Var(Y|X = x) = \mathbb{E}(Y^2|X = x) - \mathbb{E}(Y|X = x)^2$. Both parts of the right-hand side are separately estimated and, thus, the result is not necessarily nonnegative and also not fully adaptive to the mean function.[8]

(ii) The residual-based method consists of two stages—first, estimating the conditional mean function $m(\cdot)$ and calculating the squared residuals $\hat{\varepsilon}^2 = (Y - \hat{m}(X))^2$. Second, estimating the conditional variance function $\nu(\cdot)$ by regressing $\hat{\varepsilon}^2$ on a set of explanatory variables X. There exist different variants of residual based methods for the second step.[9]

(iii) The preferred estimators of Yu and Jones (2004) build on a localised normal likelihood and use a standard local-linear form for estimating the mean, a local log-linear form for estimating the variance, and allow for separating bandwidths for mean and variance estimation.

(iv) Finally, examples for the difference-sequence method in a fixed design can be found for the homoscedastic case in Wang and Yu (2017) and citations therein. Wang et al. (2008) analyse for the heteroscedastic case the effect of the unknown (smooth) mean function on the estimation of the variance function. They also compare the performance of the residual-based estimators to a first-order-difference-based estimator. Their results indicate that it is not desirable to estimate the variance function based on the residuals from an optimal estimator of the mean in case the mean function is not smooth. Wang et al. (2008) recommend instead an estimator for the mean with minimal bias.

In the empirical part of this paper in Section 3, we show the results of the residual-based method applying a local-linear kernel smoother in both stages. As a robustness check, we have implemented in the second step the local-exponential estimator (Ziegelmann 2002) and the combined estimator (Mishra et al. 2010) getting almost always very similar results.[10] We do not consider: (i) the direct method, since it is not fully adaptive to the mean function, (ii) the re-weighted local constant estimator (Xu and Phillips 2011) due to its asymptotic similarity to the local-linear method, (iii) the method based on the assumption of normal error terms (Yu and Jones 2004), since skewness and excess kurtosis are common properties of stock returns, and (iv) the difference-sequence method, since it was not convincingly performing in a small sample study, the mean functions are rather smooth in our problem, and bias reduction is key due to sparsity.[11]

[8] It does not estimate the volatility function as efficiently as if the true mean were known.

[9] Examples of these variants are: (i) Applying a local-linear kernel smoother in both stages (Fan and Yao 1998). The result is again not necessarily nonnegative but asymptotically fully adaptive to the unknown mean function. (ii) Using the local exponential estimator to ensure nonnegativity (Ziegelmann 2002). (iii) Implementing a combined estimator (a multiplicative bias reduction technique), where a parametric guide captures some roughness features of the unknown variance function (Glad 1998; Mishra et al. 2010). (iv) Utilising a re-weighted local constant estimator maximising the empirical likelihood such that it becomes a bias-reducing moment restriction (Xu and Phillips 2011).

[10] Those results are available upon request by the authors.

[11] There is also a lack of studies using the difference-sequence method in a random design and in multivariate problems as in our case.

2.4. The Validation Criterion for the Choice of Smoothing Parameters and Model Selection

For the nonparametric technique applied in this study, we require an adequate measure of predictive power. In-sample measures, such as the classical R^2 or the adjusted R^2, are not appropriate because they either prefer the most complex model or need a degrees of freedom adjustment which is an unclear concept in nonparametric estimation. Furthermore, our focus lies on prediction. Thus, we are interested in the out-of-sample performance of a model and not in how well it explains the variation inside the sample. Therefore, our preferred measure estimates the prediction error directly.

For the purpose of model selection and optimal bandwidth choice, we use the validated R_V^2 introduced in the actuarial literature by Nielsen and Sperlich (2003) and based on a leave-k-out cross-validation. Note that this criterion is very similar to the forecast content function of Galbraith (2003) and Galbraith and Kisinbay (2005) defined as the proportionate reduction in the mean square forecast error achievable relative to the unconditional mean forecast.

Our validation criteria for the first and second step are defined as

$$R^2_{V,m} = 1 - \frac{\sum\limits_t (Z_t^{(A)} - \hat{m}_{-t})^2}{\sum\limits_t (Z_t^{(A)} - \bar{Z}_{-t}^{(A)})^2} \quad \text{and} \quad R^2_{V,\nu} = 1 - \frac{\sum\limits_t (\hat{\varepsilon}_t^2 - \hat{\nu}_{-t})^2}{\sum\limits_t (\hat{\varepsilon}_t^2 - \overline{\hat{\varepsilon}^2_{-t}})^2}. \tag{8}$$

Note that leave-k-out estimators are used: $\hat{m}_{-t}$ and $\hat{\nu}_{-t}$ for the nonparametric functions m and ν, resp., $\bar{Z}_{-t}^{(A)}$ and $\overline{\hat{\varepsilon}^2_{-t}}$ for the unconditional mean of $Z_t^{(A)}$ and $\hat{\varepsilon}_t^2$, resp. These are computed by removing $k = 2T - 1$ observations: $(T - 1)$ before the tth time point, t itself, and $(T - 1)$ after t. We need to exclude $k = 2T - 1$ data points due to the construction of the dependent variable over a horizon of T years, i.e., we use for the one-year horizon the classical leave-one-out estimator, while, for example, for the five-year horizon the leave-nine-out estimator. Note that the validated R_V^2 measures the predictive power of a model in comparison to the predictive power of the cross-validated historical mean. Thus, positive values imply that the regression model based on explanatory variables outperforms the corresponding historical average over T years. Negative values in the first step of our approach suggest that the historical mean return should be preferred over a model-based approach, while negative values in the second step indicate a constant homoscedastic conditional variance forecast. Note further that the numerator in the ratio of $R^2_{V,m}$ and $R^2_{V,\nu}$ corresponds to the classical cross-validation criterion. Thus, choosing the bandwidth which minimizes this criterion for a given set of explanatory variables is equivalent in maximizing the validated R_V^2. This means that we can use the validated R_V^2 as a single criterion for both purposes: model and bandwidth selection.[12]

It is well known from the literature that cross-validation often requires to omit more than one observation and, possibly, additional correction when the omitted fraction of data are considerable (see, for example, Burman et al. 1994). In addition, when serial correlation arises, as in our longer-horizon application, and the structure of the error terms is ignored, De Brabanter et al. (2011) show that automatic methods for the choice of smoothing parameters, such as cross-validation or plug-in, fail. The problem is that the chosen bandwidths become smaller for increasing correlations (Opsomer et al. 2001), and the corresponding model fits become progressively more under-smoothed. The bias of the predictor reduces this way and, as it contributes in a squared fashion to the prediction mean squared error—the numerator of the ratio in (8), R_V^2 increases (not because the fit is good but due to the ignored correlation structure). A misleading decision on the bandwidth or model specification, as well the set of preferred covariates is the consequence. To overcome those problems, Chu and Marron (1991) propose the use of bimodal kernel functions. Such functions are known to remove the correlation structure very effectively, but the estimator $\hat{m}$ suffers from increased mean squared error, as discussed in De Brabanter et al. (2011). They also propose correlation-corrected

[12] Model selection in the sense of composition of the set of explanatory variables.

cross-validation that consists of, first, finding the amount of data k to be left out in the estimation process when a bimodal kernel function is used; and, second, applying the actual choice of the smoothing parameter using leave-k-out cross-validation with a unimodal kernel function. In our application, we can skip the first step because k is known by construction. For example, in the five-year case, we have $Z_t^{(A)} = Y_t^{(A)} + \ldots + Y_{t+4}^{(A)}$. Now, we want to exclude the complete information included at time t, i.e., skip all $Z_s^{(A)}$ that include any of $Y_t^{(A)}, \ldots, Y_{t+4}^{(A)}$; it is easy to see that this amounts to a leave-nine-out set of $Z_{t-4}^{(A)}, \ldots, Z_{t+4}^{(A)}$ (see, for example, Kyriakou et al. 2019b, Figure 1).

2.5. A Bootstrap-Test: No Predictability vs. Predictability of the Conditional Variance

We test the null of no predictability of the conditional variance applying the tests proposed by Kreiss et al. (2008) (hereafter KNY-test) and Scholz et al. (2015) (hereafter SNS-test). Formally, this is equivalent to say that, under the null, ν is a constant function, which essentially corresponds to the historical average of the squared residuals, i.e., constant volatility. In particular, let $\nu(\cdot)$ be the true volatility function as in (2) or (5) for some specified set of regressors X_t, i.e., (4) or (7) holds. Let $\overline{\hat{\varepsilon}^2}$ be the sample mean of the squared residuals from step one in our approach. The KNY-test is based on the distance

$$\int \left|\nu\left(x\right) - \overline{\hat{\varepsilon}^2}\right|^2 w\left(x\right) dx, \tag{9}$$

for some weighting function w, which has been studied by several authors and statistics have been derived from the above, for example, in Härdle and Mammen (1993) or Kreiss et al. (2008). We use the statistic derived in Equation 2.3 of Kreiss et al. (2008)

$$h^{q/2} T \int \left| \frac{1}{T} \sum_{t=1}^{T} K_h\left(x - X_t\right) \left(\hat{\varepsilon}_t^2 - \overline{\hat{\varepsilon}^2}\right) \right|^2 w\left(x\right) dx, \tag{10}$$

where $K_h\left(x\right)$ is a symmetric kernel smoother with bandwidth h. The bandwidth is selected using R_V^2 for the Nadaraya–Watson kernel estimator rather than a local-linear one. We choose w to be proportional to the uniform density with support in the range of the sample data and replace integration by the mean over uniform independent observations $X_1', X_2', \ldots, X_N'$ in the range of the data:

$$\tau := \frac{h^{q/2} T}{N} \sum_{i=1}^{N} \left| \frac{1}{T} \sum_{t=1}^{T} K_h\left(X_i' - X_t\right) \left(\hat{\varepsilon}_t^2 - \overline{\hat{\varepsilon}^2}\right) \right|^2 . \tag{11}$$

Then, the error in the integral is $O\left(N^{-1/2}\right)$ (Geweke 1996). Under the null, the above test statistic τ is small. This choice could lead to a statistic whose power is lower than the one in Härdle and Mammen (1993) due to some implicit over-smoothing resulting in the weight function w (see comment in Kreiss et al. 2008, just after their Equation 2.5). Power may also improve by using a local-linear smoother in the test. However, the theory for this has not been developed yet, so we refrain from such extension.

Critical values for τ are best derived via wild bootstrap (Härdle and Mammen 1993). For the bootstrap critical values to be consistent, the procedure needs to be independent of whether the null is true or not. Hence, in correspondence with Equation 2.10 in Kreiss et al. (2008), for $b = 1, \ldots, B$,

$$\tau^b := \frac{h^{q/2} T}{N} \sum_{i=1}^{N} \left| \frac{1}{T} \sum_{t=1}^{T} K_h\left(X_i' - X_t\right) \left[u_t^b \left(\hat{\varepsilon}_t^2 - \hat{\nu}\left(X_t\right)\right)\right] \right|^2 , \tag{12}$$

where the u_t^b's are independent and identically distributed random variables with a mean of zero and a variance of one, for example, $u_t^b \sim N(0,1)$. To decide if we reject or not, we use as critical values the corresponding quantiles of the empirical distribution[13],

$$F^*(\tau) = \frac{1}{B}\sum_b \mathbb{1}_{\{\tau^b \leq \tau\}}. \tag{13}$$

The consistency of the procedure for stationary sequences is given in Kreiss et al. (2008).

An alternative version for a wild bootstrap test is the SNS-test proposed in Scholz et al. (2015). There the B bootstrap samples are constructed using the residuals under the null, $\iota_t^0 := \hat{\varepsilon}_t^2 - \overline{\hat{\varepsilon}^2}$, and u_t^b's as above, such that

$$\hat{\varepsilon}_t^{2,b} = \overline{\hat{\varepsilon}^2} + \iota_t^0 \cdot u_t^b.$$

Then, in each bootstrap repetition b, the cross-validated mean is calculated of the $\hat{\varepsilon}_t^{2,b}$, $t = 1, \ldots, T$, as well the estimates of the predictor-based model $\hat{v}_{-t}^b$ in order to get $R_{V,\nu}^{2,b}$ like in (8). Critical values are chosen from corresponding quantiles of the empirical distribution function similar to (13).

Both tests have their own merits. We expect the KNY-test to be more conservative and potentially with less power in comparison to the SNS-test but with clear and well-established asymptotic theory. For more discussion on standard smoothing based tests and other examples for tests of the variance function, see, for example, the survey of Gonzales-Manteiga and Crujeiras (2013).

3. Empirical Application: Conditional Variance Prediction for Stock Returns in Excess of Different Benchmarks

3.1. The Data

In this paper, we extend the analysis of Kyriakou et al. (2019b), who considered the forecasting of long-term stock returns, to conditional variance predictions. Thus, we base our predictions on the same annual US data set which is provided by Robert Shiller and can be downloaded from http://www.econ.yale.edu/~shiller/data.htm. It includes, among other variables, the Standard and Poor's (S&P) Composite Stock Price Index, the consumer price index, and interest rate data from 1872 to 2019. We use here an updated and revised version of Shiller (1989, chp. 26), which provides a detailed description of the data. Note that the risk-free rate in this data set (based on the six-month commercial paper rate until 1997 and afterwards on the six-month certificate of deposit rate, secondary market) was discontinued in 2013. We follow the strategy of Welch and Goyal (2008) and replace it by an annual yield that is based on the six-month Treasury-bill rate, secondary market, from https://fred.stlouisfed.org/series/TB6MS. This new series is only available from 1958 to 2019. In the absence of information prior to 1958, we had to estimate it. To this end, we regressed the Treasury-bill rate on the risk-free rate from Shiller's data for the overlapping period 1958 to 2013, which yielded

$$\text{Treasury-bill rate} = 0.0961 + 0.8648 \times \text{commercial paper rate}$$

with an R^2 of 98.6%. Therefore, we instrumented the risk-free rate from 1872 to 1957 with the predicted regression equation. The correlation between the actual Treasury-bill rate and the predictions for the estimation period is 99.3%. Table 1 displays standard descriptive statistics for one-year and five-year returns as well as the available covariates.

[13] The symbol $\mathbb{1}_A$ denotes the indicator function of an appropriate condition A, i.e., it is one when A is true and zero otherwise.

Table 1. US market data (1872–2019).

	Max	Min	Mean	Sd	Skew	Exc. kurt
S&P stock price index P	2789.80	3.25	277.58	558.13	2.43	5.50
Dividend accruing to index D	53.75	0.18	6.04	10.56	2.45	6.00
Earnings accruing to index E	132.39	0.16	13.96	26.31	2.43	5.35
Dividend-by-price d	9.88	1.17	4.31	1.71	0.46	0.25
Earnings-by-price e	17.75	1.72	7.28	2.75	1.05	1.39
Short-term interest rate r	14.93	0.07	3.97	2.50	0.96	2.34
Long-term interest rate l	14.59	1.88	4.53	2.27	1.81	3.63
Inflation π	20.69	−15.65	2.23	5.96	0.26	1.60
Spread s	3.64	−3.71	0.56	1.32	−0.05	0.02
One-year excess stock returns $Y^{(R)}$	42.39	−58.26	4.58	17.28	−0.57	0.68
One-year excess stock returns $Y^{(C)}$	54.04	−48.81	6.41	18.05	−0.40	0.64
Five-year excess stock returns $Z^{(R)}$	107.27	−78.54	23.49	36.69	−0.14	−0.37
Five-year excess stock returns $Z^{(C)}$	122.96	−57.34	32.34	36.42	−0.05	−0.40

3.2. Single Benchmarking Approach

In this section, we consider a single benchmarking approach as in Kyriakou et al. (2019a, 2019b), i.e., only the dependent variable S_t is benchmark adjusted, as shown in (1), while the independent variable(s) is (are) measured on the original (nominal) scale. The models (2) and (5) are estimated in both steps with a local-linear kernel smoother using the quartic kernel. The optimal bandwidths are chosen by cross-validation, i.e., by maximizing the corresponding validation measure given by (8). Given that we apply a local-linear smoother, it should be kept in mind that the nonparametric method can estimate linear functions without any bias. Thus, the linear model is automatically embedded in our approach. This is an important observation as the linear model is the usual benchmark in financial applications. In addition, in case that the true (but in advance) unknown function is really linear, our approach would exactly pick the line against all other functional alternatives. We study the $R^2_{V,\nu}$ values based on different validated scenarios shown for the one-year horizon in Table 2 and the five-year horizon in Table 3. Here, the same predictive variables X_{t-1} are used in both steps of our approach. Note that we have only about 150 observations in our records. The small sample size clearly limits the complexity of our analysis in the sense of using higher dimensional vectors of explanatory variables. In what follows, we consider only one- and two-dimensional models. For a discussion on sparsely distributed annual observations in higher dimensions and ways to circumvent the curse-of-dimensionality, see, for example, Kyriakou et al. (2019a).

Overall, we find for the one-year horizon that only a few variables have small positive validated $R^2_{V,\nu}$'s and thus possibly some low explanatory power. For example, for the benchmarks $B^{(R)}$, $B^{(L)}$, and $B^{(E)}$, the excess stock return has the largest validated $R^2_{V,\nu}$ values for one-dimensional models (2.2%, 2.4%, and 1.5%). This finding would support an ARCH-type variance structure. For the inflation benchmark $B^{(C)}$, the model with the long-term interest rate produces the largest validated $R^2_{V,\nu}$ of 0.5%. When we apply the bootstrap tests introduced in Section 2.5, the KNY-test does not reject the null of no predictability for all cases at the 5%-level. The SNS-test rejects the null only for the $Y^{(A)}_{t-1}$ covariate under the benchmarks $B^{(R)}$, $B^{(L)}$ and $B^{(E)}$ at the 5%-level.[14] Note that the two-dimensional models do not add predictive power as the validated $R^2_{V,\nu}$ values remain in the same low range.

[14] The tests were conducted with 1000 repetitions at the 5% significance level for a selected number of cases. We do not present the p-values of the tests to save space. The results are available upon request by the authors.

Table 2. Predictive power for the variance of one-year excess stock returns $Y_t^{(A)}$: the single benchmarking approach. The prediction problem is defined in (2). The same predictive variables X_{t-1} are used in the predictions for the conditional mean and variance function. The predictive power (%) is measured by $R^2_{V,\nu}$ as defined in (8). The benchmarks $B^{(A)}$ considered are based on the short-term interest rate ($A \equiv R$), long-term interest rate ($A \equiv L$), earnings-by-price ratio ($A \equiv E$), and consumer price index ($A \equiv C$). The predictive variables used are X_{t-1}, given by the dividend-by-price ratio d_{t-1}, earnings-by-price ratio e_{t-1}, short-term interest rate r_{t-1}, long-term interest rate l_{t-1}, inflation π_{t-1}, term spread s_{t-1}, excess stock return $Y_{t-1}^{(A)}$, or the possible different pairwise combinations as indicated.

Benchmark $B^{(A)}$	**Explanatory Variable(s) X_{t-1}**						
	$Y^{(A)}$	d	e	r	l	π	s
Short-term rate	2.2	−1.1	−0.6	−0.3	0.3	−1.2	−0.1
Long-term rate	2.4	−1.2	−0.6	0.3	0.6	−1.4	−0.1
Earnings-by-price	1.5	−1.3	−0.7	−0.1	0.5	−1.4	0.1
Inflation	0.2	0.1	−1.3	−0.4	0.5	−1.2	−0.6
	$(Y^{(A)}, d)$	$(Y^{(A)}, e)$	$(Y^{(A)}, r)$	$(Y^{(A)}, l)$	$(Y^{(A)}, \pi)$	$(Y^{(A)}, s)$	
Short-term rate	2.4	1.9	1.1	2.2	0.1	0.3	
Long-term rate	1.5	1.4	1.1	2.1	−0.2	0.1	
Earnings-by-price	1.6	1.4	0.9	2.0	−0.2	0.1	
Inflation	−1.0	−1.1	−0.6	0.6	−2.1	−1.0	
	(d, e)	(d, r)	(d, l)	(d, π)	(d, s)		
Short-term rate	−2.1	−1.5	−0.8	−2.4	−1.5		
Long-term rate	−2.0	−1.1	−0.6	−2.2	−1.5		
Earnings-by-price	−1.9	−1.4	−0.7	−2.3	−1.5		
Inflation	−0.4	−1.0	−0.2	−2.3	−1.3		
	(e, r)	(e, l)	(e, π)	(e, s)			
Short-term rate	−1.0	−0.4	−2.3	−0.8			
Long-term rate	−0.6	−0.2	−2.2	−0.8			
Earnings-by-price	−1.0	−0.2	−2.2	−0.8			
Inflation	−1.7	−0.9	−2.2	−1.6			
	(r, l)	(r, π)	(r, s)				
Short-term rate	1.3	−1.5	1.4				
Long-term rate	1.3	−1.0	1.4				
Earnings-by-price	1.4	−1.5	1.6				
Inflation	1.3	−1.5	1.2				
	(l, π)	(l, s)					
Short-term rate	−1.2	1.4					
Long-term rate	−0.9	1.4					
Earnings-by-price	−1.0	1.6					
Inflation	−0.9	1.3					
	(π, s)						
Short-term rate	0.2						
Long-term rate	0.2						
Earnings-by-price	−0.6						
Inflation	−0.1						

Contrary to the mean prediction, where Kyriakou et al. (2019b) find that five-year predictability improves over the one-year case, we observe that the majority of predictor based volatility models do not surpass the constant volatility alternative for the five-year horizon. Even though some models produce small positive $R^2_{V,\nu}$ values, this time both the SNS- and the KNY-test do not reject the null of no predictability. Note that our results are in line with Christoffersen and Diebold (2000) who conclude that volatility forecastability may be much less important at longer horizons.

Table 3. Predictive power for the variance of five-year excess stock returns $Z_t^{(A)}$: the single benchmarking approach. The prediction problem is defined in (5). The same predictive variables X_{t-1} are used in the predictions for the conditional mean and variance function. Additional notes: see Table 2.

Benchmark $B^{(A)}$	**Explanatory Variable(s) X_{t-1}**						
	$Y^{(A)}$	d	e	r	l	π	s
Short-term rate	0.6	−1.7	−1.7	−1.2	−1.0	−2.0	−3.0
Long-term rate	0.0	−1.5	−1.3	−1.2	−1.1	−1.2	−2.7
Earnings-by-price	0.8	−1.8	−1.1	−1.8	−2.7	−0.3	−3.8
Inflation	−1.0	−3.8	−4.7	−0.7	−1.5	1.4	0.5
	$(Y^{(A)}, d)$	$(Y^{(A)}, e)$	$(Y^{(A)}, r)$	$(Y^{(A)}, l)$	$(Y^{(A)}, \pi)$	$(Y^{(A)}, s)$	
Short-term rate	−2.8	−2.5	−1.7	−1.7	−1.7	−3.9	
Long-term rate	−2.5	−2.1	−1.6	−1.8	−1.2	−3.4	
Earnings-by-price	−2.3	−2.1	−1.2	−4.1	0.4	−3.4	
Inflation	−5.1	−4.7	−1.5	−2.6	0.4	−0.9	
	(d, e)	(d, r)	(d, l)	(d, π)	(d, s)		
Short-term rate	−3.6	−3.1	−2.2	−2.8	−4.1		
Long-term rate	−3.1	−3.2	−2.7	−2.3	−4.3		
Earnings-by-price	−4.1	−4.0	−5.3	−2.3	−4.9		
Inflation	−5.2	−5.0	−8.9	−2.5	−3.2		
	(e, r)	(e, l)	(e, π)	(e, s)			
Short-term rate	−3.3	−3.3	−3.5	−4.9			
Long-term rate	−2.8	−3.3	−2.9	−4.9			
Earnings-by-price	−4.5	−5.5	−2.7	−6.5			
Inflation	−8.5	−7.8	−4.9	−6.4			
	(r, l)	(r, π)	(r, s)				
Short-term rate	−3.8	−1.7	−3.9				
Long-term rate	−4.1	−1.3	−4.2				
Earnings-by-price	−5.3	−1.9	−5.4				
Inflation	−3.9	0.3	−1.9				
	(l, π)	(l, s)					
Short-term rate	−1.7	−3.9					
Long-term rate	−1.3	−4.2					
Earnings-by-price	−2.6	−5.4					
Inflation	−1.2	−1.8					
	(π, s)						
Short-term rate	−4.4						
Long-term rate	−3.5						
Earnings-by-price	−4.8						
Inflation	−0.1						

3.3. Full Benchmarking Approach

In the next step, we consider the double benchmarking approach of Kyriakou et al. (2019a, 2019b) to analyze now whether transforming the explanatory variables can improve the predictions for the volatility function. Recall that fully nonparametric models suffer in general by the curse of dimensionality. Problems with sparsely distributed annual observations in higher dimensions, as in our framework, could be reduced or circumvented by importing more structure in the estimation process.

Here, we extend the study presented in Section 3.2 transforming both the dependent and independent variables according to the same benchmark. To this end, in our full (double) benchmarking approach, the prediction problems are reformulated as

$$Y_t^{(A)} = m(X_{t-1}^{(A)}) + \nu(X_{t-1}^{(A)})^{1/2}\zeta_t, \tag{14}$$

$$Z_t^{(A)} = m(X_{t-1}^{(A)}) + \nu(X_{t-1}^{(A)})^{1/2}\omega_t, \tag{15}$$

where we use transformed predictive variables

$$X_{t-1}^{(A)} = \begin{cases} \frac{1+X_{t-1}}{B_{t-1}^{(A)}}, \quad X \in \{d, e, r, l, \pi\} \\ \frac{s_{t-1}}{B_{t-1}^{(A)}} = \frac{l_{t-1}-r_{t-1}}{B_{t-1}^{(A)}} \\ Y_{t-1}^{(A)} \end{cases}, \quad A \in \{R, L, E, C\}. \tag{16}$$

This approach can be interpreted as a simple way of reducing the dimensionality of the estimation procedure. The adjusted variable $X_{t-1}^{(A)}$ includes now an additional predictive variable, the benchmark itself. Results of this empirical study are presented for the one-year horizon in Table 4 and for the five-year horizon in Table 5.

We find that, in comparison to the single-benchmarking approach in the one-year case, the double benchmarking improves in 15 out of 82 models (in the sense of producing a positive and higher $R^2_{V,\nu}$ as before). However, predictability is still questionable. The best model under the long-term interest rate benchmark $B^{(L)}$ uses the pair $(Y_{t-1}^{(L)}, e_{t-1}^{(L)})$ and yields $R^2_{V,\nu} = 3.0$, while the best model under $B^{(E)}$ uses the pair $(Y_{t-1}^{(E)}, l_{t-1}^{(E)})$ and yields $R^2_{V,\nu} = 2.5$. The SNS-test rejects for both the null of no predictability, while the KNY-test does not. For the rest of the new combinations of predictive variables in all benchmarks, both tests again do not reject.

For the five-year case, we find that in comparison to the single-benchmarking the double benchmarking improves in 11 out of 82 models. The best model under $B^{(E)}$ uses $d_{t-1}^{(E)}$ and yields $R^2_{V,\nu} = 1.8$, while under $B^{(C)}$ the covariates $d_{t-1}^{(C)}$ and $l_{t-1}^{(C)}$ both yield $R^2_{V,\nu} = 1.6$. Nevertheless, we do not find any combination of covariates with statistically significant predictive power.

Table 4. Predictive power for the variance of one-year excess stock returns $Y_t^{(A)}$: the double benchmarking approach. The prediction problem is defined in (14). The same predictive variables $X_{t-1}^{(A)}$ are used in the predictions for the conditional mean and variance. The predictive power (%) is measured by $R^2_{V,\nu}$ as defined in (8). The benchmarks $B^{(A)}$ considered are based on the short-term interest rate ($A \equiv R$), long-term interest rate ($A \equiv L$), earnings-by-price ratio ($A \equiv E$), and consumer price index ($A \equiv C$). The predictive variables used are $X_{t-1}^{(A)}$ using the indicated benchmark $B_{t-1}^{(A)}$ as shown in (16). X_{t-1} are given by the dividend-by-price ratio d_{t-1}, earnings-by-price ratio e_{t-1}, short-term interest rate r_{t-1}, long-term interest rate l_{t-1}, inflation π_{t-1}, term spread s_{t-1}, excess stock return $Y_{t-1}^{(A)}$, or the possible different pairwise combinations as indicated. "–" are not applicable cases of matched covariate with benchmark. Note: $s^{(R)}$ and $l^{(R)}$ (and their combinations with Y, d, e, π) have the same R^2_V by construction of the transformed spread according to (16). For example, $s_{t-1}^{(R)} = (l_{t-1} - r_{t-1})/B_{t-1}^{(R)} = (1 + l_{t-1})/(1 + r_{t-1}) - 1$ and $l_{t-1}^{(R)} = (1 + l_{t-1})/(1 + r_{t-1})$. The case of $s^{(L)}$ and $r^{(L)}$ is similar.

Benchmark $B^{(A)}$	**Explanatory Variable(s) X_{t-1}**						
	$Y^{(A)}$	$d^{(A)}$	$e^{(A)}$	$r^{(A)}$	$l^{(A)}$	$\pi^{(A)}$	$s^{(A)}$
Short-term rate	2.2	−0.3	0.7	–	−0.2	0.1	−0.2
Long-term rate	2.4	0.2	−0.5	−0.1	–	−0.2	−0.1
Earnings-by-price	1.5	−0.2	–	0.6	−0.2	−0.7	0.0
Inflation	0.2	−0.9	−1.2	−0.3	−0.2	–	−0.7
	$(Y^{(A)}, d^{(A)})$	$(Y^{(A)}, e^{(A)})$	$(Y^{(A)}, r^{(A)})$	$(Y^{(A)}, l^{(A)})$	$(Y^{(A)}, \pi^{(A)})$	$(Y^{(A)}, s^{(A)})$	
Short-term rate	0.8	0.7	–	0.2	0.1	0.2	
Long-term rate	1.3	3.0	0.1	–	−0.3	0.1	
Earnings-by-price	0.2	–	0.7	2.5	0.0	0.1	
Inflation	−3.1	−1.4	−1.5	−1.9	–	−1.0	
	$(d^{(A)}, e^{(A)})$	$(d^{(A)}, r^{(A)})$	$(d^{(A)}, l^{(A)})$	$(d^{(A)}, \pi^{(A)})$	$(d^{(A)}, s^{(A)})$		
Short-term rate	−1.3	–	0.9	0.0	0.9		
Long-term rate	−1.0	0.9	–	−0.7	0.9		
Earnings-by-price	–	−0.3	-0.8	−1.8	0.4		
Inflation	−1.9	0.7	1.6	–	−0.7		
	$(e^{(A)}, r^{(A)})$	$(e^{(A)}, l^{(A)})$	$(e^{(A)}, \pi^{(A)})$	$(e^{(A)}, s^{(A)})$			
Short-term rate	–	−0.4	−2.6	−0.4			
Long-term rate	−0.6	–	−2.5	−0.6			
Earnings-by-price	–	–	–	–			
Inflation	−1.6	−1.5	–	−1.6			
	$(r^{(A)}, l^{(A)})$	$(r^{(A)}, \pi^{(A)})$	$(r^{(A)}, s^{(A)})$				
Short-term rate	–	–	–				
Long-term rate	–	−1.2	–				
Earnings-by-price	−0.5	−2.1	−0.3				
Inflation	−1.9	–	−1.6				
	$(l^{(A)}, \pi^{(A)})$	$(l^{(A)}, s^{(A)})$					
Short-term rate	−1.4	–					
Long-term rate	–	–					
Earnings-by-price	−2.5	−0.5					
Inflation	–	−1.7					
	$(\pi^{(A)}, s^{(A)})$						
Short-term rate	−1.4						
Long-term rate	−1.2						
Earnings-by-price	−1.6						
Inflation	–						

Table 5. Predictive power for the variance of five-year excess stock returns $Z_t^{(A)}$: the double benchmarking approach. The prediction problem is defined in (15). The same predictive variables $X_{t-1}^{(A)}$ are used in the predictions for the conditional mean and variance. Additional notes: see Table 4.

Benchmark $B^{(A)}$	**Explanatory Variable(s) X_{t-1}**						
	$Y^{(A)}$	$d^{(A)}$	$e^{(A)}$	$r^{(A)}$	$l^{(A)}$	$\pi^{(A)}$	$s^{(A)}$
Short-term rate	0.6	−2.2	−3.2	–	−3.1	−3.2	−3.1
Long-term rate	0.0	−3.4	−2.8	−2.8	–	−1.3	−2.8
Earnings-by-price	0.8	1.8	–	−2.3	−3.2	0.6	−3.8
Inflation	−1.0	1.6	0.3	0.6	1.6	–	0.3
	$(Y^{(A)}, d^{(A)})$	$(Y^{(A)}, e^{(A)})$	$(Y^{(A)}, r^{(A)})$	$(Y^{(A)}, l^{(A)})$	$(Y^{(A)}, \pi^{(A)})$	$(Y^{(A)}, s^{(A)})$	
Short-term rate	−2.1	−4.3	–	−4.0	−1.2	−4.0	
Long-term rate	−3.8	−3.2	−3.6	–	−1.1	−3.6	
Earnings-by-price	1.1	–	−2.8	−3.8	−0.5	−3.4	
Inflation	0.3	−0.8	−0.3	0.4	–	−1.0	
	$(d^{(A)}, e^{(A)})$	$(d^{(A)}, r^{(A)})$	$(d^{(A)}, l^{(A)})$	$(d^{(A)}, \pi^{(A)})$	$(d^{(A)}, s^{(A)})$		
Short-term rate	−3.7	–	−5.4	−2.1	−5.4		
Long-term rate	−4.2	−5.8	–	−3.3	−5.8		
Earnings-by-price	–	−0.4	−2.6	0.3	−3.3		
Inflation	−4.3	−0.2	−0.8	–	−0.8		
	$(e^{(A)}, r^{(A)})$	$(e^{(A)}, l^{(A)})$	$(e^{(A)}, \pi^{(A)})$	$(e^{(A)}, s^{(A)})$			
Short-term rate	–	−5.9	−4.9	−5.9			
Long-term rate	−6.1	–	−4.1	−6.1			
Earnings-by-price	–	–	–	–			
Inflation	−4.8	−4.1	–	−2.1			
	$(r^{(A)}, l^{(A)})$	$(r^{(A)}, \pi^{(A)})$	$(r^{(A)}, s^{(A)})$				
Short-term rate	–	–	–				
Long-term rate	–	−2.3	–				
Earnings-by-price	−6.3	−3.2	−6.1				
Inflation	−1.0	–	0.5				
	$(l^{(A)}, \pi^{(A)})$	$(l^{(A)}, s^{(A)})$					
Short-term rate	−3.4	–					
Long-term rate	–	–					
Earnings-by-price	−3.6	−6.2					
Inflation	–	0.5					
	$(\pi^{(A)}, s^{(A)})$						
Short-term rate	−3.4						
Long-term rate	−2.3						
Earnings-by-price	−4.6						
Inflation	–						

3.4. Real-Income Long-Term Pension Prediction

In long-term pension planning or other asset allocation problems optimized with regard to real-income protection (Gerrard et al. (2019a, 2019b); (Merton 2014)), the econometric models should reflect those needs and use covariates net-of-inflation. Therefore, we take the inflation benchmark $B^{(C)}$ and analyse in more detail the best model found by Kyriakou et al. (2019b), which uses the earnings-by-price variable for the mean prediction and produced a $R^2_{V,m} = 12.2$ for the one-year horizon and $R^2_{V,m} = 12.4$ for the five-year horizon (see Kyriakou et al. 2019b, Tables 4 and 5) in the double benchmarking case. For this specific model, we are now interested in finding the set of covariates that best predicts the conditional variance.[15,16] The empirical findings in terms of $R^2_{V,\nu}$ are shown for the one-year horizon in Table 6 and the five-year horizon in Table 7. For the one-year horizon, we find in the double benchmarking approach when inflation is the benchmark, $B^{(C)}$ that the dividend-by-price $d^{(C)}$ together with the short-term interest-rate $r^{(C)}$ or the long-term interest-rate $l^{(C)}$ are chosen as best predictive variables in terms of $R^2_{V,\nu}$ (2.9% and 2.0%). Note that these values are rather low and that the SNS-test does reject the null of no predictability for both models, while the KNY-test does not reject. For all other combinations and also the five-year case, we do not find evidence for statistical significant predictability of the conditional variance. Therefore, we conclude that the constant volatility model is appropriate for practical purposes.

Note further that the ratio in our validation criterion for the mean prediction, $R^2_{V,m}$, in (8) compares the sample variance of the estimated residuals from our model based on earnings-by-price (the numerator) with the sample variance of the benchmarked stock returns (the denominator). For the one-year case, we find from Table 1 the latter to be equal to $0.1805^2 = 0.03258$. A simple calculation using the corresponding $R_{V,m} = 12.2\%$ leads then to $0.03258(1-0.122) = 0.02861$ or a standard deviation of 16.91% for returns based on the earnings-model. This means that the linear expression of real stock returns in terms of real earnings-by-price presented in Kyriakou et al. (2019b) as

$$\text{Real one-year stock return} = 0.004875 + 1.119 \times \text{real earnings-by-price} \tag{17}$$

gives on average 2.4% higher returns at the same risk as the historical mean $\bar{Y}^{(C)}$.[17] Similarly, for the five-year case, we get from Table 1 that $0.3642^2 = 0.1326$. From the $R_{V,m} = 12.4\%$, we obtain then $0.1326(1-0.122) = 0.1162$ or a standard deviation of 34.08% for returns based on the earnings-model. Thus, the linear expression of real stock returns in terms of real earnings-by-price presented in Kyriakou et al. (2019b) as

$$\text{Real five-year stock return} = 0.2068 + 2.264 \times \text{real earnings-by-price} \tag{18}$$

gives on average 6.1% higher returns at the same risk as the historical mean $\bar{Y}^{(C)}$.[18] Figure 1 shows the estimated nonparametric function $\hat{m}$ (red solid line) for the one-year horizon (left) and the five-year horizon (right) under the double inflation benchmark for the earnings-by-price covariate together with the corresponding historical mean (dashed green line). Figure 2 depicts histograms and a kernel density estimate (red solid line) of the standardized predicted returns for the one-year horizon (left) and the five-year horizon (right). The similarity for both horizons is striking and driven by the fact that the ratio of the slope of the regression lines in (17) and (18) with the corresponding standard deviation given above yields almost the same value of 6.63.

[15] Note that until now we have used the same set of covariates in both steps of our analysis to reduce the overwhelming number of models. It is also clear that not all combinations of variables are practically relevant. Now, we relax this restriction for the model with the highest predictive power for the returns.

[16] Tables 6 and 7 also present the results for the short- and long-term interest benchmarks $B^{(R)}$ and $B^{(L)}$. However, it is again hard to find predictability at all in these cases. Note that the benchmark using the earnings-by-price variable $B^{(E)}$ is not applicable since it matches the covariate and the benchmark in the first step.

[17] Here, we use the Sharpe-ratio for the comparison. From Table 1, we get $\bar{Y}^{(C)} = 6.41\%$ and divide it either by 18.05% or by 16.91%. We obtain 0.355 and 0.379, which corresponds to a difference of 2.4% points.

[18] Here, we use again the Sharpe-ratio for the comparison. From Table 1, we get $\bar{Y}^{(C)} = 32.34\%$ and divide it either by 36.42% or by 34.08%. We obtain 0.888 and 0.949, which corresponds to a difference of 6.1% points.

Finally, we consider a simple mean-reverting autoregressive model of order one for the real earnings-by-price—the main drivers of real returns in Equations (17) and (18)—and estimate it with ordinary least squares (OLS) [19]:

$$\begin{aligned}&\text{Change in real earnings-by-price}\\ &\qquad\qquad = -0.715 \times (\text{real earnings-by-price} - \text{mean of real earnings-by-price}). \qquad (19)\end{aligned}$$

Note that, for the whole sample period (1872–2019), the mean and standard deviation of real earnings-by-price are 0.0524 and 0.0595, resp. Moreover, using the current (30/09/2019) value of real earnings-by-price of 0.0278, model (19) predicts a change in real earnings-by-price of 0.0176, i.e., an expected value of real earnings-by-price of 0.0454 for 2020Q3, which is still below the long-term average.[20]

We subsequently calculate the correlation between the estimated residuals of models (17) and (19) to be −0.014. A standard stationary block-bootstrap (Politis and Romano 1994) based on 10,000 repetitions and a block-length of 12 suggests that this correlation is not statistically significantly different from zero. The correlation structure between returns and their drivers is important while searching for optimal investment strategies in a dynamic market, see Kim and Omberg (1996). Gerrard et al. (2019c) follow the approach of Kim and Omberg (1996) in a long-term return setting and show that the above correlation is very hard to estimate with precision. Sometimes, it is negative and, with a slight change of data, it is positive, and a test would almost always provide that zero correlation cannot be rejected. When this added insight is provided that zero correlation significantly simplifies that technical calculation of the optimal dynamic strategy while significantly reducing parameter uncertainty, the conclusion seems clear: we should work with zero correlation unless there is a strong argument not to do that. In our case—which is a discrete analogue to the continuous models considered in Gerrard et al. (2019c) and Kim and Omberg (1996)—it is, therefore, comforting that we can provide a simple zero-correlation econometric model to guide the market dynamics. In further work, we expect the simple econometric model of this paper to be used while generalizing the non-dynamic new approach to pension products of Gerrard et al. (2019a, 2019b).

[19] The estimated coefficient is significant at the 0.1%-level (with a corresponding standard error of 0.08), the residual standard error of the regression is 0.0572, and its R^2 has a value of 0.357.

[20] The following values are used for the calculation of the current real earnings-by-price: $P = 2976.74$, $E = 135.53$, $B^{(C)} = 1.0173$.

Table 6. Predictive power for the variance of one-year excess stock returns $Y_t^{(A)}$: the double benchmarking approach for the conditional mean model with earnings-by price as single covariate. The prediction problem is defined in (14). The predictive power (%) is measured by $R^2_{V,\nu}$ as defined in (8). The benchmarks $B^{(A)}$ considered are based on the short-term interest rate ($A \equiv R$), long-term interest rate ($A \equiv L$), and consumer price index ($A \equiv C$). The predictive variables used are $X_{t-1}^{(A)}$ using the indicated benchmark $B_{t-1}^{(A)}$ as shown in (16). X_{t-1} are given by the dividend-by-price ratio d_{t-1}, earnings-by-price ratio e_{t-1}, short-term interest rate r_{t-1}, long-term interest rate l_{t-1}, inflation π_{t-1}, term spread s_{t-1}, excess stock return $Y_{t-1}^{(A)}$, or the possible different pairwise combinations as indicated. "–" are not applicable cases of matched covariate with benchmark. Note: $s^{(R)}$ and $l^{(R)}$ (and their combinations with Y, d, e, π) have the same $R^2_{V,\nu}$ by construction of the transformed spread according to (16). For example, $s_{t-1}^{(R)} = (l_{t-1} - r_{t-1})/B_{t-1}^{(R)} = (1 + l_{t-1})/(1 + r_{t-1}) - 1$ and $l_{t-1}^{(R)} = (1 + l_{t-1})/(1 + r_{t-1})$. Similar is the case of $s^{(L)}$ and $r^{(L)}$.

Benchmark $B^{(A)}$	**Explanatory Variable(s) X_{t-1}**						
	$Y^{(A)}$	$d^{(A)}$	$e^{(A)}$	$r^{(A)}$	$l^{(A)}$	$\pi^{(A)}$	$s^{(A)}$
Short-term rate	1.0	0.3	0.7	–	0.1	−0.4	0.1
Long-term rate	1.4	0.1	−0.5	0.9	–	−0.1	0.9
Inflation	0.4	−0.6	−1.2	−0.4	−0.1	–	0.8
	$(Y^{(A)}, d^{(A)})$	$(Y^{(A)}, e^{(A)})$	$(Y^{(A)}, r^{(A)})$	$(Y^{(A)}, l^{(A)})$	$(Y^{(A)}, \pi^{(A)})$	$(Y^{(A)}, s^{(A)})$	
Short-term rate	0.6	0.7	–	0.2	−0.5	0.2	
Long-term rate	0.7	2.0	0.7	–	−0.6	0.7	
Inflation	−1.7	−1.6	−1.5	−1.7	–	−0.4	
	$(d^{(A)}, e^{(A)})$	$(d^{(A)}, r^{(A)})$	$(d^{(A)}, l^{(A)})$	$(d^{(A)}, \pi^{(A)})$	$(d^{(A)}, s^{(A)})$		
Short-term rate	0.0	–	−0.5	−0.4	−0.5		
Long-term rate	−1.0	0.3	–	−1.4	0.3		
Inflation	−1.9	2.9	2.0	–	1.5		
	$(e^{(A)}, r^{(A)})$	$(e^{(A)}, l^{(A)})$	$(e^{(A)}, \pi^{(A)})$	$(e^{(A)}, s^{(A)})$			
Short-term rate	–	0.5	−2.2	0.5			
Long-term rate	−0.7	–	−2.5	−0.7			
Inflation	−0.9	−1.7	–	−0.3			
	$(r^{(A)}, l^{(A)})$	$(r^{(A)}, \pi^{(A)})$	$(r^{(A)}, s^{(A)})$				
Short-term rate	–	–	–				
Long-term rate	–	−0.4	–				
Inflation	−0.5	–	0.7				
	$(l^{(A)}, \pi^{(A)})$	$(l^{(A)}, s^{(A)})$					
Short-term rate	0.1	–					
Long-term rate	–	–					
Inflation	–	−0.2					
	$(\pi^{(A)}, s^{(A)})$						
Short-term rate	0.1						
Long-term rate	−0.4						
Inflation	–						

Table 7. Predictive power for the variance of five-year excess stock returns $Z_t^{(A)}$: the double benchmarking approach for the conditional mean model with earnings-by price as single covariate. The prediction problem is defined in (15). Additional notes: see Table 6.

Benchmark $B^{(A)}$	**Explanatory Variable(s) X_{t-1}**						
	$Y^{(A)}$	$d^{(A)}$	$e^{(A)}$	$r^{(A)}$	$l^{(A)}$	$\pi^{(A)}$	$s^{(A)}$
Short-term rate	0.1	−1.8	−3.2	–	−4.5	−2.5	−4.5
Long-term rate	0.6	−3.9	−2.8	−4.2	–	−1.1	−4.2
Inflation	0.0	−0.1	0.3	−0.4	−0.1	–	−2.6
	$(Y^{(A)}, d^{(A)})$	$(Y^{(A)}, e^{(A)})$	$(Y^{(A)}, r^{(A)})$	$(Y^{(A)}, l^{(A)})$	$(Y^{(A)}, \pi^{(A)})$	$(Y^{(A)}, s^{(A)})$	
Short-term rate	−1.7	−4.6	–	−5.7	−3.7	−5.7	
Long-term rate	−4.5	−4.5	−4.2	–	−2.5	−4.2	
Inflation	−1.9	−1.8	−1.9	−1.7	–	−3.9	
	$(d^{(A)}, e^{(A)})$	$(d^{(A)}, r^{(A)})$	$(d^{(A)}, l^{(A)})$	$(d^{(A)}, \pi^{(A)})$	$(d^{(A)}, s^{(A)})$		
Short-term rate	−6.2	–	−7.1	−4.3	−7.1		
Long-term rate	−4.5	−7.9	–	−5.2	−7.9		
Inflation	−3.9	−2.1	−3.2	–	−2.8		
	$(e^{(A)}, r^{(A)})$	$(e^{(A)}, l^{(A)})$	$(e^{(A)}, \pi^{(A)})$	$(e^{(A)}, s^{(A)})$			
Short-term rate	–	−8.1	−5.8	−8.1			
Long-term rate	−6.6	–	−4.9	−6.6			
Inflation	−2.8	−3.4	–	−2.6			
	$(r^{(A)}, l^{(A)})$	$(r^{(A)}, \pi^{(A)})$	$(r^{(A)}, s^{(A)})$				
Short-term rate	–	–	–				
Long-term rate	–	−5.7	–				
Inflation	−3.0	–	−3.1				
	$(l^{(A)}, \pi^{(A)})$	$(l^{(A)}, s^{(A)})$					
Short-term rate	−6.5	–					
Long-term rate	–	–					
Inflation	–	−3.0					
	$(\pi^{(A)}, s^{(A)})$						
Short-term rate	−6.5						
Long-term rate	−5.7						
Inflation	–						

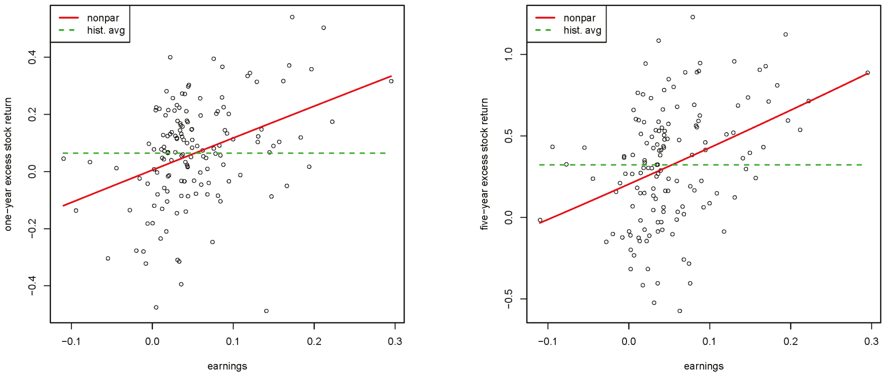


Figure 1. Double inflation benchmark. Relation between real stock returns and real earnings-by-price. Estimated nonparametric function $\hat{m}$ (red solid line) and historical average (dashed green line). **Left**: one-year horizon. **Right**: five-year horizon. Period: 1872–2019. Data: annual S&P 500.

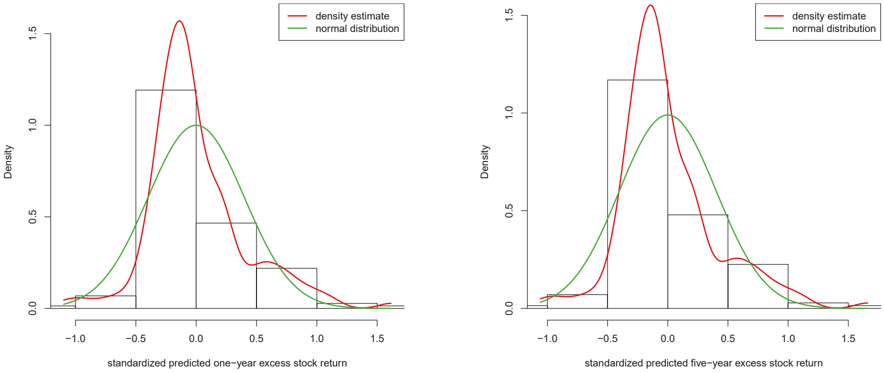


Figure 2. Standardized predicted stock returns in excess of the inflation benchmark (based on the model using earnings-by-price as covariate for mean-prediction; double benchmarking). Histogram, kernel density estimate (red), and fitted normal distribution (green). **Left**: one-year horizon. **Right**: five-year horizon. Period: 1872–2019. Data: annual S&P 500.

4. Conclusions

In this paper, we extend the original working framework of Kyriakou et al. (2019a, 2019b) of forecasting stock returns to modelling their conditional variance and test for predictability in this context. We consider returns of one-year and five-year horizons in excess of different benchmarks, considering the short- and long-term rate, the earnings-by-price ratio, and the inflation rate. We use popular explanatory variables with predictive power such as the dividend-by-price ratio, the earnings-by-price ratio, the short- and long-term interest rates, the term spread, the inflation rate, as well as the lagged excess stock return, in one- and two-dimensional settings, with the returns benchmarked or also the covariates used to predict them.

In our analysis, we find only little to no evidence of model-based volatility predictability for the one-year and five-year horizon. Only for a few of the models considered under different benchmarks, we get validation measures that are positive and significantly different from zero but of a rather small magnitude. We thus conclude that volatility forecastability is much less important at longer horizons regardless of the chosen combination of explanatory variables. The homoscedastic historical average of the squared return prediction errors gives an adequate approximation of the unobserved realised conditional variance for both the one-year and five-year horizon.

In the practically important double inflation benchmarking case, we find that the model with the largest predictive power is not only of linear functional form based on real earnings-by-price but also has a constant variance for both horizons. A simple mean-reverting linear AR1-model for the real-earnings-by-price allows then to analyse the correlation structure between returns and their main drivers. We find zero correlation which significantly simplifies the econometric modelling to guide market dynamics. This is an important observation and a relatively simple starting point when constructing forecasting models for real-value pension prognoses for long-term saving strategies.

Author Contributions: All authors contributed equally to this work.

Funding: The authors thank: (i) the Institute and Faculty of Actuaries in the UK for funding this research through the grant "Minimizing Longevity and Investment Risk while Optimizing Future Pension Plans", and (ii) the University of Graz for the Open Access Funding.

Conflicts of Interest: The authors declare no conflict of interest.

References

Andersen, Torben G., Tim Bollerslev, Francis X. Diebold, and Heiko Ebens. 2001. The distribution of realized stock return volatility. *Journal of Finacial Economics* 61: 43–76. [CrossRef]

Burman, Prabir, Edmond Chow, and Deborah Nolan. 1994. A cross-validatory method for dependent data. *Biometrika* 81: 351–58. [CrossRef]

Chen, Qingqing, and Yongmiao Hong. 2010. *Predictability of Equity Returns Over Different Time Horizons: A Nonparametric Approach*. Working Paper. Ithaca: Cornell University/Department of Economics.

Cheng, Tingting, Jiti Gao, and Oliver Linton. 2019. *Nonparametric Predictive Regressions for Stock Return Predictions*. Cambridge Working Papers in Economics: 1932. Cambridge: Faculty of Economics, University of Cambridge.

Christoffersen, Peter F., and Francis X. Diebold. 2000. How relevant is volatility forecasting for financial risk management? *Review of Economics and Statistics* 82: 12–22. [CrossRef]

Chu, C. K., and J. S. Marron. 1991. Comparison of two bandwidth selectors with dependent errors. *The Annals of Statistics* 19: 1906–18. [CrossRef]

De Brabanter, Kris, Jos De Brabanter, Johan A.K. Suykens, and Bart De Moor. 2011. Kernel regression in the presence of correlated errors. *Journal of Machine Learning Research* 12: 1955–76.

Fan, Jianqing, and Qiwei Yao. 1998. Efficient estimation of conditional variance functions in stochastic regression. *Biometrika* 85: 645–60. [CrossRef]

Galbraith, John W. 2003. Content horizons for univariate time-series forecasts. *International Journal of Forecasting* 19: 43–55. [CrossRef]

Galbraith, John W., and Turgut Kisinbay. 2005. Content horizons for conditional variance forecasts. *International Journal of Forecasting* 21: 249–60. [CrossRef]

Geller, Juliane, and Michael H. Neumann. 2018. Improved local polynomial estimation in time series regression. *Journal of Nonparametric Statistics* 30: 1–27. [CrossRef]

Gerrard, Russell, Munir Hiabu, Ioannis Kyriakou, and Jens Perch Nielsen. 2019a. Communication and personal selection of pension saver's financial risk. *European Journal of Operational Research* 274: 1102–11. [CrossRef]

Gerrard, Russell, Munir Hiabu, Ioannis Kyriakou, and Jens Perch Nielsen. 2019b. Self-selection and risk sharing in a modern world of life-long annuities. *British Actuarial Journal* 23: e30. [CrossRef]

Gerrard, Russell, Munir Hiabu, Jens Perch Nielsen, and Peter Vodicka. 2019c. *Long-Term Real Dynamic Investment Planning*. Working Paper. London: Cass Business School.

Geweke, John F. 1996. Monte carlo simulation and numerical integration. In *Handbook of Computational Economics*. Edited by Hans M. Amman, David A. Kendrick and John Rust. Amsterdam: Elsevier, vol. I, pp. 731–800.

Glad, Ingrid K. 1998. Parametrically guided non-parametric regression. *Scandinavian Journal of Statistics* 25: 649–68. [CrossRef]

Gonzales-Manteiga, Wenceslao, and Rosa M. Crujeiras. 2013. An updated review of goodness-of-fit tests for regression models. *Test* 22: 361–411. [CrossRef]

Härdle, Wolfgang K., and Enno Mammen. 1993. Comparing nonparametric versus parametric regression fits. *Annals of Statistics* 21: 1926–47. [CrossRef]

Härdle, Wolfgang K., and Alexandre B. Tsybakov. 1997. Local polynomial estimators of the volatility function in nonparametric autoregression. *Journal of Econometrics* 81: 223–42. [CrossRef]

Kim, Tong Suk, and Edward Omberg. 1996. Dynamic nonmyopic portfolio behavior. *The Review of Financial Studies* 9: 141–61. [CrossRef]

Kothari, S. P., Jonathan Lewellen, and Jerold B. Warner. 2006. Stock returns, aggregate earnings surprises, and behavioral finance. *Journal of Financial Economics* 79: 537–68. [CrossRef]

Kreiss, Jens-Peter, Michael H. Neumann, and Qiwei Yao. 2008. Bootstrap tests for simple structures in nonparametric time series regression. *Statistics and Its Interfaces* 1: 367–80. [CrossRef]

Kyriakou, Ioannis, Parastoo Mousavi, Jens Perch Nielsen, and Michael Scholz. 2019a. Forecasting benchmarks of long-term stock returns via machine learning. *Annals of Operations Research*. doi:10.1007/s10479-019-03338-4. [CrossRef]

Kyriakou, Ioannis, Parastoo Mousavi, Jens Perch Nielsen, and Michael Scholz. 2019b. *Machine Learning for Forecasting Excess Stock Returns—The Five-year View*. Graz Economics Papers 2019-06. Graz: University of Graz, Departmemt of Economics.

Lettau, Martin, and Stijn Van Nieuwerburgh. 2008. Reconciling the return predictability evidence. *Review of Financial Studies* 21: 1607–52. [CrossRef]

Linton, Oliver, and Benoit Perron. 2003. The shape of the risk premium: Evidence from a semiparametric generalized autoregressive conditional heteroscedasticity model. *Journal of Business & Economic Statistics* 21: 354–67.

Linton, Oliver B., and Yang Yan. 2011. Semi- and nonparametric arch processes. *Journal of Probability and Statistics* 2011: 906212. [CrossRef]

Linton, Oliver B., and Enno Mammen. 2008. Nonparametric transformation to white noise. *Journal of Econometrics* 142: 241–64. [CrossRef]

Merton, Robert C. 1973. An intertemporal capital asset pricing model. *Econometrica* 41: 867–87. [CrossRef]

Merton, Robert C. 2014. The crisis in retirement planning. *Harvard Business Review* 92: 43–50.

Mishra, Santosh, Liangjun Su, and Aman Ullah. 2010. Semiparametric estimator of time series conditional variance. *Journal of Business & Economic Statistics* 28: 256–74.

Munk, Claus, and Jesper Rangvid. 2018. New assumptions of a pension forecast model: Background, level and consequences for individuals forecasted pension. *Finans/Invest* 6: 6–14.

Nielsen, Jens Perch, and Stefan Sperlich. 2003. Prediction of stock returns: A new way to look at it. *ASTIN Bulletin* 33: 399–417. [CrossRef]

Opsomer, Jean, Yuedong Wang, and Yuhong Yang. 2001. Nonparametric regression with correlated errors. *Statistical Science* 16: 134–53.

Pagan, Adrian R., and Yong-Sik Hong. 1991. Nonparametric estimation and the risk premium. In *Nonparametric and Semiparametric Methods in Econometrics and Statistics*. Edited by William A. Barnett, James Powell and George E. Tauchen. Cambridge: Cambridge University Press, pp. 51–76.

Pagan, Adrian R., and Aman Ullah. 1988. The econometric analysis of models with risk terms. *Journal of Applied Econometrics* 3: 87–105. [CrossRef]

Politis, Dimitris N., and Joseph P. Romano. 1994. The stationary bootstrap. *Journal of the American Statistical Association* 89: 1303–13. [CrossRef]

Rapach, David, and Guofu Zhou. 2013. Forecasting stock returns. In *Handbook of Economic Forecasting*. Edited by Graham Elliott and Allan Timmerman. Amsterdam: Elsevier, vol. 2A, pp. 328–83.

Scholz, Michael, Jens Perch Nielsen, and Stefan Sperlich. 2015. Nonparametric prediction of stock returns based on yearly data: The long-term view. *Insurance: Mathematics and Economics* 65: 143–55. [CrossRef]

Scholz, Michael, Stefan Sperlich, and Jens Perch Nielsen. 2016. Nonparametric long term prediction of stock returns with generated bond yields. *Insurance: Mathematics and Economics* 69: 82–96. [CrossRef]

Shiller, Robert J. 1989. *Market Volatility*. Cambridge: MIT Press.

Su, Liangjun, and Aman Ullah. 2006. More efficient estimation in nonparametric regression with nonparametric autocorrelated errors. *Econometric Theory* 22: 98–126. [CrossRef]

Wang, Lie, Lawrence D. Brown, T. Tony Cai, and Michael Levine. 2008. Effect of mean on variance function estimation in nonparametric regression. *The Annals of Statistics* 36: 646–64. [CrossRef]

Wang, WenWu, and Ping Yu. 2017. Asymptotically optimal differenced estimators of error variance in nonparametric regression. *Computational Statistics and Data Analysis* 105: 125–43. [CrossRef]

Welch, Ivo, and Amit Goyal. 2008. A comprehensive look at the empirical performance of equity premium prediction. *Review of Financial Studies* 21: 1455–508. [CrossRef]

Xiao, Zhijie, Oliver B. Linton, Raymond J. Carroll, and Enno Mammen. 2003. More efficient local polynomial estimation in nonparametric regression with autocorrelated errors. *Journal of the American Statistical Association* 98: 980–92. [CrossRef]

Xu, Ke Li, and Peter C. B. Phillips. 2011. Tilted nonparametric estimation of volatility functions with empirical applications. *Journal of Business & Economic Statistics* 29: 518–28.

Yu, Keming, and M.C. Jones. 2004. Likelihood-based local linear estimation of the conditional variance function. *Journal of the American Statistical Association* 99: 139–44. [CrossRef]

Ziegelmann, Flavio A. 2002. Nonparametric estimation of volatility functions: The local exponential estimator. *Econometric Theory* 18: 985–91. [CrossRef]

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).

risks

MDPI

Article

A Likelihood Approach to Bornhuetter–Ferguson Analysis

Valandis Elpidorou [1], Carolin Margraf [2], María Dolores Martínez-Miranda [3,*] and Bent Nielsen [4]

[1] Arch Reinsurance Europe Underwriting dac Ireland, Dublin 4, Ireland; elpidoros@gmail.com
[2] Cass Business School, University of London, London EC1Y 8TZ, UK; Carolin.Margraf.1@cass.city.ac.uk
[3] Department of Statistics and Operations Research, University of Granada, 18071 Granada, Spain
[4] Nuffield College, University of Oxford, Oxford OX1 1NF, UK; bent.nielsen@nuffield.ox.ac.uk
* Correspondence: mmiranda@ugr.es

Received: 23 October 2019; Accepted: 3 December 2019; Published: 10 December 2019

Abstract: A new Bornhuetter–Ferguson method is suggested herein. This is a variant of the traditional chain ladder method. The actuary can adjust the relative ultimates using externally estimated relative ultimates. These correspond to linear constraints on the Poisson likelihood underpinning the chain ladder method. Adjusted cash flow estimates were obtained as constrained maximum likelihood estimates. The statistical derivation of the new method is provided in the generalised linear model framework. A related approach in the literature, combining unconstrained and constrained maximum likelihood estimates, is presented in the same framework and compared theoretically. A data illustration is described using a motor portfolio from a Greek insurer.

Keywords: chain ladder; Bornhuetter–Ferguson; maximum likelihood; exponential families; canonical parameters; prior knowledge

1. Introduction

While high dimensional data and its validation is important for many machine learning experiments, it is also true that many machine learning applications combine mathematical statistical methods with prior knowledge. The difficulty is to include this prior knowledge to upgrade the statistical analysis without violating the fundamental principles of mathematical statistics. These kinds of applications are omnipresent in insurance reserving, which often cite the original paper of Bornhuetter and Ferguson from 1972. This combination of prior knowledge and mathematical statistics is the purpose of this paper, where we are able to make it while sticking to the classical maximum likelihood technique of mathematical statistics.

The chain ladder method is the basic actuarial tool for reserving in general insurance. This method is based on the paid run-off triangle and provides estimates for the ultimate reserve along with development factors that are used for determining cash flow. In practice, the actuary usually adjusts the ultimates using additionally available information. With the Bornhuetter et al. (1972) method the chain ladder ultimates are adjusted using prior knowledge while the adjusted cash flow is proportional to the original chain ladder cash flow. Mack (2000) gave a credibility interpretation of the Bornhuetter–Ferguson method.

The adjustment of the ultimates can be done in two ways. Either by correcting the levels of the ultimates or the relative levels of the ultimates. By this, we distinguish between the situation where the actuary has an estimate for the ultimate for a given policy year and the situation where the actuary is more comfortable with the forecast that the ultimate for a given policy year is 10% higher, say, than in the previous year. Such an estimate could, for instance, come from chain ladder analysis of incurred data. Indeed, we provide an empirical illustration where this is the case. The levels approach is most common in the literature; see for instance (Mack 2000, 2006), Taylor (2000), Verrall (2004), Wüthrich and Merz (2008) and Heberle and Thomas (2016). The relative levels approach is more recent; see (Martínez-Miranda et al. 2013, 2015).

There are potentially two concerns with the traditional Bornhuetter–Ferguson correction. It may move the reserves too much, and the cash flow distribution is not adjusted in light of the external information. Verrall (2004) addressed this in a Bayesian setup and Mack (2006) proposed an alternative approach where new weights are computed by combining actual payments and the externally estimated reserves.

Our proposal is related to that of Mack (2006), but with weights derived from a likelihood function. Adjusting relative ultimates as opposed to level ultimates is natural when working with the likelihood function in the same way as traditional chain ladder development factors are concerned with relative effects. A feature of our approach is, therefore, that external information is linked directly to the parameters of the underlying Poisson model and it is possible to express the Bornhuetter–Ferguson adjustment in terms of adjustments to the development factors. Another feature of this approach is that we can evaluate how much the adjustment moves the reserves and establish inequalities relating our approach and the traditional Bornhuetter–Ferguson adjustments.

A fundamental interpretation of the Bornhuetter–Ferguson method arises when combining chain ladder with credibility formulas. Credibility formulas have been investigated in reserving by, for instance, de Vylder (1982), Mack (2000). and more recently, Bühlmann and Moriconi (2015). We have been particularly influenced by Mack (2000), who gives a credibility formula showing that adjusting the ultimates with prior knowledge yields a partial adjustment of the reserves. He then continues to show that the iterations of the credibility formula leads to the Benktander (1976) approach. These ideas are taken a step further by Gigante et al. (2013), whereas Taylor (2000) and Wüthrich and Merz (2008) give general overviews of the Bornhuetter–Ferguson method. Our first contribution is to show that the credibility formula also applies when adjusting the relative levels of the ultimates.

It is useful to recall that the chain ladder method has the nice interpretation as maximum likelihood in a Poisson model. Kremer (1985) (see also Mack 1991) showed that the chain ladder forecasts are maximum likelihood. These forecasts are the product of observed accident year row sums and functions of the development factors; see (4). Renshaw and Verrall (1998) showed that the development factors themselves are maximum likelihood estimators in a conditional Poisson model conditioning on row sums, while Kuang et al. (2009) showed that they are also maximum likelihod in the unconditional Poisson model. The maximum likelihood result means that it is possible to compute the chain ladder estimates using generalised linear model methods. In practice the Poisson assumption is not realistic as the paid data typically have considerable over-dispersion; see for instance England and Verrall (2002). Nonetheless, the chain ladder method provides good reserve estimates that are, at least, anchored in a quasi-likelihood.

The main idea of our approach is to impose the externally estimated relative ultimates on the Poisson likelihood. Initially, it is useful to work with the standard parametrisation of the generalised linear model as opposed to the development factors. We can then formulate the relative ultimates' constraint as a linear constraint on the parameters and derive maximum likelihood estimators. Subsequently, we translate these estimators into adjusted development factors.

The constrained maximum likelihood approach satisfies a monotonicity result. If, for instance, all the relative ultimates are increased relative to the chain ladder ultimates, then it follows that the reserves are increased. However, these new reserves increase less than the traditional Bornhuetter–Ferguson reserves that would arise by combining the adjusted relative ultimates with the chain ladder development factors.

In this paper we focus on classical mathematical statistics through the maximum likelihood method. Recent work in reserving has emerged in the literature using modern machine learning techniques. Kuo (2019) proposes deep neural networks to joint modelling paid data and total claims outstanding, claiming that no manual input is required during model updates or forecasting. Additionally, using neural networks Gabrielli and Wüthrich (2018) develop a "stochastic simulation machine" that generates individual claims histories of non-life insurance claims. Another individual claims approach but based on the classification and regression trees is suggested by De Felice and Moriconi (2019). Chukhrova and Johannssen (2017) describe a state space model for cumulative payments (that extend the chain ladder method) in combination with the Kalman filter.

The rest of the paper is organised as follows. In Section 2 we first describe the chain ladder forecasts in terms of the development factors and certain weights. Using this formulation we describe two Bornhuetter–Ferguson approaches in the literature: the interpretation offered by Mack (2000) that uses levels of ultimates, and the approach of Martínez-Miranda et al. (2013) using relative ultimates. From these two approaches the future cash flow is not influenced by the external information. We then present our proposed Bornhuetter–Ferguson reserves which are determined by a Poisson likelihood constrained by the external information. The formal derivation of our proposal is provided in Section 3 in the generalised, lineal model framework. Our proposal is derived as the solution of a constrained maximum likelihood approach, where the constraint is given by the imposed external information. Later, in Section 3.5 we show that the approach by Martínez-Miranda et al. (2013) is a mixed approach which combines unconstrained and constrained maximum likelihood estimators. Reserve forecasts and cash flow are described from our proposal, the mixed approach and the traditional chain ladder method. Only our proposed cash flow is affect by the external information, which is explicitly shown in terms of some pseudo development factors introduced in Section 3.6. In Section 4 we illustrate our proposal using a motor portfolio from a Greek insurer. These data include both paid and incurred triangles. In addition, an external estimate of the reserve is available so that this example nicely illustrates the practical issues that lead to the use of the Bornhuetter–Ferguson method. Conclusions and final remarks are provided in Section 5.

2. The Bornhuetter–Ferguson Problem

We present two standard Bornhuetter–Ferguson approaches. For now we will not formulate a statistical model, but just use the standard chain ladder formulas.

2.1. Data Structure

Consider a standard run-off triangle of paid amounts. The dimension is denoted k and we use the incremental form of the triangle. Each entry is denoted Y_{ij} so that i is the accident year index and j is the development year index. The indices vary in the upper triangle with indices $1 \leq i, j \leq k$ and $i + j - 1 \leq k$. This is the area I in Figure 1. The objective is to forecast values of Y_{ij} in the lower triangle with indices $1 \leq i, j \leq k$ and $k + 1 \leq i + j - 1 \leq 2k - 1$. This is the area J in Figure 1.

In the analysis we will be interested in row sums, column sums and rectangular sums

$$R_i = \sum_{j=1}^{k+1-i} Y_{ij}, \qquad C_j = \sum_{i=1}^{k+1-j} Y_{ij}, \qquad S_r = \sum_{\ell=1}^{r} \sum_{j=1}^{k-r} Y_{\ell j}, \tag{1}$$

for $1 \leq i, j \leq k$ and $1 \leq r \leq k - 1$. In practice the payments Y_{ij} may be negative. This is at odds with the Poisson model interpretation of the chain ladder method which requires the payments to be non-negative. With the Poisson formulation we need the further requirement that the rectangular sums, the row sums and the column sums are positive Kuang et al. (2009) (Theorem 2); that is,

$$S_1, \ldots, S_{k-1}, R_2, \ldots, R_k, C_2, \ldots, C_k > 0. \tag{2}$$

We will assume these constraints throughout the paper, while noting that constraints in (2) may be satisfied even if some payments Y_{ij} are negative.

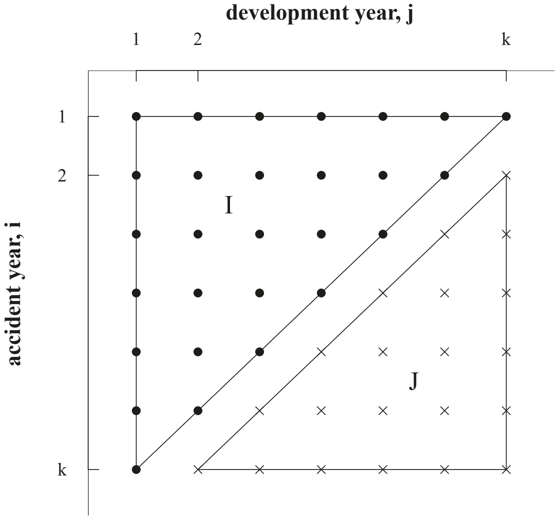


Figure 1. Illustration of data layout.

2.2. Chain Ladder Method

The chain ladder method is computed from row sums or cumulative payments R_i as in (1) and development factors

$$F_j = \frac{\sum_{i=1}^{k+1-j} \sum_{\ell=1}^{j} Y_{i\ell}}{\sum_{i=1}^{k+1-j} \sum_{\ell=1}^{j-1} Y_{i\ell}} \quad \text{for } j = 2, \dots, k. \tag{3}$$

The development factors are larger than one under the constraint (2) to R_i, C_j, S_r; see Theorem 2 in Kuang et al. (2009). The chain ladder forecasts of the amounts in the lower triangle are then

$$\widetilde{Y}_{ij} = R_i(F_j - 1) \prod_{\ell=k+2-i}^{j-1} F_\ell. \tag{4}$$

From this we compute the reserve for accident year i, for $i = 2, \dots, k$, as

$$V_i = \sum_{j=k+2-i}^{k} \widetilde{Y}_{ij} = R_i(F_i^{prod} - 1) \qquad \text{where} \qquad F_i^{prod} = \prod_{\ell=k+2-i}^{k} F_\ell, \tag{5}$$

and the predicted ultimate payment as; see also Section 2.1.3 of England and Verrall (2002),

$$U_i = R_i + V_i = R_i F_i^{prod} \quad \text{for } i = 2, \dots, k. \tag{6}$$

If we use the convention that empty products are unity, this matches with $U_1 = R_1$ and $V_1 = 0$, so that the in-sample prediction of the sum of the payments for accident year one equals the observation.

It will be convenient to express the above formulas in terms of certain weights. Thus, define weights, for $i = 2, \ldots, k$, $j = k + 2 - i, \ldots, k$,

$$W_{ij} \quad = \quad (F_j - 1)\frac{\prod_{\ell=k+2-i}^{j-1} F_\ell}{F_i^{prod}} = \frac{F_j - 1}{\prod_{\ell=j}^{k} F_\ell}, \tag{7}$$

$$W_i \quad = \quad \frac{F_i^{prod} - 1}{F_i^{prod}} = \sum_{j=k+2-i}^{k} W_{ij}. \tag{8}$$

These are numbers between zero and unity when the development factors are larger than unity. The weights W_i approach unity when the product of the development factors approaches infinity. We can then write the forecasts for each cell and each row in the lower triangle as

$$\widetilde{Y}_{ij} = U_i W_{ij}, \qquad V_i = U_i W_i, \qquad i = 2, \ldots, k. \tag{9}$$

These formulas show how the reserve V_i can be found as a fraction of the predicted ultimate U_i, while Y_{ij} indicates how the cash flow is distributed.

The chain ladder is maximum likelihood in a Poisson model that will be presented in Section 3. A feature of the likelihood function (25) is that it is symmetrical in the indices for accident year i and development year j. This observation leads to a new expression for the forecast of the reserve, which will be proved in the Appendix A. Traditionally, we forecast by computing row sums R_i of the data and multiplying by the column wise forward factors F_j as in (4). Alternatively, we can compute columns sums C_j as in (1) and row-wise forward factors

$$G_i = \frac{\sum_{j=1}^{k+1-i} \sum_{\ell=1}^{i} Y_{\ell j}}{\sum_{j=1}^{k+1-i} \sum_{\ell=1}^{i-1} Y_{\ell j}} \quad \text{for } i = 2, \ldots, k \tag{10}$$

and combine these to get the forecasts for the lower triangle, proved in the Appendix A,

$$\widetilde{Y}_{ij} = R_i (F_j - 1) \prod_{\ell=k+2-i}^{j-1} F_\ell = C_j (G_i - 1) \prod_{\ell=k+2-j}^{i-1} G_\ell. \tag{11}$$

2.3. Bornhuetter–Ferguson Using Levels of Ultimates

This section presents the Bornhuetter–Ferguson interpretation offered by Mack (2000); see also England and Verrall (2002) and Alai et al. (2009).

England and Verrall present the Bornhuetter–Ferguson idea as follows. Suppose we replace the chain ladder ultimate U_i by an externally estimated reserve U_i^{level} in the Formula (9). Then we get the level-based Bornhuetter–Ferguson reserve

$$V_i^{BF,level} = U_i^{level} W_i \quad \text{for } i = 2, \ldots, k.. \tag{12}$$

Thus, the Bornhuetter–Ferguson reserve is the proportion W_i of the externally estimated level of the ultimate. In a similar fashion the Bornhuetter–Ferguson cash flow is given by

$$\widetilde{Y}_{ij}^{BF,level} = U_i^{level} W_{ij} \quad \text{for } i = 2, \ldots, k, j = k + 2 - i, \ldots, k.. \tag{13}$$

The predicted ultimate payout turns out to be a convex combination of the chain ladder reserve U_i and the externally generated number U_i^{level}. To see this, use the Formulas (6) and (8) to write the cumulated payments as $R_i = U_i / F_i^{prod} = U_i(1 - W_i)$. It then follows that

$$U_i^{BF,level} = R_i + V_i^{BF,level} = U_i(1 - W_i) + U_i^{level} W_i. \tag{14}$$

Mack (2000) refers to this a credibility formula and traces it back to Benktander (1976). He points out that it can be iterated by replacing U_i^{level} by $U_i^{BF,level}$. Another consequence is the following ordering, assuming $0 < W_i < 1$,

$$U_i < U_i^{level} \qquad \Rightarrow \qquad U_i < U_i^{BF,level} < U_i^{level}. \tag{15}$$

2.4. Bornhuetter–Ferguson Using Relative Ultimates

This section presents the Bornhuetter–Ferguson approach of Martínez-Miranda et al. (2013). The idea is now to replace the relative ultimates rather than levels of ultimates. We then rewrite (9) as

$$V_i = R_1 \frac{U_i}{U_1} W_i \quad \text{for } i = 2, \ldots, k, \tag{16}$$

recalling that $U_1 = R_1$. We now replace U_i/U_1 by some external measure U_i^{rel}/U_1^{rel}, which only provides information about the relative ultimates, such as the figure for year i being 10% higher than that for year $i-1$. This results in the relative level-based Bornhuetter–Ferguson reserve

$$V_i^{BF,rel} = R_1 \frac{U_i^{rel}}{U_1^{rel}} W_i \quad \text{for } i = 2, \ldots, k. \tag{17}$$

The corresponding cash flow is then

$$\widetilde{Y}_{ij}^{BF,rel} = R_1 \frac{U_i^{rel}}{U_1^{rel}} W_{ij} \quad \text{for } i = 2, \ldots, k, j = k+2-i, \ldots, k. \tag{18}$$

The relative Bornhuetter–Ferguson reserve also satisfies an actuarial credibility formula. To see this define $U_1 = R_1$, write $R_i = R_1(R_i/U_1)$ and combine it with $R_i = U_i(1 - W_i)$ as before, to get

$$U_i^{BF,rel} = R_i + V_i^{BF,rel} = R_1 \left\{ \frac{U_i}{U_1}(1 - W_i) + \frac{U_i^{rel}}{U_1^{rel}} W_i \right\}. \tag{19}$$

Once again, we have the ordering, for $i = 2, \ldots k$ and assuming $0 < W_i < 1$,

$$\frac{U_i}{U_1} < \frac{U_i^{rel}}{U_1^{rel}} \qquad \Rightarrow \qquad U_i < U_i^{BF,rel}. \tag{20}$$

Martínez-Miranda et al. (2013) suggested that the relative external numbers could be computed from an incurred triangle. They extended this further to allow for reporting delays using a double chain ladder method. However, in the present paper we focus on the consequences of a Bornhuetter–Ferguson correction rather than how the external numbers are generated.

2.5. Proposed Bornhuetter–Ferguson Reserves

With the above approaches the future cash flow is determined by the chain ladder method through the weights W_{ij} and not influenced by the external information. As argued by Verrall (2004) and Mack (2006) it may be desirable that the cash flow is also influenced by the external information. Our proposal allows the cash flow to be determined by a Poisson likelihood, constrained by the external information. Before we give the derivation it is useful to give a brief overview of the results.

The proposed Bornhuetter–Ferguson approach evolves around the chain ladder reserving Formula (11) involving column sums C_j and row-wise forward factors G_i. Suppose we have externally given relative ultimates U_i^{rel}/U_1^{rel} for $i = 2, \ldots, k$, with the convention that $U_i^{rel}/U_1^{rel} = 1$ for $i = 1$. We then construct Bornhuetter–Ferguson row-wise forward factors

$$\Gamma_i^{rel} = \frac{\sum_{\ell=1}^{i} (U_\ell^{rel}/U_1^{rel})}{\sum_{\ell=1}^{i-1} (U_\ell^{rel}/U_1^{rel})} \qquad \text{for } i = 2, \ldots, k. \tag{21}$$

In Section 3.3 we show that Γ_i^{rel} naturally comes from a constrained likelihood. For now, the intuition is that the traditional development factors F_j, and G_i, are relative effects computed as ratios of sums over data rectangles of different sizes. This compensates for the fact that the data are only available in triangular form, so that column and row lengths are unbalanced. Once we impose the relative ultimates, which are relative row effects, then the unbalanced row lengths are essentially eliminated and we can capture relative row effects in a simpler fashion, as shown in (21).

The Bornhuetter–Ferguson forecasts of individual payments and of reserves are, then,

$$\widetilde{Y}_{ij} = C_j(\Gamma_i^{rel} - 1) \prod_{\ell=k+2-j}^{i-1} \Gamma_\ell^{rel}, \qquad V_i^{rel} = \sum_{j=k+2-i}^{k} \widetilde{Y}_{ij}. \tag{22}$$

3. Generalised Linear Model Framework

We present a Generalised Linear Model framework for Bornhuetter–Ferguson analysis. The usual chain ladder estimators are maximum likelihood in a Poisson model; see Kremer (1985). In practice, reserving data have considerable over-dispersion; see England and Verrall (2002), so that Poisson likelihood becomes a quasi likelihood. In the present paper this distinction is not so important as we will only be concerned with point forecasts. Now, if we maximise the likelihood while imposing constraints from external relative levels of ultimates, we get a closed form cash flow forecast that adapts to both data and the imposed constraints.

Next we describe the unconstrained and constrained Poisson likelihood. The first one provides the chain ladder forecasts without external information in Section 3.2, and the second one provides our proposed forecasts in Section 3.3. The approach of Martínez-Miranda et al. (2013) is shown in Section 3.5 as a mixed approach that combines constrained and unconstrained maximum likelihood estimates. Our proposed forecasts have an equivalent expression involving new column-wise development factors provided in Section 3.6. These development factors will be different for different accident years. For this reason we refer to them as pseudo development factors. This kind of formulation is also possible for the mixed approach, but with the standard chain ladder development factors, as we show in Section 3.7. A monotonicity result provided in Section 3.8 gives some insight about the effect that the imposed external information has on our proposed forecasts and those from the mixed approach.

3.1. Statistical Model

We assume that the incremental observations Y_{ij} are independent Poissons with log expectation $\mathsf{E}Y_{ij} = \exp(\mu_{ij})$, where the predictor is given by

$$\mu_{ij} = \alpha_i + \beta_j + \delta. \tag{23}$$

Here α_i is the level of the accident year effect, β_j is the level of the development year effect and δ is an overall level. The parametrisation presented in (23) does not identify the distribution, so we switch to the invariant parametrisation of Kuang et al. (2009); that is,

$$\mu_{ij} = \mu_{11} + \sum_{\ell=2}^{i} \Delta\alpha_\ell + \sum_{\ell=2}^{j} \Delta\beta_\ell, \tag{24}$$

with the convention that empty sums are zero. Here $\Delta\alpha_i = \alpha_i - \alpha_{i-1}$ is the relative accident year effect and $\Delta\beta_j = \beta_j - \beta_{j-1}$ is the relative development year effect, while the overall level is determined by μ_{11}. The Poisson log likelihood function is

$$\ell(\mu_{11}, \Delta\alpha_i, \Delta\beta_j) = \sum_{1 \le i,j,i+j-1 \le k} \{\mu_{ij} Y_{ij} - \exp(\mu_{ij}) - \log(Y_{ij}!)\}. \tag{25}$$

This is a regular exponential family with canonical parameters $\mu_{11}, \Delta\alpha_i, \Delta\beta_j$.

3.2. The Chain Ladder

The chain ladder arises by maximising the unconstrained likelihood. Theorem 3 in Kuang et al. (2009) shows that the maximum likelihood estimators are

$$\begin{aligned} \Delta\widehat{\alpha}_i &= \Delta \log R_i + \log F_{k+2-i} \quad \text{for } i = 2, \dots k, & (26)\\ \Delta\widehat{\beta}_j &= \Delta \log C_j + \log G_{k+2-j} \quad \text{for } j = 2, \dots k, & (27)\\ \widehat{\mu}_{11} &= \log R_1 - \sum_{j=2}^{k} \log F_j. & (28) \end{aligned}$$

When inserting these estimators into Equation (23) we get estimators $\widehat{\mu}_{ij}$. In turn, the relative ultimates are estimated by

$$\frac{U_i}{U_1} = \frac{\sum_{j=1}^{k} \exp(\widehat{\mu}_{ij})}{\sum_{j=1}^{k} \exp(\widehat{\mu}_{1j})} = \exp(\sum_{\ell=2}^{i} \Delta\widehat{\alpha}_\ell) \quad \text{for } i = 2, \dots k, \tag{29}$$

which are the relative ultimates entering in Equation (16). It is convenient to define $U_1 = R_1$, as this says that the ultimate for first accident year equals the claims observed. With this definition, we find that $R_1 = U_1$ is the maximum likelihood estimator for the expected ultimates $\mathsf{E}R_1$ for the first accident year. In turn, the maximum likelihood estimators for the ultimate levels satisfy

$$U_i = U_1 \frac{U_i}{U_1} = U_1 \exp(\sum_{\ell=2}^{i} \Delta\widehat{\alpha}_\ell) \quad \text{for } i = 2, \dots k. \tag{30}$$

Now, insert the expression for $\Delta\widehat{\alpha}_i$ in (26) to get

$$U_i = U_1 \prod_{\ell=2}^{i} (\frac{R_\ell}{R_{\ell-1}} F_{k+2-\ell}) = U_1 \frac{R_i}{R_1} \prod_{\ell=2}^{i} F_{k+2-\ell} = R_i \prod_{\ell=k+2-i}^{k} F_\ell, \tag{31}$$

which are the ultimates in (6). Thus, in both cases the ultimate formulas are closely linked to the estimated relative accident year effects $\Delta\widehat{\alpha}_i$.

An additional result from Theorem 3 in Kuang et al. (2009) is that the forward factors F_j and G_i can be viewed as maximum likelihood estimators for certain combinations of the canonical parameters $\Delta\beta_j$ and $\Delta\alpha_i$, respectively. These combinations are, for $i, j = 2, \dots, k$,

$$\Phi_j = \frac{\sum_{\ell=1}^{j} \exp(\sum_{h=2}^{\ell} \Delta\beta_h)}{\sum_{\ell=1}^{j-1} \exp(\sum_{h=2}^{\ell} \Delta\beta_h)}, \qquad \Gamma_i = \frac{\sum_{\ell=1}^{i} \exp(\sum_{h=2}^{\ell} \Delta\alpha_h)}{\sum_{\ell=1}^{i-1} \exp(\sum_{h=2}^{\ell} \Delta\alpha_h)}, \tag{32}$$

with the convention that empty sums are zero. The development factors are the corresponding maximum likelihood estimators; that is, $F_j = \widehat{\Phi}_j$ and $G_i = \widehat{\Gamma}_i$.

3.3. Imposing External Information on the Relative Ultimates

Suppose some external values are available for the relative ultimates, U_i^{rel}/U_1^{rel}. Equivalently, we have external values for the relative accident year effects $\Delta\alpha_i^\dagger$; that is,

$$\Delta\alpha_i^\dagger = \log(U_i^{rel}/U_{i-1}^{rel}). \tag{33}$$

We could impose these as a constraint on the likelihood (25). The constraint is linear and the likelihood remains that of a regular exponential family.

The constrained maximum likelihood estimators have a simple analytical form. In line with the parameters Γ_i defined in (32), define

$$\Gamma_i^\dagger = \frac{\sum_{\ell=1}^{i}\exp(\sum_{h=2}^{\ell}\Delta\alpha_h^\dagger)}{\sum_{\ell=1}^{i-1}\exp(\sum_{h=2}^{\ell}\Delta\alpha_h^\dagger)}. \tag{34}$$

We then have the following result, which is proven in the Appendix A.

Theorem 1. *Consider the Poisson likelihood (25) with known $\Delta\alpha_i = \Delta\alpha_i^\dagger$ for $i = 2,\ldots,k$ and define $\Gamma_i^\dagger$ as (32), computed using $\Delta\alpha_i^\dagger$. The constrained maximum likelihood estimator is unique if and only if $C_j > 0$ for all $j = 1,\ldots k$ and given by*

$$\Delta\widehat{\beta}_j^\dagger = \Delta\log C_j + \log\Gamma_{k+2-j}^\dagger \qquad \textit{for } j = 2,\ldots,k, \tag{35}$$

$$\widehat{\mu}_{11}^\dagger = \log C_1 - \log\{1+\sum_{i=2}^{k}\exp(\sum_{\ell=2}^{i}\Delta\alpha_\ell^\dagger)\} = \log C_1 - \sum_{\ell=2}^{k}\log\Gamma_\ell^\dagger. \tag{36}$$

As a consequence, the out-of-sample forecast from the constrained chain ladder has a simple explicit form, as shown in the following result, which is proven in the Appendix A. The result resembles the forecast in the unrestricted chain ladder computed from column sums and row-wise development factors as described in (11).

Theorem 2. *Consider the setup in Theorem 1. Point forecasts for the lower triangle are given by*

$$\widetilde{Y}_{ij}^\dagger = C_j(\Gamma_i^\dagger - 1)\prod_{\ell=k+2-j}^{i-1}\Gamma_\ell^\dagger. \tag{37}$$

We can now compute a Bornhuetter–Ferguson reserve based on Theorem 2. For each accident year we get

$$V_i^\dagger = \sum_{j=k+2-i}^{k}\widetilde{Y}_{ij}^\dagger. \tag{38}$$

In the case where we impose external relative ultimates, the above expressions reduce to those presented previously in (22). In the above expression the notation reflects that the external information is concerned with the relative accident year parameters $\Delta\alpha_i^\dagger$. Now, suppose the relative ultimates U_i^{rel}/U_1^{rel} are taken as given. We then apply the Formula (30) to get cumulated relative accident parameters $\exp(\sum_{\ell=2}^{i}\Delta\alpha_\ell^\dagger) = U_i^{rel}/U_1^{rel}$. Inserting this in the expression (34) for $\Gamma_i^\dagger$ in (34) gives

$$\Gamma_i^\dagger = \frac{\sum_{\ell=1}^{i}\exp(\sum_{h=2}^{\ell}\Delta\alpha_h^\dagger)}{\sum_{\ell=1}^{i-1}\exp(\sum_{h=2}^{\ell}\Delta\alpha_h^\dagger)} = \frac{\sum_{\ell=1}^{i}U_\ell^{rel}/U_1^{rel}}{\sum_{\ell=1}^{i-1}U_\ell^{rel}/U_1^{rel}} = \Gamma_i^{rel}, \tag{39}$$

which is the expression for Γ_i^{rel} in (21). Since $\Gamma_i^\dagger = \Gamma_i^{rel}$ we see that the point forecast $\widetilde{Y}_{ij}^\dagger$ in (37) equals the point forecast $\widetilde{Y}_{ij}$ in (22). In turn the reserve $V_i^\dagger$ in (38) equals the reserve V_i^{rel} in (22).

3.4. Implementation in GLM Software

The constrained model can also be estimated using ready-made algorithms for generalised linear models. The analysis presented above shows that the constrained model is a regular exponential family so the algorithms should perform well.

For the implementation we organise the triangle Y as a vector $\mathbf{Y}$, say, of dimension $k(k+1)/2$. A design matrix $\mathbf{X}$ can be constructed from the formula (24). It has dimension $\{k(k+1)/2\} \times (2k-1)$ and the row corresponding to entry i, j is given by

$$X'_{ij} = \{1, 1_{(2\leq i)}, \ldots, 1_{(k\leq i)}, 1_{(2\leq j)}, \ldots, 1_{(k\leq j)}\}, \tag{40}$$

where the indicator function $1_{(m\leq i)}$ takes the value unity if $m \leq i$ and zero otherwise. The unrestricted model is then estimated through a generalised linear model regression of $\mathbf{Y}$ on $\mathbf{X}$ using the Poisson distribution with a log-link function.

In the constrained model the parameters $\theta_{known} = (\Delta\alpha_2^\dagger, \ldots, \Delta\alpha_k^\dagger)'$ are known. Deleting the corresponding columns from $\mathbf{X}$ gives a design matrix $\mathbf{X}_{reduced}$ with k columns. The deleted columns are collected as $\mathbf{X}_{known}$, say. The model is then estimated as a generalised linear model regression of $\mathbf{Y}$ on $\mathbf{X}_{reduced}$ using the Poisson distribution with a log-link function and offset given by $\mathbf{X}_{known}\theta_{known}$.

3.5. A Mixed Approach

By now we have two maximum likelihood approaches: the classical chain ladder and the restricted maximum likelihood approach derived above. These give different point forecasts for the lower triangle. A third type of point forecast arises from the Bornhuetter–Ferguson double chain ladder (BDCL) method in Martínez-Miranda et al. (2013). In the following it is shown how the three are connected.

Let us first summarise the results we obtained so far in terms of the log likelihood. In the classical chain ladder approach, we maximise the unrestricted likelihood in (25), which leads to the unrestricted estimator

$$\widehat{\xi} = \max_{\xi} \ell(\xi) = (\widehat{\mu}_{11}, \Delta\widehat{\alpha}_i, \Delta\widehat{\beta}_j)'. \tag{41}$$

The restricted likelihood from Section 3.3 with restriction $\Delta\alpha_i = \Delta\alpha_i^\dagger$ has a restricted likelihood maximum likelihood estimator given by

$$\widehat{\xi}^\dagger = \max_{\xi:\Delta\alpha=\Delta\alpha^\dagger} \ell(\xi) = (\widehat{\mu}_{11}^\dagger, \Delta\alpha_i^\dagger, \Delta\widehat{\beta}_j^\dagger)'. \tag{42}$$

Notice, that if $\Delta\alpha_i^\dagger = \Delta\widehat{\alpha}_i$, then $\widehat{\mu}_{11}^\dagger = \widehat{\mu}_{11}$ and $\Delta\widehat{\beta}_j^\dagger = \Delta\widehat{\beta}_j$.

A third estimator is achieved by mixing the above estimators. This combines the unrestricted estimators for μ_{11} and β_j with the given $\Delta\alpha_i^\dagger$, such that

$$\widehat{\xi}^\ddagger = (\widehat{\mu}_{11}, \Delta\alpha_i^\dagger, \Delta\widehat{\beta}_j)'. \tag{43}$$

In the following, parameters resulting from this mixed approach will be marked with the index "‡," just as parameters resulting from the constrained method will be marked with "†." The forecast for future payments computed from $\widehat{\xi}^\ddagger$ is

$$\widetilde{Y}_{ij}^\ddagger = \exp(\widehat{\mu}_{11} + \sum_{h=2}^{i} \Delta\alpha_h^\dagger + \sum_{h=2}^{j} \Delta\widehat{\beta}_h). \tag{44}$$

In the Appendix A we prove the identities

$$\widetilde{Y}_{ij}^{\ddagger} = \widetilde{Y}_{ij} \frac{\exp(\sum_{h=2}^{i} \Delta\alpha_h^{\dagger})}{\exp(\sum_{h=2}^{i} \Delta\widehat{\alpha}_h)} = \widetilde{Y}_{ij}^{\dagger} \frac{\sum_{\ell=2}^{k+1-j} \exp(\sum_{h=2}^{\ell} \Delta\alpha_h^{\dagger})}{\sum_{\ell=2}^{k+1-j} \exp(\sum_{h=2}^{\ell} \Delta\widehat{\alpha}_h)}. \tag{45}$$

In the case when the known accident parameters are derived by applying chain ladder on the incurred data, such that $\Delta\alpha_i^{\dagger} = \Delta\widehat{\alpha}_i^{inc}$, this method gives exactly the same results as the Bornhuetter–Ferguson double chain ladder (BDCL) method in Martínez-Miranda et al. (2013).

The log likelihood function evaluated in the three points satisfies

$$\ell(\widehat{\xi}) \geq \ell(\widehat{\xi}^{\dagger}) \geq \ell(\widehat{\xi}^{\ddagger}).$$

The first inequality holds since $\widehat{\xi}$ is maximum likelihood, while $\widehat{\xi}^{\dagger}$ is restricted maximum likelihood. The second inequality holds since $\widehat{\xi}^{\ddagger}$ satisfies the restriction, but it is not maximum likelihood.

3.6. Pseudo Development Factors

It is common practice to think about the classical chain ladder method in terms of row sums R_i and column wise development factors F_j given in (1) and (3). For the restricted maximum likelihood approach there are no natural development factors in a maximum likelihood sense. Since development factors are important in daily actuarial work it is of interest to develop pseudo-development factors that keep the chain ladder pattern.

In the classical chain ladder, the forecasts for the lower triangle are computed using the Formula (11) by forwarding the row sums R_i using the factors F_j. However, in this classical setting the predicted value for the row sum equals the row sum. In the likelihood analysis, this stems from a likelihood equation of the type $R_i = \mathsf{E}(R_i)$; see Equation (20) in Kuang et al. (2009). Thus, we can also interpret the chain ladder forecast as forwarding the predicted row sums.

Once we have imposed external information on the relative ultimates, then the forecast changes and we break the link to the original row sums and development factors. We can, however, construct pseudo forecasts of the row sums and pseudo forward factors that satisfy a relationship like (11) but with the new forecasts.

Under the constraint that $\Delta\alpha = \Delta\alpha^{\dagger}$ we compute estimates $\widehat{\mu}_{11}^{\dagger}$ and $\Delta\widehat{\beta}_j^{\dagger}$ using (35) and (36) in Theorem 1. From these we compute pseudo forward factors from (32); that is,

$$F_j^{\dagger} = \frac{\sum_{\ell=1}^{j} \exp(\sum_{h=2}^{\ell} \Delta\widehat{\beta}_h^{\dagger})}{\sum_{\ell=1}^{j-1} \exp(\sum_{h=2}^{\ell} \Delta\widehat{\beta}_h^{\dagger})}, \tag{46}$$

and a pseudo first row sum from (28) as

$$\log R_1^{\dagger} = \widehat{\mu}_{11}^{\dagger} + \sum_{j=2}^{k} \log F_j^{\dagger}, \tag{47}$$

and then the remaining pseudo row sums from (26) as

$$\Delta \log R_i^{\dagger} = \Delta\alpha_i^{\dagger} - \log F_{k+2-i}^{\dagger}. \tag{48}$$

We show in the Appendix A that the forecast from (37) can be computed as

$$\widetilde{Y}_{ij}^{\dagger} = R_i^{\dagger}(F_j^{\dagger} - 1) \prod_{\ell=k+2-i}^{j-1} F_{\ell}^{\dagger}. \tag{49}$$

The above formulas for predicted reserve and the cash flow can also be written in the credibility format we saw in (17) and (18). To see this introduce the weights

$$W_{ij}^{\dagger} = (F_j^{\dagger} - 1)\frac{\prod_{\ell=k+2-i}^{j-1} F_\ell^{\dagger}}{F_i^{prod\dagger}}, \qquad W_i^{\dagger} = \frac{F_i^{prod\dagger} - 1}{F_i^{prod\dagger}},$$

where, as before, $F_i^{prod\dagger} = \prod_{\ell=k+2-i}^{k} F_\ell^{\dagger}$. Introducing the ultimates and relative ultimates

$$U_i^{\dagger} = R_i^{\dagger} F_i^{prod\dagger}, \qquad \frac{U_i^{\dagger}}{U_{i-1}^{\dagger}} = \frac{R_i^{\dagger}}{R_{i-1}^{\dagger}} F_{k+2-i}^{\dagger} = \exp(\Delta\alpha_i^{\dagger})$$

we can then write the predicted reserve and cash flow as

$$\widetilde{Y}_{ij}^{\dagger} = U_i^{\dagger} W_{ij}^{\dagger}, \qquad V_i^{\dagger} = U_i^{\dagger} W_i^{\dagger}.$$

3.7. Chain Ladder Forecasts with the Mixed Approach

In the mixed approach we follow a similar procedure to satisfy a relationship like (11) in order to obtain the new forecasts. The difference to the constrained method is that we can keep the forward factors from the unconstrained chain ladder model, F_j. However, we need to construct pseudo row sums $R_i^{\ddagger}$ as follows.

We fix the pseudo first row sum as

$$\log R_1^{\ddagger} = \log R_1, \tag{50}$$

and then compute the remaining pseudo row sums from (26) as

$$\Delta \log R_i^{\ddagger} = \Delta\alpha_i^{\dagger} - \log F_{k+2-i}. \tag{51}$$

We show in the Appendix A that the forecast from (44) can be computed as

$$\widetilde{Y}_{ij}^{\ddagger} = R_i^{\ddagger}(F_j - 1) \prod_{\ell=k+2-i}^{j-1} F_\ell. \tag{52}$$

The forecast can be written in terms of weights, as before. Since the cash flow is derived from the chain ladder development factors, the weights are as defined in (7) and (8). In particular we have the ultimates and relative ultimates

$$U_i^{\ddagger} = R_i^{\ddagger} F_i^{prod}, \qquad \frac{U_i^{\ddagger}}{U_{i-1}^{\ddagger}} = \frac{R_i^{\ddagger}}{R_{i-1}^{\ddagger}} F_{k+2-i} = \exp(\Delta\alpha_i^{\dagger}).$$

We can then write the forecast of future payments and the cash flow as

$$\widetilde{Y}_{ij}^{\ddagger} = U_i^{\ddagger} W_{ij}, \qquad V_i^{\ddagger} = U_i^{\ddagger} W_i. \tag{53}$$

3.8. Monotonicity

The idea of the Bornhuetter–Ferguson approach is to first compute the chain ladder, and then adjust it by imposing values for the ultimates. This is a quite complicated approach and it is not immediately clear what the effect is. However, when all adjustments are in the same direction it is actually possible to show a monotonicity result for the effect of the Bornhuetter–Ferguson adjustment.

Let us consider the case when the known accident parameters, $\Delta\alpha_i^{\dagger}$, are bigger than the accident parameters we obtain from the chain ladder method on paid data, $\Delta\widehat{\alpha}_i$. The following theorem, proved in the Appendix A, shows monotonicity results regarding the remaining parameters given in Theorem 1, and the resulting forecasts we obtain from the constrained approach in Section 3.3, $\widetilde{Y}_{ij}^{\dagger}$, and the mixed approach in Section 3.5, $\widetilde{Y}_{ij}^{\ddagger}$.

Theorem 3. *Suppose $\Delta\alpha_i^{\dagger} > \Delta\widehat{\alpha}_i$ for all $2 \leq i \leq k$. Then,*

(*a*) $\Gamma_i^{\dagger} > G_i$ *for all* $2 \leq i \leq k$;

(*b*) $\Delta\widehat{\beta}_j^{\dagger} > \Delta\widehat{\beta}_j$ *for all* $2 \leq j \leq k$;

(*c*) $\widehat{\mu}_{11}^{\dagger} < \widehat{\mu}_{11}$;

(*d*) $\widetilde{Y}_{ij}^{\ddagger} > \widetilde{Y}_{ij}^{\dagger} > \widetilde{Y}_{ij}$ *for all* i, j *so that* $k < i + j - 1 < 2k$;

(*e*) $F_j^{\dagger} > F_j$ *for all* $2 \leq j \leq k$;

(*f*) $R_i^{\ddagger} > R_i^{\dagger}$ *for all* $2 \leq i \leq k$;

(*g*) $R_i^{\ddagger} > R_i$ *for all* $2 \leq i \leq k$.

To interpret this, suppose all imposed relative ultimates $U_i^{rel} / U_{i-1}^{rel} = \exp(\Delta\alpha_i^{\dagger})$ are larger than the chain ladder forecasts of the relative ultimates $U_i / U_{i-1} = \exp(\Delta\widehat{\alpha}_i)$. Suppose also that the imposed relative ultimates are taken from incurred estimates as in the Bornhuetter–Ferguson double chain ladder (BDCL) method in Martínez-Miranda et al. (2013). We then get that the point forecasts for the lower triangle are ordered so that the Bornhuetter–Ferguson double chain ladder forecast $\widetilde{Y}_{ij}^{\ddagger}$ is larger than the Bornhuetter–Ferguson-restricted maximum likelihood forecast $\widetilde{Y}_{ij}^{\dagger}$, which is larger than the chain ladder forecast $\widetilde{Y}_{ij}$. This will be the situation in the empirical illustration in Section 4.

4. Empirical Illustration

We illustrate the new methods by an example where the external knowledge comes from incurred payments. In practice, the external knowledge may also come from incurred counts, from other business lines or from other sources.

We used data from a Greek non-life insurer for motor third party liability, aggregated over bodily injury and property damage. The data are presented as cumulative run-off triangles for accident years from 2005 to 2013. Table 1 shows payments, while Table 2 shows incurred amounts.

Table 1. Payments in Euros.

2005	34,492,471	47,124,007	55,244,404	59,817,460	62,550,940	66,042,036	69,311,560	70,992,659	72,265,079
2006	39,467,733	54,003,286	61,349,336	69,986,825	76,412,887	81,768,759	86,684,598	90,726,054	
2007	38,928,855	57,087,550	65,905,902	77,128,507	84,158,380	92,436,441	97,838,371		
2008	34,202,332	50,932,726	60,560,484	68,566,905	76,409,739	82,082,804			
2009	35,657,409	52,397,264	59,849,582	66,698,806	72,724,524				
2010	25,404,394	37,040,589	42,371,049	50,709,319					
2011	21,268,516	31,311,410	35,973,015						
2012	17,404,447	27,786,399							
2013	17,676,374								

Table 2. Incurred amounts in Euros.

2005	54,018,141	56,699,807	60,273,204	61,112,600	63,729,660	67,142,341	69,733,859	71,980,196	72,738,376
2006	68,706,483	70,534,436	70,254,136	75,919,965	77,900,147	83,401,774	88,690,144	92,171,660	
2007	64,613,205	72,600,950	76,163,387	82,388,057	87,424,383	96,246,891	102,854,340		
2008	58,071,632	66,701,421	69,420,629	75,280,537	81,978,240	89,923,269			
2009	60,368,719	67,868,349	72,528,239	80,726,223	85,339,588				
2010	47,282,519	56,488,940	60,896,832	65,900,623					
2011	49,905,225	54,801,141	60,026,903						
2012	48,425,940	52,652,928							
2013	47,449,977								

Table 3 shows parameter estimates for the paid data computed using the chain ladder and the Bornhuetter–Ferguson constrained model. For the moment we focus on the canonical parameters $\Delta\alpha_i$ for the relative accident year effect, $\Delta\beta_j$ for the relative development year effect and μ_{11} for the overall level. First, the chain ladder estimates are reported as $\Delta\widehat{\alpha}_i$, $\Delta\widehat{\beta}_j$ and $\Delta\widehat{\mu}_{11}$. Second, for the constrained model we first applied chain ladder to the incurred data. The estimates for the relative accident year effect are reported as $\Delta\alpha_i^\dagger$. The estimates $\Delta\widehat{\beta}_j^\dagger$ and $\Delta\widehat{\mu}_{11}^\dagger$ were then computed from the paid data using Theorem 1. We note that the ordering $\Delta\alpha_i^\dagger > \Delta\widehat{\alpha}_i$ applies for these data for all $i = 2, ..., k = 9$. Thus, the monotonicity results from Theorem 3 apply. In particular, we see that $\Delta\widehat{\beta}_j^\dagger > \Delta\widehat{\beta}_j$ for all $j = 2, ..., k = 9$ and $\widehat{\mu}_{11}^\dagger < \widehat{\mu}_{11}$ in Table 3.

Table 3. Estimates.

$\Delta\widehat{\alpha}_i$	$\Delta\alpha_i^\dagger$	$\Delta\widehat{\beta}_j$	$\Delta\widehat{\beta}_j^\dagger$
0.24526809	0.247261682	−0.80044252	−0.76965582
0.11149938	0.145178053	−0.68857388	−0.65777806
−0.12057425	−0.077312634	0.02370846	0.06137844
−0.04769497	0.027019249	−0.32208939	−0.29855013
−0.27637689	−0.204202408	−0.05908884	−0.03399479
−0.21412347	−0.018592530	−0.22363447	−0.20684905
−0.11353717	−0.078902778	−0.37786842	−0.36440835
−0.08135422	−0.005083078	−0.68021278	−0.67909386
$\widehat{\mu}_{11} = 17.18463300$		$\widehat{\mu}_{11}^\dagger = 17.00538277$	

A third approach is to use the mixed approach outlined in Section 3.5. Here we use the external estimate $\Delta\alpha_i^\dagger$ for the relative accident year effects along with the chain ladder estimates $\Delta\widehat{\beta}_j$ and $\Delta\widehat{\mu}_{11}$. When the external estimate is based on the incurred data, as in here, this is the same as the Bornhuetter–Ferguson double chain ladder (BDCL) approach of Martínez-Miranda et al. (2013).

Table 4 presents the estimated (pseudo) forward factors and the (pseudo) row sums. For the chain ladder, we have the observed row sums R_i and the traditional forward factors F_j computed by (1) and (3). For the Bornhuetter–Ferguson constrained model we have the pseudo row sums $R_i^\dagger$ and the pseudo forward factors $F_j^\dagger$ computed by (46)–(48). For the mixed approach we have the pseudo row sums $R_i^\ddagger$ computed by (50) and (51) and the traditional forward factors F_j. Once again we see that the monotonicity results from Theorem 3 apply so that $R_i^\ddagger > R_i^\dagger$ and $F_j^\dagger > F_j$.

Table 4. Row sums and forward factors.

i,j	R_i	$R_i^\dagger$	$R_i^\ddagger$	F_j	$F_j^\dagger$
1	72,265,079	63,989,145	72,265,079		
2	90,726,054	80,309,654	90,907,105	1.449130	1.463172
3	97,838,371	80,309,654	101,391,484	1.155676	1.163975
4	82,082,804	77,559,430	88,824,492	1.137937	1.149793
5	72,724,524	73,428,364	84,802,647	1.087838	1.096652
6	50,709,319	54,589,726	63,556,691	1.076112	1.085188
7	35,973,015	46,603,309	54,823,701	1.056555	1.063832
8	27,786,399	37,000,367	43,839,471	1.036684	1.041678
9	17,676,374	25,159,556	30,098,881	1.017923	1.020288

Table 5 shows the reserves resulting from the classical chain ladder method, $\sum_{i=2}^{k} V_i$ from (5); the constrained approach, $\sum_{i=2}^{k} V_i^\dagger$ from (38); and the mixed approach, $\sum_{i=2}^{k} V_i^\ddagger$ from (53). We see that the ordering from Theorem 3 applies. For comparison we note that this portfolio was evaluated at 137 million by an external actuary, with the comment that this figure may be slightly too low. This valuation is based on the information that since 2009, the case reserves incurred were gradually increased, but the gap between incurred and paid reserves was not fully closed as of 2014. In light of this, the Bornhuetter–Ferguson constrained method appears to apply rather well in this situation.

Table 5. Reserves in million Euros.

$\sum_{i=1}^{k} V_i$	External Valuation	$\sum_{i=1}^{k} V_i^\dagger$	$\sum_{i=1}^{k} V_i^\ddagger$
110.1	137	149.1	156.6

5. Conclusions

The paper introduces a Bornhuetter–Ferguson approach that replaces the relative ultimates rather than levels of ultimates. This approach has been suggested in the Bornhuetter–Ferguson double chain ladder (BDCL) method in Martínez-Miranda et al. (2013). The traditional Bornhuetter–Ferguson method uses chain ladder weights, whereas we have estimated weights.

We made use of the fact that the chain ladder method has a nice interpretation as maximum likelihood in a Poisson model, and we formulated the relative ultimates constraint as a linear constraint on the parameters and derived maximum likelihood estimators. Furthermore, we followed this approach to reproduce the results of the BDCL method in a mixed approach, combining the constrained method with the classical chain ladder.

Monotonicity results compare the constrained method, the mixed approach and the original chain ladder results. An example illustrates the mentioned results with data from a Greek general insurer. The example shows that, when comparing all methods mentioned above, including chain ladder, the reserve given by the constrained method is in fact the closest estimate to the number given by an external expert.

Our proposal incorporates prior knowledge in a transparent way, keeping the standard principles of maximum likelihood and its well known mathematical properties. In this sense we recommend our approach over traditional Bornuetter-Ferguson adjustments as a formal statistical method for the same purpose, which keeps the simplicity and the intuition of traditional reserving. This is further shown in the convenient formulation of the forecasts in terms of the pseudo development factors provided above. Apart from this, one would also benefit from the practical advantages of using maximum likelihood that include standard inference and distribution forecasting. This cannot be done with such a level of formality in the classical approach, while for the BDCL method it has been done using intense bootstrap techniques. Another advantage of our approach is that it can, unlike the BDCL method, be applied using only one triangle, usually the payments triangle. On the other hand, this has the disadvantage of not being able to distinguish between IBNR and RBNS reserves as the BDCL method does. Another limitation of our proposal is that it cannot handle negative cells, as it is sometimes the case in the payments triangle. Further refinements are required to deal with this problem.

An outstanding problem is to provide distribution forecasts of Bornhuetter–Ferguson adjusted reserves. In practice the data will have considerable over-dispersion. By modelling that we could complement the point forecasts with distribution forecasts. Recently, Harnau and Nielsen (2018) developed an asymptotic distribution theory for the chain ladder within an over-dispersed Poisson framework. The present situation is a special case of their setup so it could potentially be extended with Bornhuetter–Ferguson adjustments. That was beyond the scope of this paper though.

Finally, because there is a full statistical model specification incorporating prior knowledge, one could implement the same type of cash-flow data validation as in Agbeko et al. (2014), based on back-testing (see also De Felice and Moriconi 2019). However, this approach has several drawbacks, more so for small datasets. Controversy also exits about which error criteria should be considered. We did not consider empirical validation in this paper and focused on theoretical statistical properties when comparing reserving methods under the generalised linear models framework. A recent discussion on empirical validation methods in reserving can be found in Matinek (2019). These can be potentially used in the context of this paper.

Author Contributions: C.M. and B.N. developed the theory and R-code. V.E. provided the data and practical insights. M.D.M.-M. prepared the final version of the manuscript.

Funding: This research was funded by the programme for Economic Modelling, Oxford and European Research Council, grant AdG 694262, and the Spanish Ministry of Economy and Competitiveness, grant MTM2016-76969P, which includes support from the European Regional Development Fund (ERDF).

Acknowledgments: The authors thank three anonymous reviewers for helpful comments. Discussions with J.P. Nielsen are gratefully acknowledged.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Proofs of Theorems

Proof of Equation (11). Consider the Poisson model. The predictor is given in (24) and has the form $\mu_{ij} = \mu_{11} + \sum_{\ell=2}^{i} \Delta\alpha_\ell + \sum_{\ell=2}^{j} \Delta\beta_\ell$, while the log likelihood is as in (25). This results in maximum likelihood estimators $\Delta\widehat{\alpha}_i$, $\Delta\widehat{\beta}_j$ and $\widehat{\mu}_{11}$ presented in (26)–(28), and in turn, the development factor F_j is maximum likelihood estimator for Φ_j given in (32). When combining these we get the chain ladder forecast $\widetilde{Y}_{ij} = R_i(F_j - 1)\prod_{\ell=k+2-i}^{j-1} F_\ell$ in (11).

The derivations sketched above are symmetric in row and column. Suppose we transpose the data triangle by swapping rows and columns, so that i, j become j, i and R_i, C_j become C_j, R_i, while $\Delta\alpha_i, \Delta\beta_j$ become $\Delta\beta_j, \Delta\alpha_i$. Correspondingly, $\Delta\widehat{\alpha}_i, \Delta\widehat{\beta}_j$ and G_i, F_j become $\Delta\widehat{\beta}_j, \Delta\widehat{\alpha}_i$ and F_j, G_i; and then, second expression for the chain ladder forecast arises; that is, $\widetilde{Y}_{ij} = C_j(G_i - 1)\prod_{\ell=k+2-j}^{i-1} G_\ell$. □

Proof of Theorem 1. *The likelihood.* When the $\Delta\alpha_i^\dagger$s are known the likelihood is

$$\ell(\mu_{11}, \Delta\beta) = \mu_{11}\sum_{j=1}^{k} C_j + \sum_{j=2}^{k} \Delta\beta_j \sum_{\ell=j}^{k} C_j - \kappa(\mu_{11}, \Delta\beta) + h(data),$$

where h is a function of the data, not depending on the unknown parameters, while

$$\kappa(\mu_{11}, \Delta\beta) = \sum_{i=1}^{k}\sum_{j=1}^{k+1-i} \exp(\mu_{ij}) = \sum_{i=1}^{k}\sum_{j=1}^{k+1-i} \exp(\mu_{11} + \sum_{\ell=2}^{i} \Delta\alpha_\ell^\dagger + \sum_{\ell=2}^{j} \Delta\beta_\ell),$$

is the cumulant generating function. Empty sums are zero.

Uniqueness of the estimator. For a full exponential family the likelihood has a maximum if and only if the natural statistic is interior to its convex support, and then the maximum likelihood estimator is unique Barndorff-Nielsen (1978) (Theorem 9.13). The natural statistic $T_k^\dagger = \sum_{i,j\in\mathcal{I}}(Y_{ij}, C_2, \dots, C_k)'$ arises through a bijective, linear mapping of $(C_1, \dots, C_k)'$. Since $Y_{ij} \geq 0$ by the Poisson assumption, $C_j \geq 0$, with $C_j = 0$ as a possible outcome. Since $C_1, \dots, C_k$ are based on unrelated observations, the interior of the convex support is given by the condition that $C_j > 0$ for all $j = 1, \dots, k$.

Likelihood equations. Since the exponential family is regular, the k likelihood equations are $T_k^\dagger = \mathsf{E}T_k^\dagger$ Barndorff-Nielsen (1978) (Corollary 9.6). Since $\sum_{i=1}^k \sum_{j=1}^{k+1-i} Y_{ij} = \sum_{j=1}^k C_j$, this in turn implies the equations

$$C_j = \mathsf{E}C_j, \qquad \text{for } j = 1, \dots k. \tag{A1}$$

Estimating the level. The expression for $\hat{\mu}_{11}^\dagger$ arises from the first likelihood equation

$$C_1 = \mathsf{E}C_1 = \exp(\mu_{11}) \sum_{i=1}^k \exp(\alpha_i - \alpha_1),$$

since the parameters $\alpha_i - \alpha_1 = \sum_{\ell=2}^i \Delta\alpha_\ell$ are known.

Estimating the development parameters. The expression for $\Delta\hat{\beta}_j^\dagger$ arises by combining the $(j-1)$th and jth likelihood equations

$$\frac{C_j}{C_{j-1}} = \frac{\mathsf{E}C_j}{\mathsf{E}C_{j-1}} = \frac{\exp(\mu_{11} + \beta_j - \beta_1)\sum_{i=1}^{k+1-j}\exp(\alpha_i - \alpha_1)}{\exp(\mu_{11} + \beta_{j-1} - \beta_1)\sum_{i=1}^{k+2-j}\exp(\alpha_i - \alpha_1)}.$$

Recalling the expression for Γ_i in (10) this reduces to

$$\frac{C_j}{C_{j-1}} = \frac{\exp\,(\Delta\beta_j)}{\Gamma_{k+2-j}},$$

which has the desired solution. □

Proof of Theorem 2. Use the expressions from Theorem 1 to get

$$\begin{aligned} \tilde{Y}_{ij}^\dagger &= \exp(\hat{\mu}_{11}^\dagger + \alpha_i^\dagger - \alpha_1^\dagger + \hat{\beta}_j^\dagger - \hat{\beta}_1^\dagger) \\ &= \frac{C_1}{\prod_{\ell=2}^k \Gamma_\ell^\dagger}(\Gamma_i^\dagger - 1)\left(\prod_{\ell=2}^{i-1}\Gamma_\ell^\dagger\right)\frac{C_j}{C_1}\prod_{\ell=2}^{j}\Gamma_{k+2-\ell}^\dagger = C_j(\Gamma_i^\dagger - 1)\frac{\prod_{\ell=k+2-j}^k \Gamma_\ell^\dagger}{\prod_{\ell=i}^k \Gamma_\ell^\dagger}. \end{aligned}$$

We get the desired result by simplifying the last fraction using $i > k + 2 - j$. □

Proof of Equation (45). *First identity*. Combine the forecasts; see (44).

$$\tilde{Y}_{ij}^\ddagger = \exp(\hat{\mu}_{11} + \sum_{h=2}^i \Delta\alpha_h^\dagger + \sum_{h=2}^j \Delta\hat{\beta}_h), \qquad \tilde{Y}_{ij} = \exp(\hat{\mu}_{11} + \sum_{h=2}^i \Delta\hat{\alpha}_h + \sum_{h=2}^j \Delta\hat{\beta}_h).$$

Second identity. From (11) we have $\tilde{Y}_{ij} = C_j(G_i - 1)\prod_{\ell=k+2-j}^{i-1} G_\ell$. Write $G_i = \hat{N}_i/\hat{N}_{i-1}$ where $\hat{N}_i = \sum_{\ell=1}^i \exp(\sum_{h=2}^\ell \Delta\hat{\alpha}_h)$ and $\hat{N}_i - \hat{N}_{i-1} = \exp(\sum_{h=2}^i \Delta\hat{\alpha}_h)$. Then, we get

$$\tilde{Y}_{ij} = C_j \frac{\hat{N}_i - \hat{N}_{i-1}}{\hat{N}_{i-1}} \prod_{\ell=k+2-j}^{i-1} \frac{\hat{N}_\ell}{\hat{N}_{\ell-1}} = C_j \frac{\hat{N}_i - \hat{N}_{i-1}}{\hat{N}_{k+1-j}} = C_j \frac{\exp(\sum_{h=2}^i \Delta\hat{\alpha}_h)}{\sum_{\ell=1}^{k+1-j}\exp(\sum_{h=2}^\ell \Delta\hat{\alpha}_h)}.$$

Correspondingly, we get from (37), that

$$\tilde{Y}_{ij}^\dagger = C_j \frac{\exp(\sum_{h=2}^i \Delta\alpha_h^\dagger)}{\sum_{\ell=1}^{k+1-j}\exp(\sum_{h=2}^\ell \Delta\alpha_h^\dagger)}.$$

Then, combine the expressions for $\widetilde{Y}^{\dagger}_{ij}$ and $\widetilde{Y}^{\dagger}_{ij}$. □

Proof of Equation (49). The point forecast is $\tilde{Y}^{\dagger}_{ij} = \exp(\widehat{\mu}^{\dagger}_{11} + \sum_{h=2}^{i} \Delta\alpha^{\dagger}_{h} + \sum_{h=2}^{j} \Delta\widehat{\beta}^{\dagger}_{h})$. Insert the expression for $\widehat{\mu}^{\dagger}_{11}$ from (47), for $\Delta\alpha^{\dagger}_{i}$ from (48) and $\exp(\sum_{h=2}^{j} \Delta\widehat{\beta}^{\dagger}_{h}) = (F^{\dagger}_{j} - 1) \prod_{\ell=2}^{j-1} F^{\dagger}_{\ell}$, which follows from (46), to get

$$\tilde{Y}^{\dagger}_{ij} \quad = \quad \frac{R^{\dagger}_{1}}{\prod_{\ell=2}^{k} F^{\dagger}_{\ell}} \left(\frac{R^{\dagger}_{i}}{R^{\dagger}_{1}} \prod_{\ell=2}^{i} F^{\dagger}_{k+2-\ell} \right) (F^{\dagger}_{j} - 1) \prod_{\ell=2}^{j-1} F^{\dagger}_{\ell}.$$

Equation (49) follows by reducing common factors and noting that $j > k + 2 - i$. □

Proof of Equation (52). The point forecast is $\widetilde{Y}^{\ddagger}_{ij} = \exp(\widehat{\mu}_{11} + \sum_{h=2}^{i} \Delta\alpha^{\dagger}_{h} + \sum_{h=2}^{j} \Delta\widehat{\beta}_{h})$, as given in (44). Insert the expression for $\widehat{\mu}_{11}$ from (28), the expression for $\Delta\alpha^{\dagger}_{i}$ from (51) and $\exp(\sum_{h=2}^{j} \Delta\widehat{\beta}_{h}) = (F_{j} - 1) \prod_{\ell=2}^{j-1} F_{\ell}$, which follows from (32) noting that $F_j = \widehat{\Phi}_j$, to get

$$\widetilde{Y}^{\ddagger}_{ij} \quad = \quad \frac{R_{1}}{\prod_{\ell=2}^{k} F_{\ell}} \left(\frac{R^{\ddagger}_{i}}{R_{1}} \prod_{\ell=2}^{i} F_{k+2-\ell} \right) (F_{j} - 1) \prod_{\ell=2}^{j-1} F_{\ell}.$$

Equation (52) follows by reducing common factors and noting that $j > k + 2 - i$. □

Proof of Theorem 3. (a) We show that Γ_i defined in (32) increases in the $\Delta\alpha_i$'s. Write $\Gamma_i = N_i / N_{i-1}$ where $N_i = \sum_{\ell=1}^{i} \exp(\sum_{h=2}^{\ell} \Delta\alpha_h)$. Thus, we must show that the derivative of Γ_i with respect to $\Delta\alpha_n$ is positive for all $n \leq i$ and zero otherwise. It suffices to consider the numerator of that derivative, which is $\dot{N}_i N_{i-1} - N_i \dot{N}_{i-1}$. Now,

$$\dot{N}_i = \frac{\partial N_i}{\partial \Delta\alpha_n} = \sum_{\ell=n}^{i} \exp(\sum_{h=2}^{\ell} \Delta\alpha_h) = N_i - N_{n-1},$$

for $n \leq i$ and zero otherwise. This implies $\dot{N}_i N_{i-1} - N_i \dot{N}_{i-1} = N_{n-1}(N_i - N_{i-1})$, noting that the cases where $n < i$ and $n = i$ have to be checked separately. The desired result now follows by noting that N_{n-1} and $N_i - N_{i-1}$ are both positive.

(b) Using (35) and (a) we get

$$\Delta\widehat{\beta}^{\dagger}_{j} = \Delta \log C_j + \log \Gamma^{\dagger}_{k+2-j} > \Delta \log C_j + \log G_{k+2-j} = \Delta\widehat{\beta}_j,$$

where the last equality is of a similar type as (35) and comes from Theorem 3 in Kuang et al. (2009).

(c) Using (36) and (a) we get

$$\widehat{\mu}^{\dagger}_{11} = \log C_1 - \sum_{\ell=2}^{k} \log \Gamma^{\dagger}_{\ell} < \log C_1 - \sum_{\ell=2}^{k} \log G_{\ell} = \widehat{\mu}_{11},$$

where the last equality comes from from Theorem 3 in Kuang et al. (2009).

(d) First, we compare the new reserve $\widetilde{Y}^{\dagger}_{ij}$ with $\Delta\alpha^{\dagger}_{i}$ known to the old reserve $\widetilde{Y}_{ij}$ from CL. Since $1 \leq G_i < \Gamma^{\dagger}_{i}$ for $2 \leq i \leq k$, by (11), (a) and (37),

$$\widetilde{Y}_{ij} = C_j(G_i - 1) \prod_{\ell=k+2-j}^{i-1} G_{\ell} < C_j(\Gamma^{\dagger}_{i} - 1) \prod_{\ell=k+2-j}^{i-1} \Gamma^{\dagger}_{\ell} = \widetilde{Y}^{\dagger}_{ij}.$$

Second, we compare the new reserve $\widetilde{Y}^{\dagger}_{ij}$, using $\Delta\alpha^{\dagger}_{i}$, $\widehat{\mu}^{\dagger}_{11}$ and $\Delta\widehat{\beta}^{\dagger}_{j}$, to the mixed reserve $\widetilde{Y}^{\ddagger}_{ij}$, using $\Delta\alpha^{\dagger}_{i}$, $\widehat{\mu}_{11}$ and $\Delta\widehat{\beta}_{j}$. From (45) we have

$$\widetilde{Y}^{\ddagger}_{ij} = \widetilde{Y}^{\dagger}_{ij} \frac{\sum_{\ell=2}^{k+1-j} \exp(\sum_{h=2}^{\ell} \Delta\alpha^{\dagger}_{h})}{\sum_{\ell=2}^{k+1-j} \exp(\sum_{h=2}^{\ell} \Delta\widehat{\alpha}_{h})}.$$

Since $\Delta\alpha^{\dagger}_{i} > \Delta\widehat{\alpha}_{i}$, for all $2 \leq i \leq k$ it follows that $\widetilde{Y}^{\ddagger}_{ij} > \widetilde{Y}^{\dagger}_{ij}$.

(e) Similar to the argument in (a), but using the ordering for $\Delta\widehat{\beta}$ derived in (b).

(f) Equations (49) and (52) applied for any $k+2-i \leq j \leq k$ show that

$$\frac{R^{\ddagger}_{i}}{R^{\dagger}_{i}} = \frac{Y^{\ddagger}_{ij}}{Y^{\dagger}_{ij}} \frac{(F^{\dagger}_{j} - 1)\prod_{\ell=k+2-i}^{j-1} F^{\dagger}_{\ell}}{(F_{j} - 1)\prod_{\ell=k+2-i}^{j-1} F_{\ell}}.$$

Then, apply the orderings $\widetilde{Y}^{\ddagger}_{ij} > \widetilde{Y}^{\dagger}_{ij}$ and $F^{\dagger}_{j} > F_{j}$ from (d), (e).

(g) Use (32), (52) to get $R^{\ddagger}_{i}/R_{ij} = \widetilde{Y}^{\ddagger}_{i}/\widetilde{Y}_{ij}$ for all $k+2-i \leq j \leq k$. Apply (f). □

References

Agbeko, Tony, Munir Hiabu, María Dolores Martínez-Miranda, Jens P. Nielsen, and Richard J. Verrall. 2014. Validating the Double Chain Ladder Stochastic Claims Reserving Model. *Variance* 8: 138–60.

Alai, Daniel H., Michael Merz, and Mario V. Wüthrich. 2009. Mean square error of prediction in the Bornhuetter-Ferguson claims reserving method. *Annals of Actuarial Science* 1: 7–31.

Barndorff-Nielsen, Ole. 1978. *Information and Exponential Families*. New York: Wiley. [CrossRef]

Benktander, Gunnar. 1976. An approach to credibility in calculating IBNR for casualty excess reinsurance. *Actuarial Review* 3: 7–8.

Bornhuetter, Ronald L., and Ronald E. Ferguson. 1972. The actuary and IBNR. *Casualty Actuarial Society Proceedings* 59: 181–95.

Bühlmann, Hans, and Franco Moriconi. 2015. Credibility claims reserving with stochastic diagonal effects. *ASTIN Bulletin: The Journal of the IAA* 45: 309–53.

Chukhrova, Nataliya, and Arne Johannssen. 2017. State Space Models and the Kalman-Filter in Stochastic Claims Reserving: Forecasting, Filtering and Smoothing. *Risks* 5: 30.

De Felice, Massimo, and Franco Moriconi. 2019. Claim Watching and Individual Claims Reserving Using Classification and Regression Trees. *Risks* 7: 102.

De Vylder, Florian E. 1982. Estimation of IBNR claims by credibility theory. *Insurance: Mathematics and Economics* 1: 35–40. [CrossRef]

England, Peter D., and Richard J. Verrall. 2002. Stochastic claims reserving in general insurance. *British Actuarial Journal* 8: 519–44. [CrossRef]

Gabrielli, Andrea, and Mario V Wüthrich. 2018. An Individual Claims History Simulation Machine. *Risks* 6: 29. [CrossRef]

Gigante, Patrizia, Liviana Picech, and Luciano Sigalotti. 2013. Prediction error for credible claims reserves: An *h*-likelihood approach. *European Actuarial Journal* 3: 453–70. [CrossRef]

Harnau, Jonas, and Bent Nielsen. 2018. Over-dispersed age-period-cohort models. *Journal of the American Statistical Association* 113: 1722–32. [CrossRef]

Heberle, Jochen, and Anne Thomas. 2016. The fuzzy Bornhuetter-Ferguson method: An approach with fuzzy numbers. *Annals of Actuarial Science* 10: 303–21. [CrossRef]

Kremer, E. 1985. *Einführung in die Versicherungsmathematik*. Göttingen: Vandenhoeck & Ruprecht. [CrossRef]

Kuang, Di, Bent Nielsen, and Jens Perch Nielsen. 2009. Chain-Ladder as Maximum Likelihood Revisited. *Annals of Actuarial Science* 4: 105–21. [CrossRef]

Kuo, Kevin. 2019. DeepTriangle: A Deep Learning Approach to Loss Reserving. *Risks* 7: 97. [CrossRef]

Mack, Thomas. 1991. A simple parametric model for rating automobile insurance or estimating IBNR claims reserves. *ASTIN Bulletin: The Journal of the IAA* 21: 93–109. [CrossRef]

Mack, Thomas. 2000. Credible claims reserves: The Benktander method. *ASTIN Bulletin: The Journal of the IAA* 30: 333–47. [CrossRef]

Mack, Thomas. 2006. Parameter estimation for Bornhuetter/Ferguson. *Casualty Actuarial Society Forum* Fall: 141–57.

Martinek, László. 2019. Analysis of stochastic reserving models by means of naic claims data. *Risks* 7: 62.

Martínez-Miranda, María Dolores, Jens P. Nielsen, and Richard Verrall. 2013. Double Chain Ladder and Bornhuetter-Ferguson. *North American Actuarial Journal* 17: 101–13. [CrossRef]

Martínez Miranda, María Dolores, Jens P. Nielsen, Richard Verrall, and Mario V. Wüthrich. 2015. Double Chain Ladder, claims development inflation and zero-claims. *Scandinavian Actuarial Journal* 2015: 383–405. [CrossRef]

Renshaw, Arthur E., and Richard J. Verrall. 1998. A stochastic model underlying the chain-ladder technique. *British Actuarial Journal* 4: 903–23. [CrossRef]

Taylor, Gregory. 2000. *Loss Reserving: An Actuarial Perspective*. Norwel: Kluwer Academic Publishers. [CrossRef]

Verrall, Richard J. 2004. A Bayesian Generalized Linear Model for the Bornhuetter-Ferguson of claims reserving. *North American Actuarial Journal* 8: 67–89.

Wüthrich, Mario V., and Michael Merz. 2008. *Stochastic Claims Reserving Methods in Insurance*. New York: Wiley. [CrossRef]

© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).

risks

MDPI

Article

In-Sample Hazard Forecasting Based on Survival Models with Operational Time

Stephan M. Bischofberger

Cass Business School, University of London, London EC1Y 8TZ, UK; stephan.bischofberger@cass.city.ac.uk

Received: 27 November 2019; Accepted: 24 December 2019; Published: 3 January 2020

Abstract: We introduce a generalization of the one-dimensional accelerated failure time model allowing the covariate effect to be any positive function of the covariate. This function and the baseline hazard rate are estimated nonparametrically via an iterative algorithm. In an application in non-life reserving, the survival time models the settlement delay of a claim and the covariate effect is often called operational time. The accident date of a claim serves as covariate. The estimated hazard rate is a nonparametric continuous-time alternative to chain-ladder development factors in reserving and is used to forecast outstanding liabilities. Hence, we provide an extension of the chain-ladder framework for claim numbers without the assumption of independence between settlement delay and accident date. Our proposed algorithm is an unsupervised learning approach to reserving that detects operational time in the data and adjusts for it in the estimation process. Advantages of the new estimation method are illustrated in a data set consisting of paid claims from a motor insurance business line on which we forecast the number of outstanding claims.

Keywords: accelerated failure time model; chain-ladder method; local linear kernel estimation; non-life reserving; operational time

1. Introduction

The parametric accelerated failure time (AFT) model has been well established in medical statistics and other applications (Kalbfleisch and Prentice 2002) for decades. The aim of this paper is to introduce a nonparametric generalization of the one-dimensional AFT model for right-truncated data and apply it to estimate the number of outstanding claims in non-life insurance.

Given a covariate $X \in \mathbb{R}^d$ and given no failure has occurred until time t, the AFT specifies that the probability of a failure between time t and $t + \mathrm{d}t$ equals $\theta\alpha_0(\theta t)\mathrm{d}t$ with $\theta = \exp(-\beta' X)$ for an underlying hazard rate α_0 and a deterministic vector $\beta \in \mathbb{R}^d$. More formally, this model is expressed through the conditional hazard rate

$$\alpha(t|X) = \theta\alpha_0(\theta t), \qquad \theta = \exp(-\beta' X).$$

Its interpretation is straightforward, for example, in a medical context where failure time T describes the amount of time for a tumor to reach a critical stage. For each individual i, the value of θ_i depends on its covariate X_i (the patient's medical data). A value of $\theta_i = 2$, for instance, means that the development of the tumor happens twice as fast for a patient and $\theta_i = 0.9$ means 10% slower development than usual. This is in contrast to the proportional hazard model $\alpha(t|X) = \theta\alpha_0(t)$, where the interpretation of θ is non-trivial (Cox 1972). For the statistical analysis in the AFT model, one can transform the observed failure times through $T_i \mapsto \theta_i T_i$ (if one knows θ_i). The transformed survival time $\theta_i T_i$ follows the same distribution for all individuals and is independent of the covariate X_i.

The AFT model has been studied by various authors including Buckley and James (1979); Louis (1981); Miller (1976), and Ritov and Wellner (1988). Comprehensive overviews have been given in Cox and Oakes (1984) and Andersen et al. (1993). The model is still widely used and adapted to new problems in medical research. A recent modification of the AFT model has been introduced in Li and Jin (2018) and recent applications include AIDS research (Fulcher et al. 2017) and cancer research (Cho et al. 2018) among many others.

This article focuses on the one-dimensional case $d = 1$ and provides a nonparametric generalization of the parametric AFT model above assuming $\theta = 1/\varphi(X)$. We estimate φ nonparametrically and impose no structural assumption. In a finance or insurance context, the unknown function φ is often called operational time and it can accelerate or slow down the survival time T. However, with our definition, $\varphi(x)$ has the same effect as θ^{-1} in the AFT model, i.e., the effect is reversed. We can transform observed survival times T_i and covariates X_i via $T_i \mapsto T_i/\varphi(X_i) = \widetilde{T}_i$ to obtain identically distributed survival times $\widetilde{T}_i$ that are independent of their covariates X_i as in the AFT model. In our application of non-life reserving, X is the accident date of an insurance claim and T is its settlement delay which can be affected by calendar effects, seasonal effects or a trend in the speed of claims finalization over time, e.g., due to new organizational structures in the insurance company, more efficient IT systems, or changes in legislation. The latter trends over time are captured by our operational time function φ. We estimate φ and the marginal hazard rate of $\widetilde{T}$. Together, they yield an estimate of the conditional hazard rate of T given X, which contains full information of the distribution of T given X. This hazard rate is used to estimate outstanding claim numbers through extrapolation with a chain-ladder type algorithm. The proposed algorithm in this article detects the effects of operational time and adjusts for them. If there is no operational time present, the algorithm still estimates smoothed chain-ladder development factors for an optimal bandwidth that is selected through cross-validation.

The concept of operational time was originally developed for stochastic processes in Feller (1971). In actuarial research, it was first used for processes of claim numbers in Bühlmann (1970) and for non-life reserving in Reid (1978) and Taylor (1981,1982). Comprehensive summaries about operational time in reserving have been provided in Taylor et al. (2008) and Taylor and McGuire (2016). For an overview of its use in mathematical finance, we refer to Swishchuk (2016).

The algorithm in this paper is an alternative to the most widely used algorithm in non-life reserving, the chain-ladder method. The difference is that, in chain-ladder, it is assumed that accident date and settlement delay are independent, and thus chain-ladder does not account for calendar time effects like court rulings, emergence of latent claims, or changes in operational time. The first stochastic model around the chain-ladder method was introduced in Mack (1993). Chain-ladder is still widely used in the insurance industry and as a benchmark for new methods in research as explained in overviews of reserving methods in England and Verrall (2002) and, more recently, Taylor (2019). Based on the idea of chain-ladder, different multiplicative models with independent effects of accident date and settlement delay were introduced in Kremer (1982); Kuang et al. (2009); Renshaw and Verrall (1998), and Verrall (1991).

Aside from these publications, the greater part of the research on claims reserving can be summarized into two streams: a Poisson process approach and a two-dimensional kernel estimation approach for truncated data. The first (older and more extensive) stream of research focuses on Poisson process models in Antonio and Plat (2014); Avanzi et al. (2016); Huang et al. (2015); Jewell (1989, 1990); Larsen (2007), and Norberg (1993,1999). Extensions that investigate dependent covariates or marked Cox processes include Zhao and Zhou (2010); Zhao et al. (2009), or Badescu et al. (2016), respectively. A semiparametric approach very similar to operational time is given in Crevecoeur et al. (2019), in which the authors allow time on weekends and public holidays to pass faster in order to make up for less claim reports on these days while ensuring a continuous distribution of reporting delay.

The approach in this present paper fits into the second stream of reserving research based on “continuous chain-ladder” (Hiabu et al. (2016); Lee et al. 2015, 2017; Martínez-Miranda et al. (2013)). In a broader statistical context, the problem was introduced as “in-sample forecasting” (Mammen et al. 2015) and said papers applied their results to forecasting problems beyond actuarial research. These articles have in common that no distributional assumptions are made and that kernel estimation is performed under the assumption of a structural model for the joint density or conditional hazard rate. In the operational time model, Lee et al. (2017) assume the nonparametric factor $\theta = \psi(X)$, i.e., $\psi(X) = \varphi^{-1}(X)$, and estimate ψ as well as the two marginal densities of accident year and settlement delay. The latter is the closest approach to this present paper; however, we estimate a conditional hazard rate instead of a multivariate density. The advantage of our approach is that we only estimate two functions (φ and α_0) instead of three, and then extrapolate claim numbers to estimate the number of outstanding claims. This extrapolation is analogous to the algorithm in the chain-ladder method. Since our estimated conditional hazard rates are similar to chain-ladder development factors (Hiabu 2017), we consider it more natural to extrapolate in a hazard framework than to perform extrapolation with density functions. Therefore, we only forecast the effect of the accident date X and do not estimate its distribution. All mentioned continuous chain-ladder publications including this present paper focus on claim numbers instead of payment amounts. Recently, Bischofberger et al. (2019) have shown how to extend the models and estimators for payment amounts. This extension is also feasible for our approach; however, adding extensive additional technicalities is beyond the scope of this paper.

In traditional statistical learning, learning problems are classified as "supervised" and "unsupervised" (Hastie et al. 2008). For supervised learning algorithms, the goal is to predict an outcome measure for a given input. For this purpose, the algorithm trains on paired data consisting of (input, output) and then applies the learned structure to predict an output from a new input. On the other hand, in unsupervised learning, there is no output in the data and the goal of the algorithm is often to find patterns in the data minimizing a loss criterion. Nonparametric kernel estimation is used in both approaches: for nonparametric regression in supervised learning and for kernel density estimation in unsupervised learning (Hastie et al. 2008). The new forecasting procedure in this article can be classified as an unsupervised machine learning technique. Although the goal is to give an estimate of the number of outstanding claims from past data, our algorithm cannot be trained on a data set of input and output (in form of past claims and future claims) and then applied to a new input. The presented algorithm estimates the conditional distribution of settlement delay given the accident date that is specific for the data set it is used on. This estimation involves kernel hazard estimators and the minimization of a loss function.

Very recently, following a trend in applied statistics, various other machine learning approaches to claims reserving that do not belong to any of the previous streams have arisen. Soon these articles may constitute a third big stream of research. Useful machine learning techniques for reserving include regression trees (Baudry and Robert 2019; Wüthrich 2018) and neural networks (Kuo 2019) among others. These approaches also take dependence between accident date and delay into account and are thus more flexible than many of the aforementioned models. In contrast to the algorithm in this article, they are all based on supervised learning. A neural network architecture based on classical chain-ladder literature, into which the over-dispersed Poisson reserving model of Renshaw and Verrall (1998) is embedded, has been introduced in Gabrielli et al. (2019).

This article is structured as follows. The underlying mathematical model is introduced in Section 2. Section 3 explains an algorithm to estimate operational time and the baseline hazard. Section 4 illustrates how to estimate outstanding liabilities from an operational time and a baseline hazard estimate. A data-driven bandwidth selection procedure is introduced in Sections 5 and 6 containing an illustration for a real data set.

2. Model

We start with a general mathematical model for hazard rates with operational time but without filtering and afterwards adapt it to observations on a run-off triangle in the context of claims reserving. Since this particular triangular data structure can be expressed as truncated data, a counting process survival model lends itself to our cause.

2.1. General Model

Let (T, X) be a two-dimensional random variable on the square $\mathcal{S} = \{(t,x) : 0 \leq t, x \leq \mathcal{T}\})$ for $\mathcal{T} \geq 0$. Suppose that T can be written as

$$T = \widetilde{T}\varphi(X) \tag{1}$$

for a random variable $\widetilde{T}$ that is independent of X and a function $\varphi : [0, \mathcal{T}] \to [0, \mathcal{T}/\widetilde{\mathcal{T}}]$. We call φ operational time and Equation (1) operational time model. The support of $\widetilde{T}$ is $[0, \widetilde{\mathcal{T}}]$ for some $0 \leq \widetilde{\mathcal{T}} \leq \mathcal{T}$. In the sequel, we define quantities for each realization (T_i, X_i) of (T, X) with $i = 1, \ldots, n$.

In a counting process framework, we identify the survival time with T_i and treat X_i as a one-dimensional covariate. We first define the counting process setting before linking it to the random variable (T_i, X_i). Suppose we observe a counting process $\{N_i(t) : 0 \leq t \leq \mathcal{T}\}$ with respect to a suitable filtration $\{\mathcal{F}_t^i : 0 \leq t \leq \mathcal{T}\}$ (Andersen et al. 1993, p. 60). The intensity of N_i at time t is defined as

$$\lambda_i(t) = \lim_{h \downarrow 0} h^{-1} E[N_i((t+h)-) - N_i(t-)|\, \mathcal{F}_{i,t-}].$$

To illustrate the effect of operational time on the intensity, we start with a simple model for unfiltered data, denoted by the superscript $^{\text{unfilt}}$. We use the notation with superscripts since we will focus on a specific hazard later on, for which we want to reserve the plain notation λ and N. For illustration, we define the counting process $N_i^{\text{unfilt}}(t) = I(T_i \leq t)$ with the adapted filtration $\mathcal{F}_{i,t}^{\text{unfilt}} = \sigma(\{N_i^{\text{unfilt}}(s), X_i(s), s \leq t\})$. The intensity of N_i^{unfilt} given $X_i(t) = x$ satisfies Aalen's multiplicative model (Aalen 1980) with

$$\begin{aligned}\lambda_i^{\text{unfilt}}(t) &= \alpha^{\text{unfilt}}(t|x) I(t \leq T_i), \\ &= \frac{1}{\varphi(x)} \alpha_0^{\text{unfilt}}\left(\frac{t}{\varphi(x)}\right) I(t \leq T_i),\end{aligned}$$

where $\alpha^{\text{unfilt}}(t|x) = \lim_{h \downarrow 0} h^{-1} P(T \in [t, t+h)|\, T \geq t, X = x)$ is the conditional hazard of T given X and $\alpha_0^{\text{unfilt}}(\tilde{t}) = \lim_{h \downarrow 0} h^{-1} P(\widetilde{T} \in [\tilde{t}, \tilde{t}+h)|\widetilde{T} \geq \tilde{t})$ is the marginal hazard of $\widetilde{T}$. We want to emphasize that the hazard rate α_0^{unfilt} of $\widetilde{T}$ is in particular not conditioned on X because $\widetilde{T}$ and X are independent. The fact that α_0^{unfilt} is a function of just one argument is the advantage of assuming the structural Model (1) because one can now easily derive an estimator for α_0^{unfilt}. For unique identification of φ, we choose the normalization $\varphi(0) = 1$ in the sequel.

The advantage of this framework is that we can easily handle certain filtering schemes like right-censoring and left-truncation. If the observations of T are right-censored, we observe (X_i, T_i^*, δ_i) where $T_i^* = \min\{T_i, C\}$ is the censored value of T_i with respect to some censoring time C and $\delta_i = I(T_i < C)$ is the corresponding censoring variable. Moreover, suppose our observations to be left-truncated. In particular, we assume the special case of left-truncation $X_i \leq T_i$. Hence, we use the counting process $N_i^{\text{filt}}(t) = I(T_i \leq t)\delta_i$ with respect to its adapted filtration $\mathcal{F}_{i,t}^{\text{filt}} = \sigma(\{N_i^{\text{filt}}(s), X_i(s), s \leq t\})$ and with intensity

$$\lambda_i^{\text{filt}}(t) = \alpha^{\text{filt}}(t|X_i(t)) Z_i^{\text{filt}}(t),$$

for exposure $Z_i^{\text{filt}}(t) = I(X_i \leq t < T_i^*)$. The conditional hazard has the same structure as in the last case,

$$\alpha^{\text{filt}}(t|x) = \frac{1}{\varphi(x)} \alpha_0^{\text{filt}}\left(\frac{t}{\varphi(x)}\right). \tag{2}$$

The model in Equation (2) has been investigated for nonparametric regression in Linton et al. (2011); however, their model did not allow for right-truncation in run-off triangles.

The next chapter introduces the operational time hazard model for right-truncation, which will be used in the sequel.

2.2. Model on the Run-Off Triangle with Right-Truncation

When estimating future claim numbers, reserving departments in the non-life insurance industry work with data of historical claims aggregated in two dimensions: the accident date of the claim and the settlement delay, i.e., the time between accident date and payment to the policy holder. Note that, as in the chain-ladder method, we will not need the number of individuals under risk (the number of underwritten policies) for the estimation of future claim numbers. Therefore, we suppose the data contain only paid claims.

We denote by X the underwriting date of the policy and by T the settlement delay. Hence, adopting the notation from above, we follow the settlement delay as survival time and the accident date as covariate on which we will condition. In Model (1), the operational time function φ links a non-observable random delay $\widetilde{T}$ to the observed settlement delay depending on the accident date. The independent delay $\widetilde{T}$ can be seen as pure delay, cleared of all external factors. A value $\varphi(x) > 1$ implies a larger delay T with the heuristic that "time is running slower" and vice versa. This is best explained on the data set that is used in Section 6. The estimator of φ has values smaller than 1 for accident dates after January 2006 (see Section 6). This phenomenon is most likely due to the improved use of technology in the insurance company and has also been observed on the same data set in Lee et al. (2017). Instead of treating this as a special case, we let time run faster in this period and use the same delay throughout the whole range of accident dates. In particular, this does prevent discontinuities in the distribution of the delay. Since time was running faster for accident dates in 2006 and later, their actual delay effect $\widetilde{T}$, cleared of operational time, is larger (and sometimes even beyond the diagonal in the run-off triangle) in Figure 1. The operational time estimate already has a downwards trend for accident dates in 2004 and 2005; however, values in 2004 and and at the end of 2005 are larger than 1. On these dates, time was running slower in our model which is why the independent delay $\widetilde{T}$ for early accidents is slightly shorter than in the original data.

To adapt the operational time hazard Model (2) to the needs of our application, we assume pairs of observations (T_i, X_i), $i = 1, \ldots, n$, on the triangle $\mathcal{I} = \{(t, x) \in \mathcal{S} : 0 \leq x + t \leq \mathcal{T}\}$. Hence, we have right-truncated observations of T because it now holds $T_i \leq \mathcal{T} - X_i$. To circumvent this difficulty, we invert time and look at observations $(\mathcal{T} - T_i, X_i)$ which are left-truncated in $\mathcal{T} - T_i$ (Ware and DeMets 1976), so we can apply Model (2). Note that our observations (X_i, T_i) only have the same distribution as (X, T) if conditioned on $\{X + T \leq \mathcal{T}\}$. We do not assume any censoring in the following.

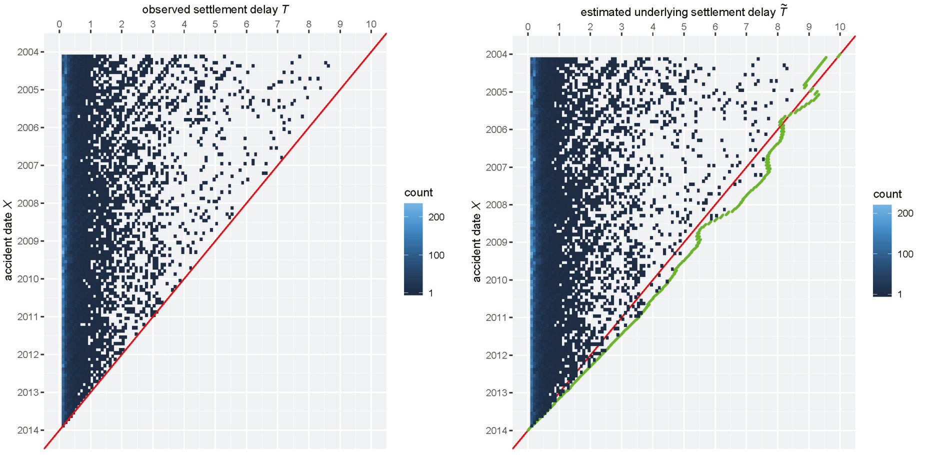
As before, we focus on a counting process $N_i(t) = I(\mathcal{T} - T_i \leq t)$. The intensity of the time-reversed counting process N_i with respect to its natural filtration now equals

$$\lambda_i(t) = \alpha(t|x) Z_i(t),$$

where $\alpha(t|x)$ is the conditional hazard of $\mathcal{T} - T$ given $X = x$ and $Z_i(t) = I(t + X_i \leq \mathcal{T}, t \leq \mathcal{T} - T_i)$. In particular, we get

$$\alpha(t|x) = \frac{1}{\varphi(x)} \alpha_0 \left(\mathcal{T} - \frac{\mathcal{T} - t}{\varphi(x)} \right), \tag{3}$$

with the marginal hazard $\alpha_0(z) = \lim_{h \downarrow 0} h^{-1} P(\mathcal{T} - \widetilde{T} \in [z, z+h) | \mathcal{T} - \widetilde{T} \geq z)$ of $\mathcal{T} - \widetilde{T}$ since $\mathcal{T} - \widetilde{T}$ and X are independent. We will also refer to α_0 as a baseline hazard in the following. The reason for the unintuitive argument of α_0 is that operational time is defined for T in "forward time"; however, α_0 is the hazard rate in reversed time but cleared of operational time, c.f., Equation (2).



(**a**) (**b**)

Figure 1. Original data and data with unobservable delay $\widetilde{T}$ cleared of operational time. The operational time in (**b**) is estimated in Section 6. Claim counts are aggregated into monthly bins for visualization, and settlement delay is displayed in years. The red line represents the date of data collection and the green points are the date of data collection cleared of operational time effects (with respect to accident date). (**a**) original data; (**b**) data cleared of operational time.

It can be easily derived that our model coincides with the structured model $f(x,t) = f_1(x)f_2(t\psi(x))$ on the joint density f considered in Lee et al. (2017) for the choice f_1 and f_2 being the marginal densities of $X, \widetilde{T}$, respectively, and $\psi(x) = \varphi(x)^{-1}$. The advantage of our approach is that we only estimate two functions φ and α_0 instead of three because we use the algorithm illustrated in Section 4 to estimate the outstanding reserve. Hence, we only forecast the effect of given underwriting data $X = x$ and do not estimate the distribution of X. For full inference on X, the roles of T and X have to be swapped.

Note that a multivariate extension of the operational time Model (1) for covariates $X \in \mathbb{R}^d$ and $\varphi : \mathbb{R}^d \to \mathbb{R}$ with $d > 1$ is possible and would result in the same hazard Model (3) with analogous baseline hazard α_0 if right-truncation is well-defined (for instance $\mathcal{T} - T_i \leq X_{i,1}$ with $X_{i,1}$ being first component of X_i). However, the estimation of φ and α_0 explained in the next section would get rather involved including a d-dimensional numerical minimization for the estimation of φ.

3. Estimation of Baseline Hazard and Operational Time

In this section, we show how to estimate the components φ and α_0 and then combine the estimators into a structured estimator of the conditional hazard. We want to recall that the whole estimation procedure is done in reversed time $\mathcal{T} - T$ instead of T for the reporting delay. Hence, the following estimators are defined for $N_i(t) = I(\mathcal{T} - T_i \leq t)$ and $Z_i(t) = I(t + X_i \leq \mathcal{T}, t \leq \mathcal{T} - T_i)$. This technical difficulty is necessary because of the right-truncation described in the last section. However, it does not constitute an issue since, once the components are estimated, we can evaluate all functions at $\mathcal{T} - t$ to get the results for t. We also want to remark again that the underwriting date X is always considered in "forward time". In the following, we see the conditional hazard $\alpha(t|x)$ as a function of two arguments $\alpha(t,x)$ and denote its estimators by $\hat{\alpha}(t,x)$. The unstructured hazard estimator in step 1 is analogously denoted by $\hat{\alpha}^{[0]}(t,x)$.

The proposed estimation procedure is as follows. The necessary expressions (7) and (10), and the loss criterion (11) will be introduced below:

1. Estimate the (unstructured) conditional hazard by $\hat{\alpha}^{[0]}(t,x)$ through Equation (7).
2. Set $\hat{\varphi}^{[0]} \equiv 1$ and $r = 1$.
3. Estimate $\hat{\alpha}_0^{[r]}$ through Equation (10) using $\hat{\varphi}^{[r-1]}$.
4. Estimate $\hat{\varphi}^{[r]}$ by minimizing the loss in (11) numerically for every x using $\hat{\alpha}_0^{[r-1]}$.
5. Repeat steps 3 and 4 for $r = 2, 3, 4, \ldots$ until the convergence criterion

   $$\int_0^{\mathcal{T}} \left(\hat{\varphi}^{[r]}(x) - \hat{\varphi}^{[r-1]}(x)\right)^2 \mathrm{d}x < 10^{-5}$$

   is satisfied in iteration r^*.
6. Set the final conditional hazard estimator to

   $$\hat{\alpha}(t,x) = \frac{1}{\hat{\varphi}(x)}\hat{\alpha}_0\left(\mathcal{T} - \frac{\mathcal{T}-t}{\hat{\varphi}(x)}\right), \tag{4}$$

   for

   $$\hat{\varphi} = \hat{\varphi}^{[r^*]}, \tag{5}$$
   $$\hat{\alpha}_0 = \hat{\alpha}_0^{[r^*]}. \tag{6}$$

   The final estimator in (non-reversed) "forward time" is set to

   $$\hat{\alpha}^f(t,x) = \hat{\alpha}(\mathcal{T}-t,x).$$

Note that the first conditional estimator $\hat{\alpha}^{[0]}(t,x)$ in step 1 is unstructured, which means that, in general, it does not satisfy Equation (3). We also want to remark that the final estimator $\hat{\alpha}^f(t,x)$ is used to extrapolate claim numbers in the next section. Despite being more intuitive, it does not occur in a well-defined model because of the right-truncation $T \leq \mathcal{T} - X$.

All estimators $\hat{\alpha}^{[0]}(t,x)$, $\hat{\alpha}_0^{[r]}$, $\hat{\varphi}^{[r]}$ are defined via integrated quadratic loss criteria and the hazard estimators $\hat{\alpha}^{[0]}(t,x)$, $\hat{\alpha}_0^{[r]}$ have closed form representations as local linear kernel estimators.

3.1. Pre-Step: Unstructured Conditional Hazard

We start with the unstructured conditional hazard estimator. Let $U_i(t) = (t, X_i(t))$ and $u = (t,x)$ to simplify the notation. For convenience, we will also write $u = (u_1, u_2)$. For any (t,x), the local linear kernel hazard estimator $\hat{\alpha}^{[0]}(t,x)$ is defined as the first component θ_0 minimizing the loss function

$$L(\theta_0,\theta_1) = \sum_{i=1}^n \int \left[\left(\frac{1}{\varepsilon}\int_s^{s+\varepsilon} \mathrm{d}N_i(v) - \theta_0 - \theta_1^T(u - U_i(s))\right)^2 - \xi(\varepsilon)\right] K_b(u - U_i(s)) Z_i(s)\mathrm{d}s,$$

for $\theta_0, \theta_1 \in \mathbb{R}$. Moreover, we use a two-dimensional kernel K and bandwidth $b = (b_1, b_2)$ for $b_1, b_2 > 0$ as well as the common notation $K_b(u_1,u_2) = b_1^{-1}b_2^{-1}K(u_1/b_1, u_2/b_2)$. The term $\xi(\varepsilon) = \left(\varepsilon^{-1}\int_s^{s+\varepsilon}\mathrm{d}N^i(s)\right)^2$ is needed to make the expression well-defined. The loss criterion L results in the closed form solution

$$\hat{\alpha}^{[0]}(t,x) = \frac{\widehat{O}(t,x)}{\widehat{E}(t,x)}, \tag{7}$$

for occurrence and exposure estimators

$$\begin{aligned} \widehat{O}(u_1,u_2) &= \frac{1}{n}\sum_{i=1}^n \int \left[1 - (u - U_i(s))D(u)^{-1}c_1(u)\right] K_b(u - U_i(s))\mathrm{d}N_i(s), \\ \widehat{E}(u_1,u_2) &= \frac{1}{n}\sum_{i=1}^n \int \left[1 - (u - U_i(s))D(u)^{-1}c_1(u)\right] K_b(u - U_i(s)) Z_i(s)\mathrm{d}s, \end{aligned} \tag{8}$$

where the components of the two-dimensional vector c_1 are

$$c_{1j}(u) = n^{-1} \sum_{i=1}^{n} \int K_b(u - U_i(s))(u_j - U_{ij}(s))Z_i(s)\mathrm{d}s, \quad j = 0, \ldots, d,$$

and the entries $(d_{jk})_{j,k=1,2}$ of the (2×2)-dimensional matrix $D(u)$ are given by

$$d_{jk}(u) = n^{-1} \sum_{i=1}^{n} \int K_b(u - U_i(s))(u_j - U_{ij}(s))(u_k - U_{ik}(s))Z_i(s)\mathrm{d}s.$$

The closed form solution $\hat{\alpha}^{[0]}$ has been derived in Nielsen (1998). This paper focusses on the local linear kernel estimator because of its good performance at boundaries. The simpler and more intuitive (Nadaraya–Watson type) local constant hazard estimator is given in Appendix A as an alternative. However, on bounded support, it is known to suffer from bias at boundaries (Nielsen 1998; Nielsen and Tanggaard 2001).

3.2. Estimation of Baseline Hazard Given Operational Time

Starting with the pilot estimator $\hat{\varphi}^{[0]} \equiv 1$, we calculate the first iteration $\hat{\alpha}_0^{[1]}$ and then recursively update $\hat{\alpha}_0^{[r]}$ making use of $\hat{\varphi}^{[r-1]}$. For the r-th iteration, we define $\hat{\alpha}_0^{[r]}$ as the hazard rate α_0 minimizing the loss

$$l(\alpha_0, \varphi, \hat{\alpha}) = \int_0^{\mathcal{T}} \int_0^{\mathcal{T}} \left[\hat{\alpha}(t,x) - \frac{1}{\varphi(x)} \alpha_0 \left(\mathcal{T} - \frac{\mathcal{T} - t}{\varphi(x)} \right) \right]^2 (\hat{\alpha}(t,x))^{-1} \widehat{E}(t,x) w(t,x) \mathrm{d}x \, \mathrm{d}t, \tag{9}$$

for operational time $\varphi = \hat{\varphi}^{[r-1]}$ and the conditional hazard estimate $\hat{\alpha} = \hat{\alpha}^{[0]}$. The loss function reflects the principle of minimizing a chi-square criterion (Berkson 1980) in which a least squares criterion is weighted by an estimate of the inverse of the asymptotic variance of $\hat{\alpha}(t,x))$, here $(\hat{\alpha}(t,x))^{-1}\widehat{E}(t,x)$. The function w is a weighting function, which is used to ensure that the resulting hazard estimator is a ratio between an occurrence estimator and an exposure estimator. It will be specified later. The minimization of (9) has the analytic solution

$$\hat{\alpha}_0(t) = \frac{\int_0^{\mathcal{T}} \widehat{E}(\hat{\varphi}_*^{[r-1]}(t,x), x) w(\hat{\varphi}_*^{[r-1]}(t,x), x) \mathrm{d}x}{\int_0^{\mathcal{T}} \widehat{E}(\hat{\varphi}_*^{[r-1]}(t,x), x) w(\hat{\varphi}_*^{[r-1]}(t,x), x) \hat{\varphi}(x)^{-1} \hat{\alpha}(\hat{\varphi}_*^{[r-1]}(t,x), x)^{-1} \mathrm{d}x},$$

where $\hat{\varphi}_*^{[r-1]}(t,x) = T - (T - t)\hat{\varphi}^{[r-1]}(x)$ for $t \in [0, \mathcal{T}]$. The derivation is analogous to Linton et al. (2011). Now, setting the weighting $w(t,x) = \hat{\varphi}^{[r-1]}(x)\hat{\alpha}(t,x)$ results in

$$\hat{\alpha}_0^{[r]}(t) = \frac{\int_0^{\mathcal{T}} \widehat{O}(\hat{\varphi}_*^{[r-1]}(t,x), x) \hat{\varphi}^{[r-1]}(x) \mathrm{d}x}{\int_0^{\mathcal{T}} \widehat{E}(\hat{\varphi}_*^{[r-1]}(t,x), x) \mathrm{d}x}. \tag{10}$$

The transformation $\varphi_*(t,x) = \mathcal{T} - (\mathcal{T} - t)\varphi(x) = t\varphi(x) + (1 - \varphi(x))\mathcal{T}$ adds the effect of operational time to occurrence and exposure estimators that were constructed with respect to $\widetilde{T}$. The function $\hat{\varphi}_*^{[r]}$ is the estimate of φ_* in the r-th iteration. Hence, we evaluate $\widehat{O}$ and $\widehat{E}$ at x but at the value of t that was corrected with the operational time effect.

It is worth pointing out that we do not get two marginal one-dimensional hazard estimator despite X and the cleared delay $\widetilde{T} = T/\varphi(X)$ being independent. This makes the implementation quite involved.

3.3. Estimation of Operational Time Given Baseline Hazard

To estimate φ in the r-th iteration, we minimize the loss function in Equation (9) in φ given the baseline hazard $\alpha_0 = \hat{\alpha}_0^{[r-1]}$ and the conditional hazard estimate $\hat{\alpha} = \hat{\alpha}^{[0]}$. Since there is no closed form solution to this problem (Linton et al. 2011), one has to minimize it numerically point-wise in x. Moreover, we set $w(t,x) = \hat{\varphi}^{[r-1]}(x)\hat{\alpha}(t,x)$ with the last estimator $\hat{\varphi}^{[r-1]}$ of φ as above. Hence, for every $x \in [0, \mathcal{T}]$, we minimize

$$l_{\hat{\varphi}^{[-1]}}(\hat{\alpha}_0^{[r-1]}, \theta, x, \hat{\alpha}) = \int_0^{\mathcal{T}} \left[\hat{\alpha}(t,x) - \frac{1}{\theta}\hat{\alpha}_0^{[r-1]}\left(\mathcal{T} - \frac{\mathcal{T}-t}{\theta}\right)\right]^2 \hat{\varphi}^{[-1]}(x)\widehat{E}(t,x)\mathrm{d}t \tag{11}$$

numerically for values $\theta \in [c_1, c_2]$. The values $c_1 \leq 1 \leq c_2$ have to be chosen manually. We define $\hat{\varphi}^{[r]}$ to be the function minimizing (11) point-wise in x. For unique identification of φ and α_0, we set the normalization $\varphi(0) = 1$.

Since there is no closed form solution of $\hat{\varphi}^{[r]}$ and the occurrence and exposure estimators $\widehat{O}$ and $\widehat{E}$ in Equation (10) depend on both t and x, asymptotic theory of our results is not straightforward and thus beyond the scope of this present paper. These difficulties arise due to the time-reversion that was necessary to derive estimators for right-truncated data. Asymptotic properties for analogous estimators on observations that are not right-truncated have been derived in Linton et al. (2011) in a non-parametric regression context. The fact that we cope with both right-truncation as present in run-off triangles and operational time distinguishes this present paper from preceding work. For a straightforward derivation of asymptotic properties of $\hat{\alpha}_0$ with standard counting process arguments as in Andersen et al. (1993), one would have to make further assumptions. A feasible approach would be to assume that φ can be estimated at a parametric $n^{-1/2}$-rate, which is possible for instance in a finite parametrization. Being against the distribution-free nature of this paper (and its benchmark the chain-ladder method), we decided against this simplification.

A modification of the proposed hazard estimator $\hat{\alpha}(t,x)$ that has been proved efficient for large sample sizes would be a two-step multiplicative bias correction, which has been introduced for local linear kernel hazard estimators in Nielsen and Tanggaard (2001). Since this paper aims at explaining a new model and estimation procedure, and a bias correction method would add a lot of notation and complexity that might distract from our new idea, such an extension is left for future research.

4. Estimating Outstanding Claim Amounts

We use our hazard estimator $\hat{\alpha}(t,x)$ to forecast outstanding claim amounts in a similar way development factors are used in the chain-ladder method. In chain-ladder with yearly aggregated data, the j-th development factor $\hat{\lambda}_j$ is effectively the ratio between claims whose payments are up to $j+1$ years delayed and those whose payments are up to j years delayed. For each claim, this yields an estimate of the probability that the payment will be $j+1$ years delayed given it has not been made within the first j years. Certainly, for more granular data, the time periods are shorter, but the principle stays the same.

In order to formally define development factors, one must first introduce the way data are aggregated in run-off triangles (England and Verrall 2002). The data are given as $(T_i, X_i) \in \mathcal{I}$, $i = 1, \ldots, n$, for the triangle $\mathcal{I} = \{(t,x) \in \mathcal{S} : 0 \leq x + t \leq \mathcal{T}\}$. The accident date X_i is given in days from the beginning of data collection and settlement delay T_i is given in days. The last day of data collection $\mathcal{T}$ is also expressed in days since day 0, and it is implicitly assumed to be the largest possible delay. The last assumption is commonly made in industry for data sets covering large enough time periods (usually if $\mathcal{T} \geq 7$ years or $\mathcal{T} \geq 10$ years). It is then said that the triangle $\mathcal{I}$ is "fully run off".

We adopt the notation of England and Verrall (2002) to introduce development factors. Suppose our data have been aggregated into $m \times m$ bins with edge length δ. In the $(m \times m)$-matrix C, we count the number of observations per bin. Its entries C_{kj} are defined as the number of claims

i for which T_i is in bin j and X_i is in bin k. In another matrix D, the cumulative numbers of events with respect to T are given by $D_{kj} = \sum_{l=1}^{j} C_{kl}$ for $j, k = 1, \dots, m$. The triangle $\{D_{kj} : j + k > \mathcal{T}\}$ represents the future and therefore contains no claim counts. This is the part we want to forecast. Now, the development factors $\{\lambda_j : j = 1, \dots, m-1\}$ are defined as

$$\hat{\lambda}_j = \frac{\sum_{k=1}^{m-j} D_{k,j+1}}{\sum_{k=1}^{m-j} D_{k,j}} = \frac{\sum_{k=1}^{m-j} \sum_{k=1}^{j+1} C_{kl}}{\sum_{k=1}^{m-j} \sum_{l=1}^{j} C_{kl}}, \qquad j = 1, \dots, m-1.$$

For the calculation of $\hat{\lambda}_j$, the last available entry with claims that were delayed $j-1$ years ($D_{m-j+2,j}$ in row $m-j+2$) is omitted, which can be seen as scaling by exposure. In the chain-ladder method, the development factors $\hat{\lambda}_j$ are then used to extrapolate the claim numbers in the cumulative matrix D into the future via

$$\begin{aligned} \hat{D}^{CL}_{k,m-k+2} &= D_{k,m-k+1}\hat{\lambda}_{m-k+1}, \\ \hat{D}^{CL}_{k,l} &= \hat{D}^{CL}_{k,l-1}\hat{\lambda}_{l-1}, \qquad l = m-k+3, \dots, m, \end{aligned} \tag{12}$$

and for $k = 2, \dots, n$. The total number of outstanding claims is then given by the last column of the estimated cumulative aggregated data $\sum_{j=2}^{m} \hat{D}^{CL}_{k,j}$.

We now link development factors to hazard estimation. Hiabu (2017) has proved the asymptotic relationship

$$\hat{\lambda}_j = \frac{1}{1 - \delta\hat{\alpha}_H(\mathcal{T} - t_j)} + o_P(1), \quad t_j \in I_j,$$

for $\hat{\alpha}_H$ being a histogram-type hazard estimator of the delay in reversed time, I_j the j-th bin of the aggregated data, and δ the bin width that satisfies $\delta = \delta_n \to 0$ for $n \to \infty$. However, this relationship was introduced under the assumption that accident date and settlement delay are independent. As an alternative for our Model (1), we define granular time-dependent development factors as

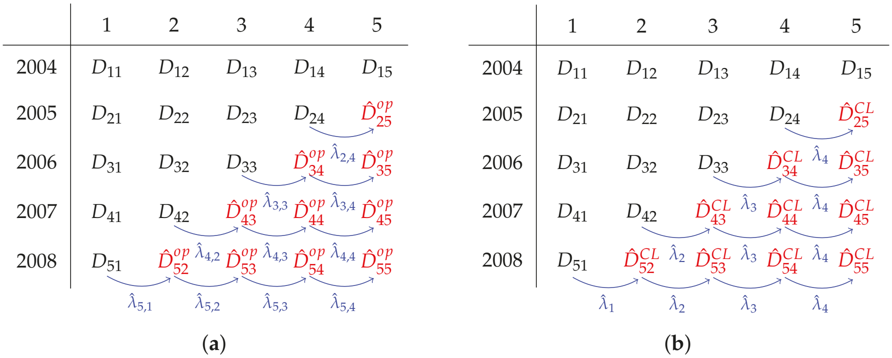
$$\hat{\lambda}_{k,j} = \frac{1}{1 - \delta\hat{\alpha}(\mathcal{T} - t_j, x_k)}, \quad (x_k, t_j) \in I_k \times J_j,$$

where I_j is the j-th bin for the delay and J_k the k-th one for accident date for $k = 2, \dots, m$. Then, we use our time-dependent development factors to forecasts reserves from a granular cumulative triangle D via

$$\begin{aligned} \hat{D}^{op}_{k,m-k+2} &= D_{k,m-k+1}\hat{\lambda}_{k,m-k+1}, \\ \hat{D}^{op}_{k,l} &= \hat{D}^{op}_{k,l-1}\hat{\lambda}_{k,l-1}, \qquad l = m-k+3, \dots, m, \end{aligned} \tag{13}$$

and for $k = 2, \dots, m$. The difference to chain-ladder is that our development factors additionally depend on the row k and that we calculate them on a finer grid, i.e., smaller δ, larger m, and more granular matrices C and D. In the application in Section 6, we use monthly aggregated data for the operational time hazard estimator and quarterly aggregated data for chain-ladder. Ideally, daily or even more granular data should be used for the proposed hazard estimator; however, this was practically computationally infeasible in our application. Analogously to chain-ladder, our final estimate for the number of outstanding payments is the last column in the estimated cumulative triangle $\sum_{j=2}^{m} \hat{D}^{op}_{k,j}$.

Figure 2 illustrates how development factors are used for extrapolation. The cumulated data is given in black, forecasts are in red and all development factors are given in blue. Our proposed time-dependent development factors can be used like traditional development factors but vary for different rows of the cumulative triangle. The illustration in Figure 2 does not show the fact that our time-dependent development factors are computed on a finer scale than for chain-ladder. Moreover, the shift x-direction through operational time $\varphi(x)$ cannot be seen in the illustration.



	1	2	3	4	5
2004	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}
2005	D_{21}	D_{22}	D_{23}	D_{24}	$\hat{D}^{op}_{25}$
2006	D_{31}	D_{32}	D_{33}	$\hat{D}^{op}_{34}$	$\hat{D}^{op}_{35}$
2007	D_{41}	D_{42}	$\hat{D}^{op}_{43}$	$\hat{D}^{op}_{44}$	$\hat{D}^{op}_{45}$
2008	D_{51}	$\hat{D}^{op}_{52}$	$\hat{D}^{op}_{53}$	$\hat{D}^{op}_{54}$	$\hat{D}^{op}_{55}$

	1	2	3	4	5
2004	D_{11}	D_{12}	D_{13}	D_{14}	D_{15}
2005	D_{21}	D_{22}	D_{23}	D_{24}	$\hat{D}^{CL}_{25}$
2006	D_{31}	D_{32}	D_{33}	$\hat{D}^{CL}_{34}$	$\hat{D}^{CL}_{35}$
2007	D_{41}	D_{42}	$\hat{D}^{CL}_{43}$	$\hat{D}^{CL}_{44}$	$\hat{D}^{CL}_{45}$
2008	D_{51}	$\hat{D}^{CL}_{52}$	$\hat{D}^{CL}_{53}$	$\hat{D}^{CL}_{54}$	$\hat{D}^{CL}_{55}$

Figure 2. Forecasting outstanding claim numbers with time-dependent development factors and chain-ladder development factors. Illustrative example with five accident years and maximum settlement delay of five years. (**a**) forecasting with time-dependent development factors via Equation (13); (**b**) forecasting with chain-ladder development factors via Equation (12).

5. Bandwidth Selection

For computational reasons, bandwidth selection is done via K-fold cross-validation (Lee et al. 2017) for $K = 20$. The set of observations is randomly split into K disjoint parts of equal size via $\{1, \dots, n\} = I_1 \dot{\cup} \dots \dot{\cup} I_K$. To find the optimal bandwidth, we minimize the score function

$$\hat{Q}(b) = n^{-1} \sum_{j=1}^{K} \hat{Q}_j(b)$$

for partial validation scores

$$\hat{Q}_j(b) = \left(\sum_{i \in I_j} \int_0^{\mathcal{T}} \left(\hat{\alpha}_b^{[-I_j]}(t, U_i) \right)^2 Y_i(s) \mathrm{d}s - 2 \sum_{i \in I_j} \int_0^{\mathcal{T}} \hat{\alpha}_b^{[-I_j]}(t, U_i) \mathrm{d}N_i(t) \right).$$

The estimator $\hat{\alpha}_b^{[-I_j]}$ is the estimator $\hat{\alpha}$ defined in Equation (4) with bandwidth b, but computed for observations $i \in \{1, \dots, n\} \setminus I_j$ only. It is being validated against the observations I_j. Being asymptotically equivalent, the estimate $\hat{Q}(b)$ is a proxy to the first two terms of the validation score

$$Q(b) = n^{-1} \sum_{i=1}^{n} \int_0^{\mathcal{T}} \left(\hat{\alpha}_b(t, U_i) - \alpha(t, U_i) \right)^2 Y_i(s) \mathrm{d}s$$

that occur after solving the quadratic expression in the integral, in which the true hazard α is unknown (Gámiz et al. 2013; Nielsen and Linton 1995). The preferred alternative, leave-one-out cross-validation, is practically unfeasible since the algorithm in Section 3 is too computationally expensive.

6. Application: Estimation of Outstanding Liabilities

We apply our estimation procedure on a data set from a Cypriot motor insurance business line. This data set contains $n = 51{,}216$ paid claims that were recorded between 1 January 2004 and 31 December 2013. First, we estimate operational time φ and the baseline hazard α_0 on the data set.

Making use of the resulting structured conditional hazard estimate $\hat{\alpha}$, we estimate outstanding liabilities through the approach with time-dependent development factors $\hat{\lambda}_{k,j}$ illustrated in Section 4.

For each claim $i = 1, \ldots, n$, the data set contains the accident date and the payment date. Instead of the settlement date, we define the settlement delay as the difference between payment date and accident date. Afterwards, we normalize the data such that the accident date X_i and settlement delay T_i take values in $0, \ldots, 3652$. Now, the data are arranged on a triangular shaped support $\mathcal{I}^{\text{daily}} = \{(x,t) \in \mathcal{S} : x + t \leq \mathcal{T}\}$ for $\mathcal{T} = 3652$ days with accident date x and settlement delay t as described in Section 2.2. For computational reasons, the data are aggregated into a monthly run-off triangle

$$\mathcal{I} = \{C_{j,k} : j, k = 1, \ldots, 120; j + k - 1 \leq 120\},$$

on which φ and α_0 are estimated. As the kernel function, we choose a multiplicative kernel $K(u_1, u_2) = k(u_1)k(u_2)$ with k being the Epanechnikov kernel $k(s) = 0.75(1 - s^2)I(|s| \leq 1)$. The data-driven bandwidth selection procedure in Section 5 leads to the optimal bandwidths $b_1 = 5$ months and $b_2 = 8$ months for delay and accident date, respectively. For the estimation of φ, we minimize the loss functions (11) in the interval $[0.5, 1.5]$ for every $x = 1, \ldots, 120$ in every iteration of the algorithm.

The estimated baseline hazard and operational time are shown in Figure 3. For the operational time estimate $\hat{\varphi}$ in Figure 3a, the settlement delay at 1 January 2004 is used as benchmark and claim settlement for most accident dates between February 2004 and December 2005 is slightly slower than this benchmark. In November 2004, the operational time estimator catches a trend towards faster settlement of claims despite short declines in 2005 and 2009. This phenomenon is most likely due to the improved use of technology in the insurance company and has also been observed on the same data set in Lee et al. (2017). The decrease of speed in claims finalization at the end of 2005 and 2009 could be due to new employees in the reserving department who are training in their first months. The average accident that happened after January 2006 was settled faster than our benchmark with the value of the operational time estimate $\hat{\varphi}$ being below 1 for this period. After 2010, our model shows the fastest processing and payments of claims. Due to high variation in the estimation of φ in the lower corner of the run-off triangle, we recommend to set $\hat{\varphi}$ to the value of the previous month for the last five months (about the last 5% of the support of φ). Note that this adjustment is still in the spirit of our approach to improve in the estimation by chain-ladder (and even multiplicative nonparametric methods as in Martínez-Miranda et al. (2013) and Hiabu et al. (2016)) because a constant operational time value corresponds to the case where T and X are independent and we still allow for dependency through operational time for 95% of the accident dates. We want to remark that this issue does not occur if un-truncated data (on a squared support instead of a triangular one) is given. The baseline hazard estimate $\hat{\alpha}_0(\mathcal{T} - t)$ of the payment delay (in forward time) in Figure 3b has the expected shape with a steep decrease for short delays and a value close to zero for delays larger than 1.5 years. This shape indicates that the vast majority of the claims in this data set were paid off within the first year as can be seen in Figure 1.

The estimated outstanding liabilities by accident year and by payment year are given in Table 1. The results from the chain-ladder method with quarterly aggregated data are used as a benchmark. The shift through operational time yields less claims than chain-ladder for all payment years except for 2016. Since the value of the operational time estimate (Figure 3a) is below the benchmark 1 for all claims with accident year later than 2005, these claims were settled faster than older claims. These claims constitute the majority of outstanding claims since most claims are estimated to be settled within one and a half years (Figure 3b). Hence, most claims are expected to be paid out earlier than estimated through average payment delay in the chain-ladder method. The same effect can be seen with respect to accident years. On 31 December 2013, the date of data collection, our operational time estimator forecasts old claims from accidents before 2009 to be paid off since their settlement delay is expected to be shorter than average settlement. On the other hand, chain-ladder still estimates a few claims from accidents between 2005 and 2008. In total, for this data set, the estimated number of outstanding payments by operational time is lower (1054) than the reserve estimate by quarterly chain-ladder (1414).

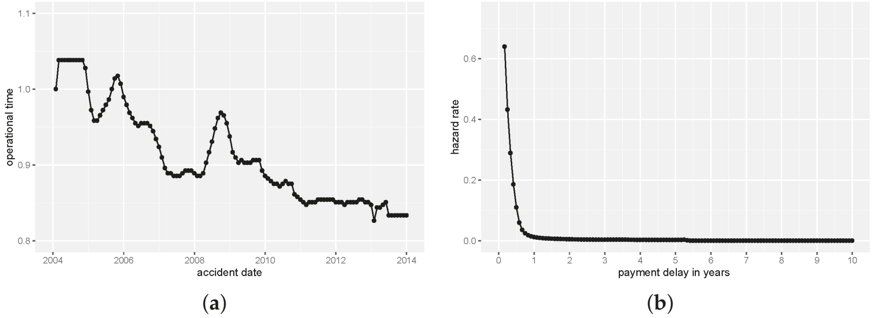


Figure 3. Estimated components of hazard rate of the payment delay T: (**a**) operational time estimate $\hat{\varphi}(t)$ with optimal bandwidths; (**b**) baseline hazard estimate $\hat{\alpha}_0(\mathcal{T} - t)$ of payment delay (in forward time) with optimal bandwidths.

Table 1. Estimated number of outstanding claims through hazard with operational time (op. time) and quarterly chain-ladder (CL) by accident year and payment year.

Accident Year	**2004**	**2005**	**2006**	**2007**	**2008**	**2009**	**2010**	**2011**	**2012**	**2013**	**Total**
Op. Time	0	0	0	0	0	23	92	171	254	513	1054
CL	0	2	8	20	32	54	75	128	224	871	1414
Payment Year	**2014**	**2015**	**2016**	**2017**	**2018**	**2019**	**2020**	**2021**	**2022**	**2023**	**Total**
Op. Time	590	256	143	58	5	0	0	0	0	0	1054
CL	856	261	130	71	45	27	16	7	2	0	1414

Although the comparison might seem unfair at first due to different levels of aggregation, more granular aggregation for chain-ladder would not improve the quality of its estimates. As shown in a simulation study in Baudry and Robert (2019), even when enough data are available for monthly aggregation, chain-ladder reserve estimates based on monthly data show very high variance, making them effectively unreliable in practice; however, monthly data are necessary for chain-ladder if one is interested, for instance, in the estimation of monthly cash-flows. This phenomenon has been confirmed in a simulation in Bischofberger et al. (2019), in which kernel estimators picked larger bandwidths while still being able to yield monthly cash-flow predictions. Furthermore, chain-ladder is typically used on at least quarterly aggregated data to prevent columns that contain only zeros in the run-off triangle. Where the chain-ladder algorithm cannot handle this issue, our operational time hazard estimator can cope with it.

In an independence test based on Conditional Kendall's tau for truncated data (Austin and Betensky 2014; Martin and Betensky 2005), the hypothesis of independent settlement delay T and accident date X was rejected. Hence, the assumptions of the chain-ladder model of Mack (1993) are violated (Hiabu 2017) and one cannot rely on its estimate in this data set. Since the chain-ladder model with independent variables is nested within our prosed operational time Model (1), we recommend our model—although inference for our operational time structure has not been carried out. With the hazard Model (3) being rather involved, the theory for a hypothesis test for the operational time structure is beyond the scope of this article.

Choices of bandwidths with higher validation scores can lead to unrealistic reserve estimates that differ from the chain-ladder estimate by up to 100%. On the one hand, the operational time hazard estimator is sensitive to the choice of bandwidth. On the other hand, the result obtained through cross-validation is stable with four bandwidth choices close to the optimal validation score $\hat{Q}$ resulting in very similar estimates of the number of outstanding claims.

7. Conclusions

We introduced a new hazard model that allows for operational time in right-truncated data as present in run-off triangles. In a structured hazard model, the conditional hazard rate of the settlement delay given the accident date is expressed through operational time (a function of the accident date) and the baseline hazard of the settlement delay (cleared of effects from accident date). Minimizing an integrated squared loss, we define nonparametric estimators of operational time and the baseline hazard. These estimators are calculated through an iterative algorithm that updates the estimates of operational time and the baseline hazard in each iteration until it converges. If no right-truncation is present, our hazard model is a nonparametric extension of the accelerated failure model with a one-dimensional covariate.

Our estimation procedure detects operational time in the data and corrects for it in the estimation process. Therefore, it can be classified as an unsupervised machine learning technique. Since operational time is a common source of dependence between accident date and settlement date in the data, we recommend the approach illustrated here if one cannot prove independent covariates in the date through hypothesis testing (and other structural dependencies like seasonal effects can be ruled out). Even if the accident date and settlement are independent, our estimator works and estimates operational time $\varphi \approx 1$. However, in the latter case or if independence is not rejected by a statistical test, estimation via chain-ladder tends to be more stable than our operational time hazard estimates and should be considered.

In an application in a real data set of paid claims, we forecast the number of outstanding claims for a motor insurance business line. For this purpose, we suggested to transform our operational time and baseline hazard estimators into time-dependent development factors. These are then used to extrapolate the claim numbers in the data set analogously to what is done in the chain-ladder method.

The downsides of the approach illustrated here are computational complexity and numerical instability of the operational time estimator on the data in the last 5–10% of accident dates, i.e., in the lower corner of the run-off triangle. The latter issue also arises in many other approaches to non-life claims reserving. Our suggested way to deal with it in our model is to set the value of operational time to the last stable value for the affected dates, which corresponds to the assumption of independent accident date and settlement delay on the most recent accident dates. Therefore, our approach still corrects for operational time on more than 90–95% of the data and in the remaining data it is as good as kernel hazard methods that assume independent variables.

Funding: This work was supported by the Deutsche Forschungsgemeinschaft (DFG) through the Research Training Group RTG 1953.

Acknowledgments: The author would like to thank Munir Hiabu and Enno Mammen for valuable advice.

Conflicts of Interest: The author declares no conflict of interest.

Appendix A. Alternative Local Constant Estimators

As an alternative to the local linear estimator of $\alpha(t,x)$ in Equation (7), one could use the local constant estimator

$$\hat{\alpha}^{LC}(t,x) = \frac{\widehat{O}^{LC}(t,x)}{\widehat{E}^{LC}(t,x)},$$

with

$$\widehat{O}^{LC}(u_1,u_2) = \sum_{i=1}^{n} \int_0^{\mathcal{T}} K_b(u - U_i(s))\mathrm{d}N_i(s),$$
$$\widehat{E}^{LC}(u_1,u_2) = \sum_{i=1}^{n} \int_0^{\mathcal{T}} K_b(u - U_i(s))Z_i(s)\mathrm{d}s.$$

It is defined through the integrated squared loss minimization

$$\arg\min_{\theta\in\mathbb{R}} \sum_{i=1}^{n} \int_0^{\mathcal{T}} \left[\left(\frac{1}{\varepsilon} \int_s^{s+\varepsilon} \mathrm{d}N_i(v) - \theta \right)^2 - \xi(\varepsilon) \right] K_b(u - U_i(s))Z_i(s)\mathrm{d}s,$$

for $u = (t,x)$ and $U_i(t) = (t, X_i(t))$ as before. The term $\xi(\varepsilon) = \left(\varepsilon^{-1} \int_s^{s+\varepsilon} \mathrm{d}N^i(s)\right)^2$ is again needed to make the expression well-defined.

References

Aalen, Odd O. 1980. A model for nonparametric regression analysis of counting processes. In *Mathematical Statistics and Probability Theory. Lecture Notes in Statistics*. Edited by Witold Klonecki, Andrzej Kozek and Jan Rosiński. New York: Springer, vol. 2, pp. 1–25.

Andersen, Per K., Ørnulf Borgan, Richard D. Gill, and Niels Keiding. 1993. *Statistical Models Based on Counting Processes*. New York: Springer.

Antonio, Katrien, and Richard Plat. 2014. Micro-level stochastic loss reserving for general insurance. *Scandinavian Actuarial Journal* 2014: 649–69. [CrossRef]

Austin, Matthew D., and Rebecca A. Betensky. 2014. Eliminating bias due to censoring in kendall's tau estimators for quasi-independence of truncation and failure. *Computational Statistics & Data Analysis* 73: 16–26.

Avanzi, Benjamin, Bernard Wong, and Xinda Yang. 2016. A micro-level claim count model with overdispersion and reporting delays. *Insurance: Mathematics and Economics* 71: 1–14. [CrossRef]

Badescu, Andrei L., X. Sheldon Lin, and Dameng Tang. 2016. A marked Cox model for the number of IBNR claims: Theory. *Insurance: Mathematics and Economics* 69: 29–37. [CrossRef]

Baudry, Maximilien, and Christian Y. Robert. 2019. A machine learning approach for individual claims reserving in insurance. *Applied Stochastic Models in Business and Industry* 35: 1127–55. [CrossRef]

Berkson, Joseph 1980. Minimum chi-square, not maximum likelihood! *The Annals of Statistics* 8: 457–87. [CrossRef]

Bischofberger, Stephan M., Munir Hiabu, and Alex Isakson. 2019. Continuous chain-ladder with paid data. *Scandinavian Actuarial Journal*. [CrossRef]

Buckley, Jonathan, and Ian James. 1979. Linear regression with censored data. *Biometrika* 66: 429–36. [CrossRef]

Bühlmann, Hans 1970. *Mathematical Methods in Risk Theory*. Berlin: Springer.

Cho, Youngjoo, Chen Hu, and Debashis Ghosh. 2018. Covariate adjustment using propensity scores for dependent censoring problems in the accelerated failure time model. *Statistics in Medicine* 37: 390–404. [CrossRef]

Cox, David R. 1972. Regression models and life tables. *Journal of the Royal Statistical Society: Series B* 34: 187–220.

Cox, David R., and David Oakes. 1984. *Analysis of Survival Data*, 1st ed. Boca Raton: Chapman & Hall/CRC.

Crevecoeur, Jonas, Katrien Antonio, and Roel Verbelen. 2019. Modeling the number of hidden events subject to observation delay. *European Journal of Operational Research* 277: 930–44. [CrossRef]

England, Peter D., and Richard J. Verrall. 2002. Stochastic claims reserving in general insurance. *British Actuarial Journal* 8: 443–544. [CrossRef]

Feller, William. 1971. *An Introduction to Probability Theory and Its Applications*. New York: John Wiley & Sons, vol. 2.

Fulcher, Isabel R., Eric Tchetgen Tchetgen, and Paige L. Williams. 2017. Mediation analysis for censored survival data under an accelerated failure time model. *Epidemiology* 28: 660–66. [CrossRef] [PubMed]

Gabrielli, Andrea, Ronald Richman, and Mario V. Wüthrich. 2019. Neural network embedding of the over-dispersed Poisson reserving model. *Scandinavian Actuarial Journal* [CrossRef]

Gámiz, María Luz, Lena Janys, María Dolores Martínez-Miranda, and Jens Perch Nielsen. 2013. Bandwidth selection in marker dependent kernel hazard estimation. *Computational Statistics & Data Analysis* 68: 155–69.

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2008. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. New York: Springer.

Hiabu, Munir. 2017. On the relationship between classical chain ladder and granular reserving. *Scandinavian Actuarial Journal* 2017: 708–29. [CrossRef]

Hiabu, Munir, Enno Mammen, María Dolores Martínez-Miranda, and Jens Perch Nielsen. 2016. In-sample forecasting with local linear survival densities. *Biometrika* 103: 843–59. [CrossRef]

Huang, Jinlong, Chunjuan Qiu, Xianyi Wu, and Xian Zhou. 2015. An individual loss reserving model with independent reporting and settlement. *Insurance: Mathematics and Economics* 64: 232–45. [CrossRef]

Jewell, William S. 1989. Predicting IBNYR events and delays I. Continuous time. *ASTIN Bulletin* 19: 25–55. [CrossRef]

Jewell, William S. 1990. Predicting IBNYR events and delays II. Discrete time. *ASTIN Bulletin* 20: 93–111. [CrossRef]

Kalbfleisch, John D., and Ross L. Prentice. 2002. *The Statistical Analysis of Failure Time Data*, 2nd ed. Wiley Series in Probability and Statistics. Hoboken: John Wiley & Sons.

Kremer, Erhard. 1982. IBNR-claims and the two-way model of ANOVA. *Scandinavian Actuarial Journal* 1982: 47–55. [CrossRef]

Kuang, Di, Bent Nielsen, and Jens Perch Nielsen. 2009. Chain-ladder as maximum likelihood revisited. *Annals of Actuarial Science* 4: 105–21. [CrossRef]

Kuo, Kevin. 2019. Deeptriangle: A deep learning approach to loss reserving. *Risks* 7: 97. [CrossRef]

Larsen, Christian Roholte. 2007. An individual claims reserving model. *ASTIN Bulletin* 37: 113–32. [CrossRef]

Lee, Young K., Enno Mammen, Jens Perch Nielsen, and Byeong U. Park. 2015. Asymptotics for in-sample density forecasting. *The Annals of Statistics* 43: 620–51. [CrossRef]

Lee, Young K., Enno Mammen, Jens Perch Nielsen, and Byeong U. Park. 2017. Operational time and in-sample density forecasting. *The Annals of Statistics* 45: 1312–41. [CrossRef]

Li, Jialiang, and Baisuo Jin. 2018. Multi-threshold accelerated failure time model. *The Annals of Statistics* 46: 2657–82. [CrossRef]

Linton, Oliver B., Enno Mammen, Jens Perch Nielsen, and Ingrid Van Keilegom. 2011. Nonparametric regression with filtered data. *Bernoulli* 17: 60–87. [CrossRef]

Louis, Thomas A. 1981. Nonparametric analysis of an accelerated failure time model. *Biometrika* 68: 381–90. [CrossRef]

Mack, Thomas. 1993. Distribution-free calculation of the standard error of chain ladder reserve estimates. *ASTIN Bulletin* 23: 213–25. [CrossRef]

Mammen, Enno, María Dolores Martínez-Miranda, and Jens Perch Nielsen. 2015. In-sample forecasting applied to reserving and mesothelioma. *Insurance: Mathematics and Economics* 61: 76–86. [CrossRef]

Martin, Emily C., and Rebecca A. Betensky. 2005. Testing quasi-independence of failure and truncation times via conditional Kendall's tau. *Journal of the American Statistical Association* 100: 484–92. [CrossRef]

Martínez-Miranda, María Dolores, Jens Perch Nielsen, Stefan Sperlich, and Richard J. Verrall. 2013. Continuous chain ladder: Reformulating and generalising a classical insurance problem. *Expert Systems with Applications* 40: 5588–603. [CrossRef]

Miller, Rupert G. 1976. Least squares regression with censored data. *Biometrika* 63: 449–64. [CrossRef]

Nielsen, Jens Perch. 1998. Marker dependent kernel hazard estimation from local linear estimation. *Scandinavian Actuarial Journal* 1998: 113–24. [CrossRef]

Nielsen, Jens Perch, and Oliver B. Linton. 1995. Kernel estimation in a non-parametric marker dependent hazard model. *The Annals of Statistics* 23: 1735–48. [CrossRef]

Nielsen, Jens Perch, and Carsten Tanggaard. 2001. Boundary and bias correction in kernel hazard estimation. *Scandinavian Journal of Statistics* 28: 675–98. [CrossRef]

Norberg, Ragnar. 1993. Prediction of outstanding liabilities in non-life insurance. *ASTIN Bulletin* 23: 95–115. [CrossRef]

Norberg, Ragnar. 1999. Prediction of outstanding liabilities II. Model variations and extensions. *ASTIN Bulletin* 29: 5–25. [CrossRef]

Reid, D. H. 1978. Claim reserves in general insurance. *Journal of the Institute of Actuaries* 105: 211–315. [CrossRef]

Renshaw, Arthur E., and Richard J. Verrall. 1998. A stochastic model underlying the chain-ladder technique. *British Actuarial Journal* 4: 903–23. [CrossRef]

Ritov, Ya'acov, and Jon A. Wellner. 1988. Censoring, martingales, and the cox model. *Contemporary Mathematics* 80: 191–219.

Swishchuk, Anatoliy. 2016. *Change of Time Methods in Quantitative Finance*. New York: Springer.

Taylor, Greg. 2019. Loss reserving models: Granular and machine learning forms. *Risks* 7: 82. [CrossRef]

Taylor, Greg, and Gráinne McGuire. 2016. *Stochastic Loss Reserving Using Generalized Linear Models*. Arlington: Casualty Actuarial Society. CAS Monograph Series, Number 3.

Taylor, Greg, Gráinne McGuire, and James Sullivan. 2008. Individual claim loss reserving conditioned by case estimates. *Annals of Actuarial Science* 3: 215–56. [CrossRef]

Taylor, Greg. 1981. Speed of finalization of claims and claims runoff analysis. *ASTIN Bulletin* 12: 81–100. [CrossRef]

Taylor, Greg. 1982. Zehnwirth's comments on the see-saw method: A reply. *Insurance: Mathematics and Economics* 1: 105–108. [CrossRef]

Verrall, Richard J. 1991. Chain ladder and maximum likelihood. *Journal of the Institute of Actuaries* 118: 489–99. [CrossRef]

Ware, James H., and David L. DeMets. 1976. Reanalysis of some baboon descent data. *Biometrics* 32: 459–63. [CrossRef]

Wüthrich, Mario V. 2018. Machine learning in individual claims reserving. *Scandinavian Actuarial Journal* 2018: 465–80. [CrossRef]

Zhao, Xiao Bing, and Xian Zhou. 2010. Applying copula models to individual claim loss reserving methods. *Insurance: Mathematics and Economics* 46: 290–99. [CrossRef]

Zhao, Xiao Bing, Xian Zhou, and Jing Long Wang. 2009. Semiparametric model for prediction of individual claim loss reserving. *Insurance: Mathematics and Economics* 45: 1–8. [CrossRef]

© 2020 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).

risks

MDPI

Article

Modelling Unobserved Heterogeneity in Claim Counts Using Finite Mixture Models

Lluís Bermúdez [1,*], Dimitris Karlis [2] and Isabel Morillo [1]

[1] Departament de Matemàtica Econòmica, Financera i Actuarial, Universitat de Barcelona, Diagonal 690, 08034 Barcelona, Spain; imorillo@ub.edu

[2] Department of Statistics, Athens University of Economics and Business, 10434 Athens, Greece; karlis@aueb.gr

* Correspondence: lbermudez@ub.edu; Tel.: +34-93-403-4853; Fax: +34-93-403-4892

Received: 20 December 2019; Accepted: 22 January 2020; Published: 29 January 2020

Abstract: When modelling insurance claim count data, the actuary often observes overdispersion and an excess of zeros that may be caused by unobserved heterogeneity. A common approach to accounting for overdispersion is to consider models with some overdispersed distribution as opposed to Poisson models. Zero-inflated, hurdle and compound frequency models are typically applied to insurance data to account for such a feature of the data. However, a natural way to deal with unobserved heterogeneity is to consider mixtures of a simpler models. In this paper, we consider k-finite mixtures of some typical regression models. This approach has interesting features: first, it allows for overdispersion and the zero-inflated model represents a special case, and second, it allows for an elegant interpretation based on the typical clustering application of finite mixture models. k-finite mixture models are applied to a car insurance claim dataset in order to analyse whether the problem of unobserved heterogeneity requires a richer structure for risk classification. Our results show that the data consist of two subpopulations for which the regression structure is different.

Keywords: zero-inflation; overdispersion; automobile insurance; risk classification; risk selection

JEL Classification: C51

1. Introduction and Aims

In insurance datasets, for the purposes of modelling claim counts, there is a problem of unobserved heterogeneity caused by differences in driving habits and behaviour among policyholders that cannot be observed or measured by the actuary (for example, driving ability, driving aggressiveness or the degree of obeying traffic regulations). This often leads to overdispersion and a relatively large number of zeros, which cannot be fully remedied by Poisson regression models. Many attempts have been made in the actuarial literature to account for such features of the data (for example, compound frequency models, also known as mixture models, and their zero-inflated or hurdle versions). This paper aims to explore whether the problem of unobserved heterogeneity requires a richer structure, though the use of finite mixtures of regression models, than the previous models have.

In a competitive market, insurance companies need to use a pricing structure that ensures that the exact weight of each risk is fairly distributed within the portfolio. If an insurance company does not achieve at least the same success with respect to this goal as their competitors, the policyholders with lower risk will be tempted to move to another company that offers better rates for them. Such an adverse selection process would lead the less unsuccessful company to lose its financial equilibrium, with insufficient income from premiums to pay for the claims reported by the remaining policyholders with higher risk.

In most developed countries, the car insurance market is a highly competitive market. Therefore, to avoid such an adverse selection process, a particularly complex pricing structure is designed by actuaries. A thorough review of the modelling of claim counts for car insurance can be found in (Denuit et al. 2007). In general terms, to handle this problem, the actuary segments the portfolio into homogeneous classes so that all the insured parties belonging to a particular class pay the same premium. This procedure is referred to as risk classification, tariff segmentation or a priori ratemaking.

In short, the classification or segmentation of risks involves establishing different classes of risk according to the nature of claims and probability of their occurrence. To this end, factors are determined to classify each risk, and its influence on the observed number of claims is estimated. To achieve this, risk analysis based on generalized linear models (GLMs) is widely accepted. Focusing on claim frequency, a regression component is included in the claim count distribution to take individual characteristics into account.

A very common GLM used for these purposes is the Poisson regression model and its generalisations. Introduced by Dionne and Vanasse (1989) in the context of car insurance, the model can be applied if a series of classification variables, referred to as a priori variables, plus the number of claims for each individual policy are known. However, the Poisson regression model is usually rejected because of the presence of overdispersion and an excess of zeros. This rejection may be interpreted as a sign that the portfolio is still heterogeneous: not all factors influencing risk can be identified, measured and introduced into the a priori modelling. This phenomena is known as the problem of unobserved heterogeneity.

In parallel, another way to account for unobserved heterogeneity is to consider that the claims record for each insured party reveals the differences in driving habits and behaviour among policyholders that cannot be observed or measured via the a priori variables. Therefore, the idea of considering individual differences in policies within the same a priori class by using an a posteriori mechanism has emerged, i.e., tailoring an individual premium based on the claims record for each insured party. This concept has received the name of a posteriori ratemaking, experience rating or the bonus-malus system (see Denuit et al. (2007)).

One way to deal with overdispersion is to consider compound frequency models (mixture models) with some overdispersed distribution. This is best achieved by moving from the simple Poisson model to the negative binomial model (Dionne and Vanasse (1992)) or to the Poisson-inverse Gaussian model (Dean et al. (1989)). To account for the excess of zeros, some generalizations of the Poisson model have been considered. Lambert (1992) introduced the zero-inflated Poisson regression model and, since then, there has been a considerable increase in the number of applications of zero-inflated regression models based on several different distributions. A comprehensive discussion of these applications can be found in Winkelmann (2008). Similarly, hurdle models are also widely applied to insurance claim count data. A common assumption in all these models is that all policyholders behave in the same way with regard to a priori variables, and thus they all have the same regression structure.

In this paper, we examine whether this assumption is realistic. The models proposed in this paper account for unobserved heterogeneity by choosing a finite number of subpopulations. To account for overdispersion and an excess of zeros, we consider a k-finite mixture of Poisson and negative binomial regression models. As Park and Lord (2009) show for vehicle crash data analysis, a finite mixture of Poisson or negative binomial regression models is especially useful where count data are drawn from heterogeneous populations. For modelling claim counts, the idea behind this is that the data consist of subpopulations of policyholders, “caused" by the unobserved heterogeneity, for which the regression structure, used to account for the observed or a priori variables, is different. These models allow each component in the discrete mixture to have its own score, i.e., for there to be different behaviour for each group of policyholders, whereas classical claim frequency models use a single score.

To sum up, this paper aims to explore whether resolving the problem of unobserved heterogeneity requires a richer structure than that which is present in typical compound frequency models and their zero-inflated or hurdle versions. By applying finite mixtures of regression models, we will examine whether unobserved risk factors that are not considered in the a priori tariff, such as a driver's reflexes, aggressiveness, or knowledge of the Highway Code, establish the existence of subpopulations of policyholders with different a priori behaviour. To achieve this goal, the proposed models are fitted to a set of car insurance claims data to compare their goodness of fit with the traditional claim frequency models and to assess if we need to account for this extra heterogeneity. Finally, we discuss whether the proposed models help to search for better alternatives to account for unobserved heterogeneity.

In the next section, the models and computational details used are defined. In Section 3, we summarize the database obtained from a Spanish insurance company and the results from fitting the models to it. Finally, we offer some concluding remarks in Section 4.

2. Finite Mixture of Regression Models

The central idea for a finite mixture of regression models is that we assume that the entire population can be split into k subpopulations (also called clusters, components or segments). Assuming a discrete-valued response y_i for the i-th individual, we then assume that

$$P(y_i) = P(Y_i = y_i) = \sum_{j=1}^{k} \pi_j P(y_i|\theta_{ij}), \quad \theta_{ij} > 0, \;\; y_i = 0, 1, \ldots,$$

where $0 < \pi_j < 1$ with $\sum_{j=1}^{k} \pi_j = 1$ are the mixing proportions indicating the probability that a randomly selected observation belongs to the j-th subpopulation and $P(y|\theta)$ is some discrete distribution indexed by some parameter vector θ. In our case presented below, $P(y|\cdot)$ will be assumed to belong to one of the Poisson or negative binomial families. Note that we assume that for each individual we have a set of parameters θ_{ij} that depend on each component and they may depend on some covariate information for the i-th individual.

We further assume that the mean of the j-th component can be modelled by a vector of covariates containing information on the i-th individual, denoted by $\mathbf{x}_i$. In the general setting, this covariate vector that characterises the i-th individual can be different for different components, and therefore we should use also a subscript j. As, in our model, we use the same covariates for all components, we drop this second index. Assuming, without loss of generality, that $\theta = (\mu, \phi)$, where μ is the mean of the distribution (this can be easily obtained with a reparameterisation) and ϕ some parameter related to overdispersion (set equal to 1 for the Poisson distribution), we further assume that:

$$\log \mu_{ij} = \mathbf{x}_i' \beta_j$$

where now β_j is a component-specific vector of coefficients.

Note that the above formulation can be seen in the context of a GLM. However, we prefer to describe the model in a more general setting since some of the families we may use instead of Poisson and negative binomial models do not belong to the exponential family or therefore to the general GLM setting.

The above generic formulation can be expanded by allowing additional covariates to the rest of the parameters for each component, as well as to the vector of mixing proportions. A well-known model of this type is the finite mixture of Poisson regressions in Wang et al. (1996) (see also Grun and Leisch 2007, 2008). Finite mixtures of regression models have been widely used in different settings, see Hennig (2000) for a thorough discussion.

This type of modelling has some interesting features: first, the zero-inflated model is a special case; second, it allows for overdispersion; and third, it allows for a neat interpretation based on the typical clustering application of finite mixture models.

It is useful to show that if we denote by μ_j and σ_j^2 the mean and the variance of the j-th component, then the mean μ and the variance σ^2 of the mixture are given by

$$\mu = \sum_{j=1}^{k} \pi_j \mu_j, \text{ and } \sigma^2 = \sum_{j=1}^{k} \pi_j (\mu_j^2 + \sigma_j^2) - \mu^2 \tag{1}$$

These formulas will be useful later on for our calculations.

2.1. Finite Mixture of Poisson Regressions

The case of a finite mixture of Poisson regressions is by far the best known and most commonly applied in practice. It dates back to Wang et al. (1996) and assumes that

$$P(y_i|\mu_{ij}) = \sum_{j=1}^{k} \pi_j \frac{\exp(-\mu_{ij})\mu_{ij}^{y_i}}{y_i!}, \quad y_i = 0, 1, \ldots$$

with $\mu_{ij} = \exp(\mathbf{x_i}'\beta_j)$. The zero-inflated Poisson regression is a special case. The model allows for overdispersion with respect to the simple Poisson regression model. For more details see Grun and Leisch (2007); Wang et al. (1996).

2.2. Finite Mixture of Negative Binomial Regressions

For the negative binomial model, we assume

$$P(y_i|\mu_{ij}, \phi_j) \quad = \quad \frac{\Gamma(\phi_j + y_i)}{\Gamma(\phi_j) y_i!} \left(\frac{\mu_{ij}}{\phi_j + \mu_{ij}}\right)^{y_i} \left(\frac{\phi_j}{\phi_j + \mu_{ij}}\right)^{\phi_j}, \quad \phi_j > 0, \ \ y_i = 0, 1, \ldots$$

and $\mu_{ij} = \exp(\mathbf{x_i}'\beta_j)$, i.e., the probability function of a negative binomial with mean μ_{ij} and variance $\mu_{ij} + \frac{\mu_{ij}^2}{\phi_j}$.

Note that we assume a separate overdispersion parameter ϕ_j for each component. Such a model has been fitted by Byung-Jung et al. (2014) and Zou et al. (2013). With respect to the finite mixture of Poisson regressions, the model has an extra overdispersion parameter and therefore allows for more flexible distributions in terms of components.

It is evident that the negative binomial model also contains the simple Poisson model as a special case ($\phi_j \to \infty$).

2.3. Other Models

Although in this paper we focus on the two families of models introduced above, there are other models that fit into this context for which we do not present results. They relate to Poisson-inverse Gaussian regression models (Dean et al. (1989)) and finite mixtures of them; some nonparametric random effects Poisson regression models (see Aitkin (1999)), i.e., the model assumes some random effect on the intercept of the Poisson regression and thus actually fits a finite mixture of Poisson regression model where the estimated coefficients (apart from the intercept) are the same for all components; and hurdle-type models (a hurdle model is a modified count model in which the two processes generating the zeros and the positives are not constrained to be the same, see Mullahy (1986)). As mentioned above, we do not formulate zero-inflated models as we treat them as special cases of finite mixture models.

2.4. Estimation via EM Algorithm

Under the umbrella of a finite mixture, estimation for this particular family of models is rather simple. We follow the standard approach of combining the observed data $(Y_i, \mathbf{X}_i)$ with unobserved latent vectors $\mathbf{Z}_i = (Z_{i1}, \ldots, Z_{ik})$ with $Z_{ij} = 1$ if the i-th observation belongs to the j-th component, and 0 otherwise. As is typical, the EM algorithm consists of estimating the Z's by their conditional expectation and then fitting a standard regression model to the response, Y, using a weighted likelihood, based on the weights derived during the E-step. A formal description of a generic algorithm is given in what follows.

E-step: Using the current estimates, $\hat{\pi}_j$ and $\hat{\theta}_{ij}$, $i = 1, \ldots, n$ and $j = 1, \ldots, k$, calculate

$$w_{ij} = E(Z_{ij}) = \frac{\hat{\pi}_j P(y_i|\hat{\theta}_{ij})}{\sum\limits_{j=1}^{k} \hat{\pi}_j P(y_i|\hat{\theta}_{ij})}, \quad i = 1, \ldots, n, \quad j = 1, \ldots, k \tag{2}$$

and then

M-step:

M1 Update the mixing proportions using

$$\hat{\pi}_j = \frac{\sum\limits_{i=1}^{n} w_{ij}}{n}, \quad j = 1, \ldots, k$$

M2 Update the regression coefficients and the component-specific parameters by fitting a single regression model for the j-th component with response y_i, covariates $\mathbf{x}_i$ using a weighted likelihood approach with weights w_{ij}.

It is clear that the M-step is not in a closed form. Also, note that actually we fit k models with the same data but different weights. This can be run in parallel to speed up the process. All the pros and cons of the EM algorithm for finite mixtures apply. Also, standard procedures for finite mixtures are applicable, such as for example model selection. We will discuss some computational details later.

Finally, we need to emphasize the issue of identifiability. Conditions for identifiability for such finite mixtures of regression models are given in Hennig (2000). For such a finite mixture of regression models for count data, problems may occur if the covariates are categorical and they can have a small number of different combinations. In our case, we have seven binary variables leading to 2^7 combinations. Not all of them appear in the data but we still have quite a large number of distinct combinations for the model matrix. In general, it is hard to show that identifiability exists since the conditions are hard to evaluate. We believe that in our case no particular problem exists. From a practical point of view we have worked with several initial values to examine whether we became trapped with different solutions. This did not happen, adding to our belief that our model is identifiable.

2.5. Computational Details

An important aspect for the successful application of the EM algorithm is that appropriate initial values need to be selected, as otherwise one may be trapped in local rather than global maxima. We selected our initial values as follows.

We started by fitting a simple Poisson regression model. This also gave sufficient initial values for the simple negative binomial regression. Initial values for the overdispersion parameter in these two models were set equal to the observed overdispersion (as proposed in Breslow (1984)).

From here on we describe the approach for each model. Therefore, when we refer to "model", we imply either the mixture of Poisson models or the mixture of negative binomial models. Initial values for $k = 2$ were selected by perturbing the simple $(k = 1)$ regression model. Specifically, we fitted a single regression and keeping the fitted values, we split them into two components with mixing probabilities of 0.5 each, and means equal to 1.2 and 0.8 of the fitted values. Then, to fit a model with $k + 1$ components, we used the solution with k components and a new component at the centre (that of a single one-component regression), with mixing probability 0.05. The other mixing probabilities were rescaled to sum to 1. Extensive simulation has shown that this approach works well to locate the maximum. Other approaches can be found in Papastamoulis et al. (2016).

All our computations were made in `R`. We used our own code, while some of the models can be fitted using the `gamlss`, `VGAM` and `flexmix` packages in `R`. However, we found some convergence problems and less flexibility while using the standard packages.

Convergence was detected when the relative change between two successive iterations was smaller than 10^{-8}. For fitting the separate regression models we used the standard GLM approach (IRLS algorithm) for Poisson and negative binomial regression.

3. Data and Results

3.1. Data Description

The original database is a random sample of the car portfolio of a major insurance company operating in Spain in 1996. Only cars categorized as being for private use were considered. The data contain information from 80,994 policyholders. Seven exogenous variables plus the annual number of accidents recorded were considered here. For each policy, the information at the beginning of the period and the total number of claims from policyholders were reported. The definition and some descriptive statistics of the variables are presented in Table 1. This dataset has previously been used in Pinquet et al. (2001), Brouhns et al. (2003), Bolancé et al. (2003, 2008), Boucher et al. (2007, 2009), Boucher and Denuit (2008), Bermúdez (2009) and Bermúdez and Karlis (2012).

The meaning of the variables that refer to the Spanish market should also be clarified. The variable *ZON* distinguishes between driving zones of greatest risk (Madrid, Catalonia and central northern Spain) and the rest. Regarding the type of coverage provided by the of policies (variable *COV*), the classification adopted here responds to the most common types of car insurance policy available on the Spanish market. The simplest policy only includes third-party liability. This simplest type of policy makes up the baseline group, while variable *COV* equals 1 denotes policies which, apart from the guarantees contained in the simplest policies, also include comprehensive and collision coverage.

Table 1. Dependent and explanatory variables used in the models.

Variable	Definition	Mean	St. dev.
N	total number of claims reported by policyholders (0: 71,087; 1: 6,744; 2: 2,067; 3: 690; 4: 248; 5: 95; 6: 34; >6: 29)	0.1833	0.5873
GEN	equals 1 for women and 0 for men	0.1600	0.3666
URB	equals 1 when driving in urban area, 0 otherwise	0.6690	0.4706
ZON	equals 1 when driving in Madrid, Catalonia or northern Spain, 0 otherwise	0.4326	0.4954
LIC	equals 1 if the driving license is 4 or more years old, 0 otherwise	0.9766	0.1511
LOY	equals 1 if the client is in the company for more than 5 years, 0 otherwise	0.1441	0.3512
COV	equals 1 if includes comprehensive and collision coverage, 0 otherwise	0.5087	0.4999
POW	equals 1 if horsepower is greater than or equal to 5500cc, 0 otherwise	0.8058	0.3955

3.2. Fitted Models

We fitted models of increasing complexity to this dataset, starting from a simple Poisson regression model. We used AIC and BIC to select the best among a series of candidate models. All models were run in `R`. Table 2 compares the fitted models for Poisson and negative binomial distributions, resulting in the best fit being obtained with a 2-finite mixture of negative binomial regression models (2FMNB). Finite mixture models with $k > 2$ were also fitted, but no improvement in terms of AIC or BIC was achieved. This result gives rise to the conclusion that this portfolio is comprised of two groups of policyholders.

Table 2. Information criteria for selecting the best model for the data.

Model	Log-Likelihood	Parameters	AIC	BIC
Poisson	−42,585.08	8	85,186.15	85,260.57
Negative binomial	−38,453.13	9	76,924.27	77,007.98
Zero-inflated Poisson	−38,836.59	9	77,691.19	77,774.91
Zero-inflated negative binomial	−38,453.13	10	76,926.27	77,019.28
2-Finite Poisson mixture	−38,449.61	17	76,933.21	77,091.36
2-Finite negative binomial mixture	−38,347.81	19	76,733.62	76,910.36

As expected, a large improvement is obtained by moving from a simple Poisson model to a compound frequency model with some overdispersed distributions. The best fit is achieved by the negative binomial model. Zero-inflated models, while providing an improvement on the basic Poisson model, allowing for overdispersion in this case, were not helpful for the negative binomial model. It seems that the problem is not extra zeros but the existence of another group of policyholders. Therefore, assuming that we have two distinct subpopulations, we may move towards a finite mixture model.

In this case, using a 2-finite mixture of regression models, a large improvement was obtained by moving from one component Poisson to a 2-finite mixture of Poisson regression models. Note that this improvement is better than that obtained with the zero-inflated Poisson model. However, as the best fit is obtained by the 2FMNB, it seems that there is still some extra overdispersion which needs to be modelled appropriately, assuming within each component an overdispersed distribution like a negative binomial.

Figure 1 shows boxplots for the fitted mean values per component for both mixture models. We can observe that the group separation is not the same for the two models. Different models can have similar likelihoods but very different properties and potential. Comparing Poisson and negative binomial mixtures we see that they model different aspects and therefore, as they are close in likelihood terms, can focus on separate things. The first component for the Poisson model is more concentrated towards 0. The opposite is true for the second component. Clearly, 2FMNB fits the data better than the 2-finite mixture of Poisson regression models.

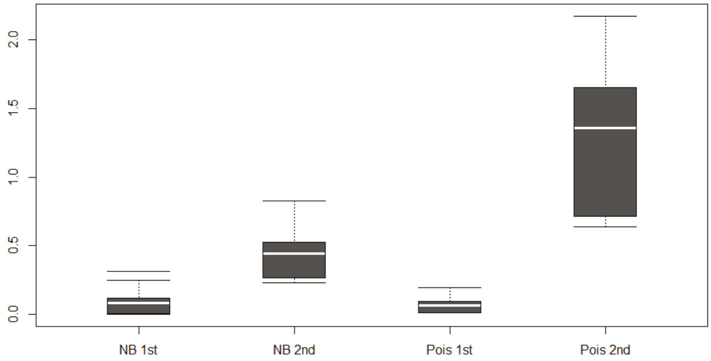


Figure 1. Boxplots of the fitted means for each of the two components for both models.

Figure 1 shows the distinct characteristics of the two assumed distributions. For the case of the Poisson distribution, as the variance is determined from the mean, we see that the two components are further away as an attempt to model the excess of variance. Recall that the total variance is in fact the sum of the between variance (how much the components differ) and the within variance (inside each component). In contrast, in the negative binomial case, the two components are closer since the extra overdispersion parameter regulates the variability. This is also a warning that use of the Poisson model can lead to an erroneous inference of the mean for each component.

From Figure 1, we can also observe that the group separation is characterised by a low mean for the first component and a high mean with higher variance for the second. One may assume that this group separation is revealed by driving characteristics, such as driving ability, aggressiveness or degree of obeying traffic regulations, that are the source of the unobserved heterogeneity. In this case, we can consider those policyholders who belong to the first component to be "good" drivers, whereas policyholders in the second component can be considered "bad" drivers.

Table 3 summaries the results for the 2FMNB and the case with $k = 1$, i.e., no mixture. For the 2FMNB, we report the estimated regression coefficients for each component and p-values for testing the hypothesis that the variable is statistically significant. For the simple negative binomial model, we report the coefficient and standard p-values based on the Wald test.

Table 3. The fitted models for both the negative binomial and the 2FMNB. The p-value for the 2FMNB refers to that of LRT when the variables is removed from both components, whereas for the simple negative binomial it refers to the Wald test.

	2FMNB				**Negative Binomial**	
	1st comp.	**2nd comp.**	***p*-Value**		**Estimate**	***p*-Value**
Intercept	−6.1420	−1.1364	<0.0001	Intercept	−2.4144	<0.0001
GEN	0.2633	0.0086	0.0124	GEN	0.0774	0.0103
URB	0.3407	−0.0762	0.0017	URB	0.0165	0.4870
ZON	0.3745	0.0564	<0.0001	ZON	0.1324	<0.0001
LIC	0.2413	−0.2423	0.0448	LIC	−0.1610	0.0230
LOY	0.3707	0.1289	<0.0001	LOY	0.2019	<0.0001
COV	3.1438	0.6373	<0.0001	COV	1.0024	<0.0001
POW	0.2502	0.1148	<0.0001	POW	0.1440	<0.0001
ϕ	0.2321	0.6051		ϕ	0.2527	
π	0.6686	0.3314				

For the 2FMNB model, to assess the significance of the variables, we calculated a Likelihood Ratio Test (LRT) statistic. Note that this, as a variable selection problem, is not standard, as each covariate appears in both components. Also note that even since standard errors can be derived through the Hessian, such a procedure can be very unstable. Also bootstrap based standard errors can be very time-consuming. Therefore, to see the importance of the covariates, we maximised the log-likelihood with and without each variables and we obtained the LRT, compared with a χ^2 distribution with 2 degrees of freedom. Also, note that covariates that perhaps were not significant for a simple model (no mixture) can be significant in the mixture model, as the two components can allow for separate effects, which are lost when combining to one model.

Comparing the models from Table 3, we can see some interesting points. First, coefficient estimates of the negative binomial model are in some way a linear combination of coefficient estimates of each component of the 2FMNB. Second, in the negative binomial model, only *URB* is not significant at a level of 95%, whereas in the 2FMNB all covariates are significant at that level: the *URB* variable that is deemed significant for the mixture is not significant in the simple model. The reason is that they have opposite signs in the mixture, and therefore when we estimate one coefficient for the simple case, the effect is cancelled out: we estimate some average effect which is close to zero and has large variance. This implies that the mixture can more clearly reflect the importance of the variables and the existence of two groups of policyholders that behave in different ways with regard to these a priori covariates. Finally, focusing on the dispersion parameters of the negative binomial distribution for each component, we can conclude that the second component presents larger dispersion than the first.

To summarize, we may assume two groups of policyholders with different regression structures. This is is particularly noticeable for the variable *URB*. For policyholders considered to be "good" drivers, driving in an urban area increases the probability of making a claim; whereas it decreases for "bad" drivers. This is reasonable: "good" drivers who make a claim are more likely to make it when driving in an urban area, and it will probably just be a small claim. In contrast, "bad" drivers are less likely to make a claim in an urban area since with their driving behaviour (more aggressive and ignoring traffic rules) they are more likely to make a claim when driving outside the urban areas.

3.3. Usage of FM Models for Actuarial Purposes

In this section, we aim to show the advantages and limitations of using a finite mixture of regression models with respect to other models, such as compound frequency models and their zero-inflated or hurdle versions.

First, with respect to compound frequency models, 2-finite mixture models account for unobserved heterogeneity more effectively, providing a better fit. 2FMNB separates the policyholders in two groups allowing for better classification and thus providing a better picture for managerial matters. Assuming that this group separation is caused by their driving capabilities or behaviour, we may consider that we have a group of "good" drivers and another of "bad" drivers.

Second, with respect to zero-inflated and hurdle models, the problem of unobserved heterogeneity is addressed in a more parsimonious way, trying to fix two issues at the same time: overdispersion and an excess of zeros. Zero-inflated and hurdle models focus on the excess of zeros and only implicitly correct for overdispersion; while finite mixture models do both explicitly. Also, note that the interpretation offered by finite mixtures is more reasonable: zero-inflation implies that some drivers will never have an accident, whereas finite mixture models say that there is still some small probability that good drivers will have an accident, this sounds more reasonable in practice from the actuarial point of view (see the discussion in Lord et al. (2007) on the usage of zero-inflated models for car accidents).

Third, the regression structure for each component provided by the 2-finite mixture models is very different from the single score given by compound frequency models and their zero-inflated or hurdle versions. This supports the aforementioned idea that the data consist of two subpopulations, "caused" by, or as a result of, the unobserved factors, for which the regression structure, used to account for the observed factors, is different. The 2-finite mixture models produce a wider picture of the portfolio and therefore offer better chances for accurate risk analysis. As mentioned above, the 2-finite mixture models enable us to see the importance of the variables more clearly. Significant variables according to the LRT test used here, with opposite signs in the mixture, may not be significant for the simple models as they only estimate one coefficient and the effect is cancelled out, estimating some average effect which is close to zero.

However, finite mixture models presents a limitation that impedes their effective use for ratemaking purposes. Although the 2-finite mixture models proposed here separate the policyholders into two types of drivers, they do not allow us to know the type of driver a particular new policyholder is. In other words, for a new customer, although we can estimate different premiums for each component, i.e., for "good" drivers and for "bad" drivers, we cannot find out in which category the new driver belongs, unless we have already observed the number of claims they have made, which is useless. This is because the model is a regression-type model and one needs to observe both the response (number of claims) together with the covariates in order to calculate the posterior probability. Also note that the mixing proportions, π_j, do not offer information on this since they refer to a randomly selected client without taking into account their characteristics. One solution might be to move some of the covariates to the mixing proportions. Therefore, for each new driver, we can have an estimate on the component that they belong to and use this to calculate their premium.

To evaluate the usefulness of finite mixture models, the differences between the 2FMNB and its respective regression model with one component (negative binomial) are analysed through the mean (a priori pure premium) and the variance (necessary for a priori loaded premium) of the number of claims per year for some profiles of the insured parties. Five different, yet representative, profiles were selected from the portfolio and classified according to their risk level. The profiles can be seen in Table 4. We selected the profiles so as to have different increasing means. The first can be classified as the best profile since it presents the lowest mean score. The second was chosen from among the profiles considered as good drivers, with a lower mean value than the mean of the portfolio. The third profile was chosen with a mean score lying very close to the mean of the portfolio. Finally, a profile considered as being for a bad driver (with a mean score above the mean of the portfolio) and the worst driver profile were selected.

Table 4. The 5 profiles used for the comparisons.

Profile Name	GEN	URB	ZON	LIC	LOY	COV	POW
Best	0	1	0	1	0	0	0
Good	1	1	0	0	0	0	1
Average	0	0	0	1	0	1	0
Bad	1	1	0	0	0	1	1
Worst	1	1	1	0	1	1	1

Table 5 shows the results for the five profiles for the two models with respect to the mean and the variance. For the finite mixture model, we have used the same mixing proportion ($\pi = (0.6686, 0.3314)$) for all profiles when we calculate the total mean (2FMNB) from the mean for each component (2FMNB-1 and 2FMNB-2). With respect to the mean, one can see that 2FMNB coincides to a great extend with the negative binomial model. However, we observe larger differences between the means for each component. The group of "good" drivers is far below the group of "bad" drivers. From a practical point of view, the means for each component can be considered as a lower bound and upper bound of the negative binomial means.

Meanwhile, the variance for 2FMNB is greater than for the negative binomial model for all the profiles. As we have commented, finite mixture models allow for unobserved heterogeneity more efficiently. In the same way as mentioned above for means, we see major differences between the variances for each component. Thus, "bad" drivers present greater dispersion than "good" drivers.

Table 5. The mean and the variance derived from the simple negative binomial model (NB) and the 2FMNB.

Profile	Mean				Variance			
	NB	2FMNB	2FMNB-1	2FMNB-2	NB	2FMNB	2FMNB-1	2FMNB-2
Best	0.077	0.080	0.004	0.233	0.101	0.183	0.004	0.323
Good	0.113	0.115	0.005	0.336	0.164	0.279	0.005	0.524
Average	0.207	0.200	0.063	0.476	0.378	0.496	0.081	0.852
Bad	0.309	0.289	0.117	0.636	0.688	0.756	0.176	1.306
Worst	0.432	0.419	0.247	0.766	1.170	1.159	0.509	1.735

Following the traditional two-step methodology, finite mixture models may open up the opportunity to evaluate the extent of a posteriori ratemaking. Bonus-malus systems are usually applied to account for the unobserved heterogeneity. In a posteriori ratemaking, actuaries consider the past claims record of each policyholder in order to update their a priori premiums, assuming that the number of claims reported by policyholders reveals unobservable risk characteristics. In this context, the mean for the first component (2FMNB-1) can be seen as the limit of the a posteriori premium with bonuses. In this assumption, we consider the group of “good” drivers as the policyholders that do not report a claim in many years. In contrast, the mean for the second component would be the limit of the a posteriori premiums with maluses.

In summary, on the basis of the 2FMNB outcome, we can conclude that the use, for ratemaking purposes, of a negative binomial model, together with a bonus-malus system to account for the unobserved heterogeneity, has at least two limitations. First, after an a priori premium is obtained with a negative binomial model, we need to take many years with no claims to reach the level of 2FMNB means for the group of “good” drivers. Second, in the mean time, we may fail to account for the effect of the a priori variables because we assume that all drivers, “good” and “bad”, behave in the same way with respect to these a priori variables.

4. Conclusions

In this paper, we propose the use of a 2-finite mixture of Poisson and negative binomial regression models to allow for the overdispersion and the excess of zeros usually detected in a car insurance dataset and commonly explained by the presence of unobserved heterogeneity. Assuming the existence of two types of clients, described separately by each component in the mixture, improves the modelling of the dataset. The idea is that the data consist of two subpopulations for which the regression structures are different.

These models are applied to a car insurance claims dataset in order to analyse whether the problem of unobserved heterogeneity requires richer structure for risk classification compared with the classical models used to allow for such a feature of the data, i.e., compound frequency models and their zero-inflated versions. From this application, we conclude the following.

First, our results show that this portfolio is comprised of two groups of policyholders or drivers. According to their driving habits or behaviour, such as driving ability, aggressiveness and degree of or obeying traffic regulations, the first group, characterised by a very low mean, can be considered the group of policyholders who are “good” drivers. In contrast, the second group, defined by a high mean with higher variance, can be considered as the group of policyholders who are “bad” drivers.

Second, the two groups of policyholders exhibits different regression structures, i.e., they behave in different ways with regard to the a priori factors. This is is highlighted particularly for the variable related to driving in an urban area or not: for policyholders considered “good” drivers, driving in a urban area increases the probability of having a claim, whereas it decreases for “bad” drivers. Furthermore, simpler models, such as a negative binomial model, fail to reflect the importance of the variables, and therefore lead to an inadequate risk classification.

Third, the two groups of policyholders have very different expected claim frequency values. When using the usual two-step ratemaking procedure, to prevent an adverse selection process, and assuming that the number of claims reported by policyholders reveals their unobservable risk characteristics, a bonus-malus system is considered to update the a priori premiums obtained with a compound frequency model. However, in this case, we would need many years without observing claims from a certain policyholder to reach the premium level provided by the 2FMNB for the group of “good” drivers.

To avoid the aforementioned limitations, we highly recommend the use of telematics devices for ratemaking purposes (see Guillén et al. (2019)). Vehicle telematics allows driving habit information to be collected that will dramatically reduce the unobserved heterogeneity caused by driving habits behavioural variation. Combining traditional a priori rating factors with the new information obtained telemetrically would make it unnecessary to use a time-consuming bonus-malus system and, simultaneously, it will lead to a more efficient risk classification. In other words, including this new information in the a priori ratemaking would allow us to differentiate between “good” and “bad” drivers from the beginning, without the need of a posteriori adjustment and taking into account the importance of all the rating factors more clearly.

Finally, although the 2-finite mixture models proposed here separate the policyholders into two types of drivers, they do not allow us to know the type of driver a particular policyholder is. This could be achieved in different ways, i.e. taking into account the past claim record of each individual or introducing covariates into the mixing probabilities of the mixtures. This may be the goal for future research.

Author Contributions: Conceptualisation, L.B., D.K. and I.M.; methodology, L.B and D.K.; software, D.K.; validation, L.B., D.K. and I.M.; formal analysis, L.B and D.K.; investigation, L.B and D.K.; resources, I.M.; data curation, L.B and I.M.; writing–original draft preparation, L.B., D.K. and I.M.; writing–review and editing, L.B. and I.M.; visualisation, D.K. and I.M.; supervision, L.B and D.K.; project administration, L.B.; funding acquisition, L.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Spanish Ministry of Economy [grant number ECO2015-66314-R].

Acknowledgments: The authors wish to acknowledge discussions with researchers at Riskcenter at the University of Barcelona and the constructive comments made by two anonymous referees, which helped to improve the quality of the paper.

Conflicts of Interest: The authors declare no conflicts of interest.

References

Aitkin, Murray. 1999. A general maximum likelihood analysis of variance components in generalized linear models. *Biometrics* 55: 117–28. [CrossRef] [PubMed]

Bermúdez, Lluís. 2009. A priori ratemaking using bivariate Poisson regression models. *Insurance: Mathematics and Economics* 44: 135–41. [CrossRef]

Bermúdez, Lluís, and Dimitris Karlis. 2012. A finite mixture of bivariate Poisson regression models with an application to insurance ratemaking. *Computational Statistics & Data Analysis* 56: 3988–99.

Bolancé, Catalina, Montserrat Guillén, and Jean Pinquet. 2003. Time-varying credibility for frequency risk models: Estimation and tests for autoregressive specifications on the random effects. *Insurance: Mathematics and Economics* 33: 273–82. [CrossRef]

Bolancé, Catalina, Montserrat Guillén, and Jean Pinquet. 2008. On the link between credibility and frequency premium. *Insurance: Mathematics and Economics* 43: 209–13. [CrossRef]

Boucher, Jean-Philippe, and Michel Denuit. 2008. Credibility premiums for the zero inflated Poisson model and new hunger for bonus interpretation. *Insurance: Mathematics and Economics* 42: 727–35. [CrossRef]

Boucher, Jean-Philippe, Michel Denuit, and Montserrat Guillén. 2007. Risk classification for claim counts: A comparative analysis of various zero-inflated mixed Poisson and hurdle models. *North American Actuarial Journal* 11: 110–31. [CrossRef]

Boucher, Jean-Philippe, Michel Denuit, and Montserrat Guillén. 2009. Number of accidents or number of claims? an approach with zero-inflated Poisson models for panel data. *Journal of Risk and Insurance* 76: 821–46. [CrossRef]

Breslow, Norman E. 1984. Extra-Poisson variation in log-linear models. *Applied Statistics* 33: 38–44. [CrossRef]

Brouhns, Natacha, Montserrat Guillén, Michael Denuit, and Jean Pinquet. 2003. Bonus-malus scales in segmented tariffs with stochastic migration between segments. *Journal of Risk and Insurance* 70: 577–99. [CrossRef]

Byung-Jung, Park, Dominique Lord, and Chungwon Lee. 2014. Finite mixture modeling for vehicle crash data with application to hotspot identification. *Accident Analysis & Prevention* 71: 319–26.

Dean, Charmaine, Jerald Lawless, and Gordon Willmot. 1989. A mixed Poisson-inverse-gaussian regression model. *Canadian Journal of Statistics* 17: 171–81. [CrossRef]

Denuit, Michael, Xavier Marechal, Sandra Pitrebois, and Jean-François Walhin. 2007. *Actuarial Modelling of Claim Counts: Risk Classification, Credibility and Bonus-Malus Systems.* New York: Wiley.

Dionne, George, and Charles Vanasse. 1989. A generalization of actuarial automobile insurance rating models: The negative binomial distribution with a regression component. *ASTIN Bulletin* 19: 199–212. [CrossRef]

Dionne, George, and Charles Vanasse. 1992. Automobile insurance ratemaking in the presence of asymmetrical information. *Journal of Applied Econometrics* 7: 149–65. [CrossRef]

Grun, Bettina, and Friedrich Leisch. 2007. Fitting finite mixtures of generalized linear regressions in R. *Computational Statistics and Data Analysis* 51: 5247–52. [CrossRef]

Grun, Bettina, and Friedrich Leisch. 2008. Flexmix version 2: Finite mixtures with concomitant variables and varying and constant parameters. *Journal of Statistical Software* 28: 1–35. [CrossRef]

Guillén, Montserrat, Jens Perch Nielsen, Mercedes Ayuso, and Ana Pérez-Marín. 2019. The use of telematics devices to improve automobile insurance rates. *Risk Analysis* 39: 662–72. [CrossRef]

Hennig, Christian. 2000. Identifiablity of models for clusterwise linear regression. *Journal of Classification* 17: 273–96. [CrossRef]

Lambert, Diane. 1992. Zero-inflated Poisson regression, with an application to defects in manufacturing. *Technometrics* 34: 1–14. [CrossRef]

Lord, Dominique, Simon Washington, and John N. Ivan. 2007. Further notes on the application of zero-inflated models in highway safety. *Accident Analysis & Prevention* 39: 53–57.

Mullahy, John. 1986. Specification and testing of some modified count data models. *Journal of Econometrics* 33: 341–65. [CrossRef]

Papastamoulis, Panagiotis, Marie-Laure Martin-Magniette, and Cathy Maugis-Rabusseau. 2016. On the estimation of mixtures of Poisson regression models with large number of components. *Computational Statistics & Data Analysis* 93: 97–106.

Park, Byung-Jung, and Dominique Lord. 2009. Application of finite mixture models for vehicle crash data analysis. *Accident Analysis and Prevention* 41: 683–91. [CrossRef] [PubMed]

Pinquet, Jean, Montserrat Guillén, and Catalina Bolancé. 2001. Long-range contagion in automobile insurance data: Estimation and implications for experience rating. *ASTIN Bulletin* 31: 337–48. [CrossRef]

Wang, Peiming, Martin L. Puterman, Iain Cockburn, and Nhu Le. 1996. Mixed Poisson regression models with covariate dependent rates. *Biometrics* 52: 381–400. [CrossRef]

Winkelmann, Rainer. 2008. *Econometric Analysis of Count Data*, 4th ed. New York: Springer.

Zou, Yajie, Yunlong Zhang, and Dominique Lord. 2013. Application of finite mixture of negative binomial regression models with varying weight parameters for vehicle crash data analysis. *Accident Analysis & Prevention* 50: 1042–51.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).

risks

MDPI

Article

Machine Learning in Least-Squares Monte Carlo Proxy Modeling of Life Insurance Companies

Anne-Sophie Krah [1,*], Zoran Nikolić [2] and Ralf Korn [1,3]

[1] Department of Mathematics, TU Kaiserslautern, Erwin-Schrödinger-Straße, Geb. 48, 67653 Kaiserslautern, Germany

[2] Mathematical Institute, University Cologne, Weyertal 86-90, 50931 Cologne, Germany; znikolic@uni-koeln.de

[3] Department Financial Mathematics, Fraunhofer ITWM, Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany; korn@mathematik.uni-kl.de

* Correspondence: anne-sophiekrah@web.de

Received: 30 December 2019; Accepted: 12 February 2020; Published: 21 February 2020

Abstract: Under the Solvency II regime, life insurance companies are asked to derive their solvency capital requirements from the full loss distributions over the coming year. Since the industry is currently far from being endowed with sufficient computational capacities to fully simulate these distributions, the insurers have to rely on suitable approximation techniques such as the least-squares Monte Carlo (LSMC) method. The key idea of LSMC is to run only a few wisely selected simulations and to process their output further to obtain a risk-dependent proxy function of the loss. In this paper, we present and analyze various adaptive machine learning approaches that can take over the proxy modeling task. The studied approaches range from ordinary and generalized least-squares regression variants over generalized linear model (GLM) and generalized additive model (GAM) methods to multivariate adaptive regression splines (MARS) and kernel regression routines. We justify the combinability of their regression ingredients in a theoretical discourse. Further, we illustrate the approaches in slightly disguised real-world experiments and perform comprehensive out-of-sample tests.

Keywords: least-squares monte carlo method; machine learning; proxy modeling; life insurance; Solvency II

1. Introduction

The Solvency II directive of the European Parliament and European Council (2009) requires from insurance companies a derivation of the solvency capital requirement (SCR) using the full probability distributions of losses over a one-year period. Some life insurers comply with this requirement by setting up internal models. Other insurers opt for the much simpler standard formula, which enables an aggregation of the company's exposures to single risks. Lacking an analytical valuation formula for the losses in a one-year period, life insurers with an internal model are supposed to utilize a Monte Carlo approach usually called nested simulations approach (Bauer et al. (2012)). In practice their cash-flow-projection (CFP) models need to be simulated several hundred thousand to several million times for a robust implementation of the nested simulations approach. But the insurers are currently far from being endowed with sufficient computational capacities to perform such expensive simulation tasks. By applying suitable approximation techniques like the least-squares Monte Carlo (LSMC) approach of Bauer and Ha (2015), the insurers are able to overcome these computational hurdles though. For example, they can implement the LSMC framework formalized by Krah et al. (2018) and applied by, for example, Bettels et al. (2014), to derive their full loss distributions. The central idea of this framework is to carry out a comparably small number of wisely chosen nested Monte

Carlo simulations and to feed the simulation results into a supervised machine learning algorithm that translates the results into a proxy function of the insurer's loss (output) with respect to the underlying risk factors (input).

Our starting point is the LSMC framework from Krah et al. (2018). In the following the same approach for the proxy derivation is assumed, we will only amend the calibration and validation steps. Therefore, we neither repeat the simulation setting nor the procedure for the full loss distribution forecast and SCR calculation here in detail. The purpose of this exposition is to introduce different machine learning methods that can be applied in the calibration step of the LSMC framework, to point out their similarities and differences and to compare their out-of-sample performances in the same slightly disguised real-world LSMC example already used in Krah et al. (2018).

We describe the data basis used for calibration and validation in Section 2.1, the structure of the calibration algorithm in Section 2.2 and our validation approach in Section 2.3. Our focus lies on out-of-sample performance rather than computational efficiency as the latter becomes only relevant if the former gives reason for it. We analyze a very realistic data basis with 15 risk factors and validate the proxy functions based on a very comprehensive and computationally expensive nested simulations test set comprising the SCR estimate.

The main idea of our approach is to combine different regression methods with an adaptive algorithm, in which the proxy functions are built up of basis functions in a stepwise fashion. In a four risk factor LSMC example, Teuguia et al. (2014) applied a full model approach, forward selection, backward elimination and a bidirectional approach as, for example, discussed in Hocking (1976) with orthogonal polynomial basis functions. They stated that only forward selection and the bidirectional approach were feasible when the number of risk factors or the polynomial degree exceeded 7, as then the resulting other models exploded. Life insurance companies covering a wide range of contracts in their portfolio are typically exposed to even more risk factors like, for example, 15. Complex business regulation frameworks such as those in Germany cause non-linear dependencies between risk factors and losses, which naturally lead to polynomials of higher degrees in the chosen proxy models. In these cases, even the standard forward selection and bidirectional approaches become infeasible as the sets of candidate terms from which the basis functions are chosen will explode then as well. We therefore follow the suggestion of Krah et al. (2018) to implement the so-called principle of marginality, an iteration-wise update technique of the set of candidate terms that lets the algorithm get along with comparably few carefully selected candidate terms.

Our main contribution is to identify, explain and illustrate a collection of regression methods and model selection criteria from the variety of regression design options that provide suitable proxy functions in the LSMC framework when applied in combination with the principle of marginality. After some general remarks in Section 3.1, we describe ordinary least-squares (OLS) regression in Section 3.2, generalized linear models (GLMs) by Nelder and Wedderburn (1972) in Section 3.3, generalized additive models (GAMs) by Hastie and Tibshirani (1986) and Hastie and Tibshirani (1990) in Section 3.4, feasible generalized least-squares (FGLS) regression in Section 3.5, multivariate adaptive regression splines (MARS) by Friedman (1991) in Section 3.6, and kernel regression by Watson (1964) and Nadaraya (1964) in Section 3.7. While some regression methods such as OLS and FGLS regression or GLMs can immediately be applied in conjunction with numerous model selection criteria such as Akaike information criterion (AIC), Bayesian information criterion (BIC), Mallow's C_P or generalized cross-validation (GCV), other regression methods such as GAMs, MARS, kernel, ridge or robust regression require well thought-through modifications thereof or work only with non-parametric alternatives such as k-fold or leave-one-out cross-validation. For adaptive approaches of FGLS, ridge and robust regression in life insurance proxy modeling, see also Hartmann (2015), Krah (2015) and Nikolić et al. (2017), respectively.

In the theory sections, we present the models with their assumptions, important properties and popular estimation algorithms and demonstrate how they can be embedded in the adaptive algorithm by proposing feasible implementation designs and combinable model selection criteria.

While we shed light on the theoretical basic concepts of the models to lay the groundwork for the application and interpretation of the later following numerical experiments, we forego describing in detail technical enhancements or peculiarities of the involved algorithms and instead refer the interested reader to further sources. Additionally we provide the practicioners with R packages containing useful implementations of the presented regression routines. We complement the theory sections by corresponding empirical results in Section 4, throughout which we perform the same Monte Carlo approximation task to make the performance of the various methods comparable. We measure the approximation quality of the resulting proxy functions by means of aggregated validation figures on three out-of-sample test sets.

Conceivable alternatives to the entire adaptive algorithm are other typical machine learning techniques such as artificial neural networks (ANNs), decision tree learning or support vector machines. In particular, the classical feed forward networks proposed by Hejazi and Jackson (2017) and applied in various ways by Kopczyk (2018), Castellani et al. (2018), Born (2018) and Schelthoff (2019) were shown to capture the complex nature of CFP models well. A major challenge here is not only to find reliable hyperparameters such as the numbers of hidden layers and nodes in the network, batch size, weight initializer probability distribution, learning rate or activation functions but also the high dependence on the random seeds. We plan to contribute to this in a further publication which will be dedicated to hyperparameter search algorithms and stabilization methods such as ensemble methods. As an alternative to feed forward networks, Kazimov (2018) suggested to use radial basis function networks albeit so far none of the tested approaches performed better than the ordinary least squares regression in Krah et al. (2018).

In decision tree learning, random forests and tree-based gradient boosting machines were considered by Kopczyk (2018) and Schoenenwald (2019). While random forests were outperformed by feed forward networks but did better than the least absolute shrinkage and selection operator (LASSO) by Tibshirani (1996) in the example of the former author, they generally performed worse than the adaptive approaches by Krah et al. (2018) with OLS regression in numerous examples of the latter author. The gradient boosting machines, requiring more parameter tuning and thus being more versatile and demanding, came overall very close to the adaptive approaches.

Castellani et al. (2018) compared support vector regression (SVR) by Drucker et al. (1997) to ANNs and the adaptive approaches by Teuguia et al. (2014) in a seven risk factor example and found the performance of SVR placed somewhere inbetween the other two approaches with the ANNs getting closest to the nested simulations benchmark. As some further non-parametric approaches, Sell (2019) tested least-squares support-vector machines (LS-SVM) by Suykens and Vandewalle (1999) and shrunk additive least-squares approximations (SALSA) by Kandasamy and Yu (2016) in comparison to ANNs and the adaptive approaches by Krah et al. (2018) with OLS regression. In his examples, SALSA was able to beat the other two approaches whereas LS-SVM was left far behind. The analyzed machine learning alternatives have in common that they require at least to some degree a fine-tuning of some model hyperparameters. Since this is often a non-trivial but crucial task for generating suitable proxy functions, finding efficient and reliable search algorithms should become a subject of future research.

2. Calibration and Validation in the LSMC Framework

2.1. Fitting and Validation Points

2.1.1. Outer Scenarios and Inner Simulations

Our starting point is the LSMC approach (Krah et al. (2018)). LSMC proxy functions are calibrated conditional on the *fitting points* generated by the Monte Carlo simulations of the CFP model. Additional out-of-sample *validation points* serve as a mean for an assessment of the goodness-of-fit. The explaining variables of a proxy function are financial and actuarial risks the insurance company is exposed to. Examples for these risks are changes in interest rates, equity, credit, mortality, morbidity, lapse and expense levels over the one-year period. The dependent variable is an economic variable like the available capital, loss of available capital or best estimate of liabilites over the one-year period. Figure 1 plots the fitting values of an exemplary economic variable with respect to a financial risk factor. By an *outer scenario* we refer to a specific realized stress level combination of these risk factors over one year, and by an *inner simulation* to a stochastic path of an outer scenario in the CFP model under the given risk-neutral probability measure. Each outer scenario is assigned the probability weighted mean value of the economic variable over the corresponding inner simulations. In the LSMC context the fitting values are the mean values over only few inner simulations whereas the validation values are derived as the mean values over many inner simulations.

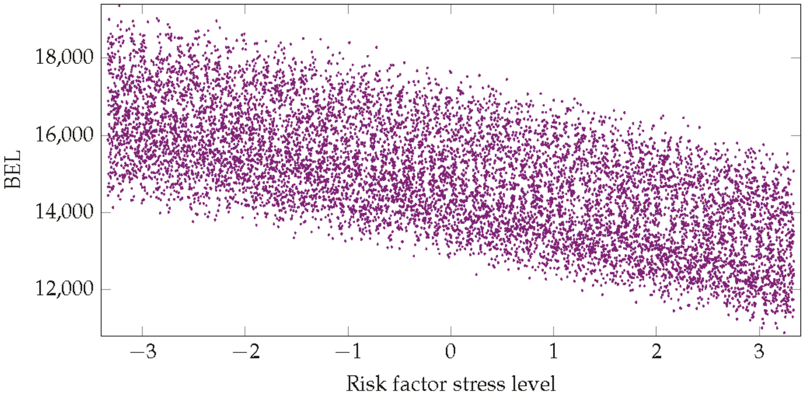


Figure 1. Fitting values of best estimate of liabilities with respect to a financial risk factor.

2.1.2. Different Trade-Off Requirements

According to the law of large numbers, this construction makes the validation values comparably stable while the fitting values are very volatile. Typically, the very limited fitting and validation simulation budgets are of similar sizes. Hence the few inner simulations in the case of the fitting points allow a great diversification among the outer scenarios whereas the many inner simulations in the case of the validation points let the validation values be quite close to their expectations but at the cost of only little diversification among the outer scenarios. These opposite ways to deal with the trade-off between the numbers of outer scenarios and inner simulations reflect the different requirements for the fitting and validation points in the LSMC approach. While the fitting scenarios should cover the domain of the real-world scenarios well to serve as a good regression basis, the validation values should approximate the expectations of the economic variable at the validation scenarios well to provide appropriate target values for the proxy functions.

2.2. Calibration Algorithm

2.2.1. Five Major Components

The calibration of the proxy function is performed by an adaptive algorithm that can be decomposed into the following five major components: (1) a set of allowed basis function types for the proxy function, (2) a regression method, (3) a model selection criterion, (4) a candidate term update principle, and (5) the number of steps per iteration and the directions of the algorithm. For illustration, we adopt the flowchart of the adaptive algorithm from Krah et al. (2018) and depict it in Figure 2. While components (1) and (5) enter the flowchart implicitly through the start proxy, candidate terms and the order of the processes and decisions in the chart, components (2), (3) and (4) are explicitly indicated through the labels “Regression”, “Model Selection Criterion” and “Get Candidate Terms”.

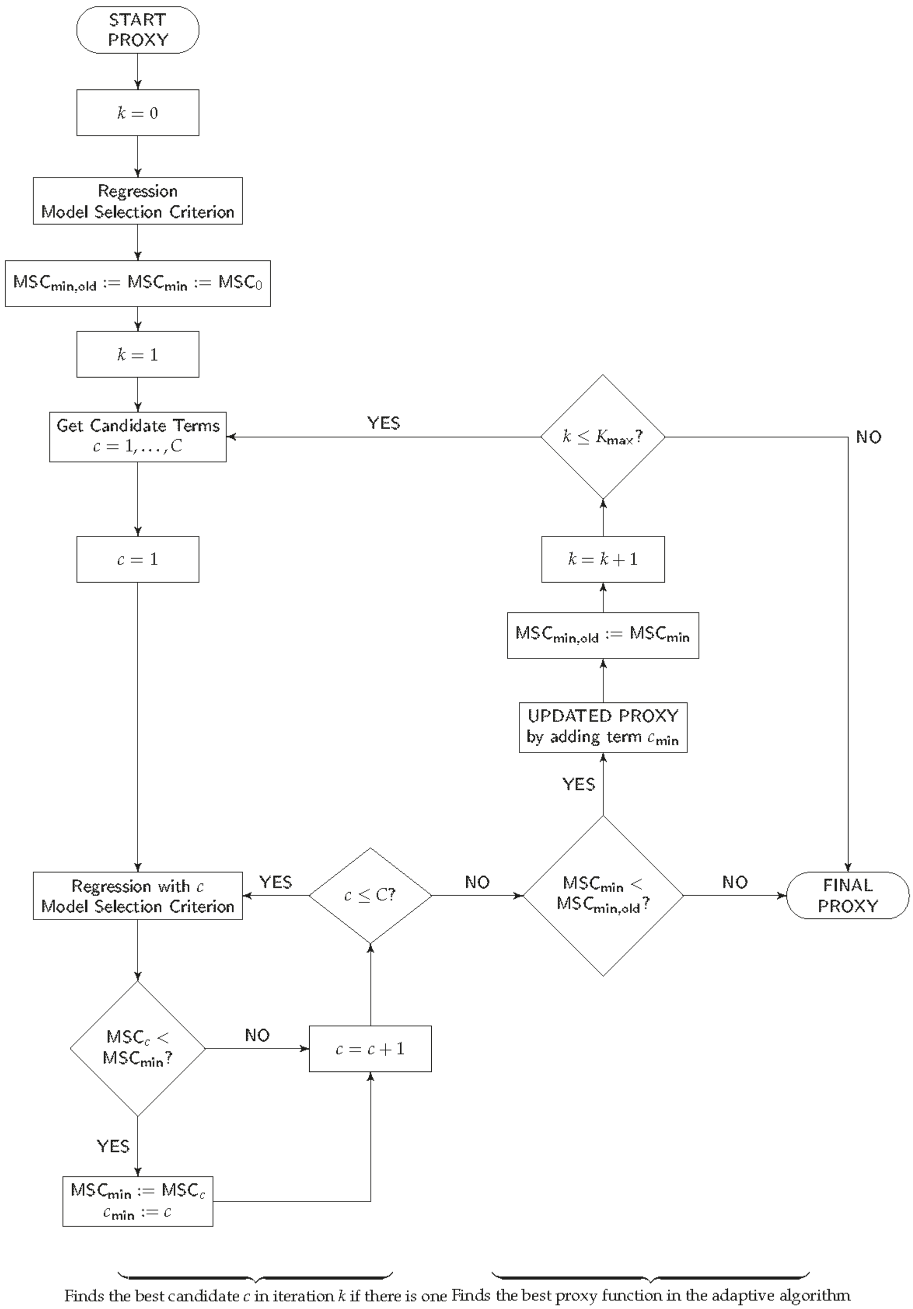

START PROXY
$k = 0$
Regression
Model Selection Criterion
$\mathsf{MSC}_{\mathsf{min,old}} := \mathsf{MSC}_{\mathsf{min}} := \mathsf{MSC}_0$
$k = 1$
Get Candidate Terms
$c = 1, \ldots, C$
YES
$k \leq K_{\mathsf{max}}$?
NO
$c = 1$
$k = k + 1$
$\mathsf{MSC}_{\mathsf{min,old}} := \mathsf{MSC}_{\mathsf{min}}$
UPDATED PROXY
by adding term c_{min}
YES
Regression with c
Model Selection Criterion
YES
$c \leq C$?
NO
$\mathsf{MSC}_{\mathsf{min}} <$
$\mathsf{MSC}_{\mathsf{min,old}}$?
NO
FINAL PROXY
$\mathsf{MSC}_c <$
$\mathsf{MSC}_{\mathsf{min}}$?
NO
$c = c + 1$
YES
$\mathsf{MSC}_{\mathsf{min}} := \mathsf{MSC}_c$
$c_{\mathsf{min}} := c$
Finds the best candidate c in iteration k if there is one
Finds the best proxy function in the adaptive algorithm

Figure 2. Flowchart of the calibration algorithm.

Let us briefly recapitulate the choice of components (1)–(5) from the successful applications of the adaptive algorithm in the insurance industry as described in Krah et al. (2018). As the function types for the basis functions (1), let only monomials be allowed. Let the regression method (2) be ordinary least-squares (OLS) regression and the model selection criterion (3) Akaike information criterion (AIC) from Akaike (1973). Let the set of candidate terms (4) be updated by the principle of marginality to which we will return in greater detail below. Lastly, when building up the proxy function iteratively, let the algorithm make only one step per iteration in the forward direction (5) meaning that in each iteration exactly one basis function is selected which cannot be removed anymore (adaptive forward stepwise selection).

2.2.2. Iterative Procedure

The algorithm starts in the upper left side of Figure 2 with the specification of the start proxy basis functions. We specify only the intercept so that the first regression ($k = 0$) reduces to averaging over all fitting values. In order to harmonize the choices of OLS regression and AIC, we assume that the errors are normally distributed and homoscedastic because then the OLS estimator coincides with the maximum likelihood estimator. AIC is a relative measure for the goodness-of-fit of the proxy function and is defined as twice the negative of the maximum log-likelihood plus twice the number of degrees of freedom. The smaller the AIC score, the better the fit, and thus the trade-off between a too complex (overfitting) and too simple model (underfitting).

At the beginning of each iteration ($k = 1, \dots, K - 1$), the set of candidate terms is updated by the *principle of marginality* which stipulates that a monomial basis function becomes a candidate if and only if all its derivatives are already included in the proxy function. The choice of a monomial basis is compatible to the principle of marginality. Using such a principle saves computational costs by selecting the basis functions conditionally on the current proxy function structure. In the first iteration ($k = 1$), all linear monomials of the risk factors become candidates as their derivatives are constant values which are represented by the intercept.

The algorithm proceeds on the lower left side of the flowchart with a loop in which all candidate terms are separately added to the proxy function structure and tested with regard to their additional explanatory power. With each candidate, the fitting values are regressed against the fitting scenarios and the AIC score is calculated. If no candidate reduces the currently smallest AIC score, the algorithm terminates, and otherwise, the proxy function is updated by the one which reduces AIC most. Then the next iteration ($k + 1$) begins with the update of the set of candidate terms, and so on. As long as no termination occurs, this procedure is repeated until the prespecified maximum number of terms $K_{\max}$ is reached.

2.3. Validation Figures

2.3.1. Validation Sets

Since it is the objective of this paper to propose suitable regression methods for the proxy function calibration in the LSMC framework, we introduce several validation figures serving as indicators for the approximation quality of the proxy functions. We measure the out-of-sample performance of each proxy function on three different validation sets by calculating five validation figures per set.

The three validation sets are a Sobol set, a nested simulations set and a capital region set. Unlike the Sobol set, the nested simulations and capital region sets do not serve as feasible validation sets in the LSMC routine as they become known only after evaluating the proxy function as explained below. Furthermore, they require massive computational capacities. Yet they can be regarded as the natural benchmark for the LSMC-based method and are thus very valuable for this analysis. Figure 3 plots the nested simulation values of an exemplary economic variable with respect to a financial risk factor. The Sobol set consists of, for example, between $L = 15$ and $L = 200$ Sobol validation points, of which the scenarios follow a Sobol sequence covering the fitting space uniformly. Thereby, the fitting space is the cube on which the outer fitting scenarios are defined. It has to cover the space of real-world scenarios used for the full loss distribution forecast sufficiently well. For interpretive reasons, sometimes the Sobol set is extended by points with, for example, one-dimensional risk scenarios or scenarios producing a risk capital close to the SCR (= 99.5% value-at-risk) in previous risk capital calculations.

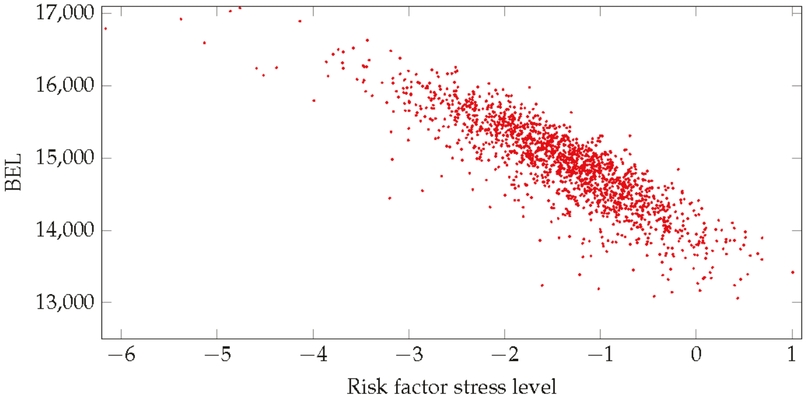


Figure 3. Nested simulation values of best estimate of liabilities with respect to a financial risk factor.

The nested simulations set comprises the, for example, $L = 820$ to $L = 6554$ validation points of which the scenarios correspond to the, for example, highest 2.5% to 5% losses from the full loss distribution forecast made by the proxy function that had been derived under the standard calibration algorithm choices described in Section 2.2. Like in the example of Chapter 5.2 in Krah et al. (2018), the order of these losses-which scenarios lead to which quantiles?following from the fourth and last step of the LSMC approach is very similar to the order following from the nested simulations approach. Therefore the scenarios of the nested simulations set are simply chosen by the order of the losses resulting from the LSMC approach. Several of these scenarios consist of stresses falling out of the fitting space. Compare Figures 1 and 3 which depict fitting and nested simulation values from the same proxy modeling task with respect to the same risk factor. Severe outliers due to extreme stresses far outside of the fitting space should be excluded from the set. The capital region set is a subset of the nested simulations set containing the nested simulations SCR estimate, that is, the scenario leading to the 99.5% loss, and the, for example, 64 losses above and below, which makes in total, for example, $L = 129$ validation points.

2.3.2. Validation Figures

The five validation figures reported in our numerical experiments comprise two normalized mean absolute errors (MAEs), one with respect to the magnitude of the economic variable itself and one with respect to the magnitude of the corresponding market value of assets. They comprise further the mean error, that is, the mean of the residuals, as well as two validation figures based on the change of the economic variable from its base value (see the definition of the base value below): the normalized MAE with respect to the magnitude of the changes and the mean error of these changes. The smaller the normalized MAEs are, the better the proxy function approximates the economic variable. However, the validation values are afflicted with Monte Carlo errors so that the normalized MAEs serve only as meaningful indicators as long as the proxy functions do not become too precise. The means of the residuals should be possibly close to zero since they indicate systematic deviations of the proxy functions from the validation values. While the first three validation figues measure how well the proxy function reflects the economic variable in the CFP model, the latter two address the approximation effects on the SCR, compare Chapter 3.4.1 of Krah et al. (2018).

Let us write the absolute value as $|\cdot|$ and let L denote the number of validation points. Then we can express the MAE of the proxy function $\widehat{f}\left(x^i\right)$ evaluated at the validation scenarios x^i versus the validation values y^i as $\frac{1}{L}\sum_{i=1}^{L}\left|y^i - \widehat{f}\left(x^i\right)\right|$. After normalizing the MAE with respect to the mean of the absolute values of the economic variable or the market value of assets, that is, $\frac{1}{L}\sum_{i=1}^{L}\left|d^i\right|$ with $d^i \in \left\{y^i, a^i\right\}$, we obtain the first two validation figures, that is,

$$\mathrm{mae} = \frac{\sum_{i=1}^{L}\left|y^i - \widehat{f}\left(x^i\right)\right|}{\sum_{i=1}^{L}\left|d^i\right|}. \tag{1}$$

In the following, we will refer to (1) with $d^i = y^i$ as the MAE with respect to the *relative metric*, and to (1) with $d^i = a^i$ as the MAE with respect to the *asset metric*. The mean of the residuals is given by

$$\mathrm{res} = \frac{1}{L}\sum_{i=1}^{L}\left(y^i - \widehat{f}\left(x^i\right)\right). \tag{2}$$

Let us refer by the *base value* y^0 to the validation value corresponding to the base scenario x^0 in which no risk factor has an effect on the economic variable. In analogy to (1) but only with respect to the relative metric, we introduce another normalized MAE by

$$\mathrm{mae}^0 = \frac{\sum_{i=1}^{L}\left|\left(y^i - y^0\right) - \left(\widehat{f}\left(x^i\right) - \widehat{f}\left(x^0\right)\right)\right|}{\sum_{i=1}^{L}\left|y^i - y^0\right|}. \tag{3}$$

The mean of the corresponding residuals is given by

$$\mathrm{res}^0 = \frac{1}{L}\sum_{i=1}^{L}\left(\left(y^i - y^0\right) - \left(\widehat{f}\left(x^i\right) - \widehat{f}\left(x^0\right)\right)\right). \tag{4}$$

In addition to these five validation figures, let us define the base residual which can be used as a substitute for (4) depending on personal taste. The base residual can easily be extracted from (2) and (4) by

$$\mathrm{res}^{\mathrm{base}} = y^0 - \widehat{f}\left(x^0\right) = \mathrm{res} - \mathrm{res}^0. \tag{5}$$

3. Machine Learning Regression Methods

3.1. General Remarks

As the main part of our work, we will compare various types of machine learning regression approaches for determining suitable proxy functions in the LSMC framework. The methods we present in this section range from ordinary and generalized least-squares regression variants over GLM and GAM approaches to multivariate adaptive regression splines and kernel regression approaches.

The performance of the newly derived proxy functions when applied to the described validation sets is one way of comparing the different methods. Another way consists of ensuring compatibility with the principle of marginality and utilizing a suitable model selection criterion such as AIC in order to be able to compare iteration-wise the candidate models inside the approaches.

We will in the following sections shortly introduce the different methods, collect some theoretical properties and then concentrate on aspects of their implementation. Their numerical performance on the different validation sets is the subject of Section 4.

Our aim in the calibration step below is to estimate the conditional expectation $Y(X)$ under the risk-neutral measure given an outer scenario X. In contrast to Krah et al. (2018) $Y(X)$ does not necessarily have to be the available capital but can instead be, for example, the best estimate of liabilites or the market value of assets. The D-dimensional fitting scenarios are always generated under the physical probability measure $\mathbb{P}'$ on the fitting space which itself is a subspace of $\mathbb{R}^D$.

3.2. Ordinary Least-Squares (OLS) Regression

3.2.1. The Regression Model

In iteration $K-1$ of the adaptive forward stepwise algorithm (as given in Section 2.2), the OLS approximation consists of a linear combination of suitable linearly independent basis functions $e_k(X) \in L^2\left(\mathbb{R}^D, \mathcal{B}, \mathbb{P}'\right),\ k = 0, 1, \ldots, K-1$, that is,

$$Y(X) \overset{K<\infty}{\approx} f(X) = \sum_{k=0}^{K-1} \beta_k e_k(X). \tag{6}$$

We call $f(X)$ the predictor of $Y(X)$ or the *systematic component*.

With the fitting points $\left(x^i, y^i\right),\ i = 1, \ldots, N$, and uncorrelated errors ϵ^i (the *random components*) having the same variance $\sigma^2 > 0$ (= homoscedastic errors), we obtain the classical linear regression model

$$y^i = \sum_{k=0}^{K-1} \beta_k e_k\left(x^i\right) + \epsilon^i, \tag{7}$$

where $e_0\left(x^i\right) = 1$ and β_0 is the intercept. Then, the ordinary least-squares (OLS) estimator $\widehat{\boldsymbol{\beta}}_{\text{OLS}}$ of the coefficients is given by

$$\widehat{\boldsymbol{\beta}}_{\text{OLS}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^K}{\arg\min} \left\{ \sum_{i=1}^{N} \left(y^i - \sum_{k=0}^{K-1} \beta_k e_k\left(x^i\right) \right)^2 \right\}. \tag{8}$$

Using the notation $z_{ik} = e_k\left(x^i\right)$ the OLS problem is solved explicitly by

$$\widehat{\boldsymbol{\beta}}_{\text{OLS}} = \left(Z^T Z\right)^{-1} Z^T \mathbf{y}. \tag{9}$$

The proxy function $\widehat{f}(X)$ for the economic variable $Y(X)$ given an outer scenario X is

$$Y(X) \overset{K,N<\infty}{\approx} \widehat{f}(X) = \sum_{k=0}^{K-1} \widehat{\beta}_{\text{OLS},k} e_k(X). \tag{10}$$

For a practical implementation see, for example, function *lm(·)* in the R package *stats* of R Core Team (2018).

3.2.2. Gauss-Markov Theorem, ML Estimation and AIC

Under the assumptions of strict exogeneity $E[\boldsymbol{\epsilon} \mid Z] = \mathbf{0}$ (A1), a spherical error variance $V[\boldsymbol{\epsilon} \mid Z] = \sigma^2 I_N$ with I_N the N-dimensional identity matrix (A2), and linearly independent basis functions (A3), we have (compare, for example, Hayashi (2000)):

- The OLS estimator is the best linear unbiased estimator (BLUE) of the coefficients in the classical linear regression model (7) (*Gauss-Markov Theorem*).
- If the errors ϵ in (7) are in addition normally distributed (A4), then the OLS estimator and the maximum likelihood (ML) estimator of the coefficients coincide.
- Under Assumptions (A1)-(A4) the Akaike information criterion (AIC) has the form

$$\text{AIC} = -2l\left(\widehat{\boldsymbol{\beta}}_{\text{OLS}}, \widehat{\sigma}^2\right) + 2(K+1) = N\left(\log\left(2\pi\widehat{\sigma}^2\right) + 1\right) + 2(K+1). \tag{11}$$

3.3. Generalized Linear Models (GLMs)

3.3.1. The Regression Model

The systematic component of a GLM (see Nelder and Wedderburn (1972) for its introduction) equals the linear predictor $\eta = f(X)$ of the model in (6). However, one uses a monotonic link function $g(\cdot)$ that relates the economic variable $Y(X)$ to the linear predictor via

$$g(\underbrace{Y(X)}_{=\,\mu}) \overset{K<\infty}{\approx} \underbrace{f(X)}_{=\,\eta} = \sum_{k=0}^{K-1} \beta_k z_k = \mathbf{z}^T \boldsymbol{\beta}, \tag{12}$$

with $\mathbf{z} = \left(e_0\left(X\right), \ldots, e_{K-1}\left(X\right)\right)^T$.

Of course, the choice of the link function $g(.)$ is a critical aspect. A possible motivation is a non-negativity requirement on $Y(X)$ that can be satisfied using $g(y) = \ln(y)$. Further comments on choices of the link function are motivated below.

3.3.2. Canonical Link Function, GLM Estimation and IRLS Algorithm

While the normal distribution assumption for the random component allowed the derivation of nice properties in the linear model of the preceding section, the GLM considers random components with (conditional) distributions from the exponential family. Its canonical form with parameter θ is given by the density function

$$\pi(y \mid \theta, \phi) = \exp\left(\frac{y\theta - b(\theta)}{a(\phi)} + c(y, \phi)\right), \tag{13}$$

where $a(\phi)$, $b(\theta)$ and $c(y, \phi)$ are specific functions. For example, a normally distributed economic variable with mean μ and variance σ^2 is given by $a(\phi) = \phi$, $b(\theta) = \frac{\theta^2}{2}$ and $c(y, \phi) = -\frac{1}{2}\left(\frac{y^2}{\sigma^2} + \log\left(2\pi\sigma^2\right)\right)$ with $\theta = \mu$ and $\phi = \sigma^2$.

For a random variable Y with a distribution from the exponential family, we have

$$E(Y) = \mu = b'(\theta), \;\; Var(Y) = b''(\theta)a(\phi) =: V\left[\mu\right] a(\phi)\,. \tag{14}$$

$a(\phi)$ is called a dispersion parameter, $V[.]$ the variance function. We will in the following make the simplifying assumption $a(\phi^i) = \phi, i = 1, ..., N$ for a constant value of ϕ (A5) and then obtain the ML estimator in the GLM from Equation (13) as

$$\widehat{\boldsymbol{\beta}}_{\mathrm{GLM}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^K}{\arg\max} \left\{ \sum_{i=1}^{N} \left(\frac{y^i\theta^i - b(\theta^i)}{\phi} + c(y^i, \phi) \right) \right\}. \tag{15}$$

Under (A5), there does in general not exist a closed-form solution for the GLM coefficient estimator (15). The resulting iterative method will be simplified for so-called canonical link functions $g(\mu) = \theta$ which due to relation (14) are given by

$$g(\mu) = (b')^{-1}(\mu), \tag{16}$$

with $b(.)$ from the definition of the exponential family. Examples of pairs of canonical link functions and corresponding distributions are $g(\mu) = \mu$ and the normal, $g(\mu) = 1/\mu$ and the gamma, and $g(\mu) = 1/\mu^2$ and the inverse Gaussian distribution.

In Chapter 2.5, McCullagh and Nelder (1989) apply Fisher's scoring method to obtain an approximation to the GLM estimator. Further, McCullagh and Nelder (1989) justify how Fisher's scoring method can be cast in the form of the iteratively reweighted least squares (IRLS) algorithm. To state the IRLS algorithm in our context, we need some notation.

Let $\widehat{\eta}^i_{(t)} = \widehat{f}\left(x^i\right)$ be the estimate for the linear predictor evaluated at fitting scenario x^i, compare (12). Let $\widehat{\mu}^i_{(t)} = g^{-1}\left(\widehat{\eta}^i_{(t)}\right)$ be the estimate for the economic variable, and $\frac{d\eta}{d\mu}\left(\widehat{\mu}^i_{(t)}\right) = g'\left(\widehat{\mu}^i_{(t)}\right)$ the first derivative of the link function with respect to the economic variable evaluated at $\widehat{\mu}^i_{(t)}$. Furthermore, we introduce the weight matrix $W^{(t)} = \operatorname{diag}\left(w^1\left(\widehat{\boldsymbol{\beta}}^{(t)}\right), \ldots, w^N\left(\widehat{\boldsymbol{\beta}}^{(t)}\right)\right)$ with components given by

$$\widehat{w}^i\left(\widehat{\boldsymbol{\beta}}^{(t)}\right) = \left(\frac{d\eta}{d\mu}\left(\widehat{\mu}^i_{(t)}\right)\right)^{-2} V\left[\widehat{\mu}^i_{(t)}\right]^{-1}, \tag{17}$$

and $V\left[\widehat{\mu}^i_{(t)}\right]$ the variance function from above evaluated at $\widehat{\mu}^i_{(t)}$. Finally, we define $D^{(t)} = \operatorname{diag}(d^1_{(t)}, ..., d^N_{(t)})$ with $d^i_{(t)} = g'\left(\widehat{\mu}^i_{(t)}\right)$ which allows us to formulate the IRLS algorithm for canonical link functions.

IRLS algorithm. *Perform the iterative approximation procedure below with an initialization of* $\widehat{\mu}^i_{(0)} = y^i + 0.1$ *and* $\widehat{\eta}^i_{(0)} = g\left(\widehat{\mu}^i_{(0)}\right)$ *as proposed by Dutang (2017) until convergence:*

$$\widehat{\boldsymbol{\beta}}^{(t+1)} = \left(Z^T W^{(t)} Z\right)^{-1} Z^T W^{(t)} \widehat{\mathbf{s}}^{(t)}\left(\widehat{\boldsymbol{\beta}}^{(t)}\right), \tag{18}$$

$$\widehat{\mathbf{s}}^{(t)}\left(\widehat{\boldsymbol{\beta}}^{(t)}\right) = Z\widehat{\boldsymbol{\beta}}^{(t)} + D^{(t)}(y - \widehat{\mu}_t) \tag{19}$$

After convergence, we set $\widehat{\boldsymbol{\beta}}_{\text{GLM}} = \widehat{\boldsymbol{\beta}}^{(t+1)}$.

Green (1984) proposes to solve the system $\left(Z^T W^{(t)} Z\right)\widehat{\boldsymbol{\beta}}^{(t+1)} = Z^T W^{(t)} \widehat{\mathbf{s}}^{(t)}$ which is equivalent to (18) via a QR decomposition to increase numerical stability. For a practical implementation of GLMs using the IRLS algorithm, see, for example, function *glm(·)* in R package *stats* of R Core Team (2018).

By inserting (17), (19) and the GLM estimator into (18) and by using (12), we obtain

$$\widehat{\boldsymbol{\beta}}_{\text{GLM}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^K}{\arg\min} \left\{ \sum_{i=1}^N V\left[\widehat{\mu}^i_{\text{GLM}}\right]\left(y^i - \widehat{\mu}^i_{\text{GLM}}\right)^2 \right\}, \tag{20}$$

that is, the GLM estimator minimizes the squared sum of raw residuals scaled by the estimated individual variances of the economic variable.

The Pearson residuals are defined as the raw residuals divided by the estimated individual standard deviations, that is,

$$\widehat{\epsilon}^i = \frac{y^i - \widehat{\mu}^i_{\text{GLM}}}{\sqrt{V\left[\widehat{\mu}^i_{\text{GLM}}\right]}}. \tag{21}$$

3.3.3. AIC and Dispersion Estimation

Since AIC depends on the ML estimators, it is combinable with GLMs in the adaptive algorithm. Here, it has the form

$$\text{AIC} = -2l\left(\widehat{\boldsymbol{\beta}}_{\text{GLM}}, \widehat{\varphi}\right) + 2\left(K + p\right), \tag{22}$$

where K is the number of coefficients and p indicates the number of the additional model parameters associated with the distribution of the random component. For instance, in the normal model, we have $p = 1$ due to the error variance/dispersion. A typical estimate of the dispersion in GLMs is the Pearson residual chi-squared statistic divided by $N-K$ as described by Zuur et al. (2009) and implemented, for example, in function *glm(·)* belonging to R package *stats*, that is,

$$\widehat{\phi} = \frac{1}{N-K}\sum_{i=1}^{N}\left(\widehat{\epsilon}^{i}\right)^{2}, \tag{23}$$

with $\widehat{\epsilon}^{i}$ given by (21). Even though this is not the ML estimator, it is a good estimate because, if the model is specified correctly, the Pearson residual chi-squared statistic divided by the dispersion is asymptotically χ^2_{N-K} distributed and the expected value of a chi-squared distribution with $N-K$ degrees of freedom is $N-K$.

3.4. Generalized Additive Models (GAMs)

3.4.1. The Regression Model

Generalized additive models (GAMs) as introduced by Hastie and Tibshirani (1986) and Hastie and Tibshirani (1990) can be regarded as richly parameterized GLMs with smooth functions. While GAMs inherit from GLMs the random component (13) and the link function (12), they inherit from the additive models of Friedman and Stuetzle (1981) the linear predictor with the smooth functions. In the adaptive algorithm, we apply GAMs of the form

$$\underbrace{g(Y(X))}_{=\mu} \overset{K<\infty}{\approx} \underbrace{f(X)}_{=\eta} = \beta_0 + \sum_{k=1}^{K-1} h_k\left(z_k\right), \tag{24}$$

where $z_k = e_k\left(X\right)$, β_0 is the intercept and $h_k\left(\cdot\right),\ k = 1, \ldots, K-1$, are the smooth functions to be estimated. In addition to the smooth functions, GAMs can also include simple linear terms of the basis functions as they appear in the linear predictor of GLMs. A smooth function $h_k\left(\cdot\right)$ can be written as a basis expansion

$$h_k\left(z_k\right) = \sum_{j=1}^{J}\beta_{kj}b_{kj}\left(z_k\right), \tag{25}$$

with coefficients β_{kj} and known basis functions $b_{kj}\left(z_k\right),\ j = 1, \ldots, J$, which should not be confused with their arguments, namely the first-order basis functions $z_k = e_k\left(X\right),\ k = 0, \ldots, K-1$. The slightly adapted Figure 4 from Wood (2006) depicts an exemplary approximation of y by a GAM with a basis expansion in one dimension z_k without an intercept. The solid colorful curves represent the pure basis functions $b_{kj}\left(z_k\right),\ j = 1, \ldots, J$, the dashed colorful curves show them after scaling with the coefficients $\beta_{kj}b_{kj}\left(z_k\right),\ j = 1, \ldots, J$, and the black curve is their sum (25).

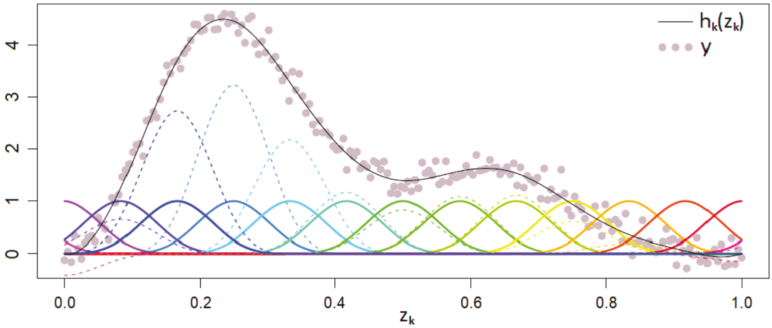


Figure 4. Generalized additive model (GAM) with a basis expansion in one dimension.

Typical examples for basis functions are thin plate regression splines, duchon splines, cubic regression splines or Eilers and Marx style P-splines. See, for example, function *gam(·)* in R package *mgcv* of Wood (2018) for a practical implementation of GAMs admitting these types of basis functions and using the PIRLS algorithm, which we present below.

In vector notation, we can write $\boldsymbol{\beta} = \left(\beta_0, \boldsymbol{\beta}_1^T, \ldots, \boldsymbol{\beta}_{K-1}^T\right)^T$ with $\boldsymbol{\beta}_k = \left(\beta_{k1}, \ldots, \beta_{kJ}\right)^T$ and $\mathbf{a} = \left(1, \mathbf{b}_1\left(z_1\right)^T, \ldots, \mathbf{b}_{K-1}\left(z_{K-1}\right)^T\right)^T$ with $\mathbf{b}_k\left(z_k\right) = \left(b_{k1}\left(z_k\right), \ldots, b_{kJ}\left(z_k\right)\right)^T$, hence (24) becomes

$$g(\underbrace{Y(X)}_{=\mu}) \overset{K<\infty}{\approx} \underbrace{f(X)}_{=\eta} = \mathbf{a}^T\boldsymbol{\beta}. \tag{26}$$

In order to make the smooth functions $h_k\left(\cdot\right), k = 1, \ldots, K-1$, identifiable, identifiability constraints $\sum_{i=1}^N h_k\left(z_{ik}\right) = 0$ with $z_{ik} = e_k\left(x^i\right)$ can be imposed. According to Wood (2006) this can be achieved by modification of the basis functions $b_{kj}\left(\cdot\right)$ with one of them being lost.

3.4.2. Penalization and GAM Estimation via PIRLS Algorithm

Let the deviance corresponding to observation y^i be $D^i\left(\boldsymbol{\beta}\right) = 2\left(l_{\text{sat}}^i - l^i\left(\boldsymbol{\beta}, \phi\right)\right)\phi$ where $D^i\left(\boldsymbol{\beta}\right)$ is independent of dispersion ϕ, where $l_{\text{sat}}^i = \max_{\boldsymbol{\beta}^i} l^i\left(\boldsymbol{\beta}^i, \phi\right)$ is the saturated log-likelihood and $l^i\left(\boldsymbol{\beta}, \phi\right)$ the log-likelihood. Then the model deviance can be written as $D\left(\boldsymbol{\beta}\right) = \sum_{i=1}^N D^i\left(\boldsymbol{\beta}\right)$. It is a generalization of the residual sum of squares for ML estimation. For instance, in the normal model the unit deviance is $\left(y^i - \mu^i\right)^2$. For given smoothing parameters $\lambda_k > 0$, $k = 1, \ldots, K-1$, the GAM estimator $\widehat{\boldsymbol{\beta}}_{\text{GAM}}$ of the coefficients is defined as the minimizer of the penalized deviance

$$\widehat{\boldsymbol{\beta}}_{\text{GAM}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{(K-1)J+1}}{\arg\min} \left\{ D\left(\boldsymbol{\beta}\right) + \sum_{k=1}^{K-1} \lambda_k \int h_k''\left(z_k\right)^2 \mathrm{d}z_k \right\}, \text{where} \tag{27}$$

$$\int h_k''\left(z_k\right)^2 \mathrm{d}z_k = \boldsymbol{\beta}_k^T \left(\int \mathbf{b}_k''\left(z_k\right) \mathbf{b}_k''\left(z_k\right)^T \mathrm{d}z_k \right) \boldsymbol{\beta}_k = \boldsymbol{\beta}_k^T \mathcal{S}_k \boldsymbol{\beta}_k$$

are the smoothing penalties. The smoothing parameters λ_k control the trade-off between a too wiggly model (overfitting) and a too smooth model (underfitting). The larger the λ_k values are, the more pronounced is the wiggliness of the basis functions reflected by their second derivatives in the minimization problem (27), and the higher is thus the penalty associated with the coefficients and the smoother is the estimated model.

A major advantage of the definition of GAMs via (24), (25), and (27) is its compatibility with information criteria and other model selection criteria such as generalized cross-validation. Besides, the resulting penalty matrix favors numerical stability in the PIRLS algorithm.

Since the saturated log-likelihood is a constant for a fixed distribution and set of fitting points, we can turn the minimization problem (27) into the maximization task of the penalized log-likelihood, that is,

$$\widehat{\boldsymbol{\beta}}_{\text{GAM}} = \underset{\boldsymbol{\beta} \in \mathbb{R}^{(K-1)J+1}}{\arg\max} \left\{ l\left(\boldsymbol{\beta}, \phi\right) - \frac{1}{2} \sum_{k=1}^{K-1} \lambda_k \boldsymbol{\beta}_k^T \mathcal{S}_k \boldsymbol{\beta}_k \right\}. \tag{28}$$

Wood (2000) points out that Fisher's scoring method can be cast in a penalized version of the iteratively reweighted least squares (PIRLS) algorithm when being used to approximate the GAM coefficient estimator (28). We formulate the PIRLS algorithm based on Marx and Eilers (1998) who indicate the iterative solution explicitly.

Let $\widehat{\boldsymbol{\beta}}^{(t)}$ now be the GAM coefficient approximation in iteration t. Then the vector of the dependent variable $\widehat{\mathbf{s}}^{(t)} = \left(\widehat{s}^1\left(\widehat{\boldsymbol{\beta}}^{(t)}\right), \ldots, \widehat{s}^N\left(\widehat{\boldsymbol{\beta}}^{(t)}\right)\right)^T$ and the weight matrix given by $W^{(t)} =$

$\operatorname{diag}\left(w^1\left(\widehat{\boldsymbol{\beta}}^{(t)}\right), \ldots, w^N\left(\widehat{\boldsymbol{\beta}}^{(t)}\right)\right)$ have the same form as in the IRLS algorithm, see (19) and (17). Additionally, let $S = \operatorname{blockdiag}\left(0, \lambda_1 \mathcal{S}_1, \ldots, \lambda_{K-1}\mathcal{S}_{K-1}\right)$ with $S_{11} = 0$ belonging to the intercept be the penalty matrix.

PIRLS algorithm. *Perform the iterative approximation procedure below with initialization of* $\widehat{\mu}^i_{(0)} = y^i + 0.1$ *and* $\widehat{\eta}^i_{(0)} = g\left(\widehat{\mu}^i_{(0)}\right)$ *until convergence occurs:*

$$\begin{aligned}\widehat{\boldsymbol{\beta}}^{(t+1)} &= \underset{\beta \in \mathbb{R}^{(K-1)J+1}}{\arg\min} \left\{ \sum_{i=1}^{N} w^i\left(\widehat{\boldsymbol{\beta}}^{(t)}\right)^{-1} \left(\widehat{s}^i\left(\widehat{\boldsymbol{\beta}}^{(t)}\right) - \beta_0 - \sum_{k=1}^{K-1}\sum_{j=1}^{J} \beta_{kj} b_{kj}\left(z_{ik}\right)\right)^2 + \sum_{k=1}^{K-1} \lambda_k \boldsymbol{\beta}_k^T \mathcal{S}_k \boldsymbol{\beta}_k \right\} \\ &= \left(Z^T W^{(t)} Z + S\right)^{-1} Z^T W^{(t)} \widehat{\mathbf{s}}^{(t)}. \end{aligned} \tag{29}$$

After convergence, we set $\widehat{\boldsymbol{\beta}}_{\mathrm{GAM}} = \widehat{\boldsymbol{\beta}}^{(t+1)}$.

3.4.3. Smoothing Parameter Selection, AIC and Stagewise Selection

The smoothing parameters λ_k can be selected such that they minimize a suitable model selection criterion, for the sake of consistency, preferably the one used in the adaptive algorithm for basis function selection. The GAM estimator (28) does not exactly maximize the log-likelihood, therefore AIC has another form for GAMs than for GLMs. Hastie and Tibshirani (1990) propose a widely used version of AIC for GAMs, which uses effective degrees of freedom df in place of the number of coefficients $(K-1)J+1$. This is

$$\mathrm{AIC} = -2l\left(\widehat{\boldsymbol{\beta}}_{\mathrm{GAM}}, \widehat{\phi}\right) + 2\left(\mathrm{df} + p\right), \tag{30}$$

where

$$\mathrm{df} = \operatorname{tr}\left(\left(I + S\right)^{-1} I\right). \tag{31}$$

Note that $I + S = Z^T W Z + S$ is already approximately calculated in the PIRLS algorithm. For GAMs, an estimate of the dispersion $\widehat{\phi}$ is obtained similarly to GLMs by (23). The parameter p is defined as in (22).

Another popular and effective smoothing parameter selection criterion invented by Craven and Wahba (1979) is generalized cross-validation (GCV), that is,

$$\mathrm{GCV} = \frac{N D\left(\widehat{\boldsymbol{\beta}}_{\mathrm{GAM}}\right)}{\left(N - \mathrm{df}\right)^2}, \tag{32}$$

with the model deviance $D\left(\widehat{\boldsymbol{\beta}}_{\mathrm{GAM}}\right)$ evaluated at the GAM estimator and the effective degrees of freedom defined just like for AIC.

Note that the adaptive forward stepwise algorithm depicted in Figure 2 can become computationally infeasible with GAMs as opposed to, for example, GLMs. In iteration k, a GAM has $(K-1)J+1$ coefficients which need to be estimated while a GLM has only K coefficients. This difference in the estimation effort is increased further due to the iterative nature of the IRLS and PIRLS algorithms. Moreover, GAMs involve the task of optimal smoothing parameter selection. To deal with this aspect, Wood (2000), Wood et al. (2015) and Wood et al. (2017) have developed practical GAM fitting methods for large data sets. However, the suitable application of these methods in the adaptive algorithm is beyond the scope of our analysis, in particular as our focus is not on computational performance. Besides parallelizing the candidate loop on the lower left side of Figure 2, we achieve the necessary performance gains in GAMs by replacing the stepwise algorithm by a stagewise algorithm. This means that in each iteration, a predefined number L or proportion of candidate basis functions is selected simultaneously until a termination criterion is fulfilled. Thereby we select in one stage those basis functions which reduce the model selection criterion of our choice most when added separately to the current proxy function structure. When there are not at least as many basis functions as targeted, the algorithm shall be terminated after the ones which lead to a reduction in the model selection criterion have been selected.

3.5. Feasible Generalized Least-Squares (FGLS) Regression

3.5.1. The Regression Model

The regression model here equals the OLS case. However, we now let the errors have the covariance matrix $\Sigma = \sigma^2\Omega$ where Ω is positive definite and known and $\sigma^2 > 0$ is unknown. We transform the generalized regression model according to Hayashi (2000) to obtain a model (*) which satisfies Assumptions (A1), (A2) and (A3) of the classical linear regression model. For this, choose an invertible matrix H with $\Omega^{-1} = H^T H$ which can, for example, be the Cholesky matrix. Then, the generalized response vector $\mathbf{y}^*$, design matrix Z^* and error vector $\boldsymbol{\epsilon}^*$ are given by

$$\mathbf{y}^* = H\mathbf{y}, \quad Z^* = HZ, \quad \boldsymbol{\epsilon}^* = \mathbf{y}^* - Z^*\boldsymbol{\beta} = H\left(\mathbf{y} - Z\boldsymbol{\beta}\right) = H\boldsymbol{\epsilon}. \tag{33}$$

In analogy to the OLS estimator, the generalized least-squares (GLS) estimator $\widehat{\boldsymbol{\beta}}_{\mathrm{GLS}}$ of the coefficients is given as the minimizer of the generalized residual sum of squares, that is,

$$\widehat{\boldsymbol{\beta}}_{\mathrm{GLS}} = \underset{\boldsymbol{\beta}\in\mathbb{R}^K}{\arg\min}\left\{\sum_{i=1}^{N}\left(\epsilon^{*,i}\right)^2\right\}. \tag{34}$$

The closed-form expression of the GLS estimator is

$$\widehat{\boldsymbol{\beta}}_{\mathrm{GLS}} = \left(Z^{*T}Z^*\right)^{-1} Z^{*T}\mathbf{y}^* = \left(Z^T\Omega^{-1}Z\right)^{-1} Z^T\Omega^{-1}\mathbf{y}, \tag{35}$$

and the proxy function becomes

$$\widehat{f}\left(X\right) = \mathbf{z}^T\widehat{\boldsymbol{\beta}}_{\mathrm{GLS}}, \tag{36}$$

where $\mathbf{z} = \left(e_0\left(X\right), \ldots, e_{K-1}\left(X\right)\right)^T$. The scalar σ^2 can be estimated in analogy to OLS regression by $s_{\mathrm{GLS}} = \frac{1}{N-K}\widehat{\boldsymbol{\epsilon}}^{*T}\widehat{\boldsymbol{\epsilon}}^*$ where $\widehat{\boldsymbol{\epsilon}}^* = \mathbf{y}^* - Z^*\widehat{\boldsymbol{\beta}}_{\mathrm{GLS}}$ is the residual vector.

3.5.2. Gauss-Markov-Aitken Theorem and ML Estimation

Under the assumptions (A1), (A3), and a covariance matrix $\Sigma = \sigma^2\Omega$ of which Ω is positive definite and known (A6), we have:

- The GLS estimator is the BLUE of the coefficients in the generalized regression model (7) (Gauss-Markov-Aitken theorem).
- If in addition we have jointly normally distributed errors conditional on the fitting scenarios (A7) then the ML coefficient estimator coincides with the GLS estimator. Further, the ML estimator of the scalar $\widehat{\sigma}^2$ can be expressed as $\frac{N}{N-K}$ times s_{GLS}.

As a consequence, given a known matrix Ω, we have a closed form solution for the GLS estimator that coincides with the ML estimator of the regression coefficients and the adaptive algorithm inside the LSMC approach goes through.

3.5.3. Unknown Ω and FGLS Estimation via ML Algorithm

In the LSMC framework, Ω is unknown. However, if a consistent estimator $\widehat{\Omega}$ exists, we can apply feasible generalized least-squares (FGLS) regression, of which the estimator

$$\widehat{\boldsymbol{\beta}}_{\mathrm{FGLS}} = \left(Z^T\widehat{\Omega}^{-1}Z\right)^{-1} Z^T\widehat{\Omega}^{-1}\mathbf{y} \tag{37}$$

has asymptotically the same properties as the GLS estimator (35).

With $\mathbf{z} = \left(e_0\left(X\right), \ldots, e_{K-1}\left(X\right)\right)^T$ the FGLS proxy function is then given as

$$\widehat{f}\left(X\right) = \mathbf{z}^T \widehat{\boldsymbol{\beta}}_{\text{FGLS}}. \tag{38}$$

For the estimation of Ω we will in the following set $\sigma^2 = 1$ which can be done without loss of generality and consider $\Sigma = \Omega$. Furthermore, we assume in addition to (A1), (A3) and (A7) that the elements of the covariance matrix Σ are twice differentiable functions of parameters $\boldsymbol{\alpha} = \left(\alpha_0, \ldots, \alpha_{M-1}\right)^T$ with $K + M \leq N$. We then write $\Sigma = \Sigma\left(\boldsymbol{\alpha}\right)$ (A8). The following result is the basis of the iterative ML algorithm for the regression coefficients and the variance matrix.

Theorem 1. *The generalized regression model (7) under Assumptions (A1), (A3), (A7) and (A8) has the following first-order ML conditions:*

$$\widehat{\boldsymbol{\beta}}_{\text{ML}} = \left(Z^T \widehat{\Sigma}^{-1} Z\right)^{-1} Z^T \widehat{\Sigma}^{-1} \mathbf{y}, \tag{39}$$

$$\frac{\partial l}{\partial \alpha_m} = \frac{1}{2} \text{tr}\left(\frac{\partial \Sigma^{-1}}{\partial \alpha_m} \Sigma\right)_{\boldsymbol{\alpha} = \widehat{\boldsymbol{\alpha}}_{\text{ML}}} - \frac{1}{2} \widehat{\boldsymbol{\epsilon}}^T \left(\frac{\partial \Sigma^{-1}}{\partial \alpha_m}\right)_{\boldsymbol{\alpha} = \widehat{\boldsymbol{\alpha}}_{\text{ML}}} \widehat{\boldsymbol{\epsilon}} = 0, \tag{40}$$

where $m = 0, \ldots, M-1$, $\widehat{\Sigma} = \Sigma\left(\widehat{\boldsymbol{\alpha}}_{\text{ML}}\right)$ *and* $\widehat{\boldsymbol{\epsilon}} = \mathbf{y} - Z\widehat{\boldsymbol{\beta}}_{\text{ML}}$.

The system in (39) and (40) is then solved iteratively (see, for example, Magnus (1978)). We start the procedure with $\boldsymbol{\beta}^{(0)}$ and then use PORT optimization routines as described in Gay (1990) and implemented in function *nlminb(·)* belonging to R package *stats* of R Core Team (2018). In this iterative routine, $\widehat{\boldsymbol{\alpha}}^{(t+1)}$ can be initialized, for example, by random numbers from the standard normal distribution.

ML algorithm. *Perform the following iterative approximation procedure with, for example, an initialization of* $\widehat{\boldsymbol{\beta}}^{(0)} = \widehat{\boldsymbol{\beta}}_{\text{OLS}}$ *until convergence:*

1. *Calculate the residual vector* $\widehat{\boldsymbol{\epsilon}}^{(t+1)} = \mathbf{y} - Z\widehat{\boldsymbol{\beta}}^{(t)}$.
2. *Substitute* $\widehat{\boldsymbol{\epsilon}}^{(t+1)}$ *into the M equations in M unknowns* α_m *given by (40) and solve them. If an explicit solution exists, set* $\widehat{\boldsymbol{\alpha}}^{(t+1)} = \boldsymbol{\alpha}\left(\widehat{\boldsymbol{\epsilon}}^{(t+1)}\right)$. *Otherwise, select the maximum likelihood solution* $\widehat{\boldsymbol{\alpha}}^{(t+1)}$ *iteratively, for example, by using PORT optimization routines.*
3. *Calculate*

$$\widehat{\Sigma}^{(t+1)} = \Sigma\left(\widehat{\boldsymbol{\alpha}}^{(t+1)}\right),$$

$$\widehat{\boldsymbol{\beta}}^{(t+1)} = \left(Z^T \left(\widehat{\Sigma}^{(t+1)}\right)^{-1} Z\right)^{-1} Z^T \left(\widehat{\Sigma}^{(t+1)}\right)^{-1} \mathbf{y}. \tag{41}$$

Continue with the next iteration.

After convergence, we set $\widehat{\boldsymbol{\beta}}_{\text{ML}} = \widehat{\boldsymbol{\beta}}^{(t+1)}$ *and* $\widehat{\boldsymbol{\alpha}}_{\text{ML}} = \widehat{\boldsymbol{\alpha}}^{(t+1)}$.

Theorem 5 of Magnus (1978) states that under some further regularity conditions the FGLS coefficient estimator can be derived as the ML coefficient estimator by the ML algorithm under Assumptions (A1), (A3), (A7) and (A8).

3.5.4. Heteroscedasticity, Variance Model Selection and AIC

Besides Assumption (A8) about the structure of the covariance matrix, we assume that the errors are uncorrelated with possibly different variances (= heteroscedastic errors), that is, $\Sigma = \text{diag}\left(\sigma_1^2, \ldots, \sigma_N^2\right)$. We model each variance σ_i^2, $i = 1, \ldots, N$, by a twice differentiable function in dependence of parameters $\boldsymbol{\alpha} = (\alpha_0, \ldots, \alpha_{M-1})^T$ and a suitable set of linearly independent basis functions $e_m(X) \in L^2(\mathbb{R}^D, \mathcal{B}, \mathbb{P}')$, $m = 0, 1, \ldots, M-1$, with $\mathbf{v}^i = \left(e_0\left(x^i\right), \ldots, e_{M-1}\left(x^i\right)\right)^T$, that is,

$$\sigma_i^2 = \sigma^2 V\left[\boldsymbol{\alpha}, \mathbf{v}^i\right], \tag{42}$$

where $V\left[\boldsymbol{\alpha}, \mathbf{v}^i\right]$ is referred to as the variance function in analogy to $V[\mu]$ for GLMs and GAMs. Without loss of generality, we set again $\sigma^2 = 1$.

Hartmann (2015) has already applied FGLS regression with different variance models in the LSMC framework. In her numerical examples, variance models with multiplicative heteroscedasticity led to the best performance of the proxy function in the validation. Therefore, we restrict our analyis on these kinds of structures, compare, for example, Harvey (1976), that is,

$$V\left[\boldsymbol{\alpha}, \mathbf{v}^i\right] = \exp\left(\mathbf{v}^{iT}\boldsymbol{\alpha}\right). \tag{43}$$

Like the proxy function, the variance function (43) has to be calibrated to apply FGLS regression, which means that the variance function has to be composed of suitable basis functions. Again, such a composition can be found with the aid of a model selection criterion. We still choose AIC, but have to take care for the fact that in FGLS regression the covariance matrix now contains M unknown parameters instead of only one in the OLS case (the same variance for all observations). Under Assumption (A7), AIC is given as

$$\begin{aligned} \text{AIC} &= -2l\left(\widehat{\boldsymbol{\beta}}_{\text{FGLS}}, \widehat{\Sigma}\right) + 2(K+M) \\ &= N\log(2\pi) + \log\left(\det\widehat{\Sigma}\right) + \left(\mathbf{y} - Z\widehat{\boldsymbol{\beta}}_{\text{FGLS}}\right)^T \widehat{\Sigma}^{-1}\left(\mathbf{y} - Z\widehat{\boldsymbol{\beta}}_{\text{FGLS}}\right) + 2(K+M). \end{aligned} \tag{44}$$

When using a variance model with multiplicative heteroscedasticity, AIC becomes

$$\text{AIC} = N\log(2\pi) + \left(\sum_{i=1}^{N} \mathbf{v}^{iT}\right)\widehat{\boldsymbol{\alpha}} + \sum_{i=1}^{N} \exp\left(-\mathbf{v}^{iT}\widehat{\boldsymbol{\alpha}}\right)\left(\widehat{\epsilon}^i\right)^2 + 2(K+M). \tag{45}$$

As an alternative or complement, the basis functions of the variance model can be selected with respect to their correlations with the final OLS residuals or based on graphical residual analysis.

For the final implementation of a variance model we use modified versions of two algorithms from Hartmann (2015). Our type I variant starts with the derivation of the proxy function by the standard adaptive OLS regression approach and then selects the variance model adaptively from the set of proxy basis functions of which the exponents sum up to at most two. The type II variant builds on the type I algorithm by taking the resulting variance model as given in its adaptive proxy basis function selection procedure with FGLS regression in each iteration.

Note further, that we should only apply FGLS regression as a substitute of OLS regression if heteroscedasticity prevails. This can be tested with the help of the Breusch-Pagan test of Breusch and Pagan (1979) for the following special structure of the variance function

$$V\left[\boldsymbol{\alpha}, \mathbf{v}^i\right] = h\left(\mathbf{v}^{i,T}\boldsymbol{\alpha}\right), \tag{46}$$

where the function $h(\cdot)$ is twice differentiable and the first element of $\mathbf{v}^i$ is $\text{v}_0^i = 1$. Further, the assumption of normally distributed errors is made. We use it in the numerical computations to check if heteroscedasticity still prevails during the iteration procedure.

3.6. Multivariate Adaptive Regression Splines (MARS)

3.6.1. The Regression Model

The multivariate adaptive regression splines (MARS) were introduced by Friedman (1991). The classical MARS model is a form of the classical linear regression model (7) where the basis functions $e_k\left(x^i\right)$ are so-called hinge functions. Therefore, the theory of OLS regression applies in this context. GLMs (12) can also be applied in conjunction with MARS models. In this case we speak of generalized MARS models.

We describe the standard MARS algorithm in the LSMC routine according to Chapter 9.4 of Hastie et al. (2017). The building blocks of MARS proxy functions are reflected pairs of piecewise linear functions with knots t as depicted in Figure 5, that is,

$$(X_d - t)_+ = \max\left(X_d - t, 0\right), \quad (t - X_d)_+ = \max\left(t - X_d, 0\right), \tag{47}$$

where the X_d, $d = 1, \ldots, D$, represent the risk factors that together form the outer scenario $X = \left(X_1, \ldots, X_D\right)^T$.

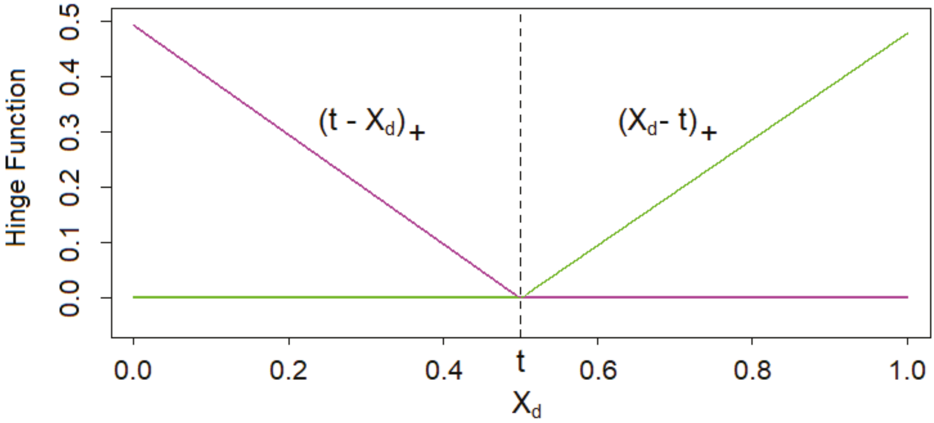


Figure 5. Reflected pair of piecewise linear functions with a knot at t.

For each risk factor, reflected pairs with knots at each fitting scenario stress x_d^i, $i = 1, \ldots, N$, are defined. All pairs are united in the following collection serving as the initial candidate basis function set of the MARS algorithm, that is,

$$C_1 = \left\{(X_d - t)_+, (t - X_d)_+\right\}_{t \in \left\{x_d^1, x_d^2, \ldots, x_d^N\right\} \,\mid\, d=1,\ldots,D}. \tag{48}$$

We call the elements of C_1 hinge functions and consider them as functions $h\left(X\right)$ over the entire input space $\mathbb{R}^D$. C_1 contains in total $2DN$ basis functions.

The adaptive basis function selection algorithm now consists of two parts, the forward and the backward pass.

3.6.2. Adaptive Forward Stepwise Selection and Forward Pass

The forward pass of the MARS algorithm can be viewed as a variation of the adaptive forward stepwise algorithm depicted in Figure 2. The start proxy function consists only of the intercept, that is, $h_0\left(X\right) = 1$. In the classical MARS model, the regression method of choice is the standard OLS regression approach with the estimator (8), where in each iteration a reflected pair of hinge functions is selected instead of $e_k\left(x^i\right)$. Similarly, the regression method of choice in the generalized MARS model is the IRLS algorithm (18). Let us denote the MARS coefficient estimator by $\widehat{\beta}_{\text{MARS}}$. Note that the theory on AIC cannot be transferred without any adjustments since the notion of the degrees of freedom has to be reconsidered due to the knots in the hinge functions acting as additional degrees of freedom.

After each iteration, the set of candidate basis functions is extended by the products of the last two selected hinge functions with all hinge functions in C_1 that depend on risk factors of which the last two selected hinge functions do not depend on. Let the reflected pair selected in the first iteration ($k = 1$) be

$$\begin{aligned} h_1(X) &= \left(X_{d_1} - t_1\right)_+, \\ h_2(X) &= \left(t_1 - X_{d_1}\right)_+. \end{aligned} \tag{49}$$

Further, let $C_{1,-} = C_1 \setminus \{h_1(X), h_2(X)\}$. Then, the set of candidate basis functions is updated at the beginning of the second iteration ($k = 2$) such that

$$\begin{aligned} C_2 = C_{1,-} &\cup \left\{(X_d - t)_+ h_1(X), (t - X_d)_+ h_1(X)\right\}_{t \in \left\{x_d^1, x_d^2, \ldots, x_d^N\right\} \mid d=1,\ldots,D,\ d \neq d_1} \\ &\cup \left\{(X_d - t)_+ h_2(X), (t - X_d)_+ h_2(X)\right\}_{t \in \left\{x_d^1, x_d^2, \ldots, x_d^N\right\} \mid d=1,\ldots,D,\ d \neq d_1}. \end{aligned} \tag{50}$$

The second set C_2 thus contains $2(DN - 1) + 4(D - 1)N$ basis functions. Often, the order of interaction is limited to improve the interpretability of the proxy functions. Besides the maximum allowed number of terms, a minimum threshold for the decrease in the residual sum of squares can be employed as a termination criterion in the forward pass. Typically, the proxy functions generated in the forward pass overfit the data since model complexity is only penalized conservatively by stipulating a maximum number of basis functions and a minimum threshold.

3.6.3. Backward Pass and GCV

Due to the overfitting tendency of the proxy function generated in the forward pass, a backward pass is executed afterwards. Apart from the direction and slight differences, the backward pass is similar to the forward pass. In each iteration, the hinge function of which the removal causes the smallest increase in the residual sum of squares is removed and the backward model selection criterion for the resulting proxy function is evaluated. By this backward procedure, we generate the "best" proxy functions of each size in terms of the residual sum of squares. Out of all these best proxy functions, we finally select the one which minimizes the backward model selection criterion. As a result, the final proxy function will not only contain reflected pairs of hinge functions but also single hinge functions of which the complements have been removed. Optionally, the backward pass can also be omitted.

Let the number of basis functions in the MARS model be K and the number of knots be T. The standard choice for the backward model selection criterion is GCV defined as

$$\text{GCV} = \frac{ND\left(\widehat{\beta}_{\text{MARS}}\right)}{(N - \text{df})^2}, \tag{51}$$

with the effective degrees of freedom $\text{df} = K + 3T$.

An especially fast MARS algorithm was later developed by Friedman (1993) and is implemented, for example, in function *earth(·)* of R package *earth* provided by Milborrow (2018).

3.7. Kernel Regression

3.7.1. The One-dimensional Regression Model

Kernel regression (which goes back to Nadaraya (1964) and Watson (1964)) is a type of locally weighted OLS regression where the weights vary with the input variable (*the target scenario*). We start with locally constant (LC) regression where for each $x_0 \in \mathbb{R}$ the fixed *univariate kernel* with given *bandwidth* $\lambda > 0$ be

$$K_\lambda\left(x_0, x^i\right) = D\left(\frac{\left|x^i - x_0\right|}{\lambda}\right), \tag{52}$$

where $D(\cdot)$ denotes the specified kernel function. Solving the corresponding least squares problem

$$\widehat{\boldsymbol{\beta}}_{\mathrm{LC}}\left(x_0\right) = \underset{\boldsymbol{\beta}(x_0)\in\mathbb{R}}{\arg\min}\left\{\sum_{i=1}^{N} K_\lambda\left(x_0, x^i\right)\left(y^i - \beta_0\left(x_0\right)\right)^2\right\}, \tag{53}$$

one obtains the Nadaraya-Watson kernel smoother as the kernel-weighted average at each x_0 over the fitting values y^i, that is,

$$\widehat{f}_{\mathrm{LC}}\left(x_0\right) = \widehat{\beta}_{\mathrm{LC}}\left(x_0\right) = \frac{\sum_{i=1}^{N} K_\lambda\left(x_0, x^i\right) y^i}{\sum_{i=1}^{N} K_\lambda\left(x_0, x^i\right)}. \tag{54}$$

Typical examples for the fixed kernel are the Epanechnikov (see the green shaded areas of Figure 6 inspired by Hastie et al. (2017)), tri-cube and uniform kernels or gaussian kernel. Note that a kernel smoother is continuous and varies over the domain of the target scenarios x_0, it needs to be estimated separately at all of them.

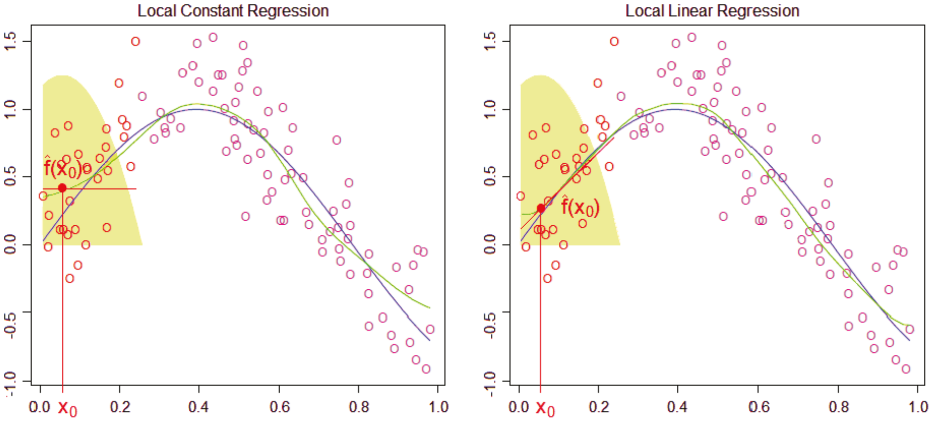


Figure 6. Locally constant (LC) and LL kernel regression using the Epanechnikov kernel with $\lambda = 0.2$ in one dimension.

The bias at the boundaries of the domain of the LC kernel estimator (53) (see the left panel of Figure 6) is mainly eliminated by fitting locally linear functions instead of locally constant functions, see the right panel of Figure 6. At each target x_0, the LL kernel estimator is defined as the minimizer of the kernel-weighted residual sum of squares, that is,

$$\widehat{\boldsymbol{\beta}}_{\mathrm{LL}}\left(x_0\right) = \underset{\boldsymbol{\beta}(x_0)\in\mathbb{R}^2}{\arg\min}\left\{\sum_{i=1}^{N} K_\lambda\left(x_0, x^i\right)\left(y^i - \beta_0\left(x_0\right) - \beta_1\left(x_0\right) x^i\right)^2\right\}, \tag{55}$$

with $\boldsymbol{\beta}\left(x_0\right) = \left(\beta_0\left(x_0\right), \beta_1\left(x_0\right)\right)^T$. The proxy function at x_0 is given by

$$\widehat{f}_{\mathrm{LL}}\left(x_0\right) = \widehat{\beta}_{\mathrm{LL},0}\left(x_0\right) + \widehat{\beta}_{\mathrm{LL},1}\left(x_0\right) x_0. \tag{56}$$

Again the minimization problem (55) must be solved separately for all target scenarios so that the coefficients of the proxy function vary across their domain. For each target scenario x_0 a weighted least-squares (WLS) problem with weights $K_\lambda\left(x_0, x^i\right)$ has to be solved. Its solution is the WLS estimator

$$\widehat{\boldsymbol{\beta}}_{\mathrm{LL}}\left(x_0\right) = \left(Z^T W\left(x_0\right) Z\right)^{-1} Z^T W\left(x_0\right) \mathbf{y}, \tag{57}$$

with $\mathbf{y}$ the response vector, $W\left(x_0\right) = \operatorname{diag}\left(K_\lambda\left(x_0, x^1\right), \ldots, K_\lambda\left(x_0, x^N\right)\right)$ the weight matrix and Z the design matrix which contains row-wise the vectors $\left(1, x^i\right)^T$. We call H the hat matrix if $\widehat{\mathbf{y}} = H\mathbf{y}$ such that $\widehat{\mathbf{y}} = \left(\widehat{f}_{\mathrm{LL}}\left(x^1\right), \ldots, \widehat{f}_{\mathrm{LL}}\left(x^N\right)\right)^T$ contains the proxy function values at their target scenarios.

When we use proxy functions in LL regression that are composed of polynomial basis functions with exponents greater than one, we could also speak of local polynomial regression.

3.7.2. The Multidimensional Regression Model

We generalize LC regression to $\mathbb{R}^K$ by expressing the kernel with respect to the basis function vector $\mathbf{z} = \left(e_0\left(X\right), \ldots, e_{K-1}\left(X\right)\right)^T$ following from the adaptive forward stepwise selection with OLS regression and small $K_{\max}$. At each target scenario vector $\mathbf{z}_0 \in \mathbb{R}^K$ with elements z_{0k}, basis function vector $\mathbf{z}^i \in \mathbb{R}^K$ with elements z_{ik} evaluated at fitting scenario x^i and given bandwidth vector $\boldsymbol{\lambda} = \left(\lambda_0, \ldots, \lambda_{K-1}\right)^T$, the multivariate kernel is defined as the product of univariate kernels, that is,

$$K_{\boldsymbol{\lambda}}\left(\mathbf{z}_0, \mathbf{z}^i\right) = \prod_{k=0}^{K-1} D\left(\frac{\left|z_{ik} - z_{0k}\right|}{\lambda_k}\right). \tag{58}$$

The LC kernel estimator in $\mathbb{R}^K$ is defined at each $\mathbf{z}_0$ as

$$\widehat{f}_{\mathrm{LC}}\left(\mathbf{z}_0\right) = \widehat{\beta}_{\mathrm{LC}}\left(\mathbf{z}_0\right) = \frac{\sum_{i=1}^N K_{\boldsymbol{\lambda}}\left(\mathbf{z}_0, \mathbf{z}^i\right) y^i}{\sum_{i=1}^N K_{\boldsymbol{\lambda}}\left(\mathbf{z}_0, \mathbf{z}^i\right)}. \tag{59}$$

Since we let $e_0\left(X\right)$ represent the intercept so that $z_{i0} = z_{00} = 1$, the corresponding univariate kernel $D\left(\frac{\left|z_{i0} - z_{00}\right|}{\lambda_0}\right) = D\left(0\right)$ is constant over all fitting points, thus cancels in (59) and can be omitted in (58).

The LL kernel estimator in $\mathbb{R}^K$ is given as the multidimensional analogue of (55) at each $\mathbf{z}_0$, that is,

$$\widehat{\boldsymbol{\beta}}_{\mathrm{LL}}\left(\mathbf{z}_0\right) = \underset{\boldsymbol{\beta}\left(\mathbf{z}_0\right) \in \mathbb{R}^K}{\arg\min} \left\{\sum_{i=1}^N K_{\boldsymbol{\lambda}}\left(\mathbf{z}_0, \mathbf{z}^i\right) \left(y^i - \mathbf{z}^{i,T} \boldsymbol{\beta}\left(\mathbf{z}_0\right)\right)^2\right\}, \tag{60}$$

with $\boldsymbol{\beta}\left(\mathbf{z}_0\right) = \left(\beta_0\left(\mathbf{z}_0\right), \ldots, \beta_{K-1}\left(\mathbf{z}_0\right)\right)^T$ and the proxy function at $\mathbf{z}_0$ is given by

$$\widehat{f}_{\mathrm{LL}}\left(\mathbf{z}_0\right) = \mathbf{z}_0^T \widehat{\boldsymbol{\beta}}_{\mathrm{LL}}\left(\mathbf{z}_0\right). \tag{61}$$

The LL kernel estimator can again be computed by WLS regression, that is,

$$\widehat{\boldsymbol{\beta}}_{\mathrm{LL}}\left(\mathbf{z}_0\right) = \left(Z^T W\left(\mathbf{z}_0\right) Z\right)^{-1} Z^T W\left(\mathbf{z}_0\right) \mathbf{y}, \tag{62}$$

where $W\left(\mathbf{z}_0\right) = \operatorname{diag}\left(K_{\boldsymbol{\lambda}}\left(\mathbf{z}_0, \mathbf{z}^1\right), \ldots, K_{\boldsymbol{\lambda}}\left(\mathbf{z}_0, \mathbf{z}^N\right)\right)$ is the weight matrix and Z the design matrix containing row-wise the vectors $\mathbf{z}^{i,T}$. The hat matrix H satisfies $\widehat{\mathbf{y}} = H\mathbf{y}$ with $\widehat{\mathbf{y}} = \left(\widehat{f}_{\mathrm{LL}}\left(\mathbf{z}^1\right), \ldots, \widehat{f}_{\mathrm{LL}}\left(\mathbf{z}^N\right)\right)^T$ containing the proxy function values at their target scenario vectors.

3.7.3. Bandwidth Selection, AIC and LOO-CV

The bandwidths λ_k in kernel regression can be selected similarly to the smoothing parameters in GAMs by minimization of a suitable model selection criterion. In fact, kernel smoothers can be interpreted as local non-parametric GLMs with identity link functions. More precisely, at each target scenario the kernel smoother can be viewed as a GLM (12) where the parametric weights $V\left[\widehat{\mu}^i_{\text{GLM}}\right]$ in (20) are the non-parametric kernel weights $K_\lambda\left(\mathbf{z}_0, \mathbf{z}^i\right)$ in (60). Since GLMs are special cases of GAMs and the bandwidths in kernel regression can be understood as smoothing parameters, kernel smoothers and GAMs are sometimes lumped together in one category. If the numbers N of the fitting points and K of the basis functions are large, from a computational perspective it might be beneficial to perform bandwidth selection based on a reduced set of fitting points.

Hurvich et al. (1998) propose to select the bandwidths $\lambda_1, \ldots, \lambda_{K-1}$ based on an improved version of AIC which works in the context of non-parametric proxy functions that can be written as linear combinations of the observations. It has the form

$$\text{AIC} = \log\left(\widehat{\sigma}^2\right) + \frac{1 + \operatorname{tr}(H)/N}{1 - (\operatorname{tr}(H) + 2)/N}, \tag{63}$$

where $\widehat{\sigma}^2 = \frac{1}{N}\left(\mathbf{y} - \widehat{\mathbf{y}}\right)^T\left(\mathbf{y} - \widehat{\mathbf{y}}\right)$ and H is the hat matrix.

As an alternative, leave-one-out cross-validation (LOO-CV) is suggested by Li and Racine (2004) for bandwidth selection. Let us refer to

$$\widehat{\boldsymbol{\beta}}_{\text{LL},-j}\left(\mathbf{z}_0\right) = \underset{\boldsymbol{\beta}(\mathbf{z}_0)\in\mathbb{R}^K}{\arg\min}\left\{\sum_{i\neq j, i=1}^{N} K_\lambda\left(\mathbf{z}_0, \mathbf{z}^i\right)\left(y^i - \mathbf{z}^{i,T}\boldsymbol{\beta}\left(\mathbf{z}_0\right)\right)^2\right\} \tag{64}$$

as the leave-one-out LL kernel estimator and to $\widehat{f}_{\text{LL},-j}\left(\mathbf{z}_0\right) = \mathbf{z}_0^T\widehat{\boldsymbol{\beta}}_{\text{LL},-j}\left(\mathbf{z}_0\right)$ as the leave-one-out proxy function at $\mathbf{z}_0$. The objective of LOO-CV is to choose the bandwidths $\lambda_1, \ldots, \lambda_{K-1}$ which minimize

$$\text{CV} = \frac{1}{N}\sum_{i=1}^{N}\left(y^i - \widehat{f}_{\text{LL},-i}\left(\mathbf{z}_0\right)\right)^2. \tag{65}$$

3.7.4. Adaptive Forward Stepwise OLS Selection

A practical implementation of kernel regression can be found, for example, via the combination of functions *npreg(·)* and *npregbw(·)* from R package *np* of Racine and Hayfield (2018).

In the other sections, basis function selection depends on the respective regression methods. Since the crucial process of bandwidth selection in kernel regression takes a very long time in the implementation of our choice, it would be infeasible to proceed here in the same way. Therefore, we derive the basis functions for LC and LL regression by adaptive forward stepwise selection based on OLS regression, by risk factor wise linear selection or a combination thereof. Thereby, we keep the maximum allowed number $K_{\max}$ of terms rather small as we aim to model the subtleties by kernel regression.

4. Numerical Experiments

4.1. General Remarks

4.1.1. Data Basis

In our slightly disguised real-world example, the life insurance company has a portfolio with a large proportion of traditional German annuity business. This choice was made in order to challenge the regression techniques since German traditional annuity business features high interest rate guarantees which may lead to large losses in low interest rate environments. We let the insurance company be exposed to $D = 15$ relevant financial and actuarial risk factors. For the derivation of the fitting points, we run its CFP model conditional on $N = 25,000$ fitting scenarios with each of these outer scenarios entailing two antithetic inner simulations. For a subset of the resulting fitting values of the best estimate of liabilities (BEL), see Figure 1, for summary statistics, the left column of Table 1, and for a histogram, the left panel of Figure 7.

Table 1. Summary statistics of fitting and nested simulation values of best estimate of liabilities (BEL).

	Fitting Values	Nested Simulation Values
Minimum:	10,883	12,479
1st quartile:	13,824	14,515
Median:	14,907	14,940
Mean:	14,922	14,922
3rd quartile:	15,989	15,330
Maximum:	19,354	17,080
Std. deviation:	1519	610
Skewness:	0.067	−0.081
Kurtosis:	2.478	3.214

The Sobol validation set is generated based on $L = 51$ validation scenarios with 1000 inner simulations, comprising 26 Sobol scenarios, 15 one-dimensional risk scenarios, 1 base scenario and 9 scenarios that turned out to be capital region scenarios in the previous year risk capital calculations. The nested simulations set which is due to its high computational costs not available in the regular LSMC approach reflects the highest 5% real-world losses and is based on $L = 1638$ outer scenarios with respectively 4000 inner simulations. From the 1638 real-world scenarios, 14 exhibit extreme stresses far beyond the bounds of the fitting space and are therefore excluded from the analysis. For the remaining nested simulation values of BEL, see Figure 3, for summary statistics, the right column of Table 1, and for a histogram, the right panel of Figure 7. The capital region set consists of the $L = 129$ nested simulations points which correspond to the nested simulations SCR estimate (= 99.5% highest loss) and the 64 losses above and below (= 99.3% to 99.7% highest losses).

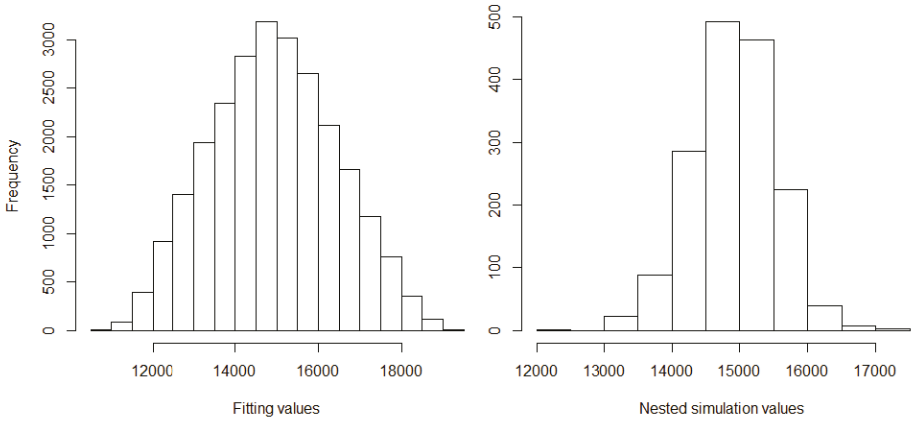


Figure 7. Histograms of fitting and nested simulation values of BEL.

4.1.2. Validation Figures

We will output validation figure (1) with respect to the relative and asset metric, and additionally figures (2)–(4). While figures (3) and (4) are evaluated with respect to a base value resulting from 1000 inner simulations on the Sobol set, that is, v.mae^0, v.res^0, they are computed with respect to a base value resulting from 16,000 inner simulations on the nested simulations set, that is, ns.mae^0, ns.res^0, and capital region set, that is, cr.mae^0, cr.res^0. The latter base value is supposed to be the more reliable validation value since it is the one associated with a lower standard error. Therefore it is worth noting here that figure v.res^0 can easily be transformed such that it is also evaluated with respect to the latter base value by subtracting from it the difference of 14 which the two different base values incur. We will not explicitly state the base residual (5) as it is just (2) minus (4).

4.1.3. Economic Variables

We derive the OLS proxy functions for two economic variables, namely for the best estimate of liabilities (BEL) and the available capital (AC) over a one-year risk horizon, that is, $Y(X) \in \{\text{BEL}(X),\ \text{AC}(X)\}$. Their approximation quality is assessed by validation figures (1) with respect to the relative and asset metric and (2). Essentially, AC is obtained as the market value of assets minus BEL, which means that AC reflects the negative behavior of BEL. Therefore, we will only derive BEL proxy functions with the other regression methods. The profit resulting from a certain risk constellation captured by an outer scenario X can be computed as $\text{AC}(X)$ minus the base AC. Validation figures (3) and (4) address the approximation quality of this difference. Taking the negative of the profit yields the loss and evaluating the loss at all real-world scenarios the real-world loss distribution from which the SCR is derived as the 99.5% value-at-risk. The out-of-sample performances of two different OLS proxy functions of BEL on the Sobol, nested simulations and capital region sets serve as the benchmark for the other regression methods.

4.1.4. Numerical Stability

Let us discuss the subject of numerical stability of QR decompositions in the OLS regression design under a monomial basis. If the weighting in the weighted least-squares problems associated with GLMs, heteroscedastic FGLS regression and kernel regression is good-natured, similar arguments apply as they can also be solved via QR decompositions according to Green (1984) where the weighting is just a scaling. However, the weighting itself raises additional numerical questions that need to be taken into consideration when making the regression design choices. In GLMs, these choices are the random component (13) and link function (12), in FGLS regression it is the functional form of the heteroscedatic variance model (42) and in kernel regression it is the kernel function (58). The following arguments do not apply to GAMs and MARS models as these are constructed out of spline functions, see (25) and (47), respectively. In GAMs, the penalty matrix increases numerical stability.

McLean (2014) justifies that from the perspective of numerical stability performing a QR decomposition on a monomial design matrix Z is asymptotically equivalent to using a Legendre design matrix Z' and transforming the resulting coefficient estimator into the monomial one. Under the assumption of an orthonormal basis, Weiß and Nikolić (2019) have derived an explicit upper bound for the condition number of non-diagonal matrix $\frac{1}{N}(Z')^T(Z')$ for $N < \infty$, where the factor $\frac{1}{N}$ is used for technical reasons. This upper bound increases in (1) the number of basis functions, (2) the Hardy-Krause variation of the basis, (3) the convergence constant of the low-discrepancy sequence, and (4) the outer scenario dimension. Our previously defined type of restriction setting controls aspect (1) through the specification of $K_{\max}$ and aspect (2) through the limitation of exponents $d_1 d_2 d_3$. Aspects (3) and (4) are beyond the scope of the calibration and validation steps of the LSMC framework and therefore left aside here.

4.1.5. Interpolation and Extrapolation

In the LSMC framework, let us refer by interpolation to prediction inside the fitting space and by extrapolation to prediction outside the fitting space. Runge (1901) found that high-degree polynomial interpolation at equidistant points can oscillate toward the ends of the interval with the approximation error getting worse the higher the degree is. In a least-squares problem, Runge's phenomenon was shown by Dahlquist and Björck (1974) not to apply to polynomials of degree d fitted based on N equidistant points if the inequality $d < 2\sqrt{N}$ holds. With N = 25,000 fitting points the inequality becomes $d < 316$ so that we clearly do not have to impose any further restrictions in OLS, FGLS and kernel regression as well as in GLMs to keep this phenomenon under control. Splines as they occur in GAMs and MARS models do not suffer from this oscillation issue by construction.

Since Runge's phenomenon concerns the ends of the interval and the real-world scenarios for the insurer's full loss distribution forecast in the fourth step of the LSMC framework partly go beyond the fitting space, its scope comprises the extrapolation area as well. High-degree polynomial extrapolation can worsen the approximation error and play a crucial role if many real-world scenarios go far beyond the fitting space.

4.1.6. Principle of Parsimony

Another problem that can occur in an adaptive algorithm is overfitting. Burnham and Anderson (2002) state that overfitted models often have needlessly large sampling variances which means that their precision of the predictions is poorer than that of more parsimonious models which are also free of bias. In cases where AIC leads to overfitting, implementing restriction settings of the form $K_{\max}$ - $d_1d_2d_3$ becomes relevant for adhering to the principle of parsimony.

4.2. Ordinary Least-Squares (OLS) Regression

4.2.1. Settings

We build the OLS proxy functions (10) of $Y(X) \in \{\text{BEL}(X),\ \text{AC}(X)\}$ with respect to an outer scenario X out of monomial basis functions that can be written as $e_k\,(X) = \prod_{l=1}^{15} X_l^{r_k^l}$ with $r_k^l \in \mathbb{N}_0$ so that each basis function can be represented by a 15-tuple $\left(r_k^1, \ldots, r_k^{15}\right)$. The final proxy function depends on the restrictions applied in the adaptive algorithm. The purpose of setting restrictions is to guarantee numerical stability, to keep the extrapolation behavior under control and the proxy functions parsimonious. In order to illustrate the impact of restrictions, we run the adaptive algorithm for BEL under two different restriction settings with the second one being so relaxed that it will not take effect in our example. Additionally, we run the adaptive algorithm under the first restriction setting for AC to give an example of how the behavior of BEL can transfer to AC. As the first ingredient of our restriction setting acts the maximum allowed number of terms $K_{\max}$. Furthermore, we limit the exponents in the monomial basis. Firstly we apply a uniform threshold to all exponents, that is, $r_k^l \leq d_1$. Secondly we restrict the degree, that is, $\sum_{l=1}^{15} r_k^l \leq d_2$. Thirdly we restrict the exponents in interaction basis functions, that is, if there are some $l_1 \neq l_2$ with $r_k^{l_1},\ r_k^{l_2} > 0$, we require $r_k^{l_1},\ r_k^{l_2} \leq d_3$. Let us denote this type of restriction setting by $K_{\max}$ - $d_1d_2d_3$.

As the first and second restriction settings, we choose 150–443 and 300–886, respectively, motivated by Teuguia et al. (2014) who found in their LSMC example in Chapter 4 with four risk factors and 50,000 fitting scenarios entailing two inner simulations that the validation error computed based on 14 validation scenarios started to stabilize at degree 4 when using monomial or Legendre basis functions in different adaptive basis function selection procedures. Furthermore, they pointed out that the LSMC approach becomes infeasible for degrees higher than 12.

We apply R function *lm(·)* implemented in R package *stats* of R Core Team (2018).

4.2.2. Results

Table A1 contains the final BEL proxy function derived under the first restriction setting 150–443 with the basis function representations and coefficients. Thereby reflect the rows the iterations of the adaptive algorithm and depict thus the sequence in which the basis functions are selected. Moreover, the iteration-wise AIC scores and out-of-sample MAEs (1) with respect to the relative metric in % on the Sobol, nested simulations and capital region sets are reported, that is, v.mae, ns.mae and cr.mae. Table A2 contains the AC counterpart of the BEL proxy function derived under 150–443 and Table A3 the final BEL proxy function derived under the more relaxed restriction setting 300–886. Tables A4 and A5 indicate respectively for the BEL and AC proxy functions derived under 150–443 the AIC scores and all five previously defined validation figures evaluated on the Sobol, nested simulations and capital region sets after each tenth iteration. Similarly, Table A6 reports these figures for the BEL proxy function derived under 300-886. Here the last row corresponds to the final iteration.

Lastly, we manipulate the validation values on all three validation sets twice insofar as we subtract respectively add pointwise 1.96 times the standard errors from respectively to them (inspired by 95% confidence interval of gaussian distribution). We then evaluate the validation figures for the final BEL proxy functions under both restriction settings on these manipulated sets of validation value estimates and depict them in Table A7 in order to assess the impact of the Monte Carlo error associated with the validation values.

4.2.3. Improvement by Relaxation

Tables A1 and A2 state that the adaptive algorithm terminates under 150–443 for both BEL and AC when the maximum allowed number of terms is reached. This gives reason to relax the restriction setting to, for example, 300–886 which eventually lets the algorithm terminate due to no further reduction in the AIC score without hitting restrictions 886, compare Table A3 for BEL. In fact, only restrictions 224–464 are hit. Except for the already very small figures cr.mae, cr.mae^a and cr.res all validation figures are further improved by the additional basis functions, see Tables A4 and A6. The largest improvement takes place between iterations 180 and 190. The result that at maximum degrees 464 are selected is consistent with the result of Teuguia et al. (2014) who conclude in their numerical examples of Chapter 4 that under a monomial, Legendre or Laguerre basis the optimum degree is probably 4 or 5. Furthermore, Bauer and Ha (2015) derive a similar result in their one risk factor LSMC example of Chapter 6 when using 50, 000 fitting scenarios and Legendre, Hermite, Chebychev basis functions or eigenfunctions.

According to our Monte Carlo error impact assessment in Table A7, the slight deterioration at the end of the algorithm is not sufficient to indicate a slight overfitting tendency of AIC. Under the standard choices of the five major components, compare Section 2.2, the adaptive algorithm manages thus to provide a numerically stable and parsimonious proxy function even without a restriction setting. Here, allowing a priori unlimited degrees of freedom is thus beneficial to capturing the complex interactions in the CFP model.

4.2.4. Reduction of Bias

Overall, the systematic deviations indicated by the means of residuals (2) and (4) are reduced significantly on the three validation sets by the relaxation but not completely eliminated. For the 300–886 OLS residuals on the three sets, see the diamond-shaped residuals in Figures 8–10, respectively. While the reduction of the bias comes along with the general improvement stated above, the remainder of the bias indicates that sample size is not sufficiently large or that the functional form is not flexible enough to replicate the complex interactions in CFP models. Note that if the functional form is correctly specified, Proposition 3.2 of Bauer and Ha (2015) states that if sample size is not sufficiently large, the AC proxy function will on average be positively biased in the tail reflecting the high losses and the BEL proxy function will thus be negatively biased there. Since Propositions 1 and 2 of Gordy and Juneja (2010) state that this result holds for the nested simulations estimators as well, the validation values of the nested simulations and capital region sets need to be more accurate in order to serve for bias detection in this case. For an illustration of such as bias, see Figures 5 and 6 of Bauer and Ha (2015). The bias in our one sample example is in the opposite systematic direction, which is an indication of insufficiency of polynomials. This is also consistent with the observations in the industry that the polynomials seem not to able to replicate the sudden changes in steepness of AC and BEL which are a consequence of regulation and complex management actions in the CFP models.

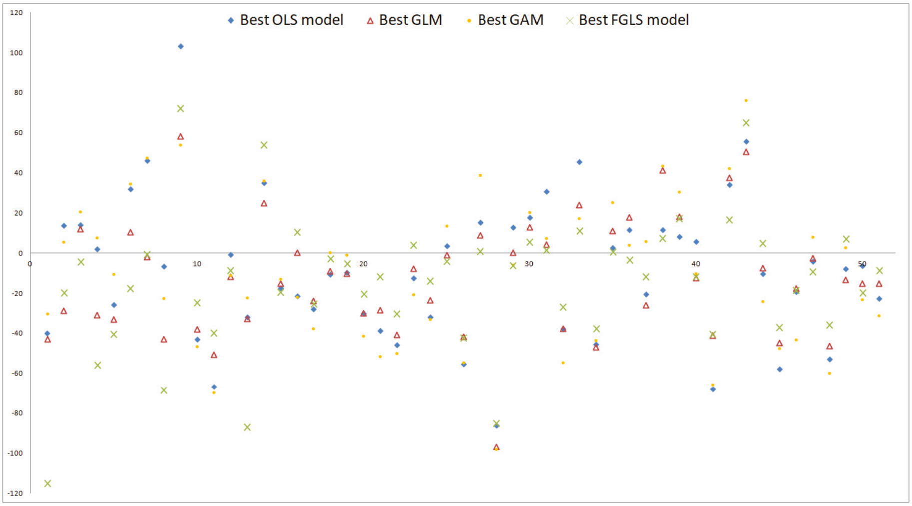


Figure 8. Residual plots on Sobol set.

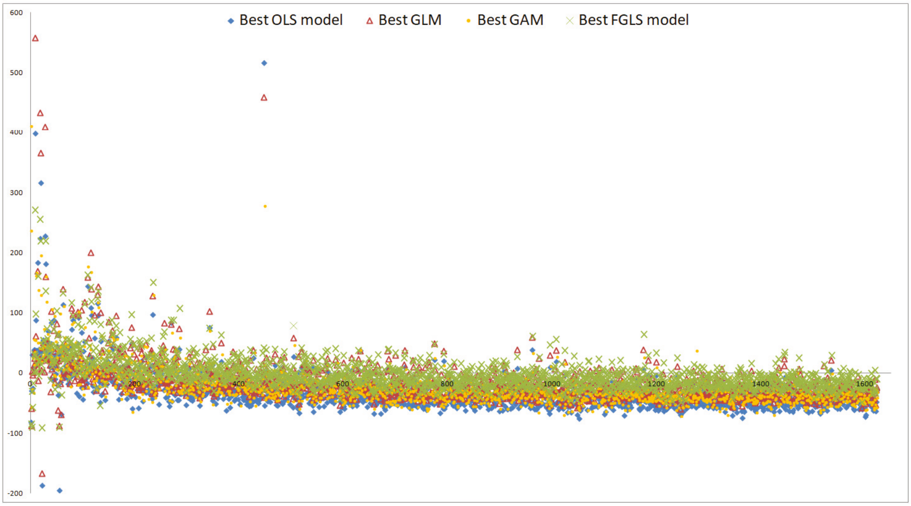


Figure 9. Residual plots on nested simulations set.

Unlike figures (1) and (2), figures (3) and (4) do not forgive a bad fit of the base value if the validation values are well approximated by a proxy function. Contrariwise, if a proxy function shows the same systematic deviation from the validation values and the base value, (3) and (4) will be close to zero whereas (1) and (2) will be not. The comparisons $|\text{v.res}| < |\text{v.res}^0|$, $|\text{cr.res}| < |\text{cr.res}^0|$ but $|\text{ns.res}| > |\text{ns.res}^0|$, holding under both restrictions settings, indicate that on the Sobol and capital region sets primarily the base value is not approximated well whereas on the nested simulations set not only the base value but also the validation values are missed. The MAEs capture this result, too, that is, $\text{v.mae}, \text{cr.mae} < \text{ns.mae}$ but $\text{ns.mae}^0 < \text{v.mae}^0, \text{cr.mae}^0$.

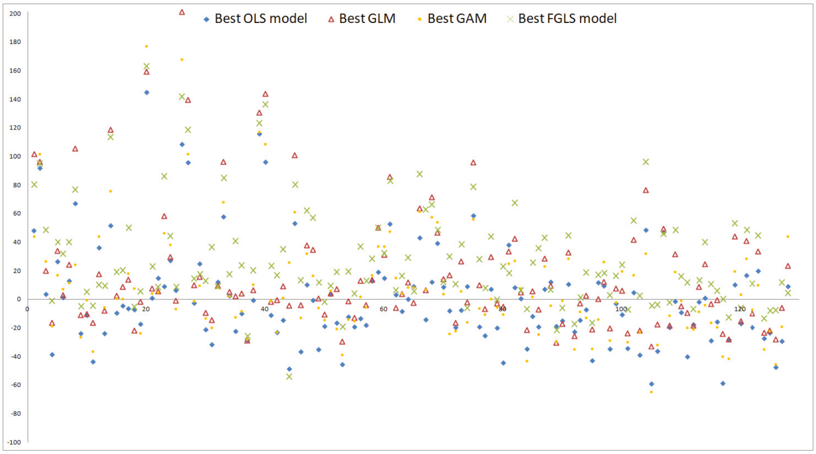


Figure 10. Residual plots on capital region set.

4.2.5. Relationship between BEL and AC

The MAEs with respect to the relative metric for BEL are much smaller than for AC since the two economic variables are subject to similar absolute fluctuations with, for example, in the base case BEL being approximately 20 times the size of AC. The similar absolute fluctuations are reflected by the iteration-wise very similar MAEs with respect to the asset metric of BEL and AC, compare v.mae^a, ns.mae^a and cr.mae^a given in % in Tables A4 and A5. Furthermore, they manifest themselves in the iteration-wise opposing means of residuals v.res, v.res^0, ns.res and cr.res as well as in the similar-sized MAEs v.mae^0, ns.mae^0 and cr.mae^0.

4.3. Generalized Linear Models (GLMs)

4.3.1. Settings

We derive the GLMs (12) of BEL under restriction settings 150–443 and 300–886 which we also employed for the derivation of the OLS proxy functions. Thereby, we run each restriction setting with the canonical choices of random components for continuous (non-negative) response variables, that is, the gaussian, gamma and inverse gaussian distributions, compare McCullagh and Nelder (1989). In cases where the economic variable can also attain negative values (for example, AC), a suitable shift of the response values in a preceding step would be required. We combine each of the three random component choices with the commonly used identity, inverse and log link functions, that is, $g\left(\mu\right) \in \left\{\mathrm{id}\left(\mu\right), \frac{1}{\mu}, \log\left(\mu\right)\right\}$, compare Hastie and Pregibon (1992). In combination with the inverse gaussian random component, we consider additionally link function $\frac{1}{\mu^2}$. Further choices are conceivable but go beyond this first shot.

We take R function *glm(·)* implemented in R package *stats* of R Core Team (2018).

4.3.2. Results

While Tables A8–A10 display the AIC scores and five previously defined validation figures after each tenth iteration for the just mentioned combinations under 150–443, Tables A11–A13 do so under 300-886 and include furthermore the final iterations. Table A14 gives an overview of the AIC scores and validation figures corresponding to all considered final GLMs and highlights in green and red respectively the best and worst values observed per figure.

4.3.3. Improvement by Relaxation

The OLS regression is the special case of a GLM with gaussian random component and identity link function which is why the first sections of Tables A8 and A11 coincide respectively with Tables A4 and A6. The adaptive algorithm terminates under 150–443 not only for this combination but also for all other ones when the maximum allowed number of terms is reached. Under 300–886 termination occurs due to no further reduction in the AIC score without hitting the restrictions-the different GLMs stop between 208–454 and 250–574.

For all GLMs except for the one with gamma random component and identity link, the AIC scores and eight most significant validation figures for measuring the approximation quality, namely leftmost figure v.mae to rightmost figure ns.res in the tables, are improved through the relaxation as can be seen in Table A14. For gamma random component with identity link, the deteriorations are negligible. Overall, figures ns.mae^0 and cr.mae^0 are deteriorated by at maximum 0.5% points and figures ns.res^0 and cr.res^0 by at maximum 4 units. Figures cr.mae and cr.mae^a are especially small under 150–443 so that slight deteriorations by at maximum 0.05% points under 300-886 towards the levels of v.mae and v.mae^a or ns.mae and ns.mae^a are not surprising. Similar arguments apply to the acceptability of the maximum deterioration of cr.res by 13 to 17 units for inverse gaussian with $\frac{1}{\mu^2}$ link. We conclude that the more relaxed restriction setting 300–886 performs better than 150–443 for all GLMs in our numerical example. This result appears plausible in comparison with the OLS result from the previous section and hence also compared to the OLS results of Teuguia et al. (2014) and Bauer and Ha (2015).

AIC cannot be said to show an overfitting tendency according to Tables A11–A13 and also Table A7 since the validation figures do not deteriorate in the late iterations more than they underly Monte Carlo fluctuations, compare the OLS interpretation. Using GLMs instead of OLS regression in the standard adaptive algorithm, compare Section 2.2, lets the algorithm thus maintain its property to yield numerically stable and parsimonious proxy functions even without restriction settings.

4.3.4. Reduction of Bias

According to Table A14, inverse gaussian with $\frac{1}{\mu^2}$ link shows the most significant decrease in v.mae by -0.088% points when moving from 150–443 to 300–886. Under 300–886 this combination even outperforms all other ones (highlighted in green) whereas under 150–443 it is vice versa (highlighted in red). Hence, the performance of a random component link combination under 150–443 does not generalize to 300–886. On the Sobol and nested simulations sets, the MAEs (1) are not only considerably lower for inverse gaussian with $\frac{1}{\mu^2}$ link than for all others but also the closest together even when the capital region set is included. This speaks for a great deal of consistency.

In fact, the systematic overestimation of 81% of the points on the nested simulations set by inverse gaussian with $\frac{1}{\mu^2}$ link is certainly smaller than, for example, that of 89% by gaussian with identity link but still very pronounced. On the capital region set, the overestimation rates for these two combinations are 41% and 56%, respectively, meaning that here the bias is negligibe. Surprisingly, for most GLMs the bias is here smaller than for inverse gaussian with $\frac{1}{\mu^2}$ link but since this result does not generalize to the nested simulations set, we regard it as a chance event and do not question the rather mediocre performance of inverse gaussian with $\frac{1}{\mu^2}$ link here further. Interpreting the mean of residuals (2) provides similar insights.

In particular, for inverse gaussian $\frac{1}{\mu^2}$ link GLM the reduction of the bias comes along with the general improvement by the relaxation. The small remainder of the bias indicates not only that this GLM is a promising choice here but also that identifying suitable regression methods and functional forms is crucial to further improving the accuracy of the proxy function. For the residuals on the three sets, see the triangle-shaped residuals in Figures 8–10, respectively.

4.3.5. Major and Minor Role of Link Function and Random Component

Apart from the just considered case, for all three random components, the relaxation to 300–886 yields the largest out-of-sample performance gains in terms of v.mae with identity link (between -0.047% and -0.058% points), closely followed by log link (between -0.033% and -0.047% points), and the least gains with inverse link (between -0.017% and -0.020% points). While with identity link the largest improvements before finalization take place for gaussian, gamma and inverse gaussian random components between iterations 180 to 190, 170 to 180, and 150 to 160, respectively, with log link they occur much sooner between iterations 120 to 130, 110 to 120, and 110 to 120, respectively, see Tables A11–A13. As a result of this behavior, under 150–443 log link performs better than identity link for gaussian and inverse gaussian whereas under 300–886 it is vice versa. Inverse link always performs worse than identity and log links, in particular under 300–886.

Applying the same link with different random components does not bring much variation under 300–886 with gamma and inverse gaussian being slightly better than gaussian for all considered links though. A possible explanation is that the distribution of BEL is slightly skewed conditional on the outer scenarios. Thereby results the skewness in the inner simulations from an asymmetric profit sharing mechanism in the CFP model. While the policyholders are entitled to participate at the profits of an insurance company, see, for example, Mourik (2003), the company has to bear its losses fully by itself. Since gaussian performs only slightly worse than the skewed distributions, it should still be considered for practical reasons because it has a closed-form solution and a great deal of statistical theory has been developed for it, compare, for example, Dobson (2002). By conclusion, the choice of the link is more important than that of the random component so that trying alternative link functions might be beneficial.

4.4. Generalized Additive Models (GAMs)

4.4.1. Settings

For the derivation of the GAMs (26) of BEL, we apply only restriction settings $K_{\max}$-443 with $K_{\max} \leq 150$ in the adaptive algorithm since we use smooth functions (25) constructed out of splines that may already have exponents greater than 1 to which the monomial first-order basis functions are raised. As the model selection criterion we take GCV (32) used by our chosen implementation by default. We vary different ingredients of GAMs while holding others fixed to carve out possible effects of these ingredients on the approximation quality of GAMs in adaptive algorithms and our application.

We rely on R function *gam(·)* implemented in R package *mgcv* of Wood (2018).

4.4.2. Results

Table A15 contains the validation figures for GAMs with varying number of spline functions per smooth function, that is, $J \in \{4, 5, 8, 10\}$, after each tenth and the finally selected smooth function. In the case of adaptive forward stepwise selection the iteration numbers coincide with the numbers of selected smooth functions. In contrast, table sections with adaptive forward stagewise selection results do not display the iteration numbers in the smooth function column k. In Table A16, we display the effective degrees of freedom, p-values and significance codes of each smooth function of the $J = 4$ and $J = 10$ GAMs from the previous table at stages $k \in \{50, 100, 150\}$. The p-values and significance codes are based on a test statistic of Marra and Wood (2012) having its foundations in the frequentist properties of Bayesian confidence intervals analyzed in Nychka (1988). Tables A17 and A18 report the validation figures respectively for GAMs with numbers $J = 5$ and $J = 10$, where the types of the spline functions are varied. Thin plate regression splines, penalized cubic regression splines, duchon splines and Eilers and Marx style P-splines are considered. Thereafter, Tables A19 and A20 display the validation figures respectively for GAMs with numbers $J = 4$ and $J = 8$ and different random component link function combinations. As in GLMs, we apply the gaussian, gamma and inverse gaussian distributions with identity, log, inverse and $\frac{1}{\mu^2}$ (only inverse gaussian) link functions.

Table A21 compares by means of two exemplary GAMs the effects of adaptive forward stagewise selection of length $L = 5$ and adaptive forward stepwise selection. Last but not least, Table A22 contains a mixture of GAMs challenging the results which we will have deduced from the other GAM tables. Table A23 gives an overview of the validation figures corresponding to all derived final GAMs and highlights in green and red respectively the best and worst values observed per figure.

4.4.3. Efficiency and Performance Gains by Tailoring the Spline Function Number

Table A15 indicates that the MAEs (1) and (3) of the exemplary GAMs built up of thin plate regression splines with gaussian random component and identity link tend to increase with the number J of spline functions per dimension until $k = 100$. Running more iterations reverses this behavior until $k = 150$. Hence, as long as comparably few smooth functions have been selected in the adaptive algorithm fewer spline functions tend to yield better out-of-sample performances of the GAMs whereas many smooth functions tend to perform better with more spline functions. A possible explanation of this observation is that an omitted-variable bias due to too few smooth functions is aggravated here by an overfitting due to too many spline functions. For more details on an omitted-variable bias, see, for example, Pindyck and Rubinfeld (1998), and for the needlessly large sampling variances and thus low estimation precision of overfitted models, see, for example, Burnham and Anderson (2002). Differently, the absolute values of the means of residuals (2) and (4) tend to become smaller with increasing J regardless of k.

According to Table A16, the components of the effective degrees of freedom (31) associated with each smooth function tend to decrease for $J = 4$ and $J = 10$ slightly in k. This is plausible as the explanatory power of each additionally selected smooth term is expected to decline by trend in the adaptive algorithm. Conditional on df > 1, that is for proportions of at least 40% of all smooth terms, the averages of the effective degrees of freedom belonging to $k \in \{50, 100, 150\}$ amount for $J = 4$ and $J = 10$ to $\{2.494, 2.399, 2.254\}$ and $\{5.366, 4.530, 4.424\}$, respectively. The values are by construction smaller than $J - 1$ since one degree of freedom per smooth function is lost to the identifiability constraints. Hence, for at least 40% of the smooth functions, on average $J = 6$ is a reasonable choice to capture the CFP model properly while maintaining computational efficiency, compare Wood (2017). The other side of the coin here is that up to 60% of the smooth functions are supposed to be replacable by simple linear terms without losing accuracy so that here tremendous efficiency gains can be realized by making the GAMs more parsimonious. Furthermore, setting J individually for each smooth function can help improve computational efficiency (if J should be set below average) and out-of-sample performance (if J should be set above average). However, such a tailored approach entails the challenge that the optimal J per smooth function is not stable across all k, compare row-wise the degrees of freedom in the table for $J = 4$ and $J = 10$.

4.4.4. Dependence of Best Spline Function Type

According to Tables A17 and A18, the adaptive algorithm terminates only due to no further decrease in GCV when the GAMs are composed of duchon splines discussed in Duchon (1977). Whether GCV has an overfitting tendency here cannot be deduced from this example since only restriction settings with $K_{\max} \leq 150$ are tested. The thin plate regression splines of Wood (2003) and penalized cubic regression splines of Wood (2017) perform similarly and significantly better than the duchon splines for both $J = 5$ and $J = 10$. For $J = 5$ the Eilers and Marx style P-splines proposed by Eilers and Marx (1996) perform by far best when $K_{\max} = 100$ smooth functions are allowed. However, for $J = 10$ they are outperformed by both the thin plate regression splines and penalized cubic regression splines when between $K_{\max} = 125$ and 150 smooth functions are allowed. This result illustrates well that the best choice of the spline function type varies with J and $K_{\max}$, meaning that it should be selected together with these parameters.

4.4.5. Minor Role of Link Function and Random Component

For GLMs, we have seen that varying the random component barely alters the validation results whereas varying the link function can make a noticeable impact. While this result mostly applies to the earlier compositions of GAMs as well, it certainly does not to the later ones. See for instance early composition $k = 40$ in Table A19. Here identity link GAMs with gamma and inverse gaussian random components perform more similar to each other than identity and log link GAMs with gamma random component or identity and log link GAMs with inverse gaussian random component do. Log link GAMs with gamma and inverse gaussian random components show such a behavior as well. However identity link GAM with the less flexible gaussian random component (no skewness) does not show at all a behavior similar to that of identity link GAMs with gamma or inverse gaussian random components. Now see later compositions $k \in \{70, 80\}$ to verify that all available GAMs in the table produce very similar validation results.

For another example see Table A20. For early composition $k = 50$, identity link GAMs with gaussian and gamma random components behave very similar to each other just like log link GAMs with gaussian and gamma random components do. For later compositions $k \in \{100, 110\}$, again all available GAMs produce very similar validation results. A possible explanation of this result is that the impact of the link function and random component decreases with the number of smooth functions as the latter take the modeling over. By conclusion, the choices of the random component and link function do not play a major role when the GAM is built up of many smooth functions.

4.4.6. Consistency of Results

Table A21 shows based on two exemplary GAMs constructed out of $J = 8$ thin plate regression splines per dimension varying in the random component and link function that the adaptive forward stagewise selection of length $L = 5$ and adaptive forward stepwise selection lead to very similar GAMs and validation results. As a result, stagewise selection should be preferred due to its considerable run time advantage. As we will see in the following, the run time can be further reduced without any drawbacks by dynamically selecting even more than 5 smooth functions per iteration.

The purpose of Table A22 is to challenge the hypotheses deduced above. Like Table A15, this table contains the results of GAMs with varying spline function number $J \in \{5, 8, 10\}$ and fixed spline function type. Instead of thin plate regression splines, now Eilers and Marx style P-splines are considered. Since adaptive forward stepwise and stagewise selection do not yield significant differences in the examples of Table A21, we do not expect that permutations thereof affect the results much here as well. This allows us to randomly assign three different adaptive forward selection approaches to the three exemplary proxy function derivation procedures. As one of these approaches, we choose a dynamic stagewise selection approach in which L is determined in each iteration as the proportion 0.25 of the size of the candidate term set. Again we see that as long as only $k \in \{90, 100\}$ smooth functions have been selected, $J = 5$ performs better than $J = 8$ and $J = 8$ better than $J = 10$. However, $k = 150$ smooth functions are not sufficient this time for $J = 10$ to catch up with the performance of $J = 5$. The observed performance order is consistent with the hypotheses of a high stability of the GAMs with respect to the adaptive selection procedure and random component link function combination.

4.4.7. Potential of Improved Interaction Modeling

Table A23 presents as the most suitable GAM the one with highest allowed maximum number of smooth functions $K_{\max} = 150$ and highest number of spline functions $J = 10$ per dimension. The slight deterioration after $k = 130$ reported by Table A15 indicates that at least one of the parameters is already comparably high. According to Table A16, there are a few smooth terms which might benefit from being composed of more than ten spline functions and increasing $K_{\max}$ might be helpful to capturing the interactions in the CFP model more appropriately, particularly in the light of the fact that the best GLM, having 250 basis functions, outperforms the best GAM on both the Sobol and nested simulations set, compare Table A14, with the best GAM showing a comparably low bias across the three validation sets though, see the dot-shaped residuals in Figures 8–10, respectively. Variations in the random component link function combination and adaptive selection procedure are not expected to change the performance much. By conclusion, we recommend the fast gaussian identity link GAMs (several expressions in the PIRLS algorithm simplify) with tailored spline function numbers per smooth function and simple linear terms under stagewise selection approaches of suitable lengths $L \geq 5$ and more relaxed restriction settings where $K_{\max} > 150$.

4.5. Feasible Generalized Least-Squares (FGLS) Regression

4.5.1. Settings

Like the OLS proxy functions and GLMs, we derive the FGLS proxy functions (38) under restriction settings 150–443 and 300–886. For the performance assessment of FGLS regression, we apply type I and II algorithms with variance models of different complexity, where type I results are obtained as a by-product of type II algorithm since the latter algorithm builds upon the former one. We control the complexity through the maximum allowed numbers of variance model terms $M_{\max} \in \{2; 6; 10; 14; 18; 22\}$.

We combine R functions *nlminb(·)* and *lm(·)* implemented in R package *stats* of R Core Team (2018).

4.5.2. Results

Tables A24 and A25 display respectively the adaptively selected FGLS variance models of BEL corresponding to maximum allowed numbers of terms $M_{\max}$ based on final 150–443 and 300–886 OLS proxy functions given in Tables A1 and A3. For reasons of numerical stability and simplicity, only basis functions with exponents summing up to at max two are considered as candidates. Additionally, the AIC scores and MAEs with respect to the relative metric are reported in the tables. By construction, these results are also the type I algorithm outcomes. Tables A26 and A27 summarize respectively under 150–443 and 300–886 all iteration-wise out-of-sample test results. The results of type II algorithm after each tenth and the final iteration of adaptive FGLS proxy function selection are respectively displayed by Tables A28 and A29. Table A30 gives an overview of the AIC scores and validation figures corresponding to all final FGLS proxy functions and highlights as in the previous overview tables in green and red respectively the best and worst values observed per figure.

4.5.3. Consistency Gains by Variance Modeling

By looking at Tables A24 and A25 we see similar out-of-sample performance patterns during adaptive variance model selection based on the basis function sets of 150–443 and 300–886 OLS proxy functions. In both cases, the p-values of Breusch-Pagan test indicate that heteroscedasticity is not eliminated but reduced when the variance models are extended, that is, when $M_{\max}$ is increased. In fact, in a more good-natured LSMC example Hartmann (2015) shows that a type I alike algorithm manages to fully eliminate heteroscedasticity. While the MAEs (1) barely change on the Sobol set, they decrease significantly on the nested simulations set and increase noticeably on the capital region set. Under 300–886 the effects are considerably smaller than under 150–443 since the capital region performance of 300–886 OLS proxy function is less extraordinarily good than that of 150–443 OLS proxy function. The three MAEs approach each other under both restriction settings. Hence the reductions in heteroscedasticity lead to consistency gains across the three validation sets.

Tables A26 and A27 complete the just discussed picture. The remaining validation figures on the Sobol set improve through type I FGLS regression slightly compared to OLS regression. Like ns.mae, figure ns.res and the base residual improve a lot with increasing $M_{\max}$ under 150–443 and a little less under 300-886 but ns.mae^0 and ns.res^0 do not alter much as the aforementioned two figures cancel each other out here. On the capital region set, the figures deteriorate or remain comparably high in absolute values. The type I FGLS figures converge fast so that increasing $M_{\max}$ successively from 10 to 22 barely affects the out-of-sample performance anymore. As a result of heteroscedasticity modeling, the proxy functions are shifted such that overall approximation quality increases. Unfortunately, this does not guarantee an improvement in the relevant region for SCR estimation as our example illustrates well.

4.5.4. Monotonicity in Complexity

Let us address the type II FGLS results under 150-443 in Table A28 now. For $M_{\max} = 2$, figures (3) and (4) are improved on all three validation sets significantly compared to OLS regression with the type I figures lying inbetween. The other validation figures are similar for OLS, type I and II FGLS regression, which traces the performance gains in (3) and (4) back to a better fit of the base value. For $M_{\max} = 6$ to 22, the type II figures show the same effects as the type I ones but more pronouncedly, see the previous two paragraphs. These effects are by trend the more distinct the more complex the variance model becomes. The type II figures stabilize less than the type I ones because of the additional variability coming along with adaptive FGLS proxy function selection. Hartmann (2015) shows in terms of Sobol figures in her LSMC example that increasing the complexity while omitting only one regressor from the simpler variance model can deteriorate the out-of-sample performance dramatically. Intuitively, it is plausible that the FGLS validation figures are the farther from the OLS figures away the more elaborately heteroscedasticity is modeled.

Now let us relate the type II FGLS results under 300-886 in Table A29 to the other FGLS results. Under 300–886 for $M_{\max} = 2$, figures (3) and (4) are already at a comparably good level with both OLS and type I FGLS regression so that they do not alter much or even deteriorate with type II FGLS regression. Like under 150–443 for $M_{\max} = 6$ to 22, the type II figures show the effects of the type I ones more pronouncedly. Under both restriction settings, ns.mae and ns.res decrease thereby significantly. While this barely causes ns.res^0 to change under 150–443, it lets ns.res^0 increase in absolute values under 300–886. The slight improvements on the Sobol set and the deteriorations on the capital region set carry over to 300–886. When $M_{\max}$ is increased up to 22, the type II FGLS validation figures under 300–886 do not stop fluctuating. The variability entailed by adaptive FGLS proxy function selection intensifies thus through the relaxation of the restriction setting in this numerical example. According to Breusch-Pagan test, heteroscedasticity is neither eliminated by the type II algorithm here nor by a type II alike approach of Hartmann (2015) in her more good-natured example.

4.5.5. Improvement by Relaxation

Among all FGLS proxy functions listed in Table A30, we consider type II with $M_{\max} = 14$ in variance model selection under 300–886 as the best performing one. Apart from nested simulations validation under type I algorithm, 300–886 performs better than 150–443. Since on the other hand type II algorithm performs better than type I algorithm under the respective restriction settings, 300–886 and type II algorithm are the most promising choices here. Differently $M_{\max} = 14$ does not constitute a stable choice due to the high variability coming along with 300–886 and type II algorithm.

While all type I FGLS proxy functions are by definition composed of the same basis functions as the OLS proxy function, the compositions of type II FGLS proxy functions vary with $M_{\max}$ because of their renewed adaptive selection. Consequently, under 300–886 all type I FGLS proxy functions hit the same restrictions 224–464 as the OLS proxy function does, whereas the restrictions hit by type II FGLS proxy functions vary between 224–454 and 258–564. This variation is consistent with the OLS and GLM results from the previous sections and hence the OLS results of Teuguia et al. (2014) and Bauer and Ha (2015).

AIC does not have an overfitting tendency according to Tables A26–A29 as the validation figures do not deteriorate in the late iterations more than they underly Monte Carlo fluctuations, compare the OLS and GLM interpretations. Using FGLS instead of OLS regression in the standard adaptive algorithm, compare Section 2.2, lets the algorithm thus yield numerically stable and parsimonious proxy functions without restriction settings as well.

4.5.6. Reduction of Bias

The type II $M_{\max} = 14$ FGLS proxy function under 300-886 reaches with 258 terms the highest observed number across all numerical experiments and not only outperforms all derived GLMs and GAMs in terms of combined Sobol and nested simulations validation, it also shows by far the smallest bias on these two validation sets and approximates the base value comparably well. This observation speaks for a high interaction complexity of the CFP model. The reduction of the bias comes again along with the general improvement by the relaxation. Given the fact that the capital region set presents the most extreme and challenging validation set in our analysis, the still mediocre performance here can be regarded as acceptable for now. Nevertheless, especially the bias on this set motivates the search for even more suitable regression methods and functional forms. For the residuals of the 300–886 FGLS proxy function on the three sets, see the x-shaped residuals in Figures 8–10, respectively.

4.6. Multivariate Adaptive Regression Splines (MARS)

4.6.1. Settings

We undertake a two-step approach to identify suitable generalized MARS models out of numerous possibilities. In the first step, we vary several MARS ingredients over a wide range and obtain in this way a large number of different MARS models. To be more specific, we vary the maximum allowed number of terms $K_{\max} \in \{50, 113, 175, 237, 300\}$ and the minimum threshold for the decrease in the residual sum of squares $t_{\min} \in \{0, 1.25, 2.5, 3.75, 5\} \cdot 10^{-5}$ in the forward pass, the order of interaction $o \in \{3, 4, 5, 6\}$, the pruning method $p \in \{'n', 'b', 'f', 's'\}$ with 'n' = 'none', 'b' = 'backward', 'f' = 'forward' and 's' = 'seqrep' in the backward pass, as well as the random component link function combination of the GLM extension. In addition to the 10 random component link function combinations applied in the numerical experiments of the GLMs, compare, for example, Table A14, we use poisson random component with identity, log and squareroot link functions. We work with the default fast MARS parameter fast.k = 20 of our chosen implementation.

We use R function *earth(·)* implemented in R package *earth* of Milborrow (2018).

4.6.2. Results

In total, these settings yield $4 \cdot 5 \cdot 5 \cdot 4 \cdot 13 = 5200$ MARS models with a lot of duplicates in our first step. We validate the 5200 MARS models on the Sobol, nested simulations and capital region sets through evaluation of the five validation figures. Then we collect the five best performing MARS models in terms of each validation figure per set which gives us in total $5 \cdot 5 = 25$ best performing models per first step validation set. Since the MAEs (1) with respect to the relative and asset metric entail the same best performing models, only $5 \cdot 4 = 20$ of the collected models per first step set are potentially different. Based on the ingredients of each of these 20 MARS models per first step set, we define $5 \cdot 5 = 25$ new sets of ingredients varying only with respect to $K_{\max}$ and $t_{\min}$ and derive the corresponding new but similar MARS models in the second step. As a result, we obtain in total $20 \cdot 25 = 500$ new MARS models per first step set. Again, we assess their out-of-sample performances through evaluation of the five validation figures on the three validation sets. Out of the 500 new MARS models per first step set, we collect then the best performing ones in terms of each validation figure per second step set. Now this gives us in total $5 \cdot 3 = 15$ best MARS models per first step set, or taking into account that the MAEs (1) with respect to the relative and asset metric entail once more the same best performing models, $4 \cdot 3 = 12$ potentially different best models per first step set. In total, this makes $12 \cdot 3 = 4 \cdot 9 = 36$ best MARS models, which can be found in Table A31 sorted by first and second step validation sets.

4.6.3. Poor Interaction Modeling and Extrapolation

In Table A31, the out-of-sample performances of all MARS models derived in our two-step approach are sorted using the first step validation set as the primary and the second step validation set as the secondary sort key. Let us address the first step second step validation set combinations by the headlines in Table A31. By construction, the combinations Sobol set^2, Nested simulations set^2 and Capital region set^2 yield respectively the MARS models with the best validation figures (1)–(4) on the Sobol, nested simulations and capital region sets. See that in the table all corresponding diagonal elements are highlighted in green. But the best MAEs (1) and (3) are not even close to what OLS regression, GLMs, GAMs and FGLS regression achieve. Finding small residuals (2) and (4) regardless of the other validation figures is not sufficient. The performances on the nested simulations and capital region sets, comprising several scenarios beyond the fitting space, are especially poor. All these results indicate that MARS models do not seem very suitable for our application. Despite the possibility to select up to 300 basis functions, the MARS algorithm selects only at maximum 148 basis functions, which suggests that without any alterations, the algorithm is not able to capture the behavior of the CFP model properly, in particular extrapolation behavior is comparably poor.

The MARS model with the set of ingredients $K_{\max} = 50$, $t_{\min} = 0$, $o = 4$, $p =$ 'b', inverse gaussian random component and identity link function is selected as the best one six times out of 36, or once for each Sobol and nested simulations first step validation set combination. Furthermore, this model performs best in terms of v.res^0, ns.mae^0 and ns.mae^a. Since there is no other MARS model with a similar high occurrence and performance, we consider it the best performing and most stable one found in our two-step approach. For illustration of a MARS model, see this one in Table A32. The fact that this best MARS model performs worse than other ones in terms of several validation figures stresses the infeasibility of MARS models in this application.

4.6.4. Limitations

Table A31 suggests that, up to a certain upper limit, the higher the maximum allowed number of terms $K_{\max}$ the higher tends the performance on the Sobol set to be. However, this result does not generalize to the nested simulations and capital region sets. Since at maximum 148 basis functions are selected here even if up to 300 basis functions are allowed, extending the range of $K_{\max}$ in the first step of this numerical experiment would not affect the output in this regard. The threshold $t_{\min}$ is an instrument controlling the number of basis functions selected in the forward pass up to $K_{\max}$ which cannot be extended below zero, meaning that its variability has already been exhausted here as well. For the interaction order o similar considerations as for $K_{\max}$ apply. The pruning method p used in the backward pass does not play a large role compared to the other ingredients as it only helps reduce the set of selected basis functions. In terms of Sobol validation, inverse gaussian random component with identity link performs best, whereas in terms of nested simulations and capital region validation, inverse gaussian random component with any link or log link with gaussian or poisson random component perform best. We conclude that if there was a suitable MARS model for our application, our two-step approach would have found it.

4.7. Kernel Regression

4.7.1. Settings

We make a series of adjustments affecting either the structure or the derivation process of the multidimensional LC and LL proxy functions (59) and (61) to get as broad a picture of the potential of kernel regression in our application as possible. Our adjustments concern the kernel function and its order, the bandwidth selection criterion, the proportion of fitting points used for bandwidth selection, and the sets of basis functions of which the local proxy functions are composed of. Thereby we combine in various ways the gaussian, Epanechnikov and uniform kernels, orders $o \in \{2, 4, 6, 8\}$, bandwidth selection criteria LOO-CV and AIC, and between 2500 (proportion bw $= 0.1$) and 25,000 (proportion bw $= 1$) fitting points for bandwidth selection.

We work with R functions *npregbw(·)* and *npreg(·)* implemented in R package *np* of Racine and Hayfield (2018).

4.7.2. Results

Furthermore, we alternate the four basis function sets contained in Tables A33 and A34. The first two basis function sets with $K_{\max} \in \{16, 27\}$ are derived by adaptive forward stepwise selection based on OLS regression, the third one with $K_{\max} = 15$ by risk factor wise linear selection and the last one with $K_{\max} = 22$ by a combination thereof. All combinations including their out-of-sample performances can be found in Table A35. Again, the best and worst values observed per validation figure are highlighted in green and red, respectively.

4.7.3. Poor Interaction Modeling and Extrapolation

We draw the following conclusions based on the validation results in Table A35. The comparisons of LC and LL regression applied with gaussian kernel and 16 basis functions or Epanechnikov kernel and 15 basis functions suggest that LL regression performs better than LC regression. However, even the best Sobol, nested simulations and capital region results of LL regression are still outperformed by OLS regression, GLMs, GAMs and FGLS regression. Possible explanations for this observation are that kernel regression is not able to model the interactions of the risk factors equally well with its few basis functions and that local regression approaches perform rather poorly close to and especially beyond the boundary of the fitting space because of the thinned out to missing data basis in this region. While the first explanation applies to all three validation sets, the latter one applies only to the nested simulations and capital region sets on which the validation figures are indeed worse than on the Sobol set. While LC regression produces interpretable results with the sets of 22 and 27 basis functions, the more complex LL regression does not in most cases.

4.7.4. Limitations

On the Sobol and capital region sets, both LC and LL regression show similar behaviors when relying on gaussian kernel and 16 basis functions compared to Epanechnikov kernel and 15 basis functions. But on the nested simulations set, gaussian kernel and 16 basis functions are the superior choices. Using a uniform kernel with LC regression deteriorates the out-of-sample performance. The results of LC regression indicate furthermore that an extension of the basis function sets from 15 to 27 only slightly affects the validation performance. With gaussian kernel switching from 16 to 27 basis functions barely has an impact and with Epanechnikov kernel only the nested simulations and capital region validation performance improve when using 27 as opposed to 15, 16 or 22 basis functions. While increasing the order of the gaussian or Epanechnikov kernel deteriorates the validation figures dramatically, for the uniform kernel the effects can go in both directions. AIC performs worse than LOO-CV when used for bandwidth selection of the gaussian kernel in LC regression. For LC regression, increasing the proportion of fitting points entering bandwidth selection improves all validation figures until a specific threshold is reached. But thereafter the nested simulations and capital region figures are deteriorated. For LL regression no such deterioration is observed.

Overall we do not see much potential in kernel regression for our practical example compared to most of the previously analyzed regression methods. Nonetheless in order to achieve comparably good kernel regression results, we consider LL regression more promising than LC regression due to the superior but still poor modeling close to and beyond the boundary of the fitting space. We would apply it with gaussian, Epanechnikov or other similar kernel functions. A high proportion of fitting points for bandwidth selection is recommended and it might be worth trying alternative comparably small basis function sets reflecting, for example, the risk factor interactions better than in our examples.

5. Conclusions

For high-dimensional variable selection applications such as the calibration step in the LSMC framework, we have presented various machine learning regression approaches ranging from ordinary and generalized least-squares regression variants over GLM and GAM approaches to multivariate adaptive regression splines and kernel regression approaches. At first we have justified the combinability of the ingredients of the regression routines such as the estimators and proposed model selection criteria in a theoretical discourse. Afterwards we have applied numerous configurations of these machine learning routines to the same slightly disguised real-world example in the LSMC framework. With the aid of different validation figures, we have analyzed the results, compared the out-of-sample performances and adviced to use certain routine designs.

In our slightly disguised real-world example and given LSMC setting, the adaptive OLS regression, GLM, GAM and FGLS regression algorithms turned out to be suitable machine learning methods for proxy modeling of life insurance companies with potential for both performance and computational efficiency gains by fine-tuning model hyperparameters and implementation designs. Differently, the MARS and kernel regression algorithms were not found to be convincing in our application. In order to study the robustness of our results, the approaches can be repeated in multiple other LSMC examples.

After all, none of our tested approaches was able to completely eliminate the bias observed in the validation figures and to yield consistent results across the three validation sets though. Investigations on whether these observations are systematic for the approaches, a result of the Monte Carlo error or a combination thereof help further narrow down the circle of recommended regression techniques. In order to assess the variance and bias of the proxy estimates conditional on an outer scenario, seed stability analyses in which the sets of fitting points are varied and convergence analyses in which sample size is increased need to be carried out. While such analyses would be computationally very costly, they would provide valuable insights into how to further improve approximation quality, that is, whether additional fitting points are necessary to reflect the underlying CFP model more accurately, whether more suitable functional forms and estimation assumptions are required for a more appropriate proxy modeling, or whether both aspects are relevant. Furthermore, one could deduce from such an analysis the sample sizes needed by the different regression algorithms to meet certain validation criteria. Since the generation of large sample sizes is currently computationally expensive for the industry, algorithms getting along with comparably few fitting points should be striven for.

Picking a suitable calibration algorithm is most important from the viewpoint of capturing the CFP model and hence the SCR appropriately. Therefore, if the bias observed in the validation figures indicates indeed issues with the functional forms of our approaches, doing further research on techniques not entailing such a bias or at least a smaller one is vital. On the one hand, one can fine-tune the approaches of this exposition and try different configurations thereof, and on the other hand, one can analyze further machine learning alternatives such as the ones mentioned in the introduction and already used in other LSMC applications. Ideally, various approaches like adaptive OLS regression, GLM, GAM and FGLS regression algorithms, artificial neural networks, tree-based methods and support vector machines would be fine-tuned and compared based on the same realistic and comprehensive data basis. Since the major challenges of machine learning calibration algorithms are hyperparameter selection and in some cases their dependence on randomness, future research should be dedicated to efficient hyperparameter search algorithms and stabilization methods such as ensemble methods.

Author Contributions: Conceptualization, A.-S.K., Z.N. and R.K.; Formal analysis, A.-S.K.; Investigation, A.-S.K.; Methodology, A.-S.K., Z.N. and R.K.; Project administration, A.-S.K.; Resources, Z.N.; Software, A.-S.K.; Supervision, R.K.; Validation, Z.N. and R.K.; Visualization, A.-S.K.; Writing–original draft, A.-S.K. and R.K.; Writing–review and editing, Z.N. and R.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The first author would like to thank Christian Weiß for his valuable comments which greatly helped to improve the paper. Furthermore, she is grateful to Magdalena Roth, Tamino Meyhöfer and her colleagues who have been supportive by providing her with academic time and computational resources. Additionally, we gratefully acknowledge very constructive comments by two anonymous reviewers.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A

Table A1. Ordinary least squares (OLS) proxy function of BEL derived under 150–443 in the adaptive algorithm with the final coefficients. Furthermore, Akaike information criterion (AIC) scores and out-of-sample mean absolute errors (MAEs) in % after each iteration.

k	r_k^1	r_k^2	r_k^3	r_k^4	r_k^5	r_k^6	r_k^7	r_k^8	r_k^9	r_k^{10}	r_k^{11}	r_k^{12}	r_k^{13}	r_k^{14}	r_k^{15}	$\widehat{\beta}_{\text{OLS},k}$	AIC	v.mae	ns.mae	cr.mae
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14,718.24	437,251	4.557	3.231	4.027
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	7850.17	386,722	2.474	0.845	0.913
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	−269.33	375,144	2.065	2.139	1.831
3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	145.21	366,567	1.656	0.444	0.496
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	−5.36	358,894	1.647	1.006	0.556
5	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	434.04	355,732	1.635	0.853	0.469
6	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1753.40	354,318	1.679	0.956	0.374
7	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	19,145.78	349,759	1.234	0.491	0.628
8	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	33.33	347,796	0.999	0.340	0.594
9	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	868.25	346,444	0.912	0.357	0.602
10	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	30.59	345,045	0.839	0.389	0.650
11	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.65	341,083	0.759	0.398	0.465
12	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	86.79	339,360	0.718	0.394	0.390
13	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	33.35	337,731	0.574	0.653	0.512
14	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	49.59	336,843	0.589	0.658	0.518
15	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	71.25	335,980	0.628	0.678	0.512
16	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	2667.92	335,351	0.609	0.671	0.503
17	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	96.43	334,876	0.579	0.701	0.545
18	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	−6.31	334,413	0.593	0.720	0.531
19	0	0	0	0	0	0	0	2	0	0	0	0	0	0	1	−47.09	333,904	0.562	0.621	0.474
20	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	48.93	333,447	0.565	0.597	0.454
21	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	−3412.68	333,116	0.553	0.543	0.407
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0.02	332,806	0.562	0.478	0.358
23	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	−0.12	332,547	0.550	0.450	0.381
24	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	43.77	332,294	0.545	0.468	0.378
25	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	118.94	332,042	0.530	0.464	0.362
26	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	−1288.45	331,687	0.522	0.453	0.355
27	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	−44.72	331,405	0.525	0.444	0.343
28	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	−24,908.99	331,136	0.499	0.405	0.327
29	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	−86.88	330,562	0.504	0.348	0.268
30	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0.55	330,361	0.518	0.418	0.264
31	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	77.26	330,163	0.512	0.443	0.272
32	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	24.78	329,988	0.508	0.443	0.264
33	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	14.33	329,834	0.477	0.491	0.286
34	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	−0.39	329,688	0.477	0.500	0.290
35	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	28.36	329,550	0.476	0.502	0.291
36	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	−370.92	329,442	0.472	0.499	0.288
37	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	−17.90	329,147	0.462	0.505	0.301
38	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	8574.53	329,043	0.472	0.518	0.300
39	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	−2.17	328,935	0.474	0.510	0.295
40	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	223.91	328,832	0.475	0.509	0.291
41	0	0	0	0	0	1	0	2	0	0	0	0	0	0	0	−1801.73	328,733	0.455	0.445	0.248
42	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	−102.10	327,927	0.372	0.345	0.237
43	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0.70	327,858	0.368	0.353	0.235
44	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0.56	327,792	0.366	0.352	0.233
45	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	−3034.32	327,729	0.365	0.356	0.228
46	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	−13,127.81	327,659	0.368	0.364	0.227
47	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	−17.54	327,603	0.368	0.366	0.226
48	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	−187.07	327,537	0.374	0.367	0.226
49	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	−300.54	327,483	0.369	0.367	0.230
50	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	−0.09	327,432	0.368	0.391	0.221
51	0	0	0	0	0	2	0	1	0	0	0	0	0	0	0	−60.84	327,382	0.359	0.390	0.228
52	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	−20.91	327,331	0.352	0.390	0.225
53	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0.00	327,287	0.346	0.377	0.206
54	0	0	0	0	0	0	0	1	0	0	0	0	0	0	2	−0.09	327,149	0.339	0.357	0.185
55	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1.44	327,105	0.315	0.321	0.173
56	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	−0.50	327,064	0.315	0.322	0.173
57	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	−6.06	327,025	0.322	0.317	0.175
58	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	−6600.49	326,986	0.317	0.310	0.172
59	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	−407.57	326,823	0.308	0.302	0.183
60	0	0	1	0	0	0	0	2	0	0	0	0	0	0	0	3378.82	326,787	0.306	0.301	0.183
61	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	205.28	326,733	0.304	0.299	0.183
62	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	−18.73	326,700	0.306	0.299	0.182
63	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	175.39	326,668	0.304	0.296	0.182
64	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	−0.20	326,638	0.304	0.298	0.181
65	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	2.45	326,610	0.301	0.296	0.183
66	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0.11	326,572	0.297	0.299	0.180
67	2	0	0	0	0	1	0	1	0	0	0	0	0	0	0	−13.02	326,545	0.292	0.286	0.169
68	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	93.69	326,519	0.292	0.287	0.172
69	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0	891.58	326,478	0.294	0.282	0.173
70	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	−6.21	326,453	0.291	0.281	0.175
71	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	−112.56	326,428	0.289	0.281	0.176
72	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	−5.27	326,398	0.284	0.282	0.173
73	1	0	0	0	0	0	0	3	0	0	0	0	0	0	0	1129.77	326,374	0.276	0.264	0.162
74	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	−0.29	326,352	0.272	0.266	0.158
75	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	−56.54	326,331	0.269	0.266	0.157

Table A1. *Cont.*

k	r_k^1	r_k^2	r_k^3	r_k^4	r_k^5	r_k^6	r_k^7	r_k^8	r_k^9	r_k^{10}	r_k^{11}	r_k^{12}	r_k^{13}	r_k^{14}	r_k^{15}	$\widehat{\beta}_{\text{OLS},k}$	AIC	v.mae	ns.mae	cr.mae
76	2	0	0	0	0	0	0	0	1	0	0	0	0	0	0	−3.02	326,313	0.271	0.266	0.155
77	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	−10.59	326,295	0.264	0.270	0.151
78	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	−6.99	326,278	0.264	0.275	0.153
79	1	0	0	0	0	2	0	0	0	0	0	0	0	0	0	−2.25	326,261	0.252	0.285	0.154
80	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	−14.77	326,245	0.263	0.309	0.157
81	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1.95	326,229	0.267	0.306	0.155
82	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	2248.54	326,214	0.266	0.307	0.156
83	0	0	0	0	0	0	0	3	0	0	0	0	0	0	1	−111.77	326,201	0.263	0.302	0.158
84	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	−0.11	326,187	0.262	0.302	0.157
85	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	−0.18	326,174	0.263	0.305	0.156
86	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	45.58	326,161	0.265	0.303	0.157
87	0	0	0	1	0	0	0	2	0	0	0	0	0	0	0	−83,291.89	326,149	0.267	0.308	0.156
88	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	−56.20	326,137	0.267	0.308	0.156
89	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	−5.32	326,126	0.267	0.310	0.156
90	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	−10.87	326,116	0.267	0.313	0.158
91	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	−32.75	326,106	0.265	0.317	0.158
92	0	0	0	0	0	0	0	2	0	0	0	0	0	0	2	−0.09	326,097	0.265	0.308	0.151
93	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	10.87	326,089	0.265	0.308	0.151
94	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	−48.93	326,081	0.264	0.306	0.148
95	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	69.57	326,073	0.256	0.288	0.141
96	0	0	0	1	0	0	0	3	0	0	0	0	0	0	0	−542,688.19	326,066	0.256	0.289	0.141
97	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	10.44	326,058	0.248	0.275	0.136
98	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	−1.08	326,051	0.248	0.276	0.136
99	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	419.05	326,045	0.249	0.275	0.136
100	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	12.80	326,038	0.250	0.276	0.136
101	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	−3.94	326,033	0.250	0.276	0.136
102	1	0	0	0	0	0	0	2	0	0	0	0	0	0	1	−10.12	326,027	0.248	0.281	0.138
103	2	0	0	0	0	0	0	1	0	0	0	0	0	0	1	−0.36	326,017	0.244	0.283	0.135
104	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	1.74	326,012	0.244	0.282	0.136
105	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0.00	326,006	0.242	0.268	0.132
106	2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	−7.09	326,001	0.238	0.265	0.131
107	2	0	0	0	0	0	1	1	0	0	0	0	0	0	0	−109.46	325,982	0.238	0.263	0.129
108	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	−0.10	325,977	0.237	0.263	0.128
109	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	5.76	325,972	0.235	0.263	0.129
110	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	54.51	325,968	0.237	0.264	0.129
111	1	0	0	0	0	0	1	2	0	0	0	0	0	0	0	−1386.73	325,963	0.235	0.264	0.129
112	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2	0.00	325,959	0.237	0.265	0.130
113	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0.11	325,955	0.235	0.265	0.130
114	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0.05	325,951	0.234	0.266	0.130
115	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	4.30	325,948	0.236	0.265	0.127
116	1	0	0	0	0	2	0	1	0	0	0	0	0	0	0	−19.81	325,944	0.237	0.262	0.126
117	2	0	0	0	0	2	0	0	0	0	0	0	0	0	0	−0.87	325,938	0.241	0.267	0.124
118	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	−0.36	325,935	0.241	0.267	0.124
119	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	−80.29	325,931	0.241	0.267	0.125
120	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	−6.95	325,928	0.241	0.267	0.124
121	0	0	0	0	0	1	0	0	0	0	0	0	0	0	2	0.00	325,925	0.243	0.259	0.121
122	0	0	0	0	0	0	0	2	0	0	1	0	0	0	0	436.56	325,923	0.241	0.259	0.121
123	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1	−0.03	325,920	0.243	0.263	0.121
124	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	2.99	325,918	0.242	0.263	0.120
125	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	−0.59	325,916	0.241	0.261	0.119
126	2	0	0	0	0	1	0	0	0	0	0	0	0	0	1	−0.02	325,908	0.247	0.265	0.124
127	0	0	0	0	0	1	0	2	0	0	0	0	0	0	1	−4.66	325,902	0.249	0.279	0.123
128	0	0	0	0	0	0	1	3	0	0	0	0	0	0	0	−8179.68	325,900	0.249	0.280	0.124
129	0	0	0	0	0	1	0	3	0	0	0	0	0	0	0	691.40	325,898	0.249	0.280	0.123
130	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.04	325,896	0.250	0.281	0.122
131	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	7.04	325,894	0.246	0.264	0.120
132	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	−27.72	325,892	0.247	0.264	0.119
133	2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1.26	325,891	0.247	0.264	0.119
134	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	−2.67	325,889	0.249	0.265	0.118
135	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1.53	325,887	0.250	0.266	0.119
136	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	−0.07	325,885	0.250	0.265	0.120
137	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	40.44	325,884	0.251	0.265	0.119
138	0	0	0	0	0	0	0	2	0	0	0	1	0	0	0	434.50	325,878	0.249	0.264	0.119
139	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	−5.99	325,877	0.248	0.264	0.119
140	0	0	0	0	0	0	0	0	2	0	0	1	0	0	0	14.64	325,873	0.246	0.263	0.120
141	0	0	0	0	0	2	0	2	0	0	0	0	0	0	0	−119.42	325,871	0.247	0.270	0.121
142	0	0	0	0	0	0	0	1	0	0	0	0	0	0	3	0.00	325,870	0.248	0.271	0.121
143	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0.07	325,868	0.248	0.271	0.121
144	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1.06	325,861	0.246	0.271	0.121
145	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	−0.74	325,859	0.247	0.271	0.121
146	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	−5.61	325,858	0.246	0.271	0.121
147	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	−0.08	325,857	0.247	0.270	0.121
148	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	−37.16	325,855	0.247	0.271	0.122
149	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0.41	325,851	0.247	0.271	0.122
150	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	−7290.99	325,850	0.247	0.271	0.122

Table A2. OLS proxy function of available capital (AC) derived under 150–443 in the adaptive algorithm with the final coefficients. Furthermore, AIC scores and out-of-sample MAEs in % after each iteration.

k	r_k^1	r_k^2	r_k^3	r_k^4	r_k^5	r_k^6	r_k^7	r_k^8	r_k^9	r_k^{10}	r_k^{11}	r_k^{12}	r_k^{13}	r_k^{14}	r_k^{15}	$\widehat{\beta}_{\mathrm{OLS},k}$	AIC	v.mae	ns.mae	cr.mae
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	745.35	391,375	60.620	97.518	257.762
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	5766.61	382,610	50.402	99.306	256.789
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	272.75	367,667	35.285	38.124	99.902
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	5.46	359,997	30.739	18.210	72.719
4	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	128.41	356,705	30.119	25.088	29.357
5	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	−1750.72	355,354	30.867	28.173	21.870
6	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	−19,127.27	351,002	22.942	14.948	44.668
7	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	−33.25	349,147	19.030	12.142	42.535
8	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	307.32	347,777	18.221	10.928	35.420
9	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	−868.05	346,423	16.662	11.527	35.941
10	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	−87.54	345,025	15.987	10.264	31.461
11	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	−30.51	343,570	14.858	11.187	34.502
12	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	−1.66	339,282	13.092	12.669	23.174
13	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	−33.33	337,648	10.427	20.976	30.402
14	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	−70.63	336,840	11.087	21.598	29.972
15	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	−41.37	336,120	11.436	21.764	30.408
16	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	−2666.44	335,495	11.088	21.543	29.890
17	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	−96.48	335,022	10.545	22.479	32.334
18	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	6.30	334,563	10.804	23.095	31.519
19	0	0	0	0	0	0	0	2	0	0	0	0	0	0	1	47.02	334,058	10.232	19.913	28.128
20	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	−48.77	333,610	10.292	19.163	26.995
21	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	3412.54	333,281	10.083	17.438	24.190
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	−0.02	332,970	10.246	15.328	21.326
23	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.12	332,714	10.020	14.436	22.671
24	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	−120.68	332,457	9.834	14.283	21.608
25	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	1287.63	332,108	9.725	13.969	21.273
26	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	44.71	331,832	9.755	13.661	20.501
27	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	24,899.66	331,569	9.275	12.462	19.873
28	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	87.04	331,004	9.292	10.757	17.022
29	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	−43.38	330,742	9.171	11.183	16.023
30	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	−0.55	330,543	9.444	13.409	15.766
31	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	−77.35	330,345	9.324	14.207	16.192
32	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	−25.20	330,161	9.246	14.203	15.692
33	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	−14.37	330,007	8.672	15.764	16.964
34	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0.39	329,859	8.682	16.031	17.223
35	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	−27.80	329,728	8.665	16.110	17.264
36	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	−8757.49	329,619	8.871	16.530	17.005
37	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	2.17	329,513	8.937	16.276	16.790
38	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	369.16	329,408	8.842	16.169	16.738
39	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	17.97	329,109	8.637	16.387	17.527
40	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	−222.55	329,008	8.656	16.359	17.271
41	0	0	0	0	0	1	0	2	0	0	0	0	0	0	0	1791.70	328,910	8.297	14.282	14.748
42	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	101.23	328,111	6.783	11.112	14.144
43	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	−0.70	328,041	6.713	11.355	14.013
44	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	−0.57	327,972	6.683	11.325	13.867
45	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	3083.05	327,905	6.654	11.456	13.595
46	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	12,863.79	327,837	6.700	11.721	13.500
47	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	17.78	327,780	6.710	11.777	13.450
48	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	190.46	327,711	6.824	11.818	13.468
49	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	300.76	327,657	6.724	11.793	13.716
50	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0.09	327,607	6.718	12.565	13.182
51	0	0	0	0	0	2	0	1	0	0	0	0	0	0	0	60.83	327,557	6.543	12.533	13.558
52	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	20.91	327,507	6.415	12.530	13.394
53	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0.00	327,463	6.314	12.118	12.252
54	0	0	0	0	0	0	0	1	0	0	0	0	0	0	2	0.08	327,327	6.176	11.486	11.049
55	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	−1.46	327,284	5.751	10.339	10.295
56	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0.50	327,242	5.746	10.367	10.287
57	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	6.08	327,203	5.871	10.211	10.450
58	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	6593.98	327,165	5.780	9.973	10.274
59	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	406.73	327,003	5.618	9.722	10.897
60	0	0	1	0	0	0	0	2	0	0	0	0	0	0	0	−3364.02	326,968	5.581	9.671	10.904
61	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	−204.12	326,914	5.542	9.626	10.921
62	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	18.90	326,881	5.588	9.611	10.837
63	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	−175.17	326,849	5.546	9.514	10.817
64	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0.21	326,818	5.540	9.597	10.799
65	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	−2.44	326,791	5.494	9.532	10.896
66	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	−0.11	326,753	5.413	9.616	10.708
67	2	0	0	0	0	1	0	1	0	0	0	0	0	0	0	12.99	326,726	5.317	9.215	10.046
68	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	−93.57	326,700	5.329	9.255	10.231
69	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0	−890.62	326,660	5.355	9.090	10.326
70	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	113.04	326,635	5.313	9.095	10.357
71	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	5.23	326,605	5.231	9.101	10.164
72	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	6.20	326,581	5.186	9.068	10.265
73	1	0	0	0	0	0	0	3	0	0	0	0	0	0	0	−1133.83	326,556	5.034	8.488	9.647
74	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0.29	326,534	4.950	8.580	9.374
75	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	56.56	326,513	4.908	8.559	9.323

Table A2. *Cont.*

k	r_k^1	r_k^2	r_k^3	r_k^4	r_k^5	r_k^6	r_k^7	r_k^8	r_k^9	r_k^{10}	r_k^{11}	r_k^{12}	r_k^{13}	r_k^{14}	r_k^{15}	$\widehat{\beta}_{\mathrm{OLS},k}$	AIC	v.mae	ns.mae	cr.mae
76	2	0	0	0	0	0	0	0	1	0	0	0	0	0	0	3.02	326,495	4.936	8.573	9.223
77	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	10.61	326,477	4.824	8.705	8.996
78	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	6.97	326,461	4.821	8.849	9.071
79	1	0	0	0	0	2	0	0	0	0	0	0	0	0	0	2.25	326,444	4.602	9.170	9.162
80	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	−1.94	326,429	4.688	9.069	8.997
81	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	−2257.40	326,414	4.676	9.099	9.070
82	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	14.06	326,399	4.853	9.831	9.278
83	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0.11	326,385	4.844	9.851	9.203
84	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.18	326,372	4.861	9.935	9.174
85	0	0	0	0	0	0	0	3	0	0	0	0	0	0	1	111.58	326,358	4.796	9.769	9.270
86	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	−45.11	326,346	4.826	9.724	9.330
87	0	0	0	1	0	0	0	2	0	0	0	0	0	0	0	82,935.66	326,334	4.871	9.865	9.284
88	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	56.00	326,322	4.867	9.862	9.267
89	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	5.35	326,311	4.857	9.938	9.258
90	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	10.88	326,301	4.870	10.043	9.414
91	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	32.81	326,291	4.833	10.156	9.394
92	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	48.96	326,283	4.812	10.085	9.185
93	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	−10.90	326,274	4.801	10.083	9.210
94	0	0	0	0	0	0	0	2	0	0	0	0	0	0	2	0.09	326,266	4.803	9.818	8.787
95	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	−69.45	326,258	4.659	9.250	8.413
96	0	0	0	1	0	0	0	3	0	0	0	0	0	0	0	543,840.26	326,251	4.663	9.269	8.393
97	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	−10.31	326,244	4.510	8.841	8.101
98	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	1.07	326,237	4.523	8.847	8.091
99	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	−417.88	326,231	4.531	8.840	8.101
100	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	−12.92	326,224	4.546	8.847	8.081
101	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	3.94	326,219	4.558	8.866	8.072
102	1	0	0	0	0	0	0	2	0	0	0	0	0	0	1	10.10	326,213	4.513	9.012	8.203
103	2	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0.36	326,204	4.453	9.084	8.035
104	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	−1.74	326,198	4.445	9.063	8.070
105	2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	7.09	326,193	4.383	8.967	8.008
106	2	0	0	0	0	0	1	1	0	0	0	0	0	0	0	109.50	326,174	4.371	8.899	7.889
107	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0.00	326,169	4.332	8.454	7.669
108	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	−5.85	326,164	4.290	8.456	7.689
109	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0.10	326,159	4.282	8.457	7.657
110	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	−54.88	326,154	4.313	8.463	7.689
111	1	0	0	0	0	0	1	2	0	0	0	0	0	0	0	1380.74	326,150	4.291	8.489	7.700
112	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2	0.00	326,146	4.315	8.498	7.751
113	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	−0.11	326,142	4.287	8.501	7.736
114	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	−4.30	326,138	4.320	8.461	7.558
115	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	−0.05	326,135	4.299	8.514	7.566
116	1	0	0	0	0	2	0	1	0	0	0	0	0	0	0	20.09	326,131	4.320	8.417	7.498
117	2	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0.87	326,125	4.393	8.561	7.371
118	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	0.36	326,122	4.389	8.564	7.409
119	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	79.51	326,118	4.394	8.560	7.411
120	0	0	0	0	0	1	0	0	0	0	0	0	0	0	2	0.00	326,115	4.430	8.304	7.187
121	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	6.91	326,113	4.420	8.305	7.176
122	0	0	0	0	0	0	0	2	0	0	1	0	0	0	0	−435.81	326,110	4.390	8.301	7.212
123	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1	0.03	326,107	4.419	8.450	7.206
124	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	−2.99	326,105	4.407	8.434	7.163
125	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0.59	326,103	4.394	8.366	7.095
126	2	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0.02	326,096	4.502	8.499	7.382
127	0	0	0	0	0	1	0	2	0	0	0	0	0	0	1	4.66	326,089	4.543	8.962	7.340
128	0	0	0	0	0	1	0	3	0	0	0	0	0	0	0	−692.59	326,088	4.537	8.961	7.248
129	0	0	0	0	0	0	1	3	0	0	0	0	0	0	0	8097.70	326,086	4.539	8.995	7.316
130	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	−0.04	326,084	4.555	9.024	7.285
131	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	2.73	326,082	4.590	9.065	7.246
132	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	−1.53	326,080	4.612	9.097	7.280
133	2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	−1.28	326,078	4.616	9.086	7.251
134	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.07	326,077	4.607	9.055	7.287
135	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	−6.96	326,075	4.533	8.527	7.230
136	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	27.74	326,073	4.556	8.520	7.115
137	0	0	0	0	0	2	0	2	0	0	0	0	0	0	0	122.08	326,071	4.571	8.746	7.171
138	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	6.00	326,070	4.556	8.745	7.190
139	0	0	0	0	0	0	0	0	2	0	0	1	0	0	0	−14.50	326,066	4.533	8.699	7.199
140	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	−0.07	326,064	4.532	8.722	7.227
141	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	−1.05	326,057	4.507	8.733	7.250
142	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0.74	326,056	4.515	8.719	7.238
143	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	5.71	326,054	4.503	8.706	7.263
144	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	−39.87	326,053	4.499	8.715	7.244
145	0	0	0	0	0	0	0	2	0	0	0	1	0	0	0	−431.71	326,047	4.470	8.669	7.215
146	0	0	0	0	0	0	0	1	0	0	0	0	0	0	3	0.00	326,046	4.488	8.698	7.207
147	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	0.08	326,045	4.494	8.694	7.223
148	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	37.33	326,043	4.496	8.703	7.236
149	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	−0.42	326,039	4.508	8.706	7.253
150	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	7224.25	326,038	4.512	8.712	7.265

Table A3. OLS proxy function of BEL derived under 300–886 in the adaptive algorithm with the final coefficients. Furthermore, AIC scores and out-of-sample MAEs in % after each iteration.

k	r_k^1	r_k^2	r_k^3	r_k^4	r_k^5	r_k^6	r_k^7	r_k^8	r_k^9	r_k^{10}	r_k^{11}	r_k^{12}	r_k^{13}	r_k^{14}	r_k^{15}	$\widehat{\beta}_{\mathrm{OLS},k}$	AIC	v.mae	ns.mae	cr.mae
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	14,689.75	437,251	4.557	3.231	4.027
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	7990.98	386,722	2.474	0.845	0.913
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	−274.24	375,144	2.065	2.139	1.831
3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	145.73	366,567	1.656	0.444	0.496
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	−5.11	358,894	1.647	1.006	0.556
5	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	416.79	355,732	1.635	0.853	0.469
6	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	2332.91	354,318	1.679	0.956	0.374
7	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	24,914.36	349,759	1.234	0.491	0.628
8	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	49.42	347,796	0.999	0.340	0.594
9	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	859.49	346,444	0.912	0.357	0.602
10	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	29.50	345,045	0.839	0.389	0.650
11	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.71	341,083	0.759	0.398	0.465
12	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	91.65	339,360	0.718	0.394	0.390
13	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	36.34	337,731	0.574	0.653	0.512
14	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	51.78	336,843	0.589	0.658	0.518
15	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	68.02	335,980	0.628	0.678	0.512
16	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	2661.47	335,351	0.609	0.671	0.503
17	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	109.14	334,876	0.579	0.701	0.545
18	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1	−12.63	334,413	0.593	0.720	0.531
19	0	0	0	0	0	0	0	2	0	0	0	0	0	0	1	−114.48	333,904	0.562	0.621	0.474
20	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	35.40	333,447	0.565	0.597	0.454
21	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0	−4570.15	333,116	0.553	0.543	0.407
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0.02	332,806	0.562	0.478	0.358
23	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	−0.26	332,547	0.550	0.450	0.381
24	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	47.17	332,294	0.545	0.468	0.378
25	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	123.47	332,042	0.530	0.464	0.362
26	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	−1240.44	331,687	0.522	0.453	0.355
27	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	−43.82	331,405	0.525	0.444	0.343
28	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	−32,661.61	331,136	0.499	0.405	0.327
29	2	0	0	0	0	0	0	1	0	0	0	0	0	0	0	−140.90	330,562	0.504	0.348	0.268
30	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0.56	330,361	0.518	0.418	0.264
31	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	87.33	330,163	0.512	0.443	0.272
32	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	25.31	329,988	0.508	0.443	0.264
33	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	14.22	329,834	0.477	0.491	0.286
34	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	−0.44	329,688	0.477	0.500	0.290
35	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	26.88	329,550	0.476	0.502	0.291
36	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	−391.81	329,442	0.472	0.499	0.288
37	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	−18.58	329,147	0.462	0.505	0.301
38	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	11,959.32	329,043	0.472	0.518	0.300
39	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	−2.15	328,935	0.474	0.510	0.295
40	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	228.32	328,832	0.475	0.509	0.291
41	0	0	0	0	0	1	0	2	0	0	0	0	0	0	0	−1938.37	328,733	0.455	0.445	0.248
42	1	0	0	0	0	1	0	1	0	0	0	0	0	0	0	−112.83	327,927	0.372	0.345	0.237
43	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0.71	327,858	0.368	0.353	0.235
44	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0.72	327,792	0.366	0.352	0.233
45	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	−4230.29	327,729	0.365	0.356	0.228
46	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	−10,720.30	327,659	0.368	0.364	0.227
47	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	−18.39	327,603	0.368	0.366	0.226
48	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	−212.78	327,537	0.374	0.367	0.226
49	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	−177.64	327,483	0.369	0.367	0.230
50	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	−0.09	327,432	0.368	0.391	0.221
51	0	0	0	0	0	2	0	1	0	0	0	0	0	0	0	−57.40	327,382	0.359	0.390	0.228
52	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	−23.55	327,331	0.352	0.390	0.225
53	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0.00	327,287	0.346	0.377	0.206
54	0	0	0	0	0	0	0	1	0	0	0	0	0	0	2	−0.08	327,149	0.339	0.357	0.185
55	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1.15	327,105	0.315	0.321	0.173
56	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	−0.65	327,064	0.315	0.322	0.173
57	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	−4.41	327,025	0.322	0.317	0.175
58	0	0	0	0	0	0	1	2	0	0	0	0	0	0	0	−6095.97	326,986	0.317	0.310	0.172
59	1	0	0	0	0	0	1	1	0	0	0	0	0	0	0	−332.88	326,823	0.308	0.302	0.183
60	0	0	1	0	0	0	0	2	0	0	0	0	0	0	0	3624.77	326,787	0.306	0.301	0.183
61	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	191.46	326,733	0.304	0.299	0.183
62	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	−17.49	326,700	0.306	0.299	0.182
63	0	0	1	0	0	1	0	1	0	0	0	0	0	0	0	183.68	326,668	0.304	0.296	0.182
64	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	−0.20	326,638	0.304	0.298	0.181
65	0	1	0	0	0	0	0	1	0	0	0	0	0	0	1	2.55	326,610	0.301	0.296	0.183
66	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0.13	326,572	0.297	0.299	0.180
67	2	0	0	0	0	1	0	1	0	0	0	0	0	0	0	−29.57	326,545	0.292	0.286	0.169
68	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	95.55	326,519	0.292	0.287	0.172
69	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0	922.48	326,478	0.294	0.282	0.173
70	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	−6.22	326,453	0.291	0.281	0.175
71	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	−134.95	326,428	0.289	0.281	0.176
72	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	−4.47	326,398	0.284	0.282	0.173
73	1	0	0	0	0	0	0	3	0	0	0	0	0	0	0	−26,186.72	326,374	0.276	0.264	0.162
74	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	−0.29	326,352	0.272	0.266	0.158
75	1	0	0	0	0	0	0	1	1	0	0	0	0	0	0	−58.01	326,331	0.269	0.266	0.157

Table A3. *Cont.*

k	r_k^1	r_k^2	r_k^3	r_k^4	r_k^5	r_k^6	r_k^7	r_k^8	r_k^9	r_k^{10}	r_k^{11}	r_k^{12}	r_k^{13}	r_k^{14}	r_k^{15}	$\widehat{\beta}_{\text{OLS},k}$	**AIC**	**v.mae**	**ns.mae**	**cr.mae**
76	2	0	0	0	0	0	0	0	1	0	0	0	0	0	0	−3.11	326,313	0.271	0.266	0.155
77	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	−2.10	326,295	0.264	0.270	0.151
78	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	−8.73	326,278	0.264	0.275	0.153
79	1	0	0	0	0	2	0	0	0	0	0	0	0	0	0	−1.93	326,261	0.252	0.285	0.154
80	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	−14.90	326,245	0.263	0.309	0.157
81	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	−1.22	326,229	0.267	0.306	0.155
82	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	3341.29	326,214	0.266	0.307	0.156
83	0	0	0	0	0	0	0	3	0	0	0	0	0	0	1	−43.84	326,201	0.263	0.302	0.158
84	1	0	0	0	0	0	0	0	1	0	0	0	0	0	1	−0.12	326,187	0.262	0.302	0.157
85	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	−0.18	326,174	0.263	0.305	0.156
86	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	67.19	326,161	0.265	0.303	0.157
87	0	0	0	1	0	0	0	2	0	0	0	0	0	0	0	−432,954.98	326,149	0.267	0.308	0.156
88	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	−34.58	326,137	0.267	0.308	0.156
89	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	−5.10	326,126	0.267	0.310	0.156
90	0	0	0	0	0	2	1	0	0	0	0	0	0	0	0	−10.78	326,116	0.267	0.313	0.158
91	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1	−66.99	326,106	0.265	0.317	0.158
92	0	0	0	0	0	0	0	2	0	0	0	0	0	0	2	−0.09	326,097	0.265	0.308	0.151
93	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0.35	326,089	0.265	0.308	0.151
94	1	0	0	0	0	1	1	1	0	0	0	0	0	0	0	−93.83	326,081	0.264	0.306	0.148
95	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	70.45	326,073	0.256	0.288	0.141
96	0	0	0	1	0	0	0	3	0	0	0	0	0	0	0	−1,073,454.04	326,066	0.256	0.289	0.141
97	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	−21.59	326,058	0.248	0.275	0.136
98	0	0	0	0	0	0	0	1	1	0	0	0	0	0	1	−1.10	326,051	0.248	0.276	0.136
99	0	0	1	0	0	0	1	1	0	0	0	0	0	0	0	398.94	326,045	0.249	0.275	0.136
100	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	22.03	326,038	0.250	0.276	0.136
101	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	−4.12	326,033	0.250	0.276	0.136
102	1	0	0	0	0	0	0	2	0	0	0	0	0	0	1	1.30	326,027	0.248	0.281	0.138
103	2	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0.20	326,017	0.244	0.283	0.135
104	1	0	0	0	0	0	0	3	0	0	0	0	0	0	1	351.11	326,009	0.245	0.289	0.138
105	0	0	1	0	0	0	0	1	0	0	0	0	0	0	1	1.09	326,003	0.244	0.288	0.139
106	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0.00	325,997	0.242	0.274	0.136
107	2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	−7.78	325,992	0.239	0.271	0.134
108	2	0	0	0	0	0	1	1	0	0	0	0	0	0	0	−126.28	325,973	0.238	0.269	0.132
109	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	−0.10	325,968	0.238	0.269	0.131
110	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	57.61	325,963	0.239	0.269	0.132
111	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	9.91	325,959	0.237	0.269	0.132
112	1	0	0	0	0	0	1	2	0	0	0	0	0	0	0	−1698.92	325,954	0.236	0.270	0.132
113	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2	−0.01	325,950	0.237	0.270	0.133
114	0	1	0	0	0	0	0	0	1	0	0	0	0	0	1	0.10	325,946	0.236	0.271	0.133
115	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	0.05	325,942	0.234	0.272	0.132
116	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	5.00	325,939	0.236	0.271	0.129
117	1	0	0	0	0	2	0	1	0	0	0	0	0	0	0	−17.60	325,935	0.238	0.268	0.127
118	2	0	0	0	0	2	0	0	0	0	0	0	0	0	0	−0.79	325,929	0.242	0.273	0.128
119	0	1	0	0	0	1	0	1	0	0	0	0	0	0	1	−0.55	325,925	0.241	0.273	0.128
120	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	−119.81	325,922	0.242	0.273	0.129
121	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	−7.16	325,919	0.241	0.273	0.128
122	0	0	0	0	0	1	0	0	0	0	0	0	0	0	2	0.00	325,916	0.243	0.265	0.124
123	0	0	0	0	0	0	0	2	0	0	1	0	0	0	0	497.02	325,914	0.241	0.265	0.125
124	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1	−0.03	325,911	0.243	0.269	0.125
125	1	0	0	0	0	1	0	1	0	0	0	0	0	0	1	−0.58	325,909	0.242	0.267	0.123
126	2	0	0	0	0	1	0	0	0	0	0	0	0	0	1	−0.02	325,901	0.248	0.271	0.129
127	0	0	0	0	0	1	0	2	0	0	0	0	0	0	1	−4.48	325,895	0.251	0.286	0.129
128	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	2.93	325,893	0.250	0.285	0.128
129	0	0	0	0	0	0	1	3	0	0	0	0	0	0	0	−5069.15	325,891	0.250	0.286	0.128
130	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0.03	325,889	0.251	0.287	0.127
131	0	0	0	0	0	1	0	3	0	0	0	0	0	0	0	2631.07	325,887	0.251	0.287	0.125
132	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	30.03	325,885	0.246	0.270	0.124
133	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	−27.79	325,883	0.248	0.270	0.123
134	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	−2.68	325,881	0.249	0.271	0.122
135	1	0	0	0	0	1	0	0	0	0	0	1	0	0	0	2.18	325,879	0.251	0.272	0.123
136	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	−0.07	325,878	0.250	0.271	0.124
137	1	0	0	0	0	0	0	1	0	0	0	1	0	0	0	52.06	325,876	0.251	0.272	0.123
138	0	0	0	0	0	0	0	2	0	0	0	1	0	0	0	507.79	325,870	0.250	0.270	0.123
139	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0.09	325,869	0.248	0.270	0.123
140	0	0	0	0	0	0	0	0	2	0	0	1	0	0	0	14.53	325,865	0.246	0.269	0.123
141	0	0	0	0	0	0	0	1	0	0	0	0	0	0	3	0.00	325,864	0.247	0.270	0.122
142	2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1.48	325,862	0.247	0.269	0.121
143	0	0	0	0	0	2	0	2	0	0	0	0	0	0	0	−98.06	325,861	0.248	0.276	0.122
144	1	0	0	0	0	0	1	1	0	0	0	0	0	0	1	−0.68	325,859	0.248	0.276	0.122
145	1	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0.08	325,858	0.248	0.276	0.122
146	0	0	0	0	0	0	0	1	0	0	0	1	0	0	1	1.10	325,850	0.247	0.277	0.122
147	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	−5.64	325,849	0.247	0.276	0.123
148	0	1	0	0	0	0	0	0	0	0	0	0	0	1	1	−0.08	325,847	0.247	0.276	0.123
149	1	0	0	1	0	0	0	0	0	0	0	0	0	0	1	20.58	325,846	0.246	0.277	0.123
150	0	0	0	1	0	0	0	1	0	0	0	0	0	0	1	−60.89	325,841	0.242	0.274	0.123

Table A3. *Cont.*

k	r_k^1	r_k^2	r_k^3	r_k^4	r_k^5	r_k^6	r_k^7	r_k^8	r_k^9	r_k^{10}	r_k^{11}	r_k^{12}	r_k^{13}	r_k^{14}	r_k^{15}	$\widehat{\beta}_{\mathrm{OLS},k}$	AIC	v.mae	ns.mae	cr.mae
151	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	−26.95	325,840	0.242	0.275	0.123
152	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0.42	325,835	0.243	0.275	0.123
153	0	1	0	1	0	0	0	1	0	0	0	0	0	0	0	−10,592.62	325,834	0.243	0.275	0.123
154	2	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0.93	325,833	0.243	0.275	0.125
155	1	0	0	0	0	0	0	0	1	0	0	1	0	0	0	2.96	325,832	0.244	0.275	0.124
156	0	0	0	0	0	1	0	0	1	0	0	1	0	0	0	−3.87	325,830	0.244	0.275	0.125
157	0	0	0	0	0	0	2	0	0	0	0	0	1	0	0	−68.29	325,829	0.243	0.277	0.125
158	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	−9773.54	325,828	0.243	0.278	0.125
159	0	0	0	1	0	0	1	0	0	0	0	0	0	0	1	120.51	325,822	0.242	0.278	0.125
160	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0.03	325,821	0.243	0.278	0.127
161	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	−19.68	325,820	0.243	0.278	0.127
162	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	−24.62	325,819	0.240	0.261	0.127
163	0	0	0	0	0	0	0	0	1	0	0	0	0	0	3	0.00	325,818	0.239	0.261	0.128
164	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	−5.28	325,817	0.239	0.262	0.128
165	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	2.36	325,816	0.240	0.262	0.129
166	1	1	0	0	0	1	0	0	0	0	0	0	0	0	1	−0.02	325,814	0.238	0.264	0.129
167	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	−5.06	325,813	0.238	0.264	0.129
168	1	0	1	0	0	1	0	1	0	0	0	0	0	0	0	20.18	325,812	0.238	0.263	0.129
169	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	−461.05	325,812	0.239	0.264	0.130
170	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	6.14	325,811	0.238	0.265	0.130
171	0	0	0	1	0	0	0	2	0	0	0	0	0	0	1	2708.64	325,810	0.237	0.265	0.130
172	0	0	0	1	0	0	0	3	0	0	0	0	0	0	1	9307.25	325,805	0.239	0.265	0.129
173	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	−0.17	325,805	0.238	0.265	0.129
174	0	1	0	0	0	0	0	0	0	0	0	0	0	2	0	5.94	325,804	0.238	0.264	0.128
175	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	−0.07	325,804	0.238	0.264	0.127
176	0	0	1	0	0	0	1	2	0	0	0	0	0	0	0	−1367.33	325,803	0.238	0.264	0.128
177	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1133.78	325,803	0.237	0.264	0.128
178	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	−1.86	325,802	0.237	0.264	0.128
179	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.99	325,802	0.241	0.274	0.131
180	3	0	0	0	0	0	0	0	0	0	0	0	0	0	1	−0.01	325,766	0.241	0.300	0.149
181	3	0	0	0	0	1	0	0	0	0	0	0	0	0	0	−0.68	325,744	0.248	0.335	0.172
182	3	0	0	0	0	0	0	1	0	0	0	0	0	0	0	−70.02	325,727	0.245	0.326	0.157
183	2	0	0	0	0	0	0	2	0	0	0	0	0	0	0	−1883.77	325,700	0.238	0.313	0.144
184	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	−1.21	325,672	0.231	0.327	0.173
185	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	−157,391.76	325,655	0.225	0.309	0.175
186	0	0	0	0	0	0	0	4	0	0	0	0	0	0	1	2127.74	325,644	0.221	0.303	0.176
187	2	0	0	0	0	0	0	2	0	0	0	0	0	0	1	21.17	325,583	0.206	0.296	0.190
188	3	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0.62	325,524	0.198	0.268	0.164
189	0	0	0	1	0	0	0	4	0	0	0	0	0	0	0	5,216,336.05	325,515	0.199	0.270	0.166
190	3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	−0.54	325,506	0.201	0.275	0.173
191	4	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0.01	325,500	0.195	0.281	0.184
192	2	0	0	0	0	0	1	2	0	0	0	0	0	0	0	136.68	325,499	0.193	0.279	0.182
193	0	0	0	0	0	0	2	1	0	0	0	0	0	0	0	−526.83	325,498	0.194	0.280	0.182
194	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	−32.63	325,494	0.192	0.270	0.178
195	0	0	0	0	0	0	2	2	0	0	0	0	0	0	0	−2791.14	325,492	0.190	0.261	0.176
196	2	0	0	0	0	0	2	0	0	0	0	0	0	0	0	11.06	325,491	0.191	0.265	0.178
197	0	0	1	0	0	1	0	0	0	0	0	0	0	0	1	0.09	325,491	0.190	0.265	0.179
198	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	13.23	325,490	0.186	0.258	0.178
199	0	0	2	0	0	0	1	0	0	0	0	0	0	0	0	143.48	325,488	0.187	0.261	0.179
200	2	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0.46	325,488	0.186	0.262	0.181
201	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0.98	325,487	0.185	0.262	0.181
202	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	8.97	325,487	0.185	0.263	0.180
203	0	0	0	1	0	0	0	4	0	0	0	0	0	0	1	−33,222.10	325,487	0.184	0.263	0.179
204	2	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0.01	325,487	0.184	0.264	0.180
205	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	−0.32	325,487	0.184	0.263	0.178
206	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0.20	325,486	0.183	0.264	0.177
207	2	0	0	0	0	1	1	0	0	0	0	0	0	0	0	−2.44	325,486	0.185	0.265	0.179
208	3	0	0	0	0	1	1	0	0	0	0	0	0	0	0	−1.76	325,485	0.184	0.261	0.173
209	2	0	0	0	0	1	1	1	0	0	0	0	0	0	0	−12.48	325,482	0.183	0.260	0.173
210	2	0	0	0	0	2	0	1	0	0	0	0	0	0	0	3.93	325,482	0.184	0.258	0.170
211	0	0	0	0	0	2	0	3	0	0	0	0	0	0	0	−495.92	325,481	0.184	0.257	0.168
212	0	0	0	0	0	1	1	2	0	0	0	0	0	0	0	−434.12	325,481	0.185	0.260	0.169
213	0	0	0	0	0	1	1	3	0	0	0	0	0	0	0	−2854.58	325,479	0.185	0.260	0.167
214	2	0	0	0	0	0	0	1	0	0	1	0	0	0	0	6.58	325,479	0.184	0.261	0.167
215	1	0	0	0	0	0	0	0	0	0	0	0	0	2	0	7.08	325,479	0.183	0.257	0.167
216	0	0	1	0	0	0	0	0	0	0	0	0	0	1	0	−20.06	325,479	0.184	0.257	0.167
217	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	11.90	325,468	0.186	0.257	0.166
218	0	0	1	0	0	0	0	0	0	0	0	0	0	1	1	0.20	325,468	0.186	0.257	0.166
219	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	18.33	325,468	0.186	0.257	0.165
220	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	9.56	325,468	0.185	0.258	0.165
221	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	37.24	325,463	0.194	0.265	0.168
222	0	1	0	0	0	0	0	0	0	0	0	0	0	3	0	17.46	325,460	0.196	0.265	0.168
223	1	0	0	0	0	0	0	0	0	0	0	0	0	3	0	−5.47	325,460	0.194	0.266	0.166
224	1	0	0	0	0	0	0	0	0	0	0	0	0	4	0	−11.21	325,459	0.194	0.268	0.168

Table A4. Out-of-sample validation figures of the OLS proxy function of BEL under 150–443 after each tenth iteration.

k	**v.mae**	**v.mae^a**	**v.res**	**v.mae^0**	**v.res^0**	**ns.mae**	**ns.mae^a**	**ns.res**	**ns.mae^0**	**ns.res^0**	**cr.mae**	**cr.mae^a**	**cr.res**	**cr.mae^0**	**cr.res^0**
0	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	0.839	0.802	0	21.468	104	0.389	0.376	23	21.659	113	0.650	0.636	89	27.112	179
20	0.565	0.540	−10	16.780	82	0.597	0.577	−75	8.274	2	0.454	0.445	−40	10.083	38
30	0.518	0.496	1	17.501	100	0.418	0.404	−47	7.970	37	0.264	0.259	1	13.378	85
40	0.475	0.454	−10	16.888	98	0.509	0.492	−66	6.234	27	0.291	0.285	−26	10.497	68
50	0.368	0.352	−15	13.268	78	0.391	0.378	−50	6.060	29	0.221	0.217	−9	10.674	69
60	0.306	0.293	−17	10.760	62	0.301	0.290	−36	5.863	29	0.183	0.179	5	10.651	69
70	0.291	0.278	−18	10.451	60	0.281	0.272	−33	6.060	30	0.175	0.171	8	10.958	72
80	0.263	0.251	−23	9.389	54	0.309	0.298	−41	4.837	22	0.157	0.154	−4	8.945	59
90	0.267	0.256	−24	9.196	54	0.313	0.303	−42	4.689	22	0.158	0.155	−7	8.587	57
100	0.250	0.239	−18	9.152	53	0.276	0.266	−35	4.637	22	0.136	0.133	0	8.606	57
110	0.237	0.226	−18	8.494	48	0.264	0.255	−34	4.144	18	0.129	0.126	−2	7.634	50
120	0.241	0.230	−16	8.896	50	0.267	0.258	−34	4.153	18	0.124	0.122	−2	7.679	51
130	0.250	0.239	−18	9.839	57	0.281	0.272	−37	4.810	24	0.122	0.120	−1	8.900	59
140	0.246	0.235	−15	9.855	57	0.263	0.254	−33	4.809	24	0.120	0.117	1	8.822	58
150	0.247	0.237	−14	9.924	57	0.271	0.262	−35	4.612	22	0.122	0.120	−1	8.537	56

Table A5. Out-of-sample validation figures of the OLS proxy function of AC under 150–443 after each tenth iteration.

k	**v.mae**	**v.mae^a**	**v.res**	**v.mae^0**	**v.res^0**	**ns.mae**	**ns.mae^a**	**ns.res**	**ns.mae^0**	**ns.res^0**	**cr.mae**	**cr.mae^a**	**cr.res**	**cr.mae^0**	**cr.res^0**
0	60.620	3.178	−296	100.000	−207	97.518	2.936	−453	100.000	−369	257.762	4.251	−653	100.000	−568
10	15.987	0.838	−1	29.161	−110	10.264	0.309	−6	32.492	−119	31.461	0.519	−67	31.704	−180
20	10.292	0.540	10	21.029	−82	19.163	0.577	75	12.240	−21	26.995	0.445	39	13.324	−57
30	9.444	0.495	−1	21.971	−100	13.409	0.404	47	15.583	−56	15.766	0.260	−1	18.759	−105
40	8.656	0.454	10	21.197	−98	16.359	0.492	67	12.740	−46	17.271	0.285	26	15.434	−87
50	6.718	0.352	15	16.655	−78	12.565	0.378	50	12.938	−47	13.182	0.217	9	15.666	−88
60	5.581	0.293	17	13.506	−62	9.671	0.291	36	12.985	−48	10.904	0.180	−5	15.640	−88
70	5.313	0.279	19	13.026	−59	9.095	0.274	34	13.289	−49	10.357	0.171	−8	15.975	−90
80	4.688	0.246	21	11.326	−51	9.069	0.273	36	11.131	−41	8.997	0.148	0	13.590	−77
90	4.870	0.255	24	11.525	−53	10.043	0.302	42	10.995	−41	9.414	0.155	7	13.285	−75
100	4.546	0.238	18	11.471	−53	8.847	0.266	35	11.041	−41	8.081	0.133	0	13.308	−76
110	4.313	0.226	18	10.650	−48	8.463	0.255	34	9.999	−37	7.689	0.127	2	12.181	−69
120	4.430	0.232	16	11.350	−51	8.304	0.250	33	10.596	−39	7.187	0.119	−1	12.763	−73
130	4.555	0.239	18	12.345	−57	9.024	0.272	37	11.491	−42	7.285	0.120	1	13.663	−78
140	4.532	0.238	15	12.470	−57	8.722	0.263	35	11.282	−42	7.227	0.119	0	13.448	−76
150	4.512	0.237	14	12.459	−57	8.712	0.262	35	11.136	−41	7.265	0.120	1	13.242	−75

Table A6. Out-of-sample validation figures of the OLS proxy function of BEL under 300–886 after each tenth and the final iteration.

k	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
0	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	0.839	0.802	0	21.468	104	0.389	0.376	23	21.659	113	0.650	0.636	89	27.112	179
20	0.565	0.540	−10	16.780	82	0.597	0.577	−75	8.274	2	0.454	0.445	−40	10.083	38
30	0.518	0.496	1	17.501	100	0.418	0.404	−47	7.970	37	0.264	0.259	1	13.378	85
40	0.475	0.454	−10	16.888	98	0.509	0.492	−66	6.234	27	0.291	0.285	−26	10.497	68
50	0.368	0.352	−15	13.268	78	0.391	0.378	−50	6.060	29	0.221	0.217	−9	10.674	69
60	0.306	0.293	−17	10.760	62	0.301	0.290	−36	5.863	29	0.183	0.179	5	10.651	69
70	0.291	0.278	−18	10.451	60	0.281	0.272	−33	6.060	30	0.175	0.171	8	10.958	72
80	0.263	0.251	−23	9.389	54	0.309	0.298	−41	4.837	22	0.157	0.154	−4	8.945	59
90	0.267	0.256	−24	9.196	54	0.313	0.303	−42	4.689	22	0.158	0.155	−7	8.587	57
100	0.250	0.239	−18	9.152	53	0.276	0.266	−35	4.637	22	0.136	0.133	0	8.606	57
110	0.239	0.229	−18	9.132	52	0.269	0.260	−35	4.577	22	0.132	0.129	−1	8.358	55
120	0.242	0.231	−16	9.519	54	0.273	0.263	−35	4.569	21	0.129	0.126	−1	8.380	55
130	0.251	0.240	−18	10.506	61	0.287	0.277	−37	5.421	27	0.127	0.125	0	9.724	64
140	0.246	0.235	−15	10.530	61	0.269	0.260	−34	5.329	27	0.123	0.120	2	9.526	63
150	0.242	0.232	−14	10.556	61	0.274	0.265	−35	5.119	26	0.123	0.120	0	9.261	61
160	0.243	0.232	−15	10.483	60	0.278	0.268	−36	5.018	25	0.127	0.124	0	9.144	60
170	0.238	0.228	−13	10.140	58	0.265	0.256	−33	4.968	24	0.130	0.127	2	8.884	59
180	0.241	0.230	−12	10.128	57	0.300	0.290	−37	4.552	18	0.149	0.146	2	8.716	58
190	0.201	0.192	−13	6.458	32	0.275	0.266	−33	4.124	−2	0.173	0.169	−4	4.721	27
200	0.186	0.178	−9	6.111	29	0.262	0.254	−29	4.460	−4	0.181	0.177	3	4.920	27
210	0.184	0.176	−9	6.210	30	0.258	0.249	−28	4.337	−3	0.170	0.167	3	4.846	28
220	0.185	0.177	−8	6.433	32	0.258	0.250	−28	4.286	−3	0.165	0.161	3	4.850	28
224	0.194	0.186	−9	6.659	34	0.268	0.259	−30	4.200	−2	0.168	0.165	1	5.007	29

Table A7. Out-of-sample validation figures of the derived OLS proxy functions of BEL under 150–443 and 300–886 after the final iteration based on three different sets of validation value estimates. Thereby emerges the first set of validation value estimates from pointwise subtraction of 1.96 times the standard errors from the original set of validation values. The second set is the original set. The third set is the addition counterpart of the first set.

k	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
150–443 figures based on validation values minus 1.96 times standard errors															
150	0.286	0.273	−30	9.878	57	0.330	0.319	−46	3.915	16	0.151	0.148	−13	7.473	49
150–443 figures based on validation values															
150	0.247	0.237	−14	9.924	57	0.271	0.262	−35	4.612	22	0.122	0.120	−1	8.537	56
150–443 figures based on validation values plus 1.96 times standard errors															
150	0.231	0.221	1	9.977	57	0.219	0.212	−24	5.473	28	0.130	0.127	11	9.591	64
300–886 figures based on validation values minus 1.96 times standard errors															
224	0.236	0.225	−24	6.757	34	0.325	0.314	−41	4.610	−8	0.191	0.187	−11	4.307	22
300–886 figures based on validation values															
224	0.194	0.186	−9	6.659	34	0.268	0.259	−30	4.200	−2	0.168	0.165	1	5.007	29
300–886 figures based on validation values plus 1.96 times standard errors															
224	0.184	0.177	7	6.625	35	0.218	0.211	−19	3.982	4	0.173	0.169	13	5.813	37

Table A8. AIC scores and out-of-sample validation figures of the gaussian generalized linear models (GLMs) of BEL with identity, inverse and log link functions under 150–443 after each tenth iteration.

k	AIC	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
Gaussian with identity link																
0	437,251	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	345,045	0.839	0.802	0	21.468	104	0.389	0.376	23	21.659	113	0.650	0.636	89	27.112	179
20	333,447	0.565	0.540	−10	16.780	82	0.597	0.577	−75	8.274	2	0.454	0.445	−40	10.083	38
30	330,361	0.518	0.496	1	17.501	100	0.418	0.404	−47	7.970	37	0.264	0.259	1	13.378	85
40	328,832	0.475	0.454	−10	16.888	98	0.509	0.492	−66	6.234	27	0.291	0.285	−26	10.497	68
50	327,432	0.368	0.352	−15	13.268	78	0.391	0.378	−50	6.060	29	0.221	0.217	−9	10.674	69
60	326,787	0.306	0.293	−17	10.760	62	0.301	0.290	−36	5.863	29	0.183	0.179	5	10.651	69
70	326,453	0.291	0.278	−18	10.451	60	0.281	0.272	−33	6.060	30	0.175	0.171	8	10.958	72
80	326,245	0.263	0.251	−23	9.389	54	0.309	0.298	−41	4.837	22	0.157	0.154	−4	8.945	59
90	326,116	0.267	0.256	−24	9.196	54	0.313	0.303	−42	4.689	22	0.158	0.155	−7	8.587	57
100	326,038	0.250	0.239	−18	9.152	53	0.276	0.266	−35	4.637	22	0.136	0.133	0	8.606	57
110	325,968	0.237	0.226	−18	8.494	48	0.264	0.255	−34	4.144	18	0.129	0.126	−2	7.634	50
120	325,928	0.241	0.230	−16	8.896	50	0.267	0.258	−34	4.153	18	0.124	0.122	−2	7.679	51
130	325,896	0.250	0.239	−18	9.839	57	0.281	0.272	−37	4.810	24	0.122	0.120	−1	8.900	59
140	325,873	0.246	0.235	−15	9.855	57	0.263	0.254	−33	4.809	24	0.120	0.117	1	8.822	58
150	325,850	0.247	0.237	−14	9.924	57	0.271	0.262	−35	4.612	22	0.122	0.120	−1	8.537	56
Gaussian with inverse link																
0	437,251	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	343,426	1.036	0.990	1	33.705	192	0.650	0.628	−63	21.481	114	0.391	0.382	44	33.482	221
20	334,985	0.689	0.659	−6	21.313	118	0.515	0.498	−62	10.319	49	0.324	0.317	−4	16.493	107
30	331,426	0.512	0.490	−16	18.836	109	0.393	0.380	−45	12.277	65	0.248	0.243	15	18.960	125
40	328,875	0.433	0.414	−5	14.354	82	0.317	0.306	−26	9.312	47	0.294	0.288	26	15.188	99
50	327,877	0.383	0.366	−8	12.959	76	0.285	0.276	−24	8.961	46	0.271	0.265	25	14.592	95
60	327,274	0.337	0.323	−16	12.572	73	0.328	0.316	−37	7.636	38	0.219	0.215	10	13.087	85
70	326,875	0.290	0.277	−14	11.248	64	0.271	0.261	−32	6.233	31	0.156	0.153	6	10.588	70
80	326,603	0.259	0.248	−16	9.976	58	0.287	0.278	−38	5.042	22	0.158	0.155	−8	8.014	52
90	326,390	0.254	0.243	−20	8.462	47	0.392	0.379	−51	4.451	1	0.220	0.215	−17	5.676	36
100	326,225	0.270	0.258	−21	8.884	49	0.393	0.379	−51	4.454	5	0.219	0.215	−12	6.732	44
110	326,152	0.272	0.260	−20	8.558	47	0.375	0.363	−48	4.441	4	0.208	0.204	−10	6.545	42
120	326,094	0.267	0.255	−19	8.418	47	0.380	0.367	−49	4.414	3	0.209	0.205	−12	6.194	40
130	326,058	0.266	0.254	−19	8.638	48	0.379	0.367	−49	4.329	4	0.203	0.199	−11	6.362	41
140	325,982	0.258	0.247	−17	8.353	45	0.363	0.351	−46	4.380	2	0.197	0.193	−10	6.059	38
150	325,952	0.258	0.247	−16	8.468	45	0.353	0.341	−44	4.282	3	0.192	0.188	−8	6.088	39
Gaussian with log link																
0	437,251	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	342,325	0.879	0.840	26	25.171	132	0.422	0.408	−17	15.628	74	0.530	0.519	52	22.034	143
20	334,417	0.661	0.632	−5	22.474	125	0.532	0.514	−64	10.764	51	0.330	0.323	−3	17.317	112
30	330,901	0.560	0.536	−3	21.780	126	0.474	0.458	−55	11.199	59	0.266	0.261	3	17.802	117
40	328,444	0.411	0.393	−10	13.639	78	0.315	0.304	−29	8.610	44	0.264	0.258	19	14.162	92
50	327,574	0.341	0.326	−16	12.936	75	0.334	0.323	−35	8.294	42	0.262	0.257	12	13.642	89
60	327,029	0.315	0.302	−17	11.991	69	0.312	0.301	−36	7.024	36	0.192	0.188	10	12.465	82
70	326,637	0.279	0.267	−16	10.620	61	0.266	0.257	−31	6.142	31	0.162	0.158	9	10.797	71
80	326,449	0.266	0.254	−21	10.069	59	0.304	0.294	−40	5.195	25	0.153	0.149	−4	9.234	61
90	326,287	0.273	0.261	−22	9.742	57	0.300	0.290	−40	5.082	25	0.141	0.138	−5	8.990	59
100	326,082	0.269	0.257	−23	8.052	45	0.370	0.358	−48	4.094	6	0.210	0.205	−13	6.314	41
110	326,021	0.258	0.247	−19	8.043	44	0.343	0.331	−43	4.102	5	0.198	0.193	−7	6.381	41
120	325,950	0.252	0.241	−17	7.891	42	0.329	0.318	−41	4.086	3	0.191	0.187	−7	5.883	37
130	325,881	0.251	0.240	−18	8.049	45	0.359	0.347	−46	4.238	2	0.194	0.190	−10	5.924	38
140	325,849	0.245	0.234	−17	7.978	44	0.340	0.328	−43	4.045	4	0.183	0.179	−7	6.131	40
150	325,823	0.240	0.229	−15	7.980	44	0.316	0.305	−38	4.014	6	0.170	0.167	−2	6.434	42

Table A9. AIC scores and out-of-sample validation figures of the gamma GLMs of BEL with identity, inverse and log link functions under 150–443 after each tenth iteration.

k	AIC	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
Gamma with identity link																
0	437,243	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	345,605	0.872	0.834	1	23.485	114	0.315	0.304	6	19.861	105	0.530	0.519	68	25.266	167
20	333,911	0.553	0.529	−12	16.265	79	0.599	0.579	−76	8.268	0	0.464	0.454	−43	9.895	34
30	330,707	0.503	0.481	0	17.404	99	0.425	0.411	−49	7.754	35	0.267	0.262	−2	12.959	82
40	328,589	0.376	0.359	−13	13.317	76	0.341	0.330	−39	7.187	35	0.238	0.233	6	12.341	80
50	327,668	0.348	0.333	−15	13.173	77	0.356	0.344	−44	6.656	34	0.227	0.222	−4	11.348	74
60	327,135	0.305	0.292	−16	11.190	65	0.304	0.294	−37	6.059	30	0.175	0.172	3	10.843	71
70	326,686	0.273	0.261	−15	9.730	55	0.257	0.249	−30	5.364	26	0.165	0.161	9	9.928	65
80	326,461	0.268	0.257	−21	9.471	54	0.287	0.277	−36	5.151	25	0.149	0.146	2	9.549	63
90	326,328	0.259	0.248	−23	8.889	52	0.304	0.293	−40	4.373	20	0.148	0.145	−6	8.255	55
100	326,246	0.238	0.227	−20	8.321	48	0.262	0.253	−34	4.279	19	0.137	0.134	−1	7.845	52
110	326,184	0.233	0.223	−18	8.045	45	0.255	0.246	−33	3.907	16	0.130	0.127	−1	7.182	47
120	326,135	0.228	0.218	−16	8.191	46	0.253	0.245	−33	3.696	15	0.129	0.126	−2	6.870	45
130	326,093	0.244	0.233	−17	9.530	55	0.272	0.263	−35	4.628	22	0.124	0.122	0	8.596	57
140	326,068	0.238	0.228	−17	9.416	54	0.271	0.261	−35	4.523	22	0.125	0.123	−1	8.371	55
150	326,041	0.236	0.226	−14	9.329	53	0.260	0.251	−33	4.321	20	0.121	0.118	1	8.206	54
Gamma with inverse link																
0	437,243	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	343,969	1.037	0.991	0	33.818	193	0.661	0.639	−64	21.601	115	0.397	0.389	44	33.752	223
20	335,495	0.679	0.649	−7	20.888	115	0.530	0.512	−65	9.637	43	0.335	0.328	−9	15.410	99
30	332,646	0.627	0.600	−9	26.098	152	0.621	0.600	−82	12.361	64	0.346	0.339	−24	18.470	122
40	329,192	0.409	0.391	−10	14.061	81	0.317	0.306	−27	9.719	50	0.289	0.283	23	15.405	101
50	328,114	0.339	0.324	−12	12.599	73	0.313	0.302	−30	8.084	40	0.271	0.265	15	13.146	85
60	327,513	0.328	0.313	−16	12.247	71	0.294	0.284	−29	8.341	43	0.240	0.235	18	13.902	91
70	327,115	0.285	0.272	−12	11.127	64	0.251	0.243	−28	6.463	33	0.166	0.162	11	10.915	72
80	326,795	0.252	0.241	−17	8.376	45	0.315	0.305	−39	4.069	9	0.196	0.192	−8	6.416	40
90	326,615	0.250	0.239	−20	8.113	45	0.384	0.371	−51	4.414	0	0.218	0.213	−16	5.478	34
100	326,445	0.263	0.252	−20	8.724	48	0.382	0.369	−49	4.410	5	0.211	0.206	−11	6.595	43
110	326,370	0.266	0.255	−19	8.251	45	0.369	0.357	−47	4.494	2	0.205	0.201	−9	6.288	40
120	326,310	0.258	0.247	−17	8.003	44	0.357	0.345	−45	4.435	2	0.196	0.192	−8	6.087	39
130	326,277	0.259	0.248	−17	8.331	47	0.357	0.344	−45	4.356	4	0.187	0.183	−7	6.509	42
140	326,246	0.262	0.250	−17	8.583	48	0.357	0.345	−45	4.304	5	0.183	0.179	−7	6.620	43
150	326,222	0.254	0.243	−15	8.410	46	0.327	0.316	−40	4.111	7	0.171	0.167	−3	6.722	44
Gamma with log link																
0	437,243	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
1	388,234	2.365	2.261	−4	67.494	277	0.773	0.747	22	54.214	287	1.193	1.168	170	65.932	435
10	342,942	0.870	0.832	21	24.998	131	0.440	0.425	−24	15.145	71	0.505	0.494	43	21.396	138
20	334,881	0.649	0.621	−5	19.899	110	0.519	0.501	−65	8.283	36	0.312	0.306	−11	14.105	90
30	331,227	0.544	0.520	−4	21.752	126	0.479	0.463	−57	11.010	58	0.262	0.257	0	17.458	115
40	328,727	0.374	0.357	−10	14.009	81	0.329	0.318	−33	8.553	43	0.268	0.263	15	13.990	91
50	327,806	0.328	0.313	−16	12.750	74	0.327	0.316	−33	8.325	42	0.272	0.266	14	13.779	90
60	327,270	0.302	0.289	−15	11.825	68	0.297	0.287	−33	7.147	37	0.197	0.193	14	12.637	83
70	326,866	0.264	0.253	−15	10.159	58	0.249	0.241	−28	6.071	31	0.165	0.162	12	10.693	70
80	326,669	0.255	0.244	−19	9.819	57	0.288	0.279	−37	5.085	24	0.146	0.143	−2	9.090	60
90	326,433	0.266	0.254	−23	8.891	51	0.327	0.316	−45	4.079	15	0.171	0.167	−12	7.353	48
100	326,302	0.265	0.253	−23	7.839	44	0.361	0.349	−47	4.030	5	0.205	0.201	−12	6.246	40
110	326,224	0.256	0.244	−18	8.139	45	0.335	0.324	−41	4.211	8	0.191	0.187	−3	7.043	46
120	326,147	0.250	0.239	−18	7.817	43	0.340	0.328	−43	4.122	4	0.188	0.184	−6	6.247	41
130	326,111	0.247	0.236	−17	7.750	43	0.341	0.329	−43	4.115	3	0.186	0.183	−7	6.060	39
140	326,050	0.247	0.236	−17	7.730	43	0.336	0.324	−42	4.073	4	0.179	0.176	−6	6.117	40
150	326,022	0.243	0.232	−15	7.820	43	0.323	0.312	−40	4.040	3	0.174	0.170	−4	6.010	39

Table A10. AIC scores and out-of-sample validation figures of the inverse gaussian GLMs of BEL with identity, inverse, log and $\frac{1}{\mu^2}$ link functions under 150–443 after each tenth iteration.

k	AIC	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
inverse gaussian with identity link																
0	437, 338	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	346, 132	0.871	0.833	1	23.559	115	0.314	0.304	7	20.269	107	0.534	0.523	70	25.673	169
20	334, 430	0.549	0.524	−13	15.996	77	0.599	0.579	−77	8.273	−1	0.468	0.458	−44	9.809	32
30	331, 453	0.488	0.467	−4	15.939	89	0.517	0.499	−67	6.532	11	0.413	0.405	−40	9.280	38
40	328, 985	0.370	0.354	−13	13.279	76	0.338	0.327	−39	7.193	35	0.238	0.233	6	12.301	80
50	328, 064	0.332	0.317	−15	12.727	74	0.338	0.327	−40	6.871	35	0.232	0.227	1	11.664	76
60	327, 533	0.298	0.285	−17	10.994	64	0.304	0.294	−37	5.868	29	0.172	0.168	3	10.646	69
70	327, 082	0.274	0.262	−15	9.387	53	0.243	0.235	−27	5.535	27	0.171	0.167	13	10.253	67
80	326, 849	0.267	0.255	−20	9.426	54	0.278	0.268	−34	5.271	25	0.152	0.148	5	9.783	65
90	326, 715	0.247	0.236	−21	8.546	49	0.275	0.266	−35	4.399	20	0.140	0.137	−1	8.302	55
100	326, 630	0.236	0.225	−20	7.879	45	0.262	0.253	−34	3.979	16	0.140	0.137	−2	7.249	48
110	326, 564	0.225	0.215	−17	7.728	43	0.243	0.235	−31	3.850	15	0.129	0.126	0	6.958	46
120	326, 507	0.237	0.226	−18	8.776	50	0.270	0.260	−35	4.120	19	0.130	0.127	−3	7.710	51
130	326, 475	0.240	0.230	−17	9.225	53	0.265	0.256	−34	4.516	21	0.123	0.120	0	8.400	55
140	326, 447	0.241	0.230	−16	9.415	54	0.270	0.261	−35	4.543	21	0.124	0.122	−1	8.426	56
150	326, 352	0.249	0.238	−17	9.375	54	0.337	0.326	−44	4.224	12	0.150	0.146	−4	7.930	52
Inverse gaussian with inverse link																
0	437, 338	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	344, 458	1.129	1.079	−25	35.685	202	1.138	1.099	−150	14.423	63	0.639	0.626	−63	22.713	149
20	336, 004	0.682	0.652	−5	21.011	117	0.534	0.516	−67	8.866	41	0.321	0.314	−12	14.895	95
30	333, 060	0.626	0.598	−10	24.463	142	0.623	0.602	−83	10.859	55	0.376	0.369	−31	16.233	107
40	329, 632	0.412	0.394	−14	15.912	93	0.345	0.333	−29	12.096	64	0.318	0.311	28	18.446	121
50	328, 515	0.335	0.320	−12	12.387	71	0.305	0.295	−29	8.122	40	0.276	0.270	18	13.333	86
60	327, 916	0.321	0.307	−15	11.970	70	0.286	0.276	−27	8.385	44	0.247	0.241	20	13.973	91
70	327, 543	0.278	0.266	−12	10.488	60	0.246	0.238	−28	6.106	31	0.164	0.161	9	10.331	67
80	327, 196	0.249	0.238	−17	8.227	45	0.308	0.297	−38	4.037	9	0.193	0.189	−7	6.381	40
90	327, 012	0.247	0.236	−19	8.016	44	0.376	0.363	−49	4.390	−1	0.212	0.207	−15	5.407	33
100	326, 837	0.261	0.250	−20	8.469	46	0.375	0.363	−48	4.428	4	0.208	0.204	−10	6.569	43
110	326, 762	0.262	0.250	−18	8.090	44	0.365	0.353	−46	4.505	2	0.201	0.197	−8	6.242	40
120	326, 699	0.259	0.248	−18	8.106	45	0.367	0.355	−47	4.402	2	0.192	0.188	−9	6.082	39
130	326, 667	0.259	0.247	−17	7.987	44	0.352	0.340	−44	4.303	2	0.187	0.183	−8	5.958	38
140	326, 642	0.258	0.246	−16	8.243	46	0.340	0.328	−42	4.228	6	0.173	0.169	−5	6.602	43
150	326, 617	0.253	0.242	−15	8.152	44	0.324	0.313	−39	4.148	5	0.172	0.169	−3	6.476	42
Inverse gaussian with log link																
0	437, 338	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	343, 530	0.866	0.828	19	24.925	131	0.450	0.435	−28	14.940	69	0.494	0.484	39	21.122	136
20	335, 355	0.644	0.616	−5	19.653	109	0.526	0.509	−67	7.947	33	0.318	0.311	−14	13.490	85
30	331, 675	0.536	0.512	−4	21.697	125	0.482	0.465	−58	10.885	57	0.262	0.256	−2	17.245	113
40	329, 140	0.366	0.350	−10	13.913	80	0.325	0.314	−32	8.604	44	0.269	0.264	16	14.011	91
50	328, 190	0.324	0.310	−16	12.640	73	0.319	0.308	−32	8.482	43	0.274	0.268	16	13.966	91
60	327, 666	0.296	0.283	−15	11.626	67	0.290	0.280	−31	7.181	37	0.201	0.197	15	12.695	83
70	327, 263	0.261	0.250	−15	9.948	57	0.244	0.236	−27	6.042	30	0.172	0.168	12	10.531	69
80	327, 061	0.251	0.240	−18	9.746	56	0.284	0.275	−37	4.988	24	0.145	0.142	−1	8.964	59
90	326, 825	0.263	0.251	−23	8.769	51	0.321	0.310	−44	4.059	15	0.168	0.165	−11	7.316	48
100	326, 695	0.261	0.249	−22	7.727	43	0.352	0.340	−45	4.048	6	0.203	0.199	−10	6.341	41
110	326, 598	0.239	0.229	−17	7.408	40	0.343	0.332	−43	4.444	−1	0.185	0.181	−7	5.572	35
120	326, 530	0.249	0.238	−18	7.520	41	0.343	0.331	−43	4.247	1	0.191	0.187	−7	5.928	38
130	326, 494	0.246	0.235	−17	7.602	42	0.337	0.326	−43	4.108	2	0.183	0.179	−6	5.964	39
140	326, 471	0.246	0.235	−17	7.772	43	0.332	0.321	−42	4.068	4	0.177	0.173	−6	6.092	39
150	326, 413	0.247	0.237	−15	7.716	42	0.324	0.313	−40	4.095	2	0.172	0.168	−4	5.892	38
Inverse gaussian with $\frac{1}{\mu^2}$ link																
0	437, 338	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	344, 467	0.985	0.941	−14	31.473	176	0.993	0.959	−130	12.573	46	0.561	0.549	−52	18.986	124
20	336, 815	0.668	0.639	−7	21.404	122	0.591	0.571	−75	9.506	38	0.372	0.364	−22	14.521	91
30	331, 792	0.478	0.457	−5	15.821	90	0.367	0.354	−28	10.573	53	0.373	0.365	33	17.496	114
40	330, 089	0.421	0.403	−1	15.183	89	0.295	0.285	−19	10.660	56	0.316	0.309	34	16.657	109
50	329, 020	0.376	0.359	−10	14.443	85	0.300	0.290	−21	11.439	60	0.320	0.313	34	17.553	115
60	328, 452	0.330	0.316	−12	12.905	75	0.290	0.280	−24	9.196	48	0.273	0.267	25	14.952	98
70	327, 925	0.316	0.302	−16	11.733	69	0.301	0.291	−35	7.090	35	0.200	0.195	6	11.701	76
80	327, 639	0.262	0.250	−18	8.128	43	0.298	0.288	−35	4.425	11	0.208	0.203	−1	7.205	45
90	327, 265	0.278	0.266	−22	8.311	46	0.355	0.343	−44	4.383	9	0.202	0.197	−7	7.090	46
100	327, 148	0.288	0.275	−22	8.166	44	0.357	0.345	−44	4.408	8	0.207	0.203	−6	7.039	46
110	327, 078	0.274	0.262	−20	7.943	43	0.354	0.342	−44	4.451	4	0.196	0.192	−7	6.434	41
120	326, 920	0.269	0.257	−18	8.350	46	0.374	0.361	−47	4.579	3	0.198	0.193	−9	6.419	41
130	326, 887	0.270	0.258	−18	8.437	47	0.360	0.348	−44	4.544	6	0.196	0.192	−4	7.151	46
140	326, 807	0.267	0.255	−18	8.193	45	0.345	0.333	−43	4.318	5	0.188	0.184	−5	6.661	43
150	326, 778	0.262	0.250	−16	8.258	44	0.332	0.321	−41	4.238	5	0.177	0.174	−3	6.518	42

Table A11. AIC scores and out-of-sample validation figures of the gaussian GLMs of BEL with identity, inverse and log link functions under 300–886 after each tenth and the final iteration.

k	AIC	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
Gaussian with identity link																
0	437,251	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	345,045	0.839	0.802	0	21.468	104	0.389	0.376	23	21.659	113	0.650	0.636	89	27.112	179
20	333,447	0.565	0.540	−10	16.780	82	0.597	0.577	−75	8.274	2	0.454	0.445	−40	10.083	38
30	330,361	0.518	0.496	1	17.501	100	0.418	0.404	−47	7.970	37	0.264	0.259	1	13.378	85
40	328,832	0.475	0.454	−10	16.888	98	0.509	0.492	−66	6.234	27	0.291	0.285	−26	10.497	68
50	327,432	0.368	0.352	−15	13.268	78	0.391	0.378	−50	6.060	29	0.221	0.217	−9	10.674	69
60	326,787	0.306	0.293	−17	10.760	62	0.301	0.290	−36	5.863	29	0.183	0.179	5	10.651	69
70	326,453	0.291	0.278	−18	10.451	60	0.281	0.272	−33	6.060	30	0.175	0.171	8	10.958	72
80	326,245	0.263	0.251	−23	9.389	54	0.309	0.298	−41	4.837	22	0.157	0.154	−4	8.945	59
90	326,116	0.267	0.256	−24	9.196	54	0.313	0.303	−42	4.689	22	0.158	0.155	−7	8.587	57
100	326,038	0.250	0.239	−18	9.152	53	0.276	0.266	−35	4.637	22	0.136	0.133	0	8.606	57
110	325,963	0.239	0.229	−18	9.132	52	0.269	0.260	−35	4.577	22	0.132	0.129	−1	8.358	55
120	325,922	0.242	0.231	−16	9.519	54	0.273	0.263	−35	4.569	21	0.129	0.126	−1	8.380	55
130	325,889	0.251	0.240	−18	10.506	61	0.287	0.277	−37	5.421	27	0.127	0.125	0	9.724	64
140	325,865	0.246	0.235	−15	10.530	61	0.269	0.260	−34	5.329	27	0.123	0.120	2	9.526	63
150	325,841	0.242	0.232	−14	10.556	61	0.274	0.265	−35	5.119	26	0.123	0.120	0	9.261	61
160	325,821	0.243	0.232	−15	10.483	60	0.278	0.268	−36	5.018	25	0.127	0.124	0	9.144	60
170	325,811	0.238	0.228	−13	10.140	58	0.265	0.256	−33	4.968	24	0.130	0.127	2	8.884	59
180	325,766	0.241	0.230	−12	10.128	57	0.300	0.290	−37	4.552	18	0.149	0.146	2	8.716	58
190	325,506	0.201	0.192	−13	6.458	32	0.275	0.266	−33	4.124	−2	0.173	0.169	−4	4.721	27
200	325,488	0.186	0.178	−9	6.111	29	0.262	0.254	−29	4.460	−4	0.181	0.177	3	4.920	27
210	325,482	0.184	0.176	−9	6.210	30	0.258	0.249	−28	4.337	−3	0.170	0.167	3	4.846	28
220	325,468	0.185	0.177	−8	6.433	32	0.258	0.250	−28	4.286	−3	0.165	0.161	3	4.850	28
224	325,459	0.194	0.186	−9	6.659	34	0.268	0.259	−30	4.200	−2	0.168	0.165	1	5.007	29
Gaussian with inverse link																
0	437,251	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	343,426	1.036	0.990	1	33.705	192	0.650	0.628	−63	21.481	114	0.391	0.382	44	33.482	221
20	334,985	0.689	0.659	−6	21.313	118	0.515	0.498	−62	10.319	49	0.324	0.317	−4	16.493	107
30	331,426	0.512	0.490	−16	18.836	109	0.393	0.380	−45	12.277	65	0.248	0.243	15	18.960	125
40	328,875	0.433	0.414	−5	14.354	82	0.317	0.306	−26	9.312	47	0.294	0.288	26	15.188	99
50	327,877	0.383	0.366	−8	12.959	76	0.285	0.276	−24	8.961	46	0.271	0.265	25	14.592	95
60	327,274	0.337	0.323	−16	12.572	73	0.328	0.316	−37	7.636	38	0.219	0.215	10	13.087	85
70	326,875	0.290	0.277	−14	11.248	64	0.271	0.261	−32	6.233	31	0.156	0.153	6	10.588	70
80	326,603	0.259	0.248	−16	9.976	58	0.287	0.278	−38	5.042	22	0.158	0.155	−8	8.014	52
90	326,390	0.254	0.243	−20	8.462	47	0.392	0.379	−51	4.451	1	0.220	0.215	−17	5.676	36
100	326,224	0.269	0.257	−21	9.365	53	0.403	0.389	−52	4.500	7	0.225	0.220	−12	7.174	47
110	326,135	0.266	0.254	−19	8.894	49	0.377	0.364	−49	4.334	5	0.205	0.201	−12	6.497	42
120	326,069	0.266	0.254	−19	8.564	48	0.381	0.368	−50	4.271	4	0.204	0.200	−14	6.102	39
130	326,033	0.265	0.253	−19	8.498	47	0.386	0.373	−50	4.445	2	0.212	0.207	−14	5.917	38
140	325,950	0.253	0.242	−17	8.151	44	0.358	0.346	−46	4.345	1	0.189	0.185	−11	5.598	35
150	325,924	0.255	0.244	−17	8.485	46	0.364	0.352	−46	4.288	3	0.192	0.188	−11	5.894	38
160	325,886	0.258	0.247	−15	8.842	48	0.349	0.337	−44	4.199	5	0.178	0.174	−8	6.359	41
170	325,869	0.249	0.238	−14	8.503	46	0.331	0.320	−40	4.254	5	0.174	0.171	−5	6.182	40
180	325,850	0.248	0.237	−12	8.505	45	0.312	0.302	−37	4.099	6	0.164	0.161	−3	6.095	40
190	325,820	0.238	0.228	−12	8.240	43	0.313	0.303	−37	4.137	4	0.169	0.166	−3	5.825	38
200	325,803	0.244	0.234	−13	8.458	45	0.320	0.309	−38	4.073	6	0.171	0.167	−4	6.132	40
210	325,800	0.241	0.231	−13	8.376	45	0.313	0.302	−36	4.059	6	0.171	0.167	−2	6.248	41
213	325,797	0.241	0.230	−12	8.325	44	0.310	0.299	−36	4.063	6	0.171	0.167	−1	6.284	41
Gaussian with log link																
0	437,251	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	342,325	0.879	0.840	26	25.171	132	0.422	0.408	−17	15.628	74	0.530	0.519	52	22.034	143
20	334,417	0.661	0.632	−5	22.474	125	0.532	0.514	−64	10.764	51	0.330	0.323	−3	17.317	112
30	330,901	0.560	0.536	−3	21.780	126	0.474	0.458	−55	11.199	59	0.266	0.261	3	17.802	117
40	328,444	0.411	0.393	−10	13.639	78	0.315	0.304	−29	8.610	44	0.264	0.258	19	14.162	92
50	327,574	0.341	0.326	−16	12.936	75	0.334	0.323	−35	8.294	42	0.262	0.257	12	13.642	89
60	327,029	0.315	0.302	−17	11.991	69	0.312	0.301	−36	7.024	36	0.192	0.188	10	12.465	82
70	326,637	0.279	0.267	−16	10.620	61	0.266	0.257	−31	6.142	31	0.162	0.158	9	10.797	71
80	326,449	0.266	0.254	−21	10.069	59	0.304	0.294	−40	5.195	25	0.153	0.149	−4	9.234	61
90	326,287	0.273	0.261	−22	9.742	57	0.300	0.290	−40	5.082	25	0.141	0.138	−5	8.990	59
100	326,082	0.269	0.257	−23	8.052	45	0.370	0.358	−48	4.094	6	0.210	0.205	−13	6.314	41
110	326,021	0.258	0.247	−19	8.043	44	0.343	0.331	−43	4.102	5	0.198	0.193	−7	6.381	41
120	325,950	0.252	0.241	−17	7.891	42	0.329	0.318	−41	4.086	3	0.191	0.187	−7	5.883	37
130	325,743	0.208	0.199	−13	6.208	30	0.310	0.299	−38	4.994	−10	0.191	0.187	−8	4.273	21
140	325,693	0.211	0.202	−13	6.620	34	0.302	0.292	−36	4.522	−3	0.186	0.182	−3	5.037	30
150	325,665	0.210	0.200	−13	6.729	35	0.298	0.288	−36	4.385	−2	0.180	0.176	−3	5.168	31
160	325,626	0.214	0.205	−14	6.549	33	0.302	0.292	−36	4.410	−3	0.183	0.179	−4	5.076	30
170	325,610	0.214	0.204	−14	6.590	33	0.291	0.281	−35	4.273	−3	0.173	0.169	−2	5.028	30
180	325,584	0.214	0.204	−13	6.587	33	0.296	0.286	−35	4.386	−4	0.176	0.172	−2	4.973	29
190	325,575	0.212	0.203	−12	6.502	32	0.283	0.273	−33	4.363	−4	0.173	0.170	0	4.950	29
200	325,567	0.201	0.192	−9	6.272	30	0.264	0.255	−29	4.491	−4	0.171	0.168	3	4.863	27
210	325,553	0.205	0.196	−9	6.655	32	0.267	0.258	−29	4.398	−2	0.176	0.173	3	5.165	30
214	325,552	0.206	0.197	−10	6.640	32	0.267	0.258	−29	4.402	−2	0.177	0.173	3	5.180	30

Table A12. AIC scores and out-of-sample validation figures of the gamma GLMs of BEL with identity, inverse and log link functions under 300–886 after each tenth and the final iteration.

k	AIC	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
Gamma with identity link																
0	437,243	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	345,605	0.872	0.834	1	23.485	114	0.315	0.304	6	19.861	105	0.530	0.519	68	25.266	167
20	333,911	0.553	0.529	−12	16.265	79	0.599	0.579	−76	8.268	0	0.464	0.454	−43	9.895	34
30	330,707	0.503	0.481	0	17.404	99	0.425	0.411	−49	7.754	35	0.267	0.262	−2	12.959	82
40	328,589	0.376	0.359	−13	13.317	76	0.341	0.330	−39	7.187	35	0.238	0.233	6	12.341	80
50	327,668	0.348	0.333	−15	13.173	77	0.356	0.344	−44	6.656	34	0.227	0.222	−4	11.348	74
60	327,135	0.305	0.292	−16	11.190	65	0.304	0.294	−37	6.059	30	0.175	0.172	3	10.843	71
70	326,686	0.273	0.261	−15	9.730	55	0.257	0.249	−30	5.364	26	0.165	0.161	9	9.928	65
80	326,461	0.268	0.257	−21	9.471	54	0.287	0.277	−36	5.151	25	0.149	0.146	2	9.549	63
90	326,328	0.259	0.248	−23	8.889	52	0.304	0.293	−40	4.373	20	0.148	0.145	−6	8.255	55
100	326,244	0.240	0.229	−20	9.273	54	0.282	0.273	−37	4.759	22	0.144	0.141	−2	8.662	57
110	326,178	0.236	0.225	−18	8.837	51	0.262	0.254	−34	4.454	20	0.135	0.132	0	8.139	54
120	326,117	0.237	0.226	−18	9.668	56	0.275	0.266	−36	4.845	24	0.129	0.126	−1	8.799	58
130	326,084	0.245	0.235	−17	10.148	59	0.270	0.260	−35	5.236	26	0.122	0.120	1	9.375	62
140	326,058	0.243	0.232	−17	10.153	58	0.273	0.264	−35	5.092	25	0.125	0.122	−1	9.122	60
150	326,031	0.239	0.229	−14	10.130	58	0.263	0.254	−33	4.914	24	0.121	0.118	2	9.014	60
160	325,871	0.232	0.222	−15	7.898	44	0.317	0.307	−39	3.918	5	0.174	0.170	−4	6.237	40
170	325,729	0.199	0.190	−13	6.235	30	0.280	0.271	−34	4.288	−5	0.176	0.172	−2	4.684	27
180	325,718	0.201	0.192	−13	6.171	30	0.279	0.270	−34	4.253	−5	0.172	0.169	−2	4.623	27
190	325,703	0.197	0.189	−12	6.158	30	0.278	0.268	−33	4.269	−5	0.171	0.168	−3	4.521	26
200	325,697	0.194	0.185	−11	5.943	28	0.264	0.255	−30	4.416	−5	0.169	0.165	0	4.470	25
210	325,689	0.190	0.181	−10	5.992	28	0.261	0.252	−29	4.381	−5	0.169	0.165	1	4.534	25
212	325,689	0.189	0.180	−11	5.975	28	0.261	0.252	−29	4.384	−5	0.169	0.165	1	4.545	25
Gamma with inverse link																
0	437,243	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	343,969	1.037	0.991	0	33.818	193	0.661	0.639	−64	21.601	115	0.397	0.389	44	33.752	223
20	335,495	0.679	0.649	−7	20.888	115	0.530	0.512	−65	9.637	43	0.335	0.328	−9	15.410	99
30	332,646	0.627	0.600	−9	26.098	152	0.621	0.600	−82	12.361	64	0.346	0.339	−24	18.470	122
40	329,192	0.409	0.391	−10	14.061	81	0.317	0.306	−27	9.719	50	0.289	0.283	23	15.405	101
50	328,114	0.339	0.324	−12	12.599	73	0.313	0.302	−30	8.084	40	0.271	0.265	15	13.146	85
60	327,513	0.328	0.313	−16	12.247	71	0.294	0.284	−29	8.341	43	0.240	0.235	18	13.902	91
70	327,115	0.285	0.272	−12	11.127	64	0.251	0.243	−28	6.463	33	0.166	0.162	11	10.915	72
80	326,795	0.252	0.241	−17	8.376	45	0.315	0.305	−39	4.069	9	0.196	0.192	−8	6.416	40
90	326,615	0.250	0.239	−20	8.113	45	0.384	0.371	−51	4.414	0	0.218	0.213	−16	5.478	34
100	326,445	0.263	0.252	−20	9.213	52	0.387	0.374	−50	4.469	8	0.219	0.214	−10	7.316	48
110	326,355	0.272	0.260	−21	8.812	49	0.384	0.371	−50	4.313	5	0.209	0.205	−14	6.489	42
120	326,297	0.267	0.255	−20	8.378	46	0.377	0.365	−48	4.470	2	0.206	0.202	−11	6.140	39
130	326,248	0.259	0.248	−17	8.210	45	0.365	0.352	−46	4.437	1	0.200	0.196	−10	5.933	38
140	326,214	0.258	0.247	−17	8.212	45	0.355	0.343	−45	4.404	3	0.192	0.188	−9	6.077	39
150	326,190	0.260	0.248	−17	8.701	49	0.349	0.337	−44	4.217	7	0.180	0.176	−7	6.781	44
160	326,147	0.247	0.236	−15	8.556	47	0.329	0.317	−40	4.091	7	0.174	0.170	−4	6.643	43
170	326,070	0.247	0.236	−15	8.355	46	0.332	0.321	−41	4.077	5	0.173	0.169	−6	6.182	40
180	326,045	0.243	0.233	−14	8.143	43	0.307	0.297	−37	4.001	6	0.164	0.160	−3	6.107	40
190	326,026	0.236	0.225	−13	7.996	42	0.305	0.295	−36	4.039	5	0.165	0.161	−2	5.973	39
200	325,979	0.239	0.229	−12	8.320	45	0.284	0.274	−31	4.162	11	0.154	0.151	5	7.110	47
208	325,969	0.234	0.223	−11	8.162	44	0.288	0.278	−31	4.185	9	0.158	0.154	5	6.832	45
Gamma with log link																
0	437,243	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	342,942	0.870	0.832	21	24.998	131	0.440	0.425	−24	15.145	71	0.505	0.494	43	21.396	138
20	334,881	0.649	0.621	−5	19.899	110	0.519	0.501	−65	8.283	36	0.312	0.306	−11	14.105	90
30	331,227	0.544	0.520	−4	21.752	126	0.479	0.463	−57	11.010	58	0.262	0.257	0	17.458	115
40	328,727	0.374	0.357	−10	14.009	81	0.329	0.318	−33	8.553	43	0.268	0.263	15	13.990	91
50	327,806	0.328	0.313	−16	12.750	74	0.327	0.316	−33	8.325	42	0.272	0.266	14	13.779	90
60	327,270	0.302	0.289	−15	11.825	68	0.297	0.287	−33	7.147	37	0.197	0.193	14	12.637	83
70	326,866	0.264	0.253	−15	10.159	58	0.249	0.241	−28	6.071	31	0.165	0.162	12	10.693	70
80	326,669	0.255	0.244	−19	9.819	57	0.288	0.279	−37	5.085	24	0.146	0.143	−2	9.090	60
90	326,433	0.266	0.254	−23	8.891	51	0.327	0.316	−45	4.079	15	0.171	0.167	−12	7.353	48
100	326,302	0.265	0.253	−23	7.839	44	0.361	0.349	−47	4.030	5	0.205	0.201	−12	6.246	40
110	326,224	0.256	0.244	−18	8.139	45	0.335	0.324	−41	4.211	8	0.191	0.187	−3	7.043	46
120	326,015	0.220	0.210	−17	6.898	36	0.317	0.306	−40	4.411	−1	0.194	0.190	−7	5.364	33
130	325,973	0.216	0.207	−15	6.654	33	0.307	0.296	−37	4.544	−4	0.196	0.192	−4	5.114	30
140	325,919	0.212	0.203	−15	6.334	31	0.302	0.292	−37	4.556	−5	0.191	0.187	−4	4.883	28
150	325,878	0.215	0.205	−14	6.486	33	0.297	0.287	−36	4.375	−3	0.181	0.177	−3	4.968	29
160	325,858	0.216	0.206	−14	6.619	34	0.299	0.289	−35	4.442	−2	0.181	0.177	−1	5.275	32
170	325,826	0.213	0.203	−14	6.485	33	0.302	0.292	−36	4.464	−4	0.183	0.180	−3	5.109	30
180	325,816	0.213	0.204	−14	6.505	33	0.300	0.290	−36	4.468	−3	0.179	0.176	−1	5.238	31
190	325,797	0.210	0.201	−14	6.580	33	0.295	0.285	−35	4.406	−3	0.179	0.176	−2	5.157	31
200	325,783	0.208	0.199	−13	6.496	32	0.290	0.280	−34	4.421	−3	0.178	0.174	−1	5.140	30
210	325,777	0.200	0.191	−10	6.260	30	0.263	0.254	−28	4.471	−3	0.176	0.173	4	5.107	30
220	325,774	0.199	0.190	−10	6.248	30	0.264	0.255	−28	4.541	−3	0.179	0.175	4	5.085	29
226	325,767	0.198	0.189	−8	6.256	29	0.249	0.241	−24	4.532	−1	0.184	0.180	8	5.417	32

Table A13. AIC scores and out-of-sample validation figures of the inverse gaussian GLMs of BEL with identity, inverse, log and $\frac{1}{\mu^2}$ link functions under 300–886 after each tenth and the final iteration.

k	AIC	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
Inverse gaussian with identity link																
0	437,338	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	346,132	0.871	0.833	1	23.559	115	0.314	0.304	7	20.269	107	0.534	0.523	70	25.673	169
20	334,430	0.549	0.524	−13	15.996	77	0.599	0.579	−77	8.273	−1	0.468	0.458	−44	9.809	32
30	331,453	0.488	0.467	−4	15.939	89	0.517	0.499	−67	6.532	11	0.413	0.405	−40	9.280	38
40	328,985	0.370	0.354	−13	13.279	76	0.338	0.327	−39	7.193	35	0.238	0.233	6	12.301	80
50	328,064	0.332	0.317	−15	12.727	74	0.338	0.327	−40	6.871	35	0.232	0.227	1	11.664	76
60	327,533	0.298	0.285	−17	10.994	64	0.304	0.294	−37	5.868	29	0.172	0.168	3	10.646	69
70	327,082	0.274	0.262	−15	9.387	53	0.243	0.235	−27	5.535	27	0.171	0.167	13	10.253	67
80	326,849	0.267	0.255	−20	9.426	54	0.278	0.268	−34	5.271	25	0.152	0.148	5	9.783	65
90	326,715	0.247	0.236	−21	8.546	49	0.275	0.266	−35	4.399	20	0.140	0.137	−1	8.302	55
100	326,627	0.234	0.224	−20	8.454	49	0.266	0.257	−34	4.414	20	0.144	0.141	−1	8.023	53
110	326,557	0.225	0.215	−17	8.350	47	0.246	0.238	−31	4.337	19	0.132	0.129	2	7.841	52
120	326,505	0.233	0.223	−17	8.897	51	0.256	0.247	−33	4.428	21	0.125	0.123	0	8.106	54
130	326,465	0.243	0.232	−16	9.965	58	0.265	0.256	−34	5.126	26	0.122	0.120	1	9.216	61
140	326,442	0.244	0.233	−16	10.175	59	0.273	0.264	−35	5.079	25	0.125	0.122	0	9.098	60
150	326,357	0.252	0.241	−16	10.133	58	0.352	0.340	−45	4.601	15	0.169	0.166	−1	8.831	58
160	326,130	0.206	0.197	−15	6.294	31	0.293	0.283	−36	4.360	−5	0.187	0.183	−4	4.711	26
170	326,112	0.204	0.195	−15	6.173	30	0.289	0.279	−35	4.284	−5	0.179	0.175	−4	4.688	27
180	326,099	0.203	0.194	−14	6.130	30	0.283	0.273	−34	4.277	−5	0.177	0.173	−3	4.654	26
190	326,088	0.204	0.195	−14	6.143	30	0.282	0.272	−34	4.280	−5	0.178	0.174	−3	4.699	27
200	326,076	0.204	0.195	−14	6.172	30	0.286	0.276	−34	4.347	−4	0.184	0.180	−3	4.823	27
210	326,071	0.199	0.190	−12	6.140	30	0.273	0.264	−32	4.277	−4	0.183	0.179	0	4.868	28
217	326,069	0.191	0.183	−11	5.967	28	0.261	0.252	−29	4.364	−5	0.178	0.175	2	4.779	27
Inverse gaussian with inverse link																
0	437,338	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	344,458	1.129	1.079	−25	35.685	202	1.138	1.099	−150	14.423	63	0.639	0.626	−63	22.713	149
20	336,004	0.682	0.652	−5	21.011	117	0.534	0.516	−67	8.866	41	0.321	0.314	−12	14.895	95
30	333,060	0.626	0.598	−10	24.463	142	0.623	0.602	−83	10.859	55	0.376	0.369	−31	16.233	107
40	329,632	0.412	0.394	−14	15.912	93	0.345	0.333	−29	12.096	64	0.318	0.311	28	18.446	121
50	328,515	0.335	0.320	−12	12.387	71	0.305	0.295	−29	8.122	40	0.276	0.270	18	13.333	86
60	327,916	0.321	0.307	−15	11.970	70	0.286	0.276	−27	8.385	44	0.247	0.241	20	13.973	91
70	327,543	0.278	0.266	−12	10.488	60	0.246	0.238	−28	6.106	31	0.164	0.161	9	10.331	67
80	327,196	0.249	0.238	−17	8.227	45	0.308	0.297	−38	4.037	9	0.193	0.189	−7	6.381	40
90	327,012	0.247	0.236	−19	8.016	44	0.376	0.363	−49	4.390	−1	0.212	0.207	−15	5.407	33
100	326,836	0.261	0.250	−20	9.073	51	0.382	0.369	−49	4.438	8	0.215	0.211	−9	7.237	47
110	326,750	0.268	0.257	−21	8.679	47	0.386	0.373	−50	4.510	4	0.217	0.212	−12	6.490	42
120	326,674	0.263	0.251	−19	8.191	45	0.378	0.365	−49	4.499	1	0.207	0.203	−12	6.011	38
130	326,636	0.261	0.250	−18	8.380	46	0.373	0.360	−48	4.402	2	0.198	0.193	−12	5.985	38
140	326,607	0.258	0.247	−17	8.253	46	0.349	0.337	−44	4.289	4	0.185	0.181	−8	6.277	40
150	326,581	0.258	0.246	−17	8.437	47	0.350	0.338	−44	4.228	6	0.183	0.179	−7	6.505	42
160	326,538	0.246	0.235	−15	8.445	47	0.326	0.315	−40	4.077	7	0.173	0.169	−4	6.572	43
170	326,522	0.249	0.238	−15	8.148	45	0.322	0.311	−39	4.119	6	0.175	0.172	−2	6.603	43
180	326,468	0.245	0.234	−14	8.583	47	0.298	0.288	−34	4.303	13	0.162	0.159	4	7.724	51
190	326,455	0.243	0.233	−14	8.506	47	0.299	0.289	−34	4.290	13	0.163	0.160	4	7.641	50
200	326,399	0.231	0.221	−12	7.918	42	0.286	0.277	−31	4.208	9	0.158	0.155	6	6.856	45
210	326,365	0.233	0.223	−12	7.983	43	0.288	0.279	−31	4.208	9	0.159	0.155	5	6.765	45
219	326,363	0.233	0.223	−11	8.040	43	0.283	0.274	−31	4.130	9	0.153	0.150	5	6.786	45

Table A13. *Cont.*

k	AIC	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
Inverse gaussian with log link																
0	437,338	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	343,530	0.866	0.828	19	24.925	131	0.450	0.435	−28	14.940	69	0.494	0.484	39	21.122	136
20	335,355	0.644	0.616	−5	19.653	109	0.526	0.509	−67	7.947	33	0.318	0.311	−14	13.490	85
30	331,675	0.536	0.512	−4	21.697	125	0.482	0.465	−58	10.885	57	0.262	0.256	−2	17.245	113
40	329,140	0.366	0.350	−10	13.913	80	0.325	0.314	−32	8.604	44	0.269	0.264	16	14.011	91
50	328,190	0.324	0.310	−16	12.640	73	0.319	0.308	−32	8.482	43	0.274	0.268	16	13.966	91
60	327,666	0.296	0.283	−15	11.626	67	0.290	0.280	−31	7.181	37	0.201	0.197	15	12.695	83
70	327,263	0.261	0.250	−15	9.948	57	0.244	0.236	−27	6.042	30	0.172	0.168	12	10.531	69
80	327,061	0.251	0.240	−18	9.746	56	0.284	0.275	−37	4.988	24	0.145	0.142	−1	8.964	59
90	326,825	0.263	0.251	−23	8.769	51	0.321	0.310	−44	4.059	15	0.168	0.165	−11	7.316	48
100	326,695	0.261	0.249	−22	7.727	43	0.352	0.340	−45	4.048	6	0.203	0.199	−10	6.341	41
110	326,589	0.240	0.230	−19	7.484	41	0.342	0.330	−44	4.124	1	0.192	0.188	−11	5.484	35
120	326,409	0.216	0.207	−16	6.397	32	0.299	0.289	−37	4.534	−2	0.195	0.191	−4	5.170	30
130	326,363	0.216	0.207	−15	6.314	31	0.308	0.298	−37	4.693	−6	0.201	0.196	−4	4.957	28
140	326,331	0.218	0.208	−15	6.537	33	0.303	0.292	−36	4.505	−3	0.195	0.191	−1	5.362	32
150	326,270	0.216	0.207	−14	6.457	32	0.302	0.291	−36	4.524	−4	0.189	0.185	−2	5.049	30
160	326,249	0.217	0.208	−14	6.596	34	0.298	0.288	−36	4.418	−2	0.182	0.178	−1	5.291	32
170	326,231	0.217	0.207	−15	6.492	32	0.296	0.286	−35	4.391	−3	0.179	0.175	−2	5.189	31
180	326,206	0.214	0.205	−15	6.426	32	0.302	0.291	−36	4.466	−4	0.179	0.175	−3	4.950	29
190	326,191	0.206	0.197	−13	6.472	33	0.288	0.279	−34	4.422	−3	0.173	0.170	0	5.149	31
200	326,176	0.208	0.199	−13	6.545	33	0.286	0.276	−33	4.430	−2	0.179	0.175	0	5.288	31
210	326,161	0.208	0.199	−13	6.501	33	0.286	0.276	−33	4.439	−2	0.184	0.180	1	5.318	32
220	326,153	0.202	0.193	−10	6.280	30	0.260	0.251	−27	4.455	−2	0.178	0.174	5	5.190	31
222	326,153	0.201	0.192	−10	6.291	30	0.261	0.252	−28	4.494	−3	0.180	0.177	5	5.176	30
Inverse gaussian with $\frac{1}{\mu^2}$ link																
0	437,338	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	344,467	0.985	0.941	−14	31.473	176	0.993	0.959	−130	12.573	46	0.561	0.549	−52	18.986	124
20	336,815	0.668	0.639	−7	21.404	122	0.591	0.571	−75	9.506	38	0.372	0.364	−22	14.521	91
30	331,792	0.478	0.457	−5	15.821	90	0.367	0.354	−28	10.573	53	0.373	0.365	33	17.496	114
40	330,089	0.421	0.403	−1	15.183	89	0.295	0.285	−19	10.660	56	0.316	0.309	34	16.657	109
50	329,020	0.376	0.359	−10	14.443	85	0.300	0.290	−21	11.439	60	0.320	0.313	34	17.553	115
60	328,452	0.330	0.316	−12	12.905	75	0.290	0.280	−24	9.196	48	0.273	0.267	25	14.952	98
70	327,925	0.316	0.302	−16	11.733	69	0.301	0.291	−35	7.090	35	0.200	0.195	6	11.701	76
80	327,639	0.262	0.250	−18	8.128	43	0.298	0.288	−35	4.425	11	0.208	0.203	−1	7.205	45
90	327,265	0.278	0.266	−22	8.311	46	0.355	0.343	−44	4.383	9	0.202	0.197	−7	7.090	46
100	327,148	0.288	0.275	−22	8.166	44	0.357	0.345	−44	4.408	8	0.207	0.203	−6	7.039	46
110	327,077	0.275	0.262	−20	7.965	42	0.366	0.353	−45	4.676	2	0.207	0.202	−7	6.410	40
120	326,916	0.274	0.262	−18	8.313	45	0.393	0.380	−47	5.133	1	0.228	0.223	−5	6.790	43
130	326,876	0.269	0.257	−18	8.133	43	0.396	0.382	−47	5.217	0	0.234	0.229	−5	6.625	42
140	326,789	0.259	0.248	−18	8.149	44	0.395	0.381	−47	5.074	1	0.249	0.244	−6	6.697	42
150	326,576	0.227	0.217	−15	6.896	34	0.341	0.329	−39	5.291	−5	0.221	0.217	−3	5.510	31
160	326,479	0.214	0.205	−16	6.274	29	0.291	0.281	−35	4.571	−6	0.206	0.202	−8	4.617	22
170	326,451	0.210	0.201	−15	6.035	26	0.285	0.275	−34	4.611	−8	0.202	0.198	−8	4.441	19
180	326,426	0.196	0.187	−13	5.753	25	0.250	0.242	−28	4.373	−6	0.187	0.183	−2	4.426	21
190	326,408	0.195	0.187	−13	5.682	24	0.249	0.241	−28	4.360	−6	0.188	0.184	−2	4.464	21
200	326,397	0.193	0.184	−13	5.686	24	0.245	0.237	−27	4.252	−5	0.186	0.182	−3	4.382	20
210	326,305	0.187	0.179	−13	5.721	27	0.237	0.229	−26	3.811	0	0.162	0.159	2	4.510	27
220	326,172	0.176	0.168	−14	5.110	26	0.197	0.191	−22	3.346	4	0.146	0.143	6	4.919	31
230	326,160	0.175	0.168	−14	4.994	25	0.206	0.199	−21	3.583	3	0.159	0.155	8	5.114	32
240	326,141	0.166	0.159	−11	5.012	24	0.197	0.190	−16	3.909	5	0.182	0.178	14	5.560	35
250	326,124	0.174	0.166	−12	5.058	25	0.193	0.186	−15	3.833	9	0.188	0.184	17	6.266	41

Table A14. AIC scores and out-of-sample validation figures of the gaussian, gamma and inverse gaussian GLMs of BEL with identity, inverse, log and $\frac{1}{\mu^2}$ link functions under 150–443 and 300–886 after the final iteration. Highlighted in green and red respectively the best and worst AIC scores and validation figures.

k	AIC	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
Gaussian with identity link under 150-443																
150	325,850	0.247	0.237	−14	9.924	57	0.271	0.262	−35	4.612	22	0.122	0.120	−1	8.537	56
Gaussian with inverse link under 150-443																
150	325,952	0.258	0.247	−16	8.468	45	0.353	0.341	−44	4.282	3	0.192	0.188	−8	6.088	39
Gaussian with log link under 150-443																
150	325,823	0.240	0.229	−15	7.980	44	0.316	0.305	−38	4.014	6	0.170	0.167	−2	6.434	42
Gamma with identity link under 150-443																
150	326,041	0.236	0.226	−14	9.329	53	0.260	0.251	−33	4.321	20	0.121	0.118	1	8.206	54
Gamma with inverse link under 150-443																
150	326,222	0.254	0.243	−15	8.410	46	0.327	0.316	−40	4.111	7	0.171	0.167	−3	6.722	44
Gamma with log link under 150-443																
150	326,022	0.243	0.232	−15	7.820	43	0.323	0.312	−40	4.040	3	0.174	0.170	−4	6.010	39
Inverse gaussian with identity link under 150-443																
150	326,352	0.249	0.238	−17	9.375	54	0.337	0.326	−44	4.224	12	0.150	0.146	−4	7.930	52
Inverse gaussian with inverse link under 150-443																
150	326,617	0.253	0.242	−15	8.152	44	0.324	0.313	−39	4.148	5	0.172	0.169	−3	6.476	42
Inverse gaussian with log link under 150-443																
150	326,413	0.247	0.237	−15	7.716	42	0.324	0.313	−40	4.095	2	0.172	0.168	−4	5.892	38
Inverse gaussian with $\frac{1}{\mu^2}$ link under 150-443																
150	326,778	0.262	0.250	−16	8.258	44	0.332	0.321	−41	4.238	5	0.177	0.174	−3	6.518	42
Gaussian with identity link under 300-886																
224	325,459	0.194	0.186	−9	6.659	34	0.268	0.259	−30	4.200	−2	0.168	0.165	1	5.007	29
Gaussian with inverse link under 300-886																
213	325,797	0.241	0.230	−12	8.325	44	0.310	0.299	−36	4.063	6	0.171	0.167	−1	6.284	41
Gaussian with log link under 300-886																
214	325,552	0.206	0.197	−10	6.640	32	0.267	0.258	−29	4.402	−2	0.177	0.173	3	5.180	30
Gamma with identity link under 300-886																
212	325,689	0.189	0.180	−11	5.975	28	0.261	0.252	−29	4.384	−5	0.169	0.165	1	4.545	25
Gamma with inverse link under 300-886																
208	325,969	0.234	0.223	−11	8.162	44	0.288	0.278	−31	4.185	9	0.158	0.154	5	6.832	45
Gamma with log link under 300-886																
226	325,767	0.198	0.189	−8	6.256	29	0.249	0.241	−24	4.532	−1	0.184	0.180	8	5.417	32
Inverse gaussian with identity link under 300-886																
217	326,069	0.191	0.183	−11	5.967	28	0.261	0.252	−29	4.364	−5	0.178	0.175	2	4.779	27
Inverse gaussian with inverse link under 300-886																
219	326,363	0.233	0.223	−11	8.040	43	0.283	0.274	−31	4.130	9	0.153	0.150	5	6.786	45
Inverse gaussian with log link under 300-886																
222	326,153	0.201	0.192	−10	6.291	30	0.261	0.252	−28	4.494	−3	0.180	0.177	5	5.176	30
Inverse gaussian with $\frac{1}{\mu^2}$ link under 300-886																
250	326,124	0.174	0.166	−12	5.058	25	0.193	0.186	−15	3.833	9	0.188	0.184	17	6.266	41

Table A15. Out-of-sample validation figures of selected generalized additive models (GAMs) of BEL with varying spline function number per dimension and fixed spline function type under 150–443 after each tenth and the finally selected smooth function.

k	$K_{\max}$	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
4 Thin plate regression splines under gaussian with identity link in stagewise selection of length 5																
0	150	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	150	0.632	0.604	28	22.019	116	0.345	0.334	−8	13.247	65	0.479	0.469	66	21.072	139
20	150	0.406	0.388	0	11.330	44	0.375	0.362	−42	7.254	−12	0.341	0.334	−6	7.709	24
30	150	0.399	0.382	−11	12.268	59	0.465	0.449	−61	5.744	−6	0.314	0.307	−26	6.116	29
40	150	0.371	0.355	−8	11.415	53	0.480	0.463	−64	6.380	−16	0.340	0.332	−34	5.283	13
50	150	0.392	0.375	−13	12.079	59	0.520	0.503	−70	5.961	−12	0.365	0.358	−39	5.368	19
60	150	0.306	0.292	−15	9.833	48	0.405	0.391	−51	5.283	−2	0.273	0.267	−10	6.484	39
70	150	0.272	0.260	−15	9.896	56	0.321	0.310	−35	5.227	22	0.232	0.228	12	10.460	69
80	150	0.249	0.238	−17	8.627	49	0.308	0.297	−36	4.588	16	0.205	0.201	9	9.100	60
90	150	0.261	0.250	−17	9.262	54	0.325	0.314	−39	4.639	18	0.195	0.191	5	9.340	62
100	150	0.254	0.243	−18	9.593	55	0.340	0.328	−42	4.626	17	0.196	0.192	3	9.312	62
110	150	0.255	0.244	−18	9.407	54	0.336	0.324	−40	4.640	18	0.207	0.203	4	9.325	62
120	150	0.243	0.233	−16	8.474	48	0.307	0.296	−38	4.023	13	0.186	0.182	1	7.819	51
130	150	0.241	0.230	−16	8.481	49	0.308	0.298	−37	4.108	13	0.183	0.179	2	8.075	53
140	150	0.235	0.225	−15	8.018	45	0.295	0.285	−35	3.865	10	0.173	0.169	2	7.182	47
150	150	0.240	0.229	−15	8.192	46	0.291	0.281	−35	3.907	13	0.176	0.172	3	7.641	50
5 Thin plate regression splines under gaussian with identity link																
0	100	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	100	0.643	0.615	27	23.278	125	0.344	0.332	−6	15.238	78	0.493	0.483	69	23.151	153
20	100	0.387	0.370	1	10.371	35	0.364	0.352	−40	7.855	−20	0.335	0.328	−6	7.454	14
30	100	0.382	0.366	−10	11.235	50	0.454	0.439	−60	6.247	−14	0.317	0.310	−28	5.603	18
40	100	0.368	0.352	−11	10.931	48	0.463	0.447	−61	6.266	−16	0.337	0.329	−33	5.343	12
50	100	0.355	0.339	−11	10.086	40	0.481	0.465	−64	7.752	−28	0.351	0.344	−37	5.481	0
60	100	0.344	0.329	−9	10.015	40	0.490	0.474	−66	8.152	−30	0.364	0.356	−38	5.593	−3
70	100	0.339	0.324	−6	10.035	45	0.476	0.460	−64	7.578	−27	0.345	0.337	−37	5.078	0
80	100	0.295	0.282	−11	9.397	49	0.404	0.390	−51	5.513	−6	0.241	0.236	−11	5.820	34
90	100	0.296	0.283	−12	9.694	52	0.393	0.380	−49	5.155	0	0.206	0.202	−7	6.605	41
100	100	0.287	0.274	−11	9.431	48	0.397	0.383	−50	5.402	−5	0.202	0.198	−9	5.945	36
8 Thin plate regression splines under gaussian with identity link																
0	150	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	150	0.639	0.611	27	23.176	125	0.340	0.329	−3	15.517	80	0.516	0.505	73	23.627	156
20	150	0.375	0.359	3	9.604	26	0.334	0.322	−33	8.378	−24	0.341	0.333	1	7.711	10
30	150	0.361	0.345	−7	10.444	41	0.415	0.401	−52	6.961	−19	0.304	0.297	−21	5.871	13
40	150	0.356	0.340	−5	10.098	36	0.425	0.410	−54	7.920	−28	0.311	0.304	−27	5.647	−1
50	150	0.339	0.324	−7	9.712	33	0.418	0.404	−53	7.746	−27	0.311	0.304	−26	5.596	0
60	150	0.325	0.311	−6	9.037	26	0.411	0.397	−52	8.706	−34	0.310	0.304	−26	5.850	−8
70	150	0.325	0.311	−4	9.180	31	0.429	0.414	−55	8.773	−34	0.326	0.319	−30	5.912	−9
80	150	0.309	0.296	−5	8.618	29	0.430	0.415	−55	8.984	−35	0.336	0.329	−29	6.382	−9
90	150	0.313	0.299	−5	8.981	32	0.384	0.371	−48	7.390	−26	0.300	0.293	−26	5.430	−4
100	150	0.328	0.313	−6	9.910	47	0.400	0.387	−51	5.572	−12	0.291	0.285	−25	5.064	13
110	150	0.256	0.245	−10	7.985	38	0.326	0.315	−40	4.655	−6	0.201	0.197	−6	5.002	28
120	150	0.253	0.242	−9	7.340	30	0.321	0.310	−39	5.542	−14	0.209	0.204	−5	4.541	20
130	150	0.252	0.241	−9	7.767	34	0.326	0.315	−40	5.197	−11	0.205	0.201	−5	4.770	24
140	150	0.245	0.234	−8	7.592	33	0.322	0.311	−41	5.315	−15	0.197	0.193	−7	4.317	20
150	150	0.217	0.208	−11	6.477	32	0.239	0.231	−26	3.652	2	0.179	0.175	6	5.578	34
10 Thin plate regression splines under gaussian with identity link																
0	150	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	150	0.642	0.614	27	23.354	126	0.344	0.332	−5	15.463	80	0.509	0.499	71	23.654	156
20	150	0.382	0.365	2	10.101	33	0.341	0.329	−34	7.780	−18	0.338	0.331	1	7.728	18
30	150	0.370	0.354	−7	10.922	45	0.416	0.402	−52	6.497	−14	0.305	0.299	−20	6.103	18
40	150	0.354	0.338	−7	10.412	39	0.404	0.391	−51	6.747	−20	0.308	0.301	−24	5.600	8
50	150	0.347	0.331	−7	10.119	38	0.426	0.412	−54	7.258	−24	0.310	0.304	−27	5.467	4
60	150	0.342	0.327	−4	9.766	34	0.400	0.387	−50	7.600	−26	0.298	0.292	−23	5.615	0
70	150	0.334	0.319	−4	9.601	35	0.428	0.414	−55	8.158	−30	0.318	0.311	−29	5.618	−5
80	150	0.315	0.301	−5	9.093	35	0.432	0.418	−55	8.113	−29	0.334	0.327	−29	6.087	−3
90	150	0.323	0.309	−5	9.436	38	0.388	0.375	−49	6.558	−20	0.297	0.291	−26	5.194	2
100	150	0.309	0.296	−6	8.722	27	0.409	0.395	−54	8.780	−36	0.261	0.255	−27	4.994	−9
110	150	0.309	0.295	−6	8.542	26	0.411	0.397	−54	8.711	−37	0.284	0.278	−33	4.768	−15
120	150	0.206	0.197	−9	5.768	25	0.216	0.209	−23	3.806	−4	0.164	0.161	5	4.519	24
130	150	0.205	0.196	−10	5.759	24	0.226	0.218	−24	3.952	−5	0.175	0.172	4	4.579	24
140	150	0.214	0.205	−10	6.761	34	0.228	0.220	−25	3.363	5	0.167	0.163	6	5.762	36
150	150	0.212	0.203	−10	7.070	37	0.230	0.223	−24	3.575	8	0.173	0.170	8	6.337	40

Table A16. Effective degrees of freedom, p-values and significance codes per dimension of GAMs of BEL built up of thin plate regression splines with gaussian random component and identity link function under 150–443 for spline function numbers $J \in \{4, 10\}$ per dimension at stages $k \in \{50, 100, 150\}$. The confidence levels corresponding to the indicated significance codes are *** = 0.001, ** = 0.01, * = 0.05, = 0.1, = 1.

	$J = 4, k = 50$			$J = 4, k = 100$			$J = 4, k = 150$			$J = 10, k = 50$			$J = 10, k = 100$			$J = 10, k = 150$		
k	**df**	**p-val**	**sign**	**df**	**p-val**	**sign**	**df**	**p-val**	**sign**	**df**	**p-val**	**sign**	**df**	**p-val**	**sign**	**df**	**p-val**	**sign**
1	2.858	2_{-16}	***	2.350	2_{-16}	***	1.948	2_{-16}	***	9.000	2_{-16}	***	8.941	2_{-16}	***	7.724	2_{-16}	***
2	3.000	2_{-16}	***	2.104	2_{-16}	***	1.000	2_{-16}	***	7.857	2_{-16}	***	4.436	2_{-16}	***	1.000	2_{-16}	***
3	3.000	2_{-16}	***	2.901	2_{-16}	***	2.922	2_{-16}	***	5.600	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***
4	2.997	2_{-16}	***	2.962	2_{-16}	***	2.998	2_{-16}	***	7.073	2_{-16}	***	6.791	2_{-16}	***	7.288	2_{-16}	***
5	2.729	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	8.679	2_{-16}	***	8.870	2_{-16}	***	8.210	2_{-16}	***
6	3.000	2_{-16}	***	3.000	2_{-16}	***	1.043	2_{-16}	***	3.417	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***
7	3.000	2_{-16}	***	2.806	2_{-16}	***	2.841	2_{-16}	***	7.990	2_{-16}	***	8.608	2_{-16}	***	1.000	2_{-16}	***
8	3.000	2_{-16}	***	2.956	2_{-16}	***	2.961	2_{-16}	***	8.282	2_{-16}	***	8.292	2_{-16}	***	8.122	2_{-16}	***
9	1.000	2_{-16}	***	1.000	2_{-16}	***	2.223	2_{-16}	***	7.710	2_{-16}	***	6.510	2_{-16}	***	6.549	2_{-16}	***
10	2.991	2_{-16}	***	2.924	2_{-16}	***	3.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***
11	2.587	2_{-16}	***	2.922	2_{-16}	***	2.889	2_{-16}	***	6.535	2_{-16}	***	7.014	2_{-16}	***	5.672	2_{-16}	***
12	2.645	2_{-16}	***	1.874	2_{-16}	***	1.000	2_{-16}	***	7.235	2_{-16}	***	7.284	2_{-16}	***	8.346	2_{-16}	***
13	2.244	2_{-16}	***	2.425	2_{-16}	***	1.000	2_{-16}	***	2.372	2_{-16}	***	2.531	2_{-16}	***	1.000	2_{-16}	***
14	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***
15	3.000	2_{-16}	***	1.000	2_{-16}	***	2.285	2_{-16}	***	5.430	2_{-16}	***	5.640	2_{-16}	***	4.437	2_{-16}	***
16	1.000	2_{-16}	***	1.000	2_{-16}	***	2.783	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***
17	2.344	2_{-16}	***	1.670	2_{-16}	***	1.646	2_{-16}	***	3.886	2_{-16}	***	1.610	2_{-16}	***	1.624	2_{-16}	***
18	3.000	2_{-16}	***	3.000	2_{-16}	***	3.000	2_{-16}	***	8.751	2_{-16}	***	8.620	1.4_{-5}	***	5.367	6.9_{-5}	***
19	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***
20	1.497	2_{-16}	***	1.501	2_{-16}	***	2.148	2_{-16}	***	1.754	2_{-16}	***	1.000	2_{-16}	***	3.141	8.1_{-16}	***
21	1.441	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***
22	1.770	2_{-16}	***	2.192	2_{-16}	***	1.400	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	3.985	1.9_{-9}	***
23	2.395	2_{-16}	***	2.746	2_{-16}	***	2.911	2_{-16}	***	2.057	2_{-16}	***	1.428	2_{-16}	***	2.663	2_{-16}	***
24	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	2.964	2_{-16}	***	1.000	3.3_{-13}	***	1.000	1.1_{-13}	***
25	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***
26	1.000	2_{-16}	***	1.485	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***
27	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2.2_{-10}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	1.6_{-10}	***
28	1.000	2_{-16}	***	2.607	2_{-16}	***	1.839	2_{-16}	***	1.000	2_{-16}	***	2.780	2_{-16}	***	1.914	2_{-16}	***
29	1.000	2_{-16}	***	1.000	2_{-16}	***	1.809	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***
30	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	6.740	2_{-16}	***	6.416	2_{-16}	***	6.508	2_{-16}	***
31	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2.4_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***
32	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***
33	1.000	2_{-16}	***	2.055	4.9_{-15}	***	1.893	2.2_{-15}	***	7.111	2_{-16}	***	7.175	6.3_{-12}	***	6.728	2_{-16}	***
34	1.000	3.2_{-16}	***	1.000	2.9_{-16}	***	1.000	8.7_{-11}	***	1.000	2_{-16}	***	1.213	2_{-16}	***	1.635	4.9_{-16}	***
35	3.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2.5_{-16}	***	4.780	2_{-16}	***	4.013	2_{-16}	***	4.224	2_{-16}	***
36	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	7.825	4.8_{-16}	***	7.867	1.1_{-15}	***	7.738	2.3_{-3}	**
37	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	2_{-16}	***	1.000	4.6_{-16}	***	1.000	7.5_{-16}	***	1.000	2_{-16}	***
38	2.512	1.1_{-14}	***	2.303	2_{-16}	***	2.057	2_{-16}	***	1.233	2_{-16}	***	1.000	2_{-16}	***	1.000	1.1_{-4}	***
39	1.000	2.7_{-12}	***	1.000	1.2_{-13}	***	1.000	1.9_{-13}	***	1.000	1.1_{-15}	***	1.000	2.6_{-16}	***	1.000	1.2_{-14}	***
40	1.826	6.4_{-11}	***	1.000	2_{-16}	***	1.915	3.6_{-15}	***	1.000	1.2_{-13}	***	1.514	2_{-16}	***	1.000	2_{-16}	***
41	2.668	7.5_{-16}	***	2.701	5.3_{-15}	***	1.787	9.8_{-7}	***	1.823	8.1_{-12}	***	1.319	9.4_{-15}	***	1.000	2_{-16}	***
42	1.000	1.1_{-15}	***	1.000	2_{-16}	***	1.000	2_{-15}	***	1.000	2.9_{-12}	***	1.000	8_{-12}	***	5.275	3.8_{-4}	***
43	1.000	3.8_{-10}	***	1.000	9.5_{-10}	***	1.000	2_{-9}	***	1.000	3.3_{-10}	***	1.000	7.7_{-11}	***	1.000	1.1_{-10}	***
44	1.713	1.3_{-8}	***	1.887	8.2_{-9}	***	1.892	6.2_{-9}	***	2.109	6_{-8}	***	1.779	5.3_{-8}	***	2.061	3.4_{-8}	***
45	1.000	5.7_{-9}	***	1.000	6.4_{-9}	***	1.000	1.9_{-8}	***	1.000	8_{-9}	***	1.000	2.1_{-8}	***	1.000	8.8_{-9}	***
46	1.917	3.5_{-9}	***	1.000	2_{-16}	***	1.000	1.3_{-15}	***	1.305	1.9_{-6}	***	1.610	1.1_{-6}	***	1.000	8.7_{-8}	***
47	1.451	1.2_{-6}	***	1.507	5.8_{-7}	***	1.234	1_{-6}	***	1.000	7.7_{-13}	***	1.000	5.5_{-13}	***	1.000	7.4_{-12}	***
48	2.753	3.2_{-7}	***	2.863	6.5_{-8}	***	2.804	2.1_{-8}	***	1.000	2.4_{-8}	***	1.000	7.8_{-8}	***	1.000	2.9_{-6}	***
49	1.000	5.5_{-7}	***	1.000	4.7_{-14}	***	1.000	1.6_{-11}	***	1.000	6.9_{-7}	***	1.000	9.6_{-12}	***	1.000	1.6_{-12}	***
50	1.000	9.2_{-7}	***	1.372	8.3_{-11}	***	1.000	1.1_{-12}	***	1.000	1.1_{-6}	***	1.000	2_{-10}	***	1.000	2_{-11}	***
51				1.004	2_{-16}	***	1.000	2_{-16}	***				1.000	1.1_{-6}	***	1.000	1.3_{-6}	***
52				2.839	2_{-16}	***	1.334	2_{-16}	***				1.000	4.3_{-13}	***	1.000	3_{-13}	***
53				2.640	2_{-16}	***	2.421	2_{-16}	***				1.000	4.7_{-10}	***	1.000	7.1_{-11}	***
54				2.664	2_{-16}	***	1.000	2_{-16}	***				3.237	2.8_{-6}	***	3.168	4.9_{-6}	***
55				1.000	9.2_{-9}	***	1.000	3.1_{-6}	***				3.906	5.8_{-8}	***	3.493	1_{-9}	***
56				1.000	2.8_{-9}	***	2.376	2.3_{-8}	***				1.098	3.5_{-5}	***	3.513	2_{-16}	***
57				1.000	3.3_{-15}	***	1.000	2.8_{-13}	***				5.574	5.1_{-3}	**	5.019	6.7_{-2}	.
58				1.000	2_{-16}	***	1.000	2_{-16}	***				1.000	7.3_{-5}	***	1.000	1_{-5}	***
59				1.000	1.2_{-11}	***	1.000	2_{-11}	***				1.000	1.8_{-6}	***	1.000	8.8_{-8}	***
60				1.000	2_{-16}	***	1.000	2_{-16}	***				3.717	5.2_{-4}	***	3.286	5.6_{-3}	**
61				1.000	7.5_{-11}	***	1.000	7.1_{-11}	***				1.000	6.7_{-5}	***	1.000	1.5_{-5}	***
62				2.613	4.2_{-4}	***	2.868	2_{-16}	***				1.000	1.1_{-5}	***	1.000	4.6_{-6}	***
63				1.000	7.9_{-15}	***	1.867	1.6_{-14}	***				4.210	6.6_{-3}	**	3.543	7.3_{-4}	***
64				1.000	2.4_{-6}	***	1.000	1.2_{-6}	***				1.000	1.7_{-4}	***	1.000	3.4_{-4}	***
65				2.960	2.3_{-13}	***	2.976	2_{-16}	***				2.799	7.1_{-3}	**	2.861	3_{-3}	**
66				1.904	2_{-16}	***	2.115	2_{-16}	***				3.054	1.7_{-3}	**	3.159	8.8_{-6}	***
67				2.859	9.1_{-14}	***	2.778	1.1_{-13}	***				3.671	7.6_{-3}	**	3.788	8.4_{-4}	***
68				1.000	2.9_{-1}		1.000	5.2_{-11}	***				1.000	4_{-4}	***	1.000	1.2_{-4}	***
69				2.797	2.8_{-3}	**	2.954	2.2_{-3}	**				1.000	2.8_{-3}	**	1.000	3.3_{-3}	**
70				1.000	2.4_{-6}	***	1.000	1.5_{-6}	***				1.000	6.7_{-3}	**	1.000	1.1_{-3}	**
71				2.957	6_{-14}	***	2.996	6.1_{-15}	***				1.000	8.6_{-3}	**	1.000	5_{-3}	**
72				2.612	1.4_{-13}	***	2.101	6.3_{-11}	***				1.000	1.2_{-2}	*	1.000	8.9_{-3}	**
73				1.196	2_{-16}	***	3.000	2_{-16}	***				1.000	1.5_{-2}	*	1.000	6.1_{-5}	***
74				2.994	3.8_{-6}	***	2.559	1.8_{-3}	**				3.644	1.2_{-1}		2.988	1.4_{-1}	
75				1.000	1.7_{-14}	***	1.000	3_{-14}	***				1.000	1.7_{-2}	*	1.000	1.8_{-2}	*

Table A16. *Cont.*

	$J = 4, k = 50$			$J = 4, k = 100$			$J = 4, k = 150$			$J = 10, k = 50$			$J = 10, k = 100$			$J = 10, k = 150$		
k	df	p-val	sign	df	p-val	sign	df	p-val	sign	df	p-val	sign	df	p-val	sign	df	p-val	sign
76				1.000	4.4_{-13}	***	2.334	3.8_{-14}	***				2.469	1_{-1}		2.077	1.8_{-1}	
77				1.353	4_{-9}	***	1.411	8.8_{-9}	***				1.000	2.5_{-2}	*	1.000	1.1_{-2}	*
78				1.000	1.5_{-5}	***	1.000	6.5_{-6}	***				1.000	2_{-16}	***	1.000	1.6_{-4}	***
79				1.000	3_{-5}	***	1.000	1.5_{-5}	***				5.186	1.5_{-6}	***	1.000	2_{-16}	***
80				1.000	1_{-7}	***	1.000	7.8_{-8}	***				1.892	2.2_{-2}	*	1.795	1.9_{-2}	*
81				2.725	1.3_{-4}	***	2.739	7.1_{-5}	***				1.000	5.2_{-6}	***	1.000	5.8_{-1}	
82				1.000	7.6_{-5}	***	2.175	1.4_{-5}	***				1.000	1.8_{-3}	**	1.000	5.1_{-1}	
83				2.240	1.3_{-3}	**	2.075	9_{-4}	***				7.020	2_{-16}	***	4.809	2.9_{-3}	**
84				1.000	6.8_{-5}	***	2.902	1.5_{-5}	***				4.003	1.5_{-1}		4.722	9.8_{-3}	**
85				1.000	7.5_{-5}	***	1.000	4_{-6}	***				1.000	1_{-9}	***	1.000	1.8_{-4}	***
86				1.000	3.7_{-4}	***	1.000	7.7_{-4}	***				3.115	1.2_{-1}		2.748	1.2_{-1}	
87				1.000	3.4_{-4}	***	1.000	9.1_{-5}	***				5.294	1.4_{-1}		5.598	1.3_{-1}	
88				1.000	1.9_{-4}	***	1.000	9.6_{-5}	***				2.263	1.5_{-1}		1.788	2.5_{-1}	
89				2.828	2.1_{-3}	**	3.000	6_{-5}	***				1.000	3.4_{-4}	***	1.000	3.3_{-4}	***
90				1.000	7.8_{-4}	***	1.000	5.6_{-4}	***				1.000	3.7_{-2}	*	1.000	3.8_{-2}	*
91				1.000	2.5_{-3}	**	1.000	2.9_{-3}	**				1.000	1.8_{-3}	**	1.000	1.2_{-3}	**
92				1.000	3.8_{-3}	**	1.000	3.5_{-3}	**				1.000	1.7_{-2}	*	1.000	1.2_{-2}	*
93				1.000	1.8_{-3}	**	1.000	1.1_{-3}	**				1.000	3.8_{-2}	*	1.000	2.8_{-2}	*
94				2.776	3.6_{-5}	***	1.000	1.8_{-7}	***				5.921	4.2_{-3}	**	3.962	2_{-16}	***
95				2.103	4.9_{-2}	*	1.974	1.3_{-1}					8.154	2_{-16}	***	2.290	2_{-16}	***
96				2.023	1.2_{-4}	***	1.000	4.6_{-10}	***				1.000	2.8_{-12}	***	1.000	1.6_{-5}	***
97				2.811	1.5_{-2}	*	2.873	5.9_{-3}	**				3.748	7.1_{-4}	***	1.000	1.2_{-6}	***
98				1.000	7.1_{-3}	**	1.000	1.1_{-2}	*				1.000	3.9_{-6}	***	7.349	2.8_{-1}	
99				1.000	1.4_{-2}	*	1.000	1.9_{-2}	*				2.149	1.2_{-3}	**	1.000	2.8_{-8}	***
100				2.764	2.9_{-2}	*	2.321	9_{-2}	.				1.000	3.1_{-3}	**	1.000	2.1_{-1}	
101							1.000	1.1_{-4}	***							1.000	8.2_{-10}	***
102							1.000	7.7_{-2}	.							1.000	1.6_{-2}	*
103							1.000	2.9_{-3}	**							4.084	5.8_{-4}	***
104							1.000	6.8_{-5}	***							1.000	3.2_{-2}	*
105							1.000	9.3_{-3}	**							1.000	6.8_{-2}	.
106							1.000	2.1_{-9}	***							1.000	5.2_{-3}	**
107							1.000	1.9_{-2}	*							3.397	1_{-1}	
108							2.187	9.6_{-2}	.							1.248	3.4_{-1}	
109							1.000	2.1_{-3}	**							3.079	3.9_{-1}	
110							1.000	4.6_{-2}	*							1.000	3.9_{-4}	***
111							1.000	2_{-16}	***							0.979	4.3_{-8}	***
112							1.000	2.9_{-2}	*							8.555	2_{-16}	***
113							1.000	9.5_{-1}								8.952	1.7_{-12}	***
114							1.644	9.6_{-2}	.							1.000	2_{-16}	***
115							1.000	2_{-2}	*							1.000	2_{-16}	***
116							1.000	1.8_{-2}	*							1.000	1.7_{-13}	***
117							1.000	4.8_{-3}	**							2.988	3.4_{-13}	***
118							1.000	2.4_{-2}	*							8.401	1.2_{-10}	***
119							2.704	8.3_{-2}	.							2.493	4.7_{-5}	***
120							1.000	1.8_{-2}	*							1.000	4.1_{-7}	***
121							1.413	6.7_{-1}								1.000	9_{-5}	***
122							1.886	6.2_{-1}								2.745	1.2_{-3}	**
123							1.000	1.4_{-5}	***							1.000	3.4_{-3}	**
124							2.499	1.8_{-1}								1.000	1.5_{-2}	*
125							1.000	3.6_{-2}	*							1.000	1.4_{-2}	*
126							2.416	1_{-1}								1.000	5.8_{-3}	**
127							1.000	5.1_{-5}	***							3.120	5.7_{-2}	.
128							1.000	3.8_{-2}	*							1.000	9.2_{-4}	***
129							1.000	1.3_{-3}	**							1.000	3.9_{-3}	**
130							1.000	5.7_{-2}	.							3.778	1.7_{-1}	
131							1.000	1.3_{-2}	*							2.752	2.7_{-2}	*
132							1.000	1.2_{-2}	*							1.000	6.9_{-3}	**
133							1.970	2.5_{-1}								1.000	4.8_{-3}	**
134							1.000	3.5_{-2}	*							1.000	5.5_{-2}	.
135							1.000	5.9_{-4}	***							1.000	3.8_{-2}	*
136							1.176	7.1_{-3}	**							5.289	1.4_{-1}	
137							2.357	3.4_{-1}								1.000	3.7_{-2}	*
138							1.000	6.7_{-2}	.							1.000	2_{-4}	***
139							1.000	7.9_{-2}	.							1.000	5.1_{-3}	**
140							1.000	6.9_{-2}	.							1.000	1.6_{-1}	
141							1.000	4.7_{-2}	*							8.453	2.5_{-3}	**
142							1.000	1.3_{-3}	**							1.000	4_{-2}	*
143							2.602	4.1_{-2}	*							3.975	1.4_{-1}	
144							1.631	4.6_{-1}								1.000	4.2_{-4}	***
145							1.000	8.3_{-2}	.							1.000	3.7_{-3}	**
146							1.000	1_{-2}	*							2.147	1.9_{-1}	
147							1.000	3.6_{-2}	*							1.000	5_{-2}	.
148							1.251	1.6_{-1}								1.000	4.1_{-2}	*
149							2.376	2.1_{-1}								1.000	5.4_{-2}	.
150							1.482	2_{-1}								1.000	6.3_{-2}	.

Table A17. Out-of-sample validation figures of selected GAMs of BEL with varying spline function type and fixed spline function number of 5 per dimension under 100–443 after each tenth and the finally selected smooth function.

k	$K_{\max}$	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
5 Thin plate regression splines under gaussian with identity link																
0	100	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	100	0.643	0.615	27	23.278	125	0.344	0.332	−6	15.238	78	0.493	0.483	69	23.151	153
20	100	0.387	0.370	1	10.371	35	0.364	0.352	−40	7.855	−20	0.335	0.328	−6	7.454	14
30	100	0.382	0.366	−10	11.235	50	0.454	0.439	−60	6.247	−14	0.317	0.310	−28	5.603	18
40	100	0.368	0.352	−11	10.931	48	0.463	0.447	−61	6.266	−16	0.337	0.329	−33	5.343	12
50	100	0.355	0.339	−11	10.086	40	0.481	0.465	−64	7.752	−28	0.351	0.344	−37	5.481	0
60	100	0.344	0.329	−9	10.015	40	0.490	0.474	−66	8.152	−30	0.364	0.356	−38	5.593	−3
70	100	0.339	0.324	−6	10.035	45	0.476	0.460	−64	7.578	−27	0.345	0.337	−37	5.078	0
80	100	0.295	0.282	−11	9.397	49	0.404	0.390	−51	5.513	−6	0.241	0.236	−11	5.820	34
90	100	0.296	0.283	−12	9.694	52	0.393	0.380	−49	5.155	0	0.206	0.202	−7	6.605	41
100	100	0.287	0.274	−11	9.431	48	0.397	0.383	−50	5.402	−5	0.202	0.198	−9	5.945	36
5 Cubic regression splines under gaussian with identity link																
0	100	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	100	0.637	0.609	28	22.739	122	0.337	0.326	−4	14.733	75	0.505	0.494	71	22.781	150
20	100	0.388	0.371	2	10.094	32	0.358	0.346	−40	8.256	−25	0.319	0.313	−5	7.161	10
30	100	0.389	0.372	−6	11.426	50	0.436	0.421	−55	6.652	−14	0.289	0.283	−19	5.849	22
40	100	0.359	0.343	−9	10.508	41	0.448	0.433	−59	7.171	−23	0.310	0.303	−29	5.175	6
50	100	0.345	0.330	−9	9.906	35	0.476	0.460	−63	8.736	−34	0.328	0.321	−34	5.373	−5
60	100	0.338	0.323	−7	9.817	34	0.475	0.459	−63	9.192	−37	0.330	0.324	−34	5.491	−8
70	100	0.307	0.294	−8	9.341	47	0.430	0.416	−58	6.081	−18	0.234	0.229	−26	3.871	15
80	100	0.289	0.277	−13	10.157	55	0.410	0.396	−53	5.106	0	0.237	0.232	−11	6.939	43
90	100	0.283	0.271	−13	10.307	56	0.407	0.394	−53	5.067	1	0.229	0.224	−10	7.035	44
100	100	0.268	0.256	−12	9.903	52	0.399	0.386	−51	5.182	−2	0.226	0.221	−9	6.533	40
5 Duchon splines under gaussian with identity link																
0	100	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	100	0.753	0.720	−4	20.570	98	0.428	0.413	−39	11.806	49	0.408	0.399	6	15.241	93
20	100	0.704	0.673	−22	17.488	74	0.441	0.426	−51	8.606	31	0.380	0.372	−16	11.600	66
30	100	0.661	0.632	−32	19.699	95	0.376	0.363	−40	14.235	73	0.319	0.312	11	19.168	124
40	100	0.663	0.634	−21	18.426	84	0.292	0.282	−18	14.138	73	0.377	0.370	33	19.007	123
50	100	0.666	0.636	−17	18.534	86	0.287	0.277	−12	14.785	76	0.410	0.402	41	19.896	130
56	100	0.666	0.636	−18	18.532	86	0.288	0.279	−14	14.643	75	0.406	0.397	40	19.757	129
5 Eilers and Marx style P-splines under gaussian with identity link																
0	100	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	100	0.643	0.615	29	22.836	123	0.344	0.332	−9	13.951	70	0.471	0.461	65	21.854	144
20	100	0.389	0.372	1	10.496	37	0.365	0.353	−41	7.778	−20	0.336	0.329	−8	7.402	13
30	100	0.384	0.367	−9	11.377	53	0.459	0.444	−60	6.138	−13	0.320	0.313	−30	5.512	17
40	100	0.371	0.354	−10	10.977	49	0.454	0.439	−60	6.095	−16	0.327	0.320	−34	5.092	11
50	100	0.357	0.341	−9	10.459	45	0.467	0.451	−62	6.909	−22	0.335	0.328	−34	5.059	6
60	100	0.339	0.324	−10	9.932	43	0.492	0.476	−66	7.640	−28	0.365	0.357	−40	5.155	−2
70	100	0.343	0.328	−10	10.523	52	0.546	0.527	−75	7.681	−27	0.366	0.358	−46	4.576	2
80	100	0.334	0.319	−7	9.920	45	0.520	0.503	−67	8.655	−29	0.346	0.339	−36	5.036	1
90	100	0.228	0.218	−10	6.973	35	0.279	0.269	−31	4.299	0	0.208	0.204	3	5.810	34
100	100	0.225	0.215	−11	6.897	34	0.256	0.248	−30	3.716	2	0.164	0.161	1	5.212	32

Table A18. Out-of-sample validation figures of selected GAMs of BEL with varying spline function type and fixed spline function number of 10 per dimension under between 100–443 and 150–443 after each tenth and the finally selected smooth function.

k	$K_{\max}$	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
10 Thin plate regression splines under gaussian with identity link																
0	150	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	150	0.642	0.614	27	23.354	126	0.344	0.332	−5	15.463	80	0.509	0.499	71	23.654	156
20	150	0.382	0.365	2	10.101	33	0.341	0.329	−34	7.780	−18	0.338	0.331	1	7.728	18
30	150	0.370	0.354	−7	10.922	45	0.416	0.402	−52	6.497	−14	0.305	0.299	−20	6.103	18
40	150	0.354	0.338	−7	10.412	39	0.404	0.391	−51	6.747	−20	0.308	0.301	−24	5.600	8
50	150	0.347	0.331	−7	10.119	38	0.426	0.412	−54	7.258	−24	0.310	0.304	−27	5.467	4
60	150	0.342	0.327	−4	9.766	34	0.400	0.387	−50	7.600	−26	0.298	0.292	−23	5.615	0
70	150	0.334	0.319	−4	9.601	35	0.428	0.414	−55	8.158	−30	0.318	0.311	−29	5.618	−5
80	150	0.315	0.301	−5	9.093	35	0.432	0.418	−55	8.113	−29	0.334	0.327	−29	6.087	−3
90	150	0.323	0.309	−5	9.436	38	0.388	0.375	−49	6.558	−20	0.297	0.291	−26	5.194	2
100	150	0.309	0.296	−6	8.722	27	0.409	0.395	−54	8.780	−36	0.261	0.255	−27	4.994	−9
110	150	0.309	0.295	−6	8.542	26	0.411	0.397	−54	8.711	−37	0.284	0.278	−33	4.768	−15
120	150	0.206	0.197	−9	5.768	25	0.216	0.209	−23	3.806	−4	0.164	0.161	5	4.519	24
130	150	0.205	0.196	−10	5.759	24	0.226	0.218	−24	3.952	−5	0.175	0.172	4	4.579	24
140	150	0.214	0.205	−10	6.761	34	0.228	0.220	−25	3.363	5	0.167	0.163	6	5.762	36
150	150	0.212	0.203	−10	7.070	37	0.230	0.223	−24	3.575	8	0.173	0.170	8	6.337	40
10 Cubic regression splines under gaussian with identity link																
0	125	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	125	0.638	0.610	27	23.397	127	0.341	0.329	−3	15.829	82	0.519	0.509	73	23.960	158
20	125	0.380	0.364	2	10.038	34	0.339	0.328	−34	7.650	−16	0.345	0.338	0	7.865	18
30	125	0.377	0.360	−6	11.458	53	0.411	0.397	−50	6.035	−5	0.309	0.302	−14	6.976	30
40	125	0.364	0.348	−10	10.929	47	0.421	0.407	−53	5.791	−10	0.315	0.308	−25	5.824	18
50	125	0.348	0.333	−11	10.437	44	0.436	0.421	−56	6.263	−15	0.319	0.312	−27	5.636	13
60	125	0.342	0.327	−5	9.791	36	0.403	0.389	−50	7.282	−23	0.308	0.302	−23	5.789	4
70	125	0.355	0.340	−3	10.502	48	0.442	0.427	−56	7.001	−20	0.327	0.320	−30	5.570	6
80	125	0.349	0.334	−2	10.275	46	0.434	0.419	−55	7.159	−22	0.326	0.319	−29	5.592	4
90	125	0.282	0.269	−5	7.978	37	0.275	0.266	−30	4.426	−3	0.215	0.210	−2	5.088	25
100	125	0.263	0.251	−5	7.109	29	0.301	0.291	−37	5.637	−17	0.200	0.196	−8	3.969	12
110	125	0.255	0.244	−7	6.999	30	0.303	0.292	−37	5.435	−15	0.202	0.198	−6	4.230	16
120	125	0.257	0.246	−7	7.052	30	0.304	0.294	−37	5.371	−14	0.200	0.196	−6	4.232	17
125	125	0.254	0.243	−7	7.139	31	0.299	0.289	−36	5.189	−13	0.197	0.192	−6	4.228	17
10 Duchon splines under gaussian with identity link																
0	100	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	100	0.786	0.752	−5	22.143	110	0.445	0.430	−44	12.588	57	0.406	0.397	1	16.238	102
20	100	0.783	0.749	−32	20.489	101	0.494	0.477	−62	11.319	58	0.357	0.350	−21	15.316	98
30	100	0.782	0.748	−39	21.134	98	0.538	0.520	−59	12.715	64	0.422	0.413	−3	18.621	121
40	100	0.816	0.780	−45	22.125	98	0.559	0.540	−63	13.071	65	0.450	0.440	−10	18.616	119
50	100	0.823	0.787	−45	21.473	96	0.555	0.536	−63	12.672	63	0.451	0.441	−10	18.114	116
53	100	0.821	0.785	−44	21.348	94	0.545	0.526	−61	12.593	62	0.446	0.437	−8	18.091	116
10 Eilers and Marx style P-splines under gaussian with identity link in stagewise selection of length 5																
0	150	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	150	0.648	0.619	27	23.688	128	0.349	0.337	−7	15.566	80	0.506	0.495	71	23.889	158
20	150	0.398	0.380	1	10.946	45	0.358	0.346	−37	7.063	−7	0.338	0.331	1	8.102	31
30	150	0.393	0.376	−9	11.983	59	0.435	0.421	−55	5.575	−2	0.299	0.293	−17	6.928	36
40	150	0.371	0.355	−8	11.374	55	0.449	0.434	−57	5.738	−9	0.314	0.308	−26	5.770	23
50	150	0.363	0.347	−9	10.956	50	0.460	0.444	−60	6.249	−14	0.315	0.308	−28	5.492	17
60	150	0.349	0.334	−8	10.479	46	0.443	0.428	−56	6.526	−17	0.305	0.298	−26	5.427	14
70	150	0.349	0.333	−6	10.629	51	0.464	0.449	−60	6.687	−17	0.325	0.318	−29	5.501	13
80	150	0.350	0.335	−7	10.465	48	0.468	0.452	−60	7.036	−19	0.335	0.328	−29	5.563	11
90	150	0.350	0.335	−7	10.639	51	0.470	0.454	−60	6.683	−17	0.330	0.323	−29	5.453	14
100	150	0.334	0.319	−8	9.960	46	0.468	0.452	−60	7.170	−20	0.339	0.332	−29	5.835	11
110	150	0.337	0.323	−9	10.249	48	0.450	0.435	−58	6.171	−15	0.329	0.322	−31	5.267	12
120	150	0.339	0.324	−7	10.283	45	0.433	0.419	−55	6.420	−17	0.320	0.313	−28	5.340	10
130	150	0.269	0.257	−13	8.912	43	0.365	0.352	−46	4.891	−4	0.244	0.238	−12	5.503	30
140	150	0.255	0.244	−12	8.157	36	0.356	0.344	−44	5.415	−10	0.246	0.241	−10	5.196	24
150	150	0.261	0.250	−12	8.514	39	0.368	0.355	−46	5.267	−9	0.245	0.240	−12	5.162	25

Table A19. Out-of-sample validation figures of selected GAMs of BEL with varying random component link function combination and fixed spline function number of 4 per dimension under between 40–443 and 150–443 after each tenth and the finally selected smooth function.

k	$K_{\max}$	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
4 Thin plate regression splines under gaussian with identity link in stagewise selection of length 5																
0	150	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	150	0.632	0.604	28	22.019	116	0.345	0.334	−8	13.247	65	0.479	0.469	66	21.072	139
20	150	0.406	0.388	0	11.330	44	0.375	0.362	−42	7.254	−12	0.341	0.334	−6	7.709	24
30	150	0.399	0.382	−11	12.268	59	0.465	0.449	−61	5.744	−6	0.314	0.307	−26	6.116	29
40	150	0.371	0.355	−8	11.415	53	0.480	0.463	−64	6.380	−16	0.340	0.332	−34	5.283	13
50	150	0.392	0.375	−13	12.079	59	0.520	0.503	−70	5.961	−12	0.365	0.358	−39	5.368	19
60	150	0.306	0.292	−15	9.833	48	0.405	0.391	−51	5.283	−2	0.273	0.267	−10	6.484	39
70	150	0.272	0.260	−15	9.896	56	0.321	0.310	−35	5.227	22	0.232	0.228	12	10.460	69
80	150	0.249	0.238	−17	8.627	49	0.308	0.297	−36	4.588	16	0.205	0.201	9	9.100	60
90	150	0.261	0.250	−17	9.262	54	0.325	0.314	−39	4.639	18	0.195	0.191	5	9.340	62
100	150	0.254	0.243	−18	9.593	55	0.340	0.328	−42	4.626	17	0.196	0.192	3	9.312	62
110	150	0.255	0.244	−18	9.407	54	0.336	0.324	−40	4.640	18	0.207	0.203	4	9.325	62
120	150	0.243	0.233	−16	8.474	48	0.307	0.296	−38	4.023	13	0.186	0.182	1	7.819	51
130	150	0.241	0.230	−16	8.481	49	0.308	0.298	−37	4.108	13	0.183	0.179	2	8.075	53
140	150	0.235	0.225	−15	8.018	45	0.295	0.285	−35	3.865	10	0.173	0.169	2	7.182	47
150	150	0.240	0.229	−15	8.192	46	0.291	0.281	−35	3.907	13	0.176	0.172	3	7.641	50
4 Thin plate regression splines under gaussian with log link in stagewise selection of length 5																
0	40	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	40	0.788	0.754	8	23.011	114	0.423	0.408	26	22.471	118	0.700	0.685	94	28.248	186
20	40	0.452	0.432	−4	12.761	50	0.421	0.406	−48	7.626	−9	0.360	0.352	−11	8.166	29
30	40	0.462	0.442	−10	14.180	72	0.527	0.509	−68	6.209	−1	0.368	0.360	−32	7.116	36
40	40	0.438	0.419	−7	13.382	66	0.524	0.506	−69	6.189	−10	0.373	0.365	−39	5.913	20
4 Thin plate regression splines under gamma with identity link in stagewise selection of length 5																
0	70	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	70	0.625	0.598	31	21.068	110	0.332	0.321	−5	12.421	60	0.486	0.475	68	19.997	132
20	70	0.394	0.377	1	10.887	41	0.357	0.345	−39	7.283	−15	0.340	0.333	−6	7.641	19
30	70	0.383	0.367	−10	11.985	56	0.467	0.451	−62	5.853	−10	0.331	0.324	−30	5.742	22
40	70	0.289	0.277	−11	9.447	45	0.346	0.335	−41	5.159	0	0.256	0.250	−2	6.682	39
50	70	0.307	0.293	−11	10.339	53	0.389	0.376	−50	4.922	0	0.252	0.247	−11	6.294	38
60	70	0.308	0.295	−14	10.455	56	0.372	0.360	−49	4.377	7	0.222	0.218	−9	7.143	46
70	70	0.270	0.259	−16	9.999	57	0.325	0.314	−36	5.280	23	0.245	0.240	10	10.416	69
4 Thin plate regression splines under gamma with log link in stagewise selection of length 5																
0	120	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	120	0.780	0.745	12	22.104	101	0.436	0.421	35	21.150	110	0.736	0.720	101	26.692	175
20	120	0.497	0.475	−1	14.721	71	0.457	0.442	−55	6.794	2	0.360	0.352	−16	8.605	41
30	120	0.437	0.418	−7	13.581	66	0.483	0.467	−61	6.042	−3	0.364	0.357	−28	7.018	31
40	120	0.418	0.400	−7	12.575	58	0.505	0.488	−67	6.530	−16	0.382	0.374	−40	5.844	11
50	120	0.416	0.397	−11	12.456	58	0.522	0.505	−70	6.310	−15	0.392	0.384	−42	5.536	12
60	120	0.407	0.390	−11	12.201	59	0.547	0.529	−74	6.706	−19	0.411	0.403	−47	5.476	8
70	120	0.407	0.390	−7	12.104	59	0.480	0.464	−64	5.741	−13	0.356	0.349	−39	5.173	12
80	120	0.274	0.262	−9	10.461	60	0.319	0.309	−31	5.409	23	0.257	0.251	16	10.636	70
90	120	0.252	0.241	−10	9.362	52	0.289	0.279	−31	4.594	17	0.195	0.191	9	8.753	58
100	120	0.239	0.229	−13	8.404	46	0.254	0.245	−26	4.423	18	0.182	0.178	13	8.710	57
110	120	0.251	0.240	−15	8.307	46	0.256	0.248	−28	4.442	19	0.174	0.171	11	8.708	57
120	120	0.252	0.241	−16	8.368	47	0.263	0.254	−29	4.585	20	0.171	0.167	9	8.830	58
4 Thin plate regression splines under inverse gaussian with identity link in stagewise selection of length 5																
0	85	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	85	0.622	0.595	33	20.643	108	0.328	0.317	−3	12.034	57	0.488	0.478	68	19.473	129
20	85	0.443	0.423	0	13.176	63	0.412	0.398	−49	6.644	−1	0.336	0.329	−11	8.149	37
30	85	0.390	0.373	−10	12.087	60	0.481	0.465	−65	5.771	−9	0.334	0.327	−33	5.777	23
40	85	0.280	0.268	−9	9.655	48	0.339	0.327	−39	5.079	4	0.255	0.250	1	7.154	44
50	85	0.296	0.283	−10	9.742	48	0.374	0.362	−48	4.933	−3	0.242	0.237	−10	5.768	34
60	85	0.310	0.297	−14	10.405	54	0.367	0.354	−48	4.592	6	0.232	0.227	−8	7.165	46
70	85	0.272	0.260	−12	10.279	58	0.313	0.303	−34	5.205	22	0.249	0.244	12	10.286	67
80	85	0.247	0.236	−14	8.583	48	0.293	0.283	−33	4.594	15	0.217	0.213	10	8.776	58
85	85	0.250	0.239	−17	8.739	50	0.325	0.314	−38	4.585	14	0.218	0.213	6	8.871	58
4 Thin plate regression splines under inverse gaussian with log link in stagewise selection of length 5																
0	75	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	75	0.778	0.744	14	21.780	95	0.446	0.431	40	20.520	106	0.756	0.740	104	25.969	170
20	75	0.491	0.470	−1	14.542	69	0.452	0.437	−55	6.759	0	0.362	0.355	−17	8.423	38
30	75	0.425	0.407	−7	13.142	62	0.472	0.456	−60	6.123	−5	0.366	0.358	−27	6.854	27
40	75	0.406	0.388	−7	12.151	54	0.499	0.482	−66	6.757	−19	0.389	0.381	−41	5.920	7
50	75	0.412	0.394	−11	12.543	56	0.513	0.495	−69	6.309	−16	0.396	0.388	−42	5.655	10
60	75	0.298	0.285	−12	9.519	47	0.392	0.379	−50	5.298	−4	0.265	0.260	−10	6.172	36
70	75	0.263	0.251	−13	9.789	56	0.298	0.288	−31	5.406	23	0.227	0.222	16	10.673	70
75	75	0.258	0.246	−14	9.181	52	0.300	0.290	−33	5.049	19	0.223	0.219	13	9.837	65
4 Thin plate regression splines under inverse gaussian with $\frac{1}{\mu^2}$ link in stagewise selection of length 5																
0	55	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	55	0.803	0.768	2	23.425	117	0.383	0.370	−24	15.197	76	0.435	0.426	27	19.713	127
20	55	0.448	0.428	8	12.645	61	0.331	0.320	−29	7.088	10	0.330	0.323	18	9.983	56
30	55	0.387	0.370	1	12.458	64	0.331	0.320	−29	6.701	20	0.311	0.304	22	11.099	70
40	55	0.341	0.326	−5	11.661	61	0.339	0.328	−35	5.920	17	0.271	0.266	11	9.851	63
45	55	0.343	0.328	−9	10.928	55	0.361	0.349	−38	6.111	12	0.300	0.294	9	9.451	59
50	55	0.336	0.321	−7	10.645	55	0.355	0.343	−40	5.319	8	0.250	0.245	7	8.525	54
55	55	0.328	0.314	−9	10.595	56	0.328	0.317	−35	5.325	15	0.241	0.236	16	10.249	67

Table A20. Out-of-sample validation figures of selected GAMs of BEL with varying random component link function combination and fixed spline function number of 8 per dimension under between 50–443 and 150–443 after each tenth and the finally selected smooth function.

k	$K_{\max}$	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
8 Thin plate regression splines under gaussian with identity link																
0	150	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	150	0.639	0.611	27	23.176	125	0.340	0.329	−3	15.517	80	0.516	0.505	73	23.627	156
20	150	0.375	0.359	3	9.604	26	0.334	0.322	−33	8.378	−24	0.341	0.333	1	7.711	10
30	150	0.361	0.345	−7	10.444	41	0.415	0.401	−52	6.961	−19	0.304	0.297	−21	5.871	13
40	150	0.356	0.340	−5	10.098	36	0.425	0.410	−54	7.920	−28	0.311	0.304	−27	5.647	−1
50	150	0.339	0.324	−7	9.712	33	0.418	0.404	−53	7.746	−27	0.311	0.304	−26	5.596	0
60	150	0.325	0.311	−6	9.037	26	0.411	0.397	−52	8.706	−34	0.310	0.304	−26	5.850	−8
70	150	0.325	0.311	−4	9.180	31	0.429	0.414	−55	8.773	−34	0.326	0.319	−30	5.912	−9
80	150	0.309	0.296	−5	8.618	29	0.430	0.415	−55	8.984	−35	0.336	0.329	−29	6.382	−9
90	150	0.313	0.299	−5	8.981	32	0.384	0.371	−48	7.390	−26	0.300	0.293	−26	5.430	−4
100	150	0.328	0.313	−6	9.910	47	0.400	0.387	−51	5.572	−12	0.291	0.285	−25	5.064	13
110	150	0.256	0.245	−10	7.985	38	0.326	0.315	−40	4.655	−6	0.201	0.197	−6	5.002	28
120	150	0.253	0.242	−9	7.340	30	0.321	0.310	−39	5.542	−14	0.209	0.204	−5	4.541	20
130	150	0.252	0.241	−9	7.767	34	0.326	0.315	−40	5.197	−11	0.205	0.201	−5	4.770	24
140	150	0.245	0.234	−8	7.592	33	0.322	0.311	−41	5.315	−15	0.197	0.193	−7	4.317	20
150	150	0.217	0.208	−11	6.477	32	0.239	0.231	−26	3.652	2	0.179	0.175	6	5.578	34
8 Thin plate regression splines under gaussian with log link in stagewise selection of length 5																
0	50	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	50	0.757	0.724	10	21.570	101	0.444	0.429	39	22.141	116	0.755	0.739	106	27.693	182
20	50	0.401	0.383	1	10.278	23	0.359	0.347	−35	9.154	−28	0.362	0.354	−1	8.110	7
30	50	0.396	0.379	−5	11.249	43	0.438	0.424	−53	7.692	−20	0.339	0.332	−19	6.803	14
40	50	0.382	0.365	−5	11.036	45	0.470	0.454	−60	7.846	−25	0.351	0.344	−31	6.234	4
50	50	0.370	0.353	−8	10.487	39	0.464	0.448	−60	8.000	−28	0.340	0.333	−32	5.901	0
8 Thin plate regression splines under gamma with identity link in stagewise selection of length 5																
0	100	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	100	0.637	0.609	29	22.743	123	0.334	0.323	−3	14.941	77	0.510	0.500	72	22.871	151
20	100	0.370	0.354	4	9.537	27	0.324	0.313	−31	8.076	−22	0.340	0.333	1	7.725	10
30	100	0.359	0.344	−8	10.558	44	0.414	0.400	−52	6.415	−15	0.305	0.298	−22	5.909	16
40	100	0.329	0.314	−9	9.643	37	0.402	0.388	−51	6.673	−21	0.321	0.314	−26	5.702	4
50	100	0.342	0.327	−7	9.631	33	0.409	0.395	−52	7.553	−27	0.326	0.320	−28	5.863	−3
60	100	0.324	0.310	−6	9.114	28	0.409	0.395	−52	8.421	−32	0.327	0.320	−28	6.067	−9
70	100	0.328	0.314	−6	9.617	41	0.451	0.435	−59	7.631	−26	0.349	0.342	−35	5.796	−2
80	100	0.270	0.258	−9	7.944	37	0.324	0.313	−38	5.068	−7	0.221	0.217	−2	5.461	29
90	100	0.279	0.267	−10	8.926	47	0.341	0.329	−40	4.595	2	0.224	0.219	−2	6.713	41
100	100	0.272	0.260	−11	8.654	44	0.335	0.324	−40	4.532	0	0.216	0.211	−2	6.397	38
8 Thin plate regression splines under gamma with log link in stagewise selection of length 5																
0	110	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	110	0.762	0.729	13	21.360	95	0.458	0.443	45	21.527	112	0.773	0.756	108	26.743	176
20	110	0.442	0.422	2	12.416	49	0.396	0.382	−44	7.515	−12	0.349	0.342	−8	8.083	24
30	110	0.387	0.370	−3	11.147	45	0.414	0.400	−49	7.058	−16	0.338	0.331	−18	6.847	16
40	110	0.372	0.356	−6	10.826	43	0.458	0.442	−59	7.546	−24	0.360	0.352	−34	6.225	1
50	110	0.357	0.342	−9	10.240	36	0.458	0.443	−60	7.977	−29	0.357	0.349	−36	6.073	−5
60	110	0.351	0.336	−5	9.866	30	0.439	0.424	−56	9.066	−36	0.353	0.346	−35	6.537	−15
70	110	0.354	0.339	−5	10.130	37	0.458	0.442	−59	8.442	−31	0.364	0.356	−37	6.271	−9
80	110	0.359	0.344	−6	10.122	37	0.463	0.447	−60	8.529	−32	0.371	0.363	−37	6.412	−9
90	110	0.282	0.270	−10	9.017	47	0.364	0.352	−44	4.991	−2	0.249	0.244	−6	6.286	36
100	110	0.268	0.256	−11	7.807	37	0.320	0.309	−38	4.748	−5	0.209	0.204	−1	5.604	32
110	110	0.259	0.247	−11	7.373	34	0.312	0.302	−37	4.801	−7	0.201	0.197	0	5.354	31

Table A21. Out-of-sample validation figures of selected GAMs of BEL in adaptive forward stepwise and stagewise selection of length 5 under between 25–443 and 100–443 after each tenth and the finally selected smooth function.

k	$K_{\max}$	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
8 Thin plate regression splines under gaussian with log link																
0	25	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	25	0.663	0.634	26	23.298	123	0.341	0.330	1	16.218	84	0.547	0.536	78	24.370	161
20	25	0.398	0.381	2	10.221	23	0.361	0.349	−35	9.380	−28	0.375	0.367	−1	8.460	6
25	25	0.411	0.393	2	11.892	47	0.410	0.397	−47	7.709	−17	0.324	0.317	−11	7.120	19
8 Thin plate regression splines under gaussian with log link in stagewise selection of length 5																
0	50	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	50	0.757	0.724	10	21.570	101	0.444	0.429	39	22.141	116	0.755	0.739	106	27.693	182
20	50	0.401	0.383	1	10.278	23	0.359	0.347	−35	9.154	−28	0.362	0.354	−1	8.110	7
30	50	0.396	0.379	−5	11.249	43	0.438	0.424	−53	7.692	−20	0.339	0.332	−19	6.803	14
40	50	0.382	0.365	−5	11.036	45	0.470	0.454	−60	7.846	−25	0.351	0.344	−31	6.234	4
50	50	0.370	0.353	−8	10.487	39	0.464	0.448	−60	8.000	−28	0.340	0.333	−32	5.901	0
8 Thin plate regression splines under gamma with identity link																
0	71	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	71	0.637	0.609	29	22.743	123	0.334	0.323	−3	14.941	77	0.510	0.500	72	22.871	151
20	71	0.386	0.369	8	10.141	31	0.310	0.299	−26	7.904	−18	0.358	0.350	8	8.140	16
30	71	0.359	0.344	−8	10.558	44	0.414	0.400	−52	6.415	−15	0.305	0.298	−22	5.909	16
40	71	0.329	0.314	−9	9.643	37	0.402	0.388	−51	6.673	−21	0.321	0.314	−26	5.702	4
50	71	0.338	0.324	−7	9.543	32	0.412	0.399	−53	7.748	−28	0.324	0.318	−29	5.805	−4
60	71	0.324	0.310	−6	9.114	28	0.409	0.395	−52	8.421	−32	0.327	0.320	−28	6.067	−9
70	71	0.327	0.313	−5	9.417	36	0.434	0.419	−56	8.017	−29	0.342	0.335	−32	5.967	−5
71	71	0.291	0.278	−4	8.639	41	0.341	0.329	−43	5.205	−12	0.196	0.192	−17	3.898	14
8 Thin plate regression splines under gamma with identity link in stagewise selection of length 5																
0	100	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	100	0.637	0.609	29	22.743	123	0.334	0.323	−3	14.941	77	0.510	0.500	72	22.871	151
20	100	0.370	0.354	4	9.537	27	0.324	0.313	−31	8.076	−22	0.340	0.333	1	7.725	10
30	100	0.359	0.344	−8	10.558	44	0.414	0.400	−52	6.415	−15	0.305	0.298	−22	5.909	16
40	100	0.329	0.314	−9	9.643	37	0.402	0.388	−51	6.673	−21	0.321	0.314	−26	5.702	4
50	100	0.342	0.327	−7	9.631	33	0.409	0.395	−52	7.553	−27	0.326	0.320	−28	5.863	−3
60	100	0.324	0.310	−6	9.114	28	0.409	0.395	−52	8.421	−32	0.327	0.320	−28	6.067	−9
70	100	0.328	0.314	−6	9.617	41	0.451	0.435	−59	7.631	−26	0.349	0.342	−35	5.796	−2
80	100	0.270	0.258	−9	7.944	37	0.324	0.313	−38	5.068	−7	0.221	0.217	−2	5.461	29
90	100	0.279	0.267	−10	8.926	47	0.341	0.329	−40	4.595	2	0.224	0.219	−2	6.713	41
100	100	0.272	0.260	−11	8.654	44	0.335	0.324	−40	4.532	0	0.216	0.211	−2	6.397	38

Table A22. Out-of-sample validation figures of selected GAMs of BEL with varying spline function number per dimension and fixed spline function type under between 91–443 and 150–443 after each tenth and the finally selected smooth function or after each dynamically stagewise selected smooth function block. Thereby furthermore a variation in the random component link function combination.

k	$K_{\max}$	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
5 Eilers and Marx style P-splines under gaussian with identity link																
0	100	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	100	0.643	0.615	29	22.836	123	0.344	0.332	−9	13.951	70	0.471	0.461	65	21.854	144
20	100	0.389	0.372	1	10.496	37	0.365	0.353	−41	7.778	−20	0.336	0.329	−8	7.402	13
30	100	0.384	0.367	−9	11.377	53	0.459	0.444	−60	6.138	−13	0.320	0.313	−30	5.512	17
40	100	0.371	0.354	−10	10.977	49	0.454	0.439	−60	6.095	−16	0.327	0.320	−34	5.092	11
50	100	0.357	0.341	−9	10.459	45	0.467	0.451	−62	6.909	−22	0.335	0.328	−34	5.059	6
60	100	0.339	0.324	−10	9.932	43	0.492	0.476	−66	7.640	−28	0.365	0.357	−40	5.155	−2
70	100	0.343	0.328	−10	10.523	52	0.546	0.527	−75	7.681	−27	0.366	0.358	−46	4.576	2
80	100	0.334	0.319	−7	9.920	45	0.520	0.503	−67	8.655	−29	0.346	0.339	−36	5.036	1
90	100	0.228	0.218	−10	6.973	35	0.279	0.269	−31	4.299	0	0.208	0.204	3	5.810	34
100	100	0.225	0.215	−11	6.897	34	0.256	0.248	−30	3.716	2	0.164	0.161	1	5.212	32
8 Eilers and Marx style P-splines under inverse gaussian with $\frac{1}{\mu^2}$ link in dynamically stagewise selection of proportion 0.25																
0	91	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
5	91	1.574	1.505	−18	41.688	233	0.732	0.708	−75	30.201	161	0.384	0.376	42	42.135	278
11	91	0.817	0.781	−3	22.381	113	0.396	0.383	−34	13.475	68	0.412	0.404	23	19.322	124
21	91	0.679	0.650	−9	24.203	138	0.763	0.738	−102	8.222	31	0.424	0.415	−44	13.548	89
37	91	0.525	0.502	1	15.485	79	0.521	0.504	−63	6.154	0	0.397	0.389	−30	7.461	33
62	91	0.505	0.482	−1	14.208	64	0.507	0.490	−61	6.842	−10	0.418	0.410	−33	7.405	18
91	91	0.309	0.296	−11	9.688	45	0.335	0.324	−36	5.239	6	0.279	0.273	2	7.420	43
10 Eilers and Marx style P-splines under gaussian with identity link in stagewise selection of length 5																
0	150	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	150	0.648	0.619	27	23.688	128	0.349	0.337	−7	15.566	80	0.506	0.495	71	23.889	158
20	150	0.398	0.380	1	10.946	45	0.358	0.346	−37	7.063	−7	0.338	0.331	1	8.102	31
30	150	0.393	0.376	−9	11.983	59	0.435	0.421	−55	5.575	−2	0.299	0.293	−17	6.928	36
40	150	0.371	0.355	−8	11.374	55	0.449	0.434	−57	5.738	−9	0.314	0.308	−26	5.770	23
50	150	0.363	0.347	−9	10.956	50	0.460	0.444	−60	6.249	−14	0.315	0.308	−28	5.492	17
60	150	0.349	0.334	−8	10.479	46	0.443	0.428	−56	6.526	−17	0.305	0.298	−26	5.427	14
70	150	0.349	0.333	−6	10.629	51	0.464	0.449	−60	6.687	−17	0.325	0.318	−29	5.501	13
80	150	0.350	0.335	−7	10.465	48	0.468	0.452	−60	7.036	−19	0.335	0.328	−29	5.563	11
90	150	0.350	0.335	−7	10.639	51	0.470	0.454	−60	6.683	−17	0.330	0.323	−29	5.453	14
100	150	0.334	0.319	−8	9.960	46	0.468	0.452	−60	7.170	−20	0.339	0.332	−29	5.835	11
110	150	0.337	0.323	−9	10.249	48	0.450	0.435	−58	6.171	−15	0.329	0.322	−31	5.267	12
120	150	0.339	0.324	−7	10.283	45	0.433	0.419	−55	6.420	−17	0.320	0.313	−28	5.340	10
130	150	0.269	0.257	−13	8.912	43	0.365	0.352	−46	4.891	−4	0.244	0.238	−12	5.503	30
140	150	0.255	0.244	−12	8.157	36	0.356	0.344	−44	5.415	−10	0.246	0.241	−10	5.196	24
150	150	0.261	0.250	−12	8.514	39	0.368	0.355	−46	5.267	−9	0.245	0.240	−12	5.162	25

Table A23. Maximum allowed numbers of smooth functions and out-of-sample validation figures of all derived GAMs of BEL under between 25–443 and 150–443 after the final iteration. Highlighted in green and red respectively the best and worst validation figures.

k	$K_{\max}$	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
4 Thin plate regression splines under gaussian with identity link																
150	150	0.240	0.229	−15	8.192	46	0.291	0.281	−35	3.907	13	0.176	0.172	3	7.641	50
5 Thin plate regression splines under gaussian with identity link																
100	100	0.287	0.274	−11	9.431	48	0.397	0.383	−50	5.402	−5	0.202	0.198	−9	5.945	36
8 Thin plate regression splines under gaussian with identity link																
150	150	0.217	0.208	−11	6.477	32	0.239	0.231	−26	3.652	2	0.179	0.175	6	5.578	34
10 Thin plate regression splines under gaussian with identity link																
150	150	0.212	0.203	−10	7.070	37	0.230	0.223	−24	3.575	8	0.173	0.170	8	6.337	40
5 Cubic regression splines under gaussian with identity link																
100	100	0.268	0.256	−12	9.903	52	0.399	0.386	−51	5.182	−2	0.226	0.221	−9	6.533	40
5 Duchon splines under gaussian with identity link																
56	100	0.666	0.636	−18	18.532	86	0.288	0.279	−14	14.643	75	0.406	0.397	40	19.757	129
5 Eilers and Marx style P-splines under gaussian with identity link																
100	100	0.225	0.215	−11	6.897	34	0.256	0.248	−30	3.716	2	0.164	0.161	1	5.212	32
10 Cubic regression splines under gaussian with identity link																
125	125	0.254	0.243	−7	7.139	31	0.299	0.289	−36	5.189	−13	0.197	0.192	−6	4.228	17
10 Duchon splines under gaussian with identity link																
53	100	0.821	0.785	−44	21.348	94	0.545	0.526	−61	12.593	62	0.446	0.437	−8	18.091	116
10 Eilers and Marx style P-splines under gaussian with identity link in stagewise selection of length 5																
150	150	0.261	0.250	−12	8.514	−39	0.368	0.355	−46	5.267	9	0.245	0.240	−12	5.162	−25
8 Thin plate regression splines under gaussian with log link																
25	25	0.411	0.393	2	11.892	47	0.410	0.397	−47	7.709	−17	0.324	0.317	−11	7.120	19
8 Thin plate regression splines under gaussian with log link in stagewise selection of length 5																
50	50	0.370	0.353	−8	10.487	39	0.464	0.448	−60	8.000	−28	0.340	0.333	−32	5.901	0
8 Thin plate regression splines under gamma with identity link																
71	71	0.291	0.278	−4	8.639	41	0.341	0.329	−43	5.205	−12	0.196	0.192	−17	3.898	14
8 Thin plate regression splines under gamma with identity link in stagewise selection of length 5																
100	100	0.272	0.260	−11	8.654	44	0.335	0.324	−40	4.532	0	0.216	0.211	−2	6.397	38
4 Thin plate regression splines under gaussian with identity link in stagewise selection of length 5																
150	150	0.240	0.229	−15	8.192	46	0.291	0.281	−35	3.907	13	0.176	0.172	3	7.641	50
4 Thin plate regression splines under gaussian with log link in stagewise selection of length 5																
40	40	0.438	0.419	−7	13.382	66	0.524	0.506	−69	6.189	−10	0.373	0.365	−39	5.913	20
4 Thin plate regression splines under gamma with identity link in stagewise selection of length 5																
70	70	0.270	0.259	−16	9.999	57	0.325	0.314	−36	5.280	23	0.245	0.240	10	10.416	69
4 Thin plate regression splines under gaussian with log link in stagewise selection of length 5																
120	120	0.252	0.241	−16	8.368	47	0.263	0.254	−29	4.585	20	0.171	0.167	9	8.830	58
4 Thin plate regression splines under inverse gaussian with identity link in stagewise selection of length 5																
85	85	0.250	0.239	−17	8.739	50	0.325	0.314	−38	4.585	14	0.218	0.213	6	8.871	58
4 Thin plate regression splines under inverse gaussian with log link in stagewise selection of length 5																
75	75	0.258	0.246	−14	9.181	52	0.300	0.290	−33	5.049	19	0.223	0.219	13	9.837	65
4 Thin plate regression splines under inverse gaussian with $\frac{1}{\mu^2}$ link in stagewise selection of length 5																
55	55	0.328	0.314	−9	10.595	56	0.328	0.317	−35	5.325	15	0.241	0.236	16	10.249	67
8 Thin plate regression splines under gamma with log link in stagewise selection of length 5																
110	110	0.259	0.247	−11	7.373	34	0.312	0.302	−37	4.801	−7	0.201	0.197	0	5.354	31
8 Eilers and Marx style P-splines under inverse gaussian with $\frac{1}{\mu^2}$ link in dynamic stagewise selection of proportion 0.25																
91	91	0.309	0.296	−11	9.688	45	0.335	0.324	−36	5.239	6	0.279	0.273	2	7.420	43

Table A24. Feasible generalized least-squares (FGLS) variance models of BEL corresponding to $M_{\max} \in \{2, 6, 10, 14, 18, 22\}$ derived by adaptive selection from the set of basis functions of the 150–443 OLS proxy function given in Table A1 with exponents summing up to at max two. Furthermore, p-values of Breusch-Pagan test, AIC scores and out-of-sample MAEs in % after each iteration.

m	r_m^1	r_m^2	r_m^3	r_m^4	r_m^5	r_m^6	r_m^7	r_m^8	r_m^9	r_m^{10}	r_m^{11}	r_m^{12}	r_m^{13}	r_m^{14}	r_m^{15}	BP.p-val	AIC	v.mae	ns.mae	cr.mae
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1_{-20}	325,850	0.238	0.252	0.154
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1_{-20}	322,452	0.238	0.246	0.122
2	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1_{-20}	315,980	0.239	0.255	0.153
3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1_{-20}	314,077	0.237	0.226	0.165
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1_{-20}	312,280	0.231	0.206	0.184
5	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1_{-20}	312,114	0.231	0.205	0.185
6	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1_{-20}	311,949	0.231	0.203	0.186
7	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1_{-20}	311,794	0.232	0.202	0.187
8	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1_{-20}	311,700	0.235	0.200	0.190
9	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1_{-20}	311,610	0.233	0.198	0.190
10	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	1_{-20}	311,363	0.227	0.194	0.195
11	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1_{-20}	311,293	0.229	0.194	0.197
12	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	1_{-20}	311,237	0.228	0.193	0.198
13	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1_{-20}	311,196	0.230	0.193	0.198
14	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1.5_{-20}	311,161	0.231	0.193	0.200
15	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	7.1_{-19}	311,136	0.231	0.191	0.202
16	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	5_{-15}	311,091	0.228	0.189	0.201
17	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	5.8_{-13}	311,067	0.228	0.188	0.203
18	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	8.3_{-13}	311,048	0.228	0.187	0.204
19	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	3.2_{-12}	311,030	0.228	0.188	0.204
20	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	2.7_{-12}	311,003	0.230	0.188	0.205
21	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1.3_{-11}	310,988	0.230	0.188	0.206
22	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	9.4_{-11}	310,974	0.230	0.187	0.207

Table A25. FGLS variance models of BEL corresponding to $M_{\max} \in \{2, 6, 10, 14, 18, 22\}$ derived by adaptive selection from the set of basis functions of the 300–886 OLS proxy function given in Table A3 with exponents summing up to at max two. Furthermore, p-values of Breusch-Pagan test, AIC scores and out-of-sample MAEs in % after each iteration.

m	r_m^1	r_m^2	r_m^3	r_m^4	r_m^5	r_m^6	r_m^7	r_m^8	r_m^9	r_m^{10}	r_m^{11}	r_m^{12}	r_m^{13}	r_m^{14}	r_m^{15}	BP.p-val	AIC	v.mae	ns.mae	cr.mae
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1_{-20}	325,459	0.195	0.275	0.175
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1_{-20}	322,077	0.199	0.273	0.166
2	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1_{-20}	315,615	0.196	0.275	0.175
3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1_{-20}	313,659	0.195	0.255	0.175
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1_{-20}	311,864	0.198	0.239	0.182
5	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1_{-20}	311,704	0.198	0.236	0.182
6	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1_{-20}	311,554	0.200	0.240	0.183
7	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1_{-20}	311,454	0.199	0.241	0.183
8	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1_{-20}	311,360	0.199	0.238	0.186
9	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	1_{-20}	311,318	0.201	0.236	0.188
10	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1_{-20}	311,287	0.203	0.234	0.189
11	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	1_{-20}	311,260	0.203	0.233	0.189
12	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	1_{-20}	311,237	0.203	0.232	0.189
13	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	3.7_{-17}	311,001	0.200	0.223	0.192
14	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1.7_{-16}	310,980	0.200	0.222	0.194
15	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	7.6_{-13}	310,934	0.200	0.220	0.196
16	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	4.2_{-11}	310,912	0.200	0.218	0.197
17	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	1.3_{-10}	310,895	0.200	0.219	0.198
18	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	2.3_{-10}	310,881	0.200	0.217	0.198
19	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	7.6_{-10}	310,867	0.200	0.218	0.197
20	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	3.4_{-9}	310,854	0.200	0.218	0.196
21	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	9.9_{-9}	310,843	0.200	0.218	0.196
22	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	3.1_{-8}	310,832	0.200	0.217	0.196

Table A26. Iteration-wise out-of-sample validation figures in adaptive variance model selection of BEL corresponding to $M_{\max} \in \{2, 6, 10, 14, 18, 22\}$ based on the 150–443 OLS proxy function given in Table A1 with exponents summing up to at max two. Simultaneously type I FGLS regression results.

m	**v.mae**	**v.mae^a**	**v.res**	**v.mae^0**	**v.res^0**	**ns.mae**	**ns.mae^a**	**ns.res**	**ns.mae^0**	**ns.res^0**	**cr.mae**	**cr.mae^a**	**cr.res**	**cr.mae^0**	**cr.res^0**
0	0.238	0.228	−15	8.103	45	0.252	0.243	−30	3.984	16	0.154	0.151	3	7.379	49
1	0.238	0.228	−15	8.668	49	0.246	0.238	−30	4.120	19	0.122	0.120	3	7.873	52
2	0.239	0.229	−16	8.147	46	0.255	0.246	−30	4.032	17	0.153	0.149	2	7.489	49
3	0.237	0.226	−15	7.789	43	0.226	0.218	−24	4.423	20	0.165	0.162	10	8.117	54
4	0.231	0.221	−13	7.684	42	0.206	0.199	−18	4.817	22	0.184	0.180	17	8.756	58
5	0.231	0.221	−13	7.666	42	0.205	0.198	−18	4.803	22	0.185	0.181	17	8.740	58
6	0.231	0.221	−13	7.577	41	0.203	0.196	−18	4.762	22	0.186	0.183	17	8.637	57
7	0.232	0.222	−12	7.661	42	0.202	0.195	−17	4.787	22	0.187	0.183	18	8.691	57
8	0.235	0.225	−12	7.774	42	0.200	0.193	−17	4.914	23	0.190	0.186	19	8.912	59
9	0.233	0.223	−11	7.692	42	0.198	0.191	−16	4.838	23	0.190	0.186	19	8.763	58
10	0.227	0.217	−10	7.460	40	0.194	0.188	−15	4.708	21	0.195	0.191	20	8.537	56
11	0.229	0.219	−10	7.447	40	0.194	0.187	−15	4.686	21	0.197	0.193	20	8.455	56
12	0.228	0.218	−10	7.426	40	0.193	0.186	−14	4.687	21	0.198	0.194	20	8.444	56
13	0.230	0.220	−9	7.513	41	0.193	0.187	−14	4.696	21	0.198	0.194	21	8.491	56
14	0.231	0.221	−9	7.527	41	0.193	0.186	−14	4.701	21	0.200	0.195	21	8.497	56
15	0.231	0.221	−9	7.523	41	0.191	0.185	−13	4.742	21	0.202	0.197	22	8.569	57
16	0.228	0.218	−9	7.437	40	0.189	0.182	−13	4.730	21	0.201	0.197	22	8.557	56
17	0.228	0.218	−9	7.421	40	0.188	0.182	−13	4.747	21	0.203	0.199	22	8.568	56
18	0.228	0.218	−9	7.433	40	0.187	0.181	−13	4.780	22	0.204	0.200	22	8.621	57
19	0.228	0.218	−9	7.435	40	0.188	0.182	−13	4.786	22	0.204	0.200	22	8.628	57
20	0.230	0.219	−9	7.442	40	0.188	0.182	−13	4.796	22	0.205	0.201	22	8.650	57
21	0.230	0.220	−9	7.466	40	0.188	0.181	−13	4.800	22	0.206	0.201	23	8.648	57
22	0.230	0.220	−8	7.436	40	0.187	0.180	−12	4.802	22	0.207	0.203	23	8.639	57

Table A27. Iteration-wise out-of-sample validation figures in adaptive variance model selection of BEL corresponding to $M_{\max} \in \{2, 6, 10, 14, 18, 22\}$ based on the 300–886 OLS proxy function given in Table A3 with exponents summing up to at max two. Simultaneously type I FGLS regression results.

m	**v.mae**	**v.mae^a**	**v.res**	**v.mae^0**	**v.res^0**	**ns.mae**	**ns.mae^a**	**ns.res**	**ns.mae^0**	**ns.res^0**	**cr.mae**	**cr.mae^a**	**cr.res**	**cr.mae^0**	**cr.res^0**
0	0.195	0.186	−9	6.468	33	0.275	0.266	−30	4.601	−3	0.175	0.171	5	5.315	32
1	0.199	0.190	−9	6.648	34	0.273	0.263	−31	4.272	−3	0.166	0.162	1	5.005	30
2	0.196	0.187	−9	6.527	33	0.275	0.266	−30	4.564	−3	0.175	0.171	5	5.401	32
3	0.195	0.186	−9	6.487	33	0.255	0.247	−27	4.350	1	0.175	0.171	9	5.916	37
4	0.198	0.189	−9	6.305	32	0.239	0.231	−23	4.262	4	0.182	0.178	13	6.303	40
5	0.198	0.190	−9	6.298	32	0.236	0.228	−22	4.252	4	0.182	0.178	14	6.336	40
6	0.200	0.191	−9	6.399	33	0.240	0.232	−23	4.292	4	0.183	0.179	13	6.389	40
7	0.199	0.190	−9	6.364	32	0.241	0.233	−23	4.304	4	0.183	0.179	13	6.324	40
8	0.199	0.190	−8	6.381	32	0.238	0.230	−22	4.313	4	0.186	0.182	14	6.407	40
9	0.201	0.193	−8	6.432	33	0.236	0.228	−22	4.313	5	0.188	0.184	15	6.521	41
10	0.203	0.194	−8	6.473	33	0.234	0.226	−21	4.310	5	0.189	0.185	16	6.621	42
11	0.203	0.195	−8	6.492	33	0.233	0.225	−21	4.303	5	0.189	0.185	16	6.628	42
12	0.203	0.194	−8	6.476	33	0.232	0.224	−21	4.294	5	0.189	0.186	16	6.641	42
13	0.200	0.191	−7	6.254	32	0.223	0.216	−19	4.252	5	0.192	0.188	17	6.615	42
14	0.200	0.191	−7	6.246	31	0.222	0.214	−19	4.257	6	0.194	0.190	18	6.697	42
15	0.200	0.191	−7	6.216	31	0.220	0.213	−18	4.243	6	0.196	0.192	19	6.773	43
16	0.200	0.191	−7	6.180	31	0.218	0.211	−18	4.239	6	0.197	0.193	19	6.753	43
17	0.200	0.192	−7	6.197	31	0.219	0.211	−18	4.249	6	0.198	0.194	19	6.804	43
18	0.200	0.191	−7	6.194	31	0.217	0.210	−18	4.250	6	0.198	0.194	19	6.801	43
19	0.200	0.191	−7	6.207	31	0.218	0.210	−18	4.238	6	0.197	0.193	19	6.787	43
20	0.200	0.191	−7	6.229	32	0.218	0.211	−18	4.226	6	0.196	0.192	19	6.793	43
21	0.200	0.192	−7	6.240	32	0.218	0.211	−18	4.224	7	0.196	0.192	19	6.814	43
22	0.200	0.192	−7	6.256	32	0.217	0.210	−18	4.223	7	0.196	0.192	19	6.844	44

Table A28. AIC scores and out-of-sample validation figures of type II FGLS proxy functions of BEL under 150–443 with variance models of varying complexity $M_{\max}$ after each tenth iteration.

k	AIC	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
$M_{\max} = 2$ in variance model selection																
0	437,251	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	336,390	1.786	1.708	184	44.082	198	1.402	1.354	209	39.152	209	2.290	2.242	344	52.033	344
20	323,883	0.826	0.790	25	22.007	111	0.424	0.409	−28	10.764	44	0.437	0.428	28	16.424	99
30	319,958	0.465	0.445	3	12.876	55	0.288	0.278	2	9.650	40	0.467	0.457	57	15.234	96
40	318,945	0.401	0.384	−16	11.036	51	0.357	0.345	−37	7.158	16	0.330	0.323	3	10.127	55
50	318,206	0.355	0.339	−24	9.270	35	0.336	0.324	−36	6.611	8	0.339	0.332	−8	8.602	36
60	317,485	0.323	0.309	−25	8.407	36	0.309	0.298	−36	5.548	11	0.279	0.273	−11	7.244	36
70	317,197	0.306	0.293	−28	7.631	28	0.345	0.334	−43	5.405	−1	0.272	0.266	−17	5.899	25
80	316,263	0.272	0.260	−24	6.946	32	0.320	0.310	−42	4.051	0	0.227	0.222	−17	4.898	25
90	316,021	0.260	0.249	−23	7.143	39	0.298	0.288	−37	3.854	10	0.173	0.169	−5	6.461	42
100	315,871	0.256	0.245	−23	7.424	41	0.294	0.284	−35	4.078	14	0.186	0.182	0	7.443	49
110	315,784	0.256	0.245	−22	7.396	41	0.302	0.292	−37	3.962	12	0.189	0.185	−3	7.013	46
120	315,719	0.257	0.245	−23	6.923	38	0.296	0.286	−36	3.870	11	0.181	0.177	−2	6.872	45
130	315,675	0.258	0.247	−25	6.506	35	0.295	0.285	−36	3.760	9	0.188	0.184	−3	6.461	42
140	315,649	0.252	0.241	−23	6.424	34	0.283	0.274	−34	3.749	9	0.184	0.180	−1	6.399	42
150	315,629	0.239	0.229	−21	6.467	34	0.261	0.252	−30	3.796	10	0.177	0.173	3	6.654	44
$M_{\max} = 6$ in variance model selection																
0	437,251	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	332,479	2.014	1.926	259	49.098	213	2.000	1.933	298	44.745	238	2.964	2.901	445	58.341	385
20	320,873	0.881	0.842	51	22.821	115	0.341	0.329	16	13.428	66	0.622	0.609	84	20.790	134
30	316,187	0.429	0.410	19	10.875	32	0.308	0.297	29	8.537	28	0.561	0.549	73	12.633	72
40	315,132	0.366	0.350	6	10.243	45	0.254	0.246	1	7.853	25	0.401	0.393	36	11.221	61
50	314,473	0.303	0.289	3	9.346	46	0.229	0.222	0	7.543	28	0.361	0.353	34	10.776	62
60	313,643	0.307	0.293	−18	7.567	28	0.251	0.242	−21	5.808	11	0.266	0.261	9	7.676	41
70	313,301	0.280	0.268	−17	7.768	30	0.222	0.214	−12	6.229	21	0.268	0.262	23	9.315	56
80	313,060	0.270	0.258	−20	7.092	28	0.230	0.222	−13	6.273	22	0.280	0.274	25	9.554	59
90	312,883	0.262	0.251	−22	6.754	29	0.239	0.231	−17	5.977	20	0.253	0.248	19	9.077	56
100	312,100	0.246	0.235	−19	6.177	29	0.202	0.195	−14	4.814	18	0.221	0.216	21	8.305	54
110	311,656	0.231	0.221	−16	6.446	33	0.189	0.182	−12	4.827	22	0.211	0.206	25	8.964	59
120	311,574	0.236	0.225	−16	6.545	34	0.209	0.202	−16	4.594	19	0.207	0.202	22	8.637	57
130	311,511	0.238	0.227	−17	6.551	35	0.207	0.200	−16	4.797	21	0.204	0.200	23	9.104	60
140	311,461	0.231	0.221	−16	6.026	31	0.189	0.183	−12	4.726	21	0.216	0.212	25	8.853	58
150	311,426	0.224	0.215	−14	5.904	31	0.177	0.171	−9	4.756	22	0.226	0.221	29	9.005	59
$M_{\max} = 10$ in variance model selection																
0	437,251	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	328,519	2.120	2.027	288	50.524	221	2.206	2.132	329	46.563	248	3.194	3.127	480	60.396	399
20	319,481	0.971	0.928	95	24.185	105	0.439	0.424	53	11.839	49	0.821	0.803	117	18.086	112
30	316,529	0.655	0.627	56	16.560	74	0.420	0.406	57	12.301	61	0.780	0.764	113	18.285	117
40	314,460	0.379	0.362	19	10.089	42	0.268	0.259	19	8.120	28	0.473	0.463	54	11.608	63
50	313,842	0.324	0.310	2	8.422	33	0.229	0.221	−4	6.420	12	0.339	0.331	20	8.600	36
60	313,022	0.297	0.284	−13	7.619	31	0.223	0.215	−13	6.123	17	0.277	0.271	14	8.292	43
70	312,692	0.282	0.269	−17	7.494	26	0.221	0.213	−5	6.762	24	0.326	0.319	35	10.467	64
80	312,443	0.271	0.259	−19	7.171	27	0.218	0.211	−7	6.625	25	0.303	0.297	33	10.306	65
90	312,264	0.261	0.249	−21	6.610	27	0.222	0.215	−11	6.300	23	0.278	0.272	28	9.806	62
100	312,187	0.262	0.250	−21	6.568	26	0.216	0.208	−10	6.265	23	0.272	0.266	28	9.707	61
110	312,108	0.256	0.244	−21	6.031	23	0.203	0.196	−5	6.324	25	0.288	0.282	31	9.754	61
120	312,043	0.261	0.250	−23	5.989	20	0.200	0.194	−4	6.287	25	0.293	0.287	33	9.857	62
130	311,078	0.226	0.216	−18	5.466	25	0.160	0.155	−4	5.115	24	0.244	0.239	32	9.192	60
140	310,918	0.220	0.210	−16	5.451	25	0.153	0.148	−4	4.820	23	0.233	0.228	31	8.859	58
150	310,868	0.212	0.203	−14	5.375	25	0.148	0.143	0	5.098	25	0.256	0.250	36	9.296	61

Table A28. *Cont.*

k	AIC	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
$M_{\max} = 14$ in variance model selection																
0	437,251	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	326,308	2.120	2.027	290	50.306	220	2.215	2.141	331	46.129	246	3.197	3.130	480	59.909	396
20	319,199	1.024	0.979	100	26.049	137	0.527	0.509	75	18.639	98	1.044	1.022	155	27.142	178
30	316,093	0.702	0.671	67	17.574	79	0.503	0.486	73	13.745	70	0.901	0.882	133	20.208	131
40	314,155	0.393	0.376	24	10.363	44	0.282	0.273	25	8.426	31	0.505	0.494	62	12.131	68
50	313,562	0.327	0.313	6	8.561	34	0.225	0.217	1	6.535	15	0.352	0.345	27	8.936	41
60	312,811	0.298	0.285	−10	7.608	29	0.203	0.196	4	7.086	29	0.336	0.329	37	10.283	62
70	312,455	0.289	0.276	−15	7.409	26	0.219	0.211	−2	6.863	25	0.343	0.335	38	10.612	65
80	312,235	0.273	0.261	−17	7.222	28	0.215	0.208	−4	6.738	26	0.322	0.316	37	10.662	67
90	312,057	0.264	0.253	−22	6.680	27	0.222	0.214	−10	6.406	24	0.283	0.277	28	9.981	63
100	311,953	0.255	0.244	−21	6.117	24	0.201	0.194	−5	6.381	25	0.290	0.284	31	9.780	61
110	311,898	0.252	0.241	−20	5.929	22	0.200	0.193	−4	6.236	24	0.293	0.287	32	9.583	60
120	311,832	0.263	0.251	−23	5.962	19	0.198	0.192	−3	6.300	25	0.303	0.296	34	9.878	62
130	310,916	0.223	0.213	−17	5.363	23	0.154	0.149	−1	5.233	25	0.263	0.257	36	9.305	61
140	310,757	0.215	0.206	−15	5.339	24	0.147	0.142	0	4.954	24	0.251	0.246	35	8.972	59
150	310,714	0.214	0.205	−14	5.368	25	0.146	0.141	−1	4.857	23	0.244	0.239	34	8.906	59
$M_{\max} = 18$ in variance model selection																
0	437,251	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	326,125	2.127	2.034	292	50.425	220	2.226	2.151	332	46.222	246	3.209	3.142	482	60.019	396
20	318,762	1.036	0.991	111	25.668	113	0.538	0.520	75	13.429	64	0.983	0.962	144	20.708	133
30	315,995	0.710	0.679	69	17.741	80	0.523	0.505	76	13.963	72	0.925	0.906	137	20.465	133
40	314,060	0.401	0.383	27	10.529	45	0.292	0.282	28	8.560	33	0.521	0.510	66	12.341	70
50	313,483	0.329	0.315	9	8.687	35	0.225	0.217	4	6.620	16	0.362	0.354	31	9.120	43
60	312,938	0.316	0.302	−5	7.840	30	0.209	0.202	5	6.855	26	0.347	0.340	41	10.297	62
70	312,363	0.270	0.258	−10	6.960	21	0.215	0.207	11	7.089	28	0.389	0.381	48	10.795	65
80	312,166	0.259	0.248	−12	6.558	22	0.204	0.198	9	7.008	29	0.369	0.361	47	10.718	67
90	311,963	0.234	0.223	−15	6.141	24	0.196	0.189	1	6.432	26	0.313	0.306	37	9.844	61
100	311,883	0.241	0.231	−18	6.031	24	0.194	0.187	−1	6.449	26	0.299	0.293	34	9.777	61
110	311,830	0.239	0.229	−18	5.836	22	0.193	0.187	0	6.298	25	0.303	0.296	35	9.610	60
120	311,766	0.244	0.234	−19	5.713	18	0.191	0.184	3	6.340	26	0.321	0.314	39	9.866	62
130	311,045	0.225	0.215	−15	5.396	23	0.148	0.143	0	5.061	24	0.259	0.254	35	8.950	59
140	310,694	0.213	0.204	−13	5.314	24	0.139	0.134	1	4.855	24	0.245	0.240	34	8.672	57
150	310,644	0.211	0.202	−14	5.131	23	0.139	0.135	1	4.816	23	0.250	0.245	35	8.618	57
$M_{\max} = 22$ in variance model selection																
0	437,251	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	325,988	2.127	2.034	292	50.414	220	2.226	2.151	332	46.259	246	3.210	3.143	482	60.061	397
20	318,926	1.034	0.988	105	26.160	137	0.569	0.550	83	19.043	101	1.098	1.075	163	27.621	181
30	315,805	0.712	0.681	71	17.763	79	0.537	0.519	78	14.063	72	0.943	0.923	140	20.603	134
40	313,973	0.409	0.391	29	10.730	46	0.301	0.291	31	8.709	34	0.539	0.527	70	12.589	72
50	313,411	0.349	0.334	7	8.950	34	0.223	0.216	3	6.618	16	0.357	0.349	30	9.081	42
60	312,873	0.308	0.295	−2	8.205	37	0.203	0.196	8	7.490	33	0.350	0.343	43	10.853	67
70	312,286	0.271	0.260	−9	6.950	21	0.217	0.210	12	7.124	28	0.398	0.389	50	10.856	66
80	312,091	0.261	0.249	−11	6.557	22	0.207	0.200	10	7.051	29	0.377	0.369	48	10.793	68
90	311,893	0.235	0.225	−15	6.043	23	0.196	0.189	1	6.367	25	0.314	0.307	36	9.683	60
100	311,815	0.238	0.228	−17	5.970	23	0.194	0.187	1	6.462	26	0.311	0.304	37	9.829	61
110	311,761	0.237	0.227	−17	5.780	21	0.194	0.188	2	6.364	25	0.313	0.307	37	9.694	60
120	311,697	0.243	0.232	−19	5.818	18	0.191	0.185	2	6.325	25	0.320	0.313	39	9.885	62
130	311,655	0.232	0.222	−17	5.688	18	0.195	0.188	8	6.714	29	0.353	0.346	46	10.509	67
140	310,748	0.215	0.206	−14	5.206	23	0.148	0.143	5	5.578	27	0.293	0.287	42	9.788	64
150	310,590	0.208	0.199	−13	5.209	23	0.139	0.134	5	5.193	26	0.275	0.270	40	9.256	61

Table A29. AIC scores and out-of-sample validation figures of type II FGLS proxy functions of BEL under 300–886 with variance models of varying complexity $M_{\max}$ after each tenth and the final iteration.

k	AIC	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
$M_{\max} = 2$ in variance model selection																
0	437,251	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	336,390	1.786	1.708	184	44.082	198	1.402	1.354	209	39.152	209	2.290	2.242	344	52.033	344
20	323,883	0.826	0.790	25	22.007	111	0.424	0.409	−28	10.764	44	0.437	0.428	28	16.424	99
30	319,958	0.465	0.445	3	12.876	55	0.288	0.278	2	9.650	40	0.467	0.457	57	15.234	96
40	318,945	0.401	0.384	−16	11.036	51	0.357	0.345	−37	7.158	16	0.330	0.323	3	10.127	55
50	318,206	0.355	0.339	−24	9.270	35	0.336	0.324	−36	6.611	8	0.339	0.332	−8	8.602	36
60	317,485	0.323	0.309	−25	8.407	36	0.309	0.298	−36	5.548	11	0.279	0.273	−11	7.244	36
70	317,197	0.306	0.293	−28	7.631	28	0.345	0.334	−43	5.405	−1	0.272	0.266	−17	5.899	25
80	316,263	0.272	0.260	−24	6.946	32	0.320	0.310	−42	4.051	0	0.227	0.222	−17	4.898	25
90	316,021	0.260	0.249	−23	7.143	39	0.298	0.288	−37	3.854	10	0.173	0.169	−5	6.461	42
100	315,871	0.256	0.245	−23	7.424	41	0.294	0.284	−35	4.078	14	0.186	0.182	0	7.443	49
110	315,784	0.256	0.245	−22	7.396	41	0.302	0.292	−37	3.962	12	0.189	0.185	−3	7.013	46
120	315,719	0.257	0.245	−23	6.923	38	0.296	0.286	−36	3.870	11	0.181	0.177	−2	6.872	45
130	315,675	0.258	0.247	−25	6.506	35	0.295	0.285	−36	3.760	9	0.188	0.184	−3	6.461	42
140	315,641	0.250	0.239	−23	6.441	34	0.284	0.275	−34	3.741	9	0.182	0.178	−2	6.338	41
150	315,622	0.238	0.228	−20	6.433	34	0.258	0.250	−29	3.821	11	0.177	0.174	4	6.740	44
160	315,599	0.233	0.223	−20	6.578	35	0.256	0.247	−28	3.920	12	0.183	0.179	6	6.988	46
170	315,573	0.232	0.222	−19	6.616	35	0.254	0.246	−28	3.880	12	0.181	0.178	5	6.927	45
180	315,535	0.225	0.215	−19	6.502	35	0.252	0.243	−28	3.773	11	0.172	0.169	5	6.797	44
190	315,523	0.229	0.219	−19	6.809	37	0.244	0.236	−26	4.020	15	0.164	0.161	9	7.607	50
200	315,507	0.215	0.206	−18	6.738	36	0.243	0.235	−26	3.969	14	0.164	0.161	9	7.387	49
210	315,500	0.214	0.205	−18	6.704	35	0.234	0.226	−24	3.989	14	0.162	0.159	10	7.323	48
220	315,492	0.217	0.207	−18	6.769	35	0.239	0.231	−26	3.930	14	0.159	0.155	9	7.277	48
224	315,491	0.209	0.199	−17	6.584	34	0.226	0.219	−22	3.999	14	0.165	0.161	12	7.290	48
$M_{\max} = 6$ in variance model selection																
0	437,251	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	332,479	2.014	1.926	259	49.098	213	2.000	1.933	298	44.745	238	2.964	2.901	445	58.341	385
20	320,873	0.881	0.842	51	22.821	115	0.341	0.329	16	13.428	66	0.622	0.609	84	20.790	134
30	316,187	0.429	0.410	19	10.875	32	0.308	0.297	29	8.537	28	0.561	0.549	73	12.633	72
40	315,132	0.366	0.350	6	10.243	45	0.254	0.246	1	7.853	25	0.401	0.393	36	11.221	61
50	314,473	0.303	0.289	3	9.346	46	0.229	0.222	0	7.543	28	0.361	0.353	34	10.776	62
60	313,643	0.307	0.293	−18	7.567	28	0.251	0.242	−21	5.808	11	0.266	0.261	9	7.676	41
70	313,301	0.280	0.268	−17	7.768	30	0.222	0.214	−12	6.229	21	0.268	0.262	23	9.315	56
80	313,060	0.270	0.258	−20	7.092	28	0.230	0.222	−13	6.273	22	0.280	0.274	25	9.554	59
90	312,883	0.262	0.251	−22	6.754	29	0.239	0.231	−17	5.977	20	0.253	0.248	19	9.077	56
100	312,100	0.246	0.235	−19	6.177	29	0.202	0.195	−14	4.814	18	0.221	0.216	21	8.305	54
110	311,656	0.231	0.221	−16	6.446	33	0.189	0.182	−12	4.827	22	0.211	0.206	25	8.964	59
120	311,574	0.236	0.225	−16	6.545	34	0.209	0.202	−16	4.594	19	0.207	0.202	22	8.637	57
130	311,507	0.234	0.223	−16	6.706	36	0.206	0.199	−16	4.801	21	0.204	0.200	23	9.094	60
140	311,456	0.226	0.216	−16	6.102	32	0.189	0.182	−12	4.717	21	0.215	0.211	25	8.827	58
150	311,419	0.224	0.214	−15	5.899	31	0.178	0.172	−10	4.712	22	0.213	0.209	27	8.971	59
160	311,355	0.217	0.207	−15	5.536	29	0.160	0.154	−4	5.013	25	0.246	0.241	33	9.420	62
170	311,308	0.198	0.189	−13	5.090	23	0.141	0.137	−4	4.144	19	0.221	0.216	27	7.491	49
180	311,266	0.202	0.193	−14	5.112	24	0.132	0.127	−3	4.433	22	0.218	0.213	27	7.868	52
190	311,248	0.208	0.198	−16	5.287	23	0.143	0.138	−5	4.163	19	0.213	0.208	25	7.630	50
200	311,228	0.202	0.193	−14	5.269	24	0.137	0.133	−4	4.148	20	0.213	0.209	27	7.639	50
210	311,196	0.192	0.184	−14	5.032	20	0.125	0.121	4	4.655	23	0.253	0.248	32	7.919	52
220	311,164	0.195	0.187	−15	5.079	21	0.122	0.118	1	4.620	23	0.237	0.232	31	8.070	53
230	311,148	0.194	0.185	−15	5.146	22	0.122	0.118	1	4.571	23	0.236	0.231	29	7.949	52
237	311,144	0.196	0.188	−15	5.342	23	0.125	0.121	0	4.765	24	0.235	0.230	30	8.243	54
$M_{\max} = 10$ in variance model selection																
0	437,251	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	331,056	2.073	1.982	273	50.085	216	2.113	2.041	315	45.714	244	3.090	3.025	464	59.451	393
20	320,199	0.924	0.884	76	23.133	101	0.375	0.362	25	10.921	35	0.655	0.641	82	15.999	92
30	316,044	0.543	0.519	31	14.068	56	0.372	0.359	45	11.729	56	0.742	0.727	107	18.450	118
40	314,821	0.385	0.368	11	10.626	47	0.256	0.248	6	8.118	28	0.424	0.415	43	11.685	65
50	314,201	0.327	0.313	2	9.206	41	0.240	0.232	−8	6.713	17	0.336	0.329	21	9.103	45
60	313,386	0.269	0.257	−5	7.831	34	0.220	0.213	6	7.506	31	0.365	0.357	46	11.223	71
70	312,986	0.290	0.278	−17	7.316	26	0.210	0.203	−4	6.646	25	0.310	0.304	33	9.955	61
80	312,722	0.280	0.268	−18	7.425	31	0.223	0.215	−8	6.792	27	0.300	0.293	33	10.652	68
90	312,545	0.270	0.259	−22	7.110	32	0.233	0.225	−13	6.634	26	0.273	0.267	27	10.450	67
100	312,469	0.265	0.253	−21	6.800	29	0.224	0.217	−11	6.420	25	0.274	0.268	29	10.128	64
110	312,397	0.254	0.243	−19	6.136	25	0.202	0.195	−4	6.360	25	0.290	0.284	33	9.940	63
120	312,346	0.247	0.236	−19	5.940	22	0.193	0.187	1	6.468	27	0.307	0.301	38	10.078	64
130	312,299	0.240	0.230	−17	5.784	21	0.192	0.185	4	6.563	28	0.329	0.322	43	10.369	66
140	312,274	0.247	0.236	−18	5.811	22	0.193	0.186	5	6.870	31	0.338	0.331	45	10.944	71
150	312,243	0.249	0.238	−19	5.950	24	0.193	0.186	3	6.872	31	0.324	0.317	43	10.984	71
160	312,222	0.255	0.244	−19	6.162	25	0.198	0.191	1	6.859	30	0.324	0.318	42	11.092	72
170	311,204	0.228	0.218	−14	5.957	31	0.161	0.156	−1	5.874	30	0.276	0.270	40	10.703	71
180	311,040	0.223	0.213	−13	6.021	31	0.154	0.149	−1	5.594	29	0.265	0.259	39	10.356	68
190	310,996	0.222	0.213	−13	6.152	32	0.154	0.149	−2	5.584	28	0.258	0.253	38	10.311	68
200	310,968	0.206	0.197	−10	6.163	32	0.144	0.139	3	5.924	31	0.285	0.279	42	10.568	70
210	310,953	0.211	0.202	−10	5.930	30	0.143	0.138	3	5.615	29	0.276	0.270	41	10.153	67
220	310,927	0.208	0.199	−11	6.353	33	0.147	0.142	−1	5.602	29	0.252	0.247	37	10.225	67
230	310,919	0.211	0.202	−11	6.454	34	0.149	0.144	−1	5.702	29	0.259	0.253	38	10.376	69
240	310,908	0.210	0.201	−11	6.559	35	0.152	0.147	−3	5.570	28	0.251	0.245	36	10.218	67
244	310,905	0.208	0.199	−11	6.577	35	0.153	0.147	−2	5.617	29	0.252	0.247	37	10.259	68

Table A29. *Cont.*

k	AIC	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
$M_{\max} = 14$ in variance model selection																
0	437,251	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	327,049	2.133	2.039	292	50.561	222	2.233	2.157	333	46.686	249	3.222	3.154	484	60.524	400
20	318,965	1.020	0.976	108	25.288	111	0.507	0.490	69	12.759	57	0.931	0.912	136	19.634	124
30	316,262	0.694	0.663	65	17.386	78	0.484	0.468	69	13.341	68	0.872	0.853	128	19.643	127
40	314,272	0.392	0.375	23	10.373	44	0.277	0.268	23	8.322	30	0.493	0.483	59	11.941	66
50	313,691	0.349	0.333	1	8.772	32	0.228	0.220	−5	6.440	12	0.335	0.328	19	8.633	36
60	312,860	0.289	0.276	−10	7.475	30	0.204	0.197	−2	6.583	24	0.302	0.295	28	9.218	53
70	312,542	0.286	0.273	−16	7.501	26	0.219	0.211	−3	6.802	24	0.334	0.327	37	10.548	64
80	312,337	0.281	0.269	−18	7.254	27	0.215	0.207	−4	6.834	27	0.323	0.316	37	10.655	67
90	312,126	0.261	0.250	−21	6.672	27	0.221	0.213	−10	6.384	23	0.286	0.280	29	9.942	62
100	312,046	0.268	0.256	−22	6.695	27	0.222	0.215	−12	6.317	24	0.270	0.265	26	9.779	61
110	311,961	0.257	0.245	−22	5.979	23	0.200	0.193	−5	6.316	25	0.284	0.278	31	9.695	61
120	311,903	0.252	0.241	−21	5.892	19	0.193	0.186	1	6.411	26	0.311	0.304	37	9.977	63
130	311,860	0.244	0.233	−19	5.886	20	0.190	0.184	3	6.562	28	0.322	0.315	41	10.344	66
140	311,824	0.243	0.232	−20	5.880	19	0.190	0.183	5	6.758	30	0.335	0.328	44	10.696	69
150	311,800	0.247	0.236	−21	6.011	20	0.185	0.179	2	6.452	28	0.309	0.303	40	10.365	66
160	310,806	0.218	0.208	−16	5.451	25	0.140	0.135	0	5.234	27	0.255	0.249	37	9.596	63
170	310,710	0.210	0.201	−15	5.473	25	0.137	0.132	0	5.077	26	0.249	0.244	36	9.359	62
180	310,682	0.206	0.197	−14	5.303	24	0.136	0.131	2	5.064	26	0.266	0.260	39	9.492	63
190	310,661	0.200	0.191	−13	5.285	23	0.144	0.139	5	5.163	26	0.298	0.292	44	9.843	65
200	310,639	0.201	0.192	−13	5.413	22	0.143	0.138	4	5.088	25	0.293	0.287	44	9.726	64
210	310,606	0.203	0.194	−13	5.599	23	0.145	0.141	6	5.459	27	0.314	0.307	47	10.294	68
220	310,525	0.183	0.174	−13	4.672	12	0.148	0.143	−3	3.744	7	0.221	0.217	30	6.238	40
230	310,513	0.179	0.171	−14	4.668	13	0.153	0.148	−6	3.729	7	0.206	0.202	27	6.113	40
240	310,475	0.172	0.164	−14	4.347	10	0.130	0.126	−1	3.523	9	0.219	0.214	30	6.154	39
250	310,462	0.171	0.163	−14	4.307	10	0.134	0.130	−2	3.480	8	0.211	0.206	28	5.958	38
258	310,443	0.172	0.165	−14	4.371	10	0.134	0.129	−2	3.504	8	0.214	0.210	28	6.063	39
$M_{\max} = 18$ in variance model selection																
0	437,251	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	325,846	2.112	2.020	290	50.142	221	2.201	2.127	328	46.153	246	3.183	3.116	478	59.925	396
20	318,985	1.027	0.982	104	25.991	136	0.566	0.547	82	18.748	99	1.089	1.066	162	27.261	179
30	315,896	0.705	0.674	69	17.595	79	0.526	0.508	76	13.871	71	0.928	0.908	137	20.356	132
40	314,044	0.404	0.386	28	10.602	45	0.296	0.286	30	8.630	34	0.531	0.519	68	12.462	71
50	313,483	0.330	0.316	9	8.715	35	0.225	0.217	5	6.643	17	0.365	0.358	32	9.177	44
60	312,939	0.316	0.302	−5	7.833	31	0.210	0.203	5	6.895	26	0.352	0.345	42	10.382	63
70	312,359	0.270	0.258	−10	6.927	21	0.216	0.208	11	7.084	27	0.393	0.385	49	10.781	65
80	312,165	0.260	0.248	−12	6.555	22	0.206	0.199	10	7.018	29	0.373	0.365	48	10.721	67
90	311,964	0.233	0.223	−15	6.130	24	0.196	0.189	1	6.433	26	0.313	0.307	37	9.838	61
100	311,882	0.237	0.227	−17	5.756	20	0.190	0.183	2	6.218	24	0.305	0.298	36	9.431	58
110	311,827	0.239	0.229	−18	5.733	21	0.190	0.184	1	6.305	25	0.303	0.296	36	9.588	60
120	311,769	0.245	0.234	−20	5.762	18	0.189	0.183	3	6.425	27	0.319	0.313	39	9.924	62
130	311,716	0.224	0.214	−16	5.502	15	0.190	0.183	10	6.403	27	0.350	0.342	46	9.993	63
140	311,005	0.216	0.206	−13	5.222	21	0.142	0.137	6	5.361	26	0.291	0.285	42	9.416	62
150	310,660	0.203	0.194	−12	5.094	21	0.133	0.129	7	5.158	26	0.284	0.278	42	9.129	60
160	310,611	0.201	0.192	−12	5.033	21	0.137	0.133	8	5.360	27	0.303	0.297	45	9.568	63
170	310,586	0.196	0.187	−11	4.994	21	0.136	0.132	10	5.548	28	0.316	0.310	47	9.821	65
180	310,550	0.193	0.184	−12	4.987	21	0.135	0.130	1	4.264	20	0.241	0.236	35	8.200	54
190	310,535	0.196	0.187	−14	5.087	21	0.139	0.135	−3	4.049	18	0.217	0.212	31	7.884	52
200	310,511	0.182	0.174	−11	4.965	21	0.131	0.127	0	3.992	18	0.231	0.226	34	7.810	52
210	310,467	0.185	0.177	−12	5.011	20	0.131	0.127	0	3.967	17	0.231	0.226	34	7.741	51
220	310,463	0.181	0.173	−12	5.059	20	0.130	0.125	2	4.181	19	0.246	0.241	36	8.110	54
230	310,454	0.181	0.173	−11	5.409	23	0.138	0.133	1	4.405	20	0.246	0.241	36	8.436	56
240	310,440	0.182	0.174	−11	5.398	23	0.138	0.133	1	4.457	21	0.250	0.245	37	8.559	57
250	310,431	0.181	0.173	−11	5.509	23	0.138	0.133	1	4.525	21	0.251	0.246	37	8.638	57
252	310,425	0.185	0.176	−11	5.515	23	0.138	0.133	1	4.548	22	0.253	0.248	37	8.700	57
$M_{\max} = 22$ in variance model selection																
0	437,251	4.557	4.357	−238	100.000	38	3.231	3.121	0	100.000	261	4.027	3.942	106	100.000	367
10	325,796	2.115	2.023	290	50.203	222	2.206	2.131	329	46.238	246	3.189	3.121	479	60.021	396
20	318,940	1.026	0.981	112	25.965	135	0.666	0.644	98	20.243	107	1.199	1.174	179	28.606	188
30	315,849	0.708	0.677	70	17.681	79	0.532	0.514	77	14.005	72	0.936	0.917	139	20.526	133
40	314,001	0.407	0.389	28	10.712	46	0.299	0.289	31	8.710	34	0.536	0.524	69	12.589	73
50	313,413	0.348	0.332	10	9.025	36	0.223	0.216	5	6.616	17	0.364	0.356	32	9.225	44
60	312,897	0.316	0.302	−4	7.866	31	0.211	0.203	6	6.983	27	0.358	0.351	44	10.549	65
70	312,317	0.271	0.259	−9	6.969	22	0.217	0.210	12	7.185	28	0.399	0.391	50	10.961	67
80	312,120	0.260	0.249	−11	6.565	23	0.207	0.200	10	7.119	30	0.379	0.371	49	10.896	69
90	311,920	0.235	0.224	−15	6.091	24	0.196	0.189	1	6.427	26	0.313	0.306	37	9.791	61
100	311,842	0.238	0.228	−16	6.034	23	0.194	0.187	1	6.531	27	0.311	0.304	37	9.949	63
110	311,784	0.241	0.230	−18	5.900	24	0.192	0.185	1	6.554	28	0.304	0.297	36	10.004	63
120	311,737	0.241	0.230	−18	5.809	21	0.189	0.182	2	6.395	27	0.310	0.303	38	9.924	63
130	311,690	0.227	0.217	−16	5.653	18	0.187	0.181	8	6.468	28	0.339	0.332	45	10.100	64
140	310,925	0.213	0.203	−13	5.206	22	0.140	0.136	7	5.430	27	0.293	0.286	43	9.548	63
150	310,604	0.202	0.193	−11	5.131	22	0.133	0.129	7	5.286	27	0.289	0.283	42	9.321	61
160	310,559	0.200	0.192	−11	5.063	22	0.139	0.134	9	5.507	28	0.310	0.304	46	9.791	65
170	310,532	0.189	0.181	−10	4.999	22	0.134	0.129	8	5.194	26	0.297	0.291	44	9.438	62
180	310,503	0.193	0.185	−12	5.222	24	0.132	0.128	4	5.137	26	0.270	0.264	40	9.462	62
190	310,481	0.194	0.186	−13	5.113	22	0.140	0.136	−2	4.124	19	0.220	0.215	32	8.019	53
200	310,454	0.189	0.181	−13	5.164	21	0.135	0.130	−1	4.033	18	0.224	0.220	33	7.836	52
210	310,412	0.185	0.177	−12	5.038	20	0.132	0.128	0	4.019	18	0.231	0.226	34	7.805	52
220	310,406	0.185	0.176	−12	5.067	20	0.132	0.128	1	4.062	18	0.239	0.234	35	7.981	53
224	310,404	0.184	0.176	−12	5.112	20	0.132	0.128	1	4.076	18	0.239	0.234	35	7.934	52

Table A30. AIC scores and out-of-sample validation figures of all derived FGLS proxy functions of BEL under 150–443 and 300–886 after the final iteration. Highlighted in green and red respectively the best and worst AIC scores and validation figures.

k	M_{max}	AIC	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
Type I algorithm under 150-443																	
150	2	315,980	0.239	0.229	−16	8.147	46	0.255	0.246	−30	4.032	17	0.153	0.149	2	7.489	49
150	6	311,949	0.231	0.221	−13	7.577	41	0.203	0.196	−18	4.762	22	0.186	0.183	17	8.637	57
150	10	311,363	0.227	0.217	−10	7.460	40	0.194	0.188	−15	4.708	21	0.195	0.191	20	8.537	56
150	14	311,161	0.231	0.221	−9	7.527	41	0.193	0.186	−14	4.701	21	0.200	0.195	21	8.497	56
150	18	311,048	0.228	0.218	−9	7.433	40	0.187	0.181	−13	4.780	22	0.204	0.200	22	8.621	57
150	22	310,974	0.230	0.220	−8	7.436	40	0.187	0.180	−12	4.802	22	0.207	0.203	23	8.639	57
Type I algorithm under 300-886																	
224	2	315,615	0.196	0.187	−9	6.527	33	0.275	0.266	−30	4.564	−3	0.175	0.171	5	5.401	32
224	6	311,554	0.200	0.191	−9	6.399	33	0.240	0.232	−23	4.292	4	0.183	0.179	13	6.389	40
224	10	311,287	0.203	0.194	−8	6.473	33	0.234	0.226	−21	4.310	5	0.189	0.185	16	6.621	42
224	14	310,980	0.200	0.191	−7	6.246	31	0.222	0.214	−19	4.257	6	0.194	0.190	18	6.697	42
224	18	310,881	0.200	0.191	−7	6.194	31	0.217	0.210	−18	4.250	6	0.198	0.194	19	6.801	43
224	22	310,832	0.200	0.192	−7	6.256	32	0.217	0.210	−18	4.223	7	0.196	0.192	19	6.844	44
Type II algorithm under 150-443																	
150	2	315,629	0.239	0.229	−21	6.467	34	0.261	0.252	−30	3.796	10	0.177	0.173	3	6.654	44
150	6	311,426	0.224	0.215	−14	5.904	31	0.177	0.171	−9	4.756	22	0.226	0.221	29	9.005	59
150	10	310,868	0.212	0.203	−14	5.375	25	0.148	0.143	0	5.098	25	0.256	0.250	36	9.296	61
150	14	310,714	0.214	0.205	−14	5.368	25	0.146	0.141	−1	4.857	23	0.244	0.239	34	8.906	59
150	18	310,644	0.211	0.202	−14	5.131	23	0.139	0.135	1	4.816	23	0.250	0.245	35	8.618	57
150	22	310,590	0.208	0.199	−13	5.209	23	0.139	0.134	5	5.193	26	0.275	0.270	40	9.256	61
Type II algorithm under 300-886																	
224	2	315,491	0.209	0.199	−17	6.584	34	0.226	0.219	−22	3.999	14	0.165	0.161	12	7.290	48
237	6	311,144	0.196	0.188	−15	5.342	23	0.125	0.121	0	4.765	24	0.235	0.230	30	8.243	54
244	10	310,905	0.208	0.199	−11	6.577	35	0.153	0.147	−2	5.617	29	0.252	0.247	37	10.259	68
258	14	310,443	0.172	0.165	−14	4.371	10	0.134	0.129	−2	3.504	8	0.214	0.210	28	6.063	39
252	18	310,425	0.185	0.176	−11	5.515	23	0.138	0.133	1	4.548	22	0.253	0.248	37	8.700	57
224	22	310,404	0.184	0.176	−12	5.112	20	0.132	0.128	1	4.076	18	0.239	0.234	35	7.934	52

Table A31. Settings and out-of-sample validation figures of best performing multivariate adaptive regression splines (MARS) models derived in a two-step approach sorted by first and second step validation sets. Highlighted in green and red respectively the best and worst validation figures.

k	$K_{\max}$	$t_{\min}$	o	p	glm	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
Sobol set^2																				
148	206	0	6	s	inv.g, id	0.265	0.253	−24	10.317	55	0.575	0.555	−40	16.234	−56	0.822	0.805	80	17.657	64
49	50	0	3	n	inv.g, log	0.370	0.354	0	9.168	19	0.705	0.681	−12	29.477	−102	0.525	0.514	25	16.891	−65
60	66	0	4	s	inv.g, id	0.324	0.310	−11	8.517	16	1.712	1.654	151	44.504	132	0.917	0.897	102	19.877	83
45	50	0	4	b	inv.g, id	0.347	0.332	−2	8.686	11	0.447	0.431	−36	22.702	−125	0.511	0.500	35	15.785	−54
Sobol set and nested simulations set																				
45	50	0	4	b	inv.g, id	0.347	0.332	−2	8.686	11	0.447	0.431	−36	22.702	−125	0.511	0.500	35	15.785	−54
17	19	0	4	b	inv.g, id	0.834	0.797	25	24.673	124	0.480	0.464	−4	41.356	−243	0.763	0.747	108	21.398	−132
70	81	0	4	b	inv.g, id	0.335	0.320	−22	10.872	52	0.554	0.535	−35	14.073	−38	0.875	0.857	102	18.250	99
33	34	0	3	n	inv.g, id	0.426	0.407	−10	10.871	21	1.565	1.512	108	52.384	1	0.662	0.648	32	20.997	−75
Sobol set and capital region set																				
45	50	0	3	b	pois, log	0.379	0.362	0	9.556	28	0.480	0.464	−43	24.878	−139	0.510	0.500	28	16.938	−69
31	34	0	3	b	pois, log	0.476	0.455	−13	12.752	46	0.593	0.573	−54	31.148	−175	0.661	0.647	18	23.088	−103
45	50	0	4	b	inv.g, id	0.347	0.332	−2	8.686	11	0.447	0.431	−36	22.702	−125	0.511	0.500	35	15.785	−54
59	66	0	3	b	pois, log	0.428	0.439	40	16.674	98	0.760	0.734	−12	22.511	−41	0.809	0.792	68	18.403	39
Nested simulations set and Sobol set																				
134	144	1.6_{-5}	5	n	gaus, log	0.273	0.261	−22	10.255	54	1.025	0.990	−1	28.192	−23	1.515	1.484	179	32.616	157
45	50	0	4	s	inv.g, id	0.347	0.332	−2	8.686	11	0.447	0.431	−36	22.702	−125	0.511	0.500	35	15.785	−54
60	66	0	4	s	inv.g, id	0.324	0.310	−11	8.517	16	1.712	1.654	151	44.504	132	0.917	0.897	102	19.877	83
45	50	0	4	b	inv.g, id	0.347	0.332	−2	8.686	11	0.447	0.431	−36	22.702	−125	0.511	0.500	35	15.785	−54
Nested simulations set^2																				
45	50	0	4	b	inv.g, id	0.347	0.332	−2	8.686	11	0.447	0.431	−36	22.702	−125	0.511	0.500	35	15.785	−54
146	159	9.4_{-6}	5	n	gaus, log	0.279	0.267	−24	10.008	53	1.025	0.990	0	26.779	−11	1.498	1.467	174	31.702	163
76	97	3.8_{-5}	4	b	inv.g, log	0.344	0.329	−17	10.676	52	0.538	0.520	−37	11.874	−24	0.804	0.787	88	16.584	100
107	113	0	4	n	gaus, log	0.321	0.307	−20	11.976	63	0.997	0.963	8	25.694	0	1.529	1.496	191	32.148	182
Nested simulations set and capital region set																				
45	50	0	4	s	pois, id	0.353	0.338	−3	8.891	18	0.449	0.434	−36	23.634	−131	0.504	0.493	36	16.079	−58
31	34	0	4	s	pois, id	0.437	0.418	−11	11.254	32	0.548	0.530	−45	28.444	−157	0.648	0.634	29	21.374	−84
72	82	3.1_{-5}	4	b	inv.g, inv	0.365	0.349	−16	11.181	53	0.579	0.560	−49	14.528	−51	0.700	0.685	65	14.619	64
45	50	0	4	b	inv.g, id	0.347	0.332	−2	8.686	11	0.447	0.431	−36	22.702	−125	0.511	0.500	35	15.785	−54
Capital region set and Sobol set																				
125	144	0	5	f	inv.g, inv	0.283	0.271	−20	10.336	54	0.630	0.608	−63	17.245	−76	0.675	0.660	45	14.737	32
45	50	0	4	s	gaus, log	0.382	0.365	−1	9.916	32	0.469	0.453	−41	25.487	−144	0.495	0.485	32	16.868	−71
114	144	1.9_{-5}	5	s	inv.g,$1/\mu^2$	0.313	0.299	−12	9.414	40	0.708	0.684	−77	20.115	−97	0.626	0.612	36	14.095	17
45	50	0	4	b	gaus, log	0.382	0.365	−1	9.916	32	0.469	0.453	−41	25.487	−144	0.495	0.485	32	16.868	−71
Capital region set and nested simulations set																				
45	50	0	4	f	gaus, log	0.386	0.369	−1	10.095	34	0.468	0.452	−41	25.709	−145	0.496	0.486	32	17.077	−73
64	66	0	4	n	inv.g,$1/\mu^2$	0.420	0.401	−3	11.506	39	0.840	0.811	3	25.969	−38	1.298	1.271	146	29.110	105
148	175	0	6	s	inv.g,$1/\mu^2$	0.311	0.297	−16	10.447	52	0.576	0.556	−55	14.565	−57	0.611	0.598	30	12.844	27
77	81	0	4	n	inv.g,$1/\mu^2$	0.387	0.370	−11	11.519	52	1.029	0.994	−28	25.831	−32	1.279	1.252	148	26.700	145
Capital region set^2																				
45	50	0	4	s	gaus, log	0.382	0.365	−1	9.916	32	0.469	0.453	−41	25.487	−144	0.495	0.485	32	16.868	−71
33	34	0	3	n	inv.g,$1/\mu^2$	0.564	0.539	−14	15.693	64	0.827	0.800	−54	38.645	−185	0.745	0.729	−2	26.338	−134
148	175	0	6	s	inv.g,$1/\mu^2$	0.311	0.297	−16	10.447	52	0.576	0.556	−55	14.565	−57	0.611	0.598	30	12.844	27
148	175	4.7_{-6}	5	f	inv.g, inv	0.296	0.283	−20	10.416	53	0.549	0.530	−54	18.260	−87	0.664	0.650	32	16.307	−1

Table A32. Best MARS model of BEL derived in a two-step approach with the final coefficients.

k	$h_k\,(X)$	$\widehat{\beta}_{\mathbf{MARS},k}$
0	1	15, 397.13
1	$h\,(X_8 - 0.104892)$	7901.89
2	$h\,(0.104892 - X_8)$	−8165.64
3	$h\,(0.205577 - X_1) \cdot h\,(0.104892 - X_8)$	688.83
4	$h\,(X_6 - 1.17224)$	265.08
5	$h\,(1.17224 - X_6)$	−280.94
6	$h\,(X_{15} - 53.8706)$	−2.11
7	$h\,(53.8706 - X_{15})$	1.16
8	$h\,(X_7 - -0.147599)$	−60.90
9	$h\,(-0.147599 - X_7)$	−334.77
10	$h\,(X_8 - -0.0456197)$	3183.07
11	$h\,(0.205577 - X_1) \cdot h\,(0.104892 - X_8) \cdot h\,(X_{15} - 64.6262)$	−9.48
12	$h\,(0.205577 - X_1) \cdot h\,(0.104892 - X_8) \cdot h\,(64.6262 - X_{15})$	29.85
13	$h\,(X_1 - 0.945371)$	−64.88
14	$h\,(0.945371 - X_1)$	124.45
15	$h\,(X_6 - 1.56058) \cdot h\,(0.104892 - X_8)$	−815.20
16	$h\,(1.56058 - X_6) \cdot h\,(0.104892 - X_8)$	1085.80
17	$h\,(1.44218 - X_2)$	−60.23
18	$h\,(X_1 - -1.61447) \cdot h\,(1.56058 - X_6) \cdot h\,(0.104892 - X_8)$	−233.14
19	$h\,(-1.61447 - X_1) \cdot h\,(1.56058 - X_6) \cdot h\,(0.104892 - X_8)$	415.92
20	$h\,(X_8 - 0.0159508) \cdot h\,(53.8706 - X_{15})$	8.94
21	$h\,(0.0159508 - X_8) \cdot h\,(53.8706 - X_{15})$	47.99
22	$h\,(X_9 - 0.247192)$	47.72
23	$h\,(0.247192 - X_9)$	−82.58
24	$h\,(0.993896 - X_{12})$	−63.61
25	$h\,(X_1 - 0.0195594) \cdot h\,(0.0159508 - X_8) \cdot h\,(53.8706 - X_{15})$	−12.58
26	$h\,(0.0195594 - X_1) \cdot h\,(0.0159508 - X_8) \cdot h\,(53.8706 - X_{15})$	−42.25
27	$h\,(X_7 - -0.147599) \cdot h\,(X_8 - -0.191689)$	2124.93
28	$h\,(X_7 - -0.147599) \cdot h\,(-0.191689 - X_8)$	1510.41
29	$h\,(X_3 - 0.323352) \cdot h\,(0.104892 - X_8)$	948.86
30	$h\,(0.323352 - X_3) \cdot h\,(0.104892 - X_8)$	−577.61
31	$h\,(X_1 - -1.26627) \cdot h\,(X_7 - -0.147599)$	101.15
32	$h\,(-1.26627 - X_1) \cdot h\,(X_7 - -0.147599)$	−10.00
33	$h\,(X_{14} - 0.684998)$	109.76
34	$h\,(0.684998 - X_{14})$	−37.89
35	$h\,(1.17224 - X_6) \cdot h\,(X_8 - -0.12538)$	216.62
36	$h\,(1.17224 - X_6) \cdot h\,(-0.12538 - X_8)$	2076.18
37	$h\,(0.945371 - X_1) \cdot h\,(X_8 - 0.0019988)$	−156.79
38	$h\,(0.945371 - X_1) \cdot h\,(0.0019988 - X_8)$	1262.56
39	$h\,(X_1 - -1.58818) \cdot h\,(X_6 - 1.56058) \cdot h\,(0.104892 - X_8)$	137.60
40	$h\,(1.56058 - X_6) \cdot h\,(0.104892 - X_8) \cdot h\,(X_{15} - 76.9327)$	−4.87
41	$h\,(1.56058 - X_6) \cdot h\,(0.104892 - X_8) \cdot h\,(76.9327 - X_{15})$	2.11
42	$h\,(0.205577 - X_1) \cdot h\,(X_2 - 1.43028) \cdot h\,(0.104892 - X_8)$	24, 003.07
43	$h\,(0.205577 - X_1) \cdot h\,(1.43028 - X_2) \cdot h\,(0.104892 - X_8)$	−161.88
44	$h\,(X_1 - 0.945371) \cdot h\,(X_8 - -0.0165546)$	−224.18
45	$h\,(X_1 - 0.945371) \cdot h\,(-0.0165546 - X_8)$	−987.47

Table A33. Basis function sets of LC and LL proxy functions of BEL corresponding to $K_{\max} \in \{16, 27\}$ derived by adaptive OLS selection.

k	r_k^1	r_k^2	r_k^3	r_k^4	r_k^5	r_k^6	r_k^7	r_k^8	r_k^9	r_k^{10}	r_k^{11}	r_k^{12}	r_k^{13}	r_k^{14}	r_k^{15}
$K_{\max} = 16$ in adaptive basis function selection															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
5	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
6	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
8	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
11	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
12	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
13	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
16	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
$K_{\max} = 27$ in adaptive basis function selection															
17	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0
18	1	0	0	0	0	0	0	1	0	0	0	0	0	0	1
19	0	0	0	0	0	0	0	2	0	0	0	0	0	0	1
20	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
21	1	0	0	0	0	0	0	2	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
23	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1
24	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
25	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0
27	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0

Table A34. Basis function sets of LC and LL proxy functions of BEL corresponding to $K_{\max} \in \{15, 22\}$ derived by risk factor wise or combined risk factor wise and adaptive OLS selection.

k	r_k^1	r_k^2	r_k^3	r_k^4	r_k^5	r_k^6	r_k^7	r_k^8	r_k^9	r_k^{10}	r_k^{11}	r_k^{12}	r_k^{13}	r_k^{14}	r_k^{15}
$K_{\max} = 15$ in risk factor wise basis function selection															
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
$K_{\max} = 22$ in combined risk factor wise and adaptive selection															
16	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0
17	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1
19	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1
20	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0
22	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0

Table A35. Settings and out-of-sample validation figures of LC and LL proxy functions of BEL using basis function sets from Tables A33 and A34. Highlighted in green and red respectively the best and worst validation figures.

k	bw	o	v.mae	v.mae^a	v.res	v.mae^0	v.res^0	ns.mae	ns.mae^a	ns.res	ns.mae^0	ns.res^0	cr.mae	cr.mae^a	cr.res	cr.mae^0	cr.res^0
LC regression with gaussian kernel and LOO-CV																	
16	0.1	2	0.55	0.52	−44	13	50	0.70	0.68	−86	12	−7	0.55	0.54	−35	12	45
16	0.2	2	0.40	0.38	−26	11	47	0.52	0.50	−51	11	7	0.44	0.43	5	13	63
16	0.3	2	0.37	0.35	−25	11	45	0.45	0.44	−37	11	19	0.44	0.43	5	12	60
27	0.2	2	0.39	0.38	−26	11	43	0.51	0.49	−51	11	3	0.43	0.43	4	12	58
16	0.1	4	2.80	2.68	−155	84	−407	8.05	7.78	−558	247	−825	5.04	4.94	−96	128	−363
LL regression with gaussian kernel and LOO-CV																	
16	0.1	2	0.38	0.36	−11	12	57	0.57	0.55	−68	10	−15	0.41	0.40	−22	9	31
16	0.2	2	0.34	0.33	−6	11	59	0.45	0.43	−49	8	2	0.37	0.36	5	10	55
27	0.1	2	210.30	201.06	−30,682	5209	−30,589	131.04	126.61	−18,981	3670	−18,902	4.09	4.00	−82	92	−3
27	0.2	2	2726.47	2606.74	400,254	67,487	400,306	3502.24	3383.85	422,443	98,081	422,481	1.85	1.81	−25	41	13
LC regression with gaussian kernel and AIC																	
16	0.1	2	0.57	0.55	−43	14	55	0.65	0.62	−72	12	12	0.50	0.49	−12	14	72
16	0.2	2	1.63	1.55	38	41	73	1.94	1.88	266	57	286	2.57	2.51	384	61	404
27	0.1	2	0.56	0.54	−42	14	56	0.64	0.62	−72	12	12	0.50	0.49	−12	14	72
LC regression with Epanechnikov kernel and LOO-CV																	
15	0.1	2	0.53	0.50	−36	13	41	1.05	1.02	−38	22	24	0.51	0.50	−29	11	33
15	0.2	2	0.41	0.39	−31	10	33	1.14	1.10	3	26	53	1.18	1.16	97	27	146
15	0.3	2	0.40	0.38	−30	9	23	0.96	0.93	16	23	54	0.46	0.45	−6	11	33
15	0.4	2	0.35	0.33	−22	9	18	1.11	1.08	12	28	39	0.47	0.46	−2	11	25
15	0.5	2	0.34	0.33	−18	9	37	1.24	1.20	6	30	46	0.51	0.50	−22	11	18
15	0.6	2	0.33	0.32	−17	10	50	1.16	1.12	21	27	74	0.46	0.45	−2	11	50
15	0.7	2	0.33	0.32	−16	10	41	1.17	1.13	18	28	61	0.44	0.43	−14	9	28
15	0.8	2	0.33	0.31	−16	10	45	1.21	1.17	29	29	76	1.16	1.13	101	26	148
15	0.9	2	0.32	0.30	−20	12	61	1.14	1.10	40	27	107	1.14	1.11	111	29	178
15	1.0	2	0.32	0.31	−22	10	49	1.19	1.15	52	29	109	1.13	1.11	106	27	163
16	0.1	2	0.53	0.50	−40	13	43	1.20	1.16	2	28	71	0.51	0.50	−20	12	49
16	0.2	2	0.41	0.39	−26	11	50	1.16	1.12	27	28	88	0.44	0.43	2	12	64
16	0.3	2	0.36	0.34	−27	9	29	1.07	1.03	41	27	83	0.44	0.43	1	11	43
16	0.4	2	0.33	0.32	−19	8	22	1.16	1.12	27	30	53	0.45	0.44	4	10	30
16	0.5	2	0.32	0.31	−16	9	36	1.34	1.30	30	33	67	1.22	1.19	101	27	138
16	0.1	4	0.45	0.43	−26	13	34	0.74	0.71	−68	16	−23	0.59	0.57	5	15	51
16	0.2	4	3.29	3.15	−104	160	891	7.50	7.24	−14	329	966	8.06	7.89	176	295	1157
16	0.1	6	3.31	3.16	−32	84	68	5.74	5.55	−96	158	−10	6.62	6.48	−53	148	32
16	0.2	6	3.32	3.18	−71	85	−217	9.37	9.06	73	268	−87	13.18	12.90	246	304	86
16	0.1	8	3.94	3.77	146	105	−119	10.71	10.35	−191	308	−470	8.84	8.65	−312	205	−591
16	0.2	8	8.53	8.16	397	286	−639	7.79	7.52	70	347	−980	12.37	12.11	1365	390	315
22	0.1	2	0.50	0.48	−37	12	44	1.07	1.03	−41	22	25	0.52	0.50	−30	11	37
22	0.2	2	0.42	0.40	−28	10	39	1.07	1.03	−3	25	50	1.20	1.17	106	29	159
22	0.3	2	0.39	0.37	−29	9	23	0.89	0.86	6	22	43	0.45	0.44	−3	11	34
22	0.4	2	0.35	0.33	−21	8	16	1.05	1.02	3	27	26	0.49	0.48	−4	11	19
22	0.5	2	0.33	0.31	−14	9	32	1.17	1.13	−2	28	29	0.47	0.46	−15	10	16
22	0.6	2	0.33	0.32	−17	10	46	1.09	1.06	11	25	60	0.45	0.44	−1	11	48
22	0.7	2	0.32	0.31	−15	9	39	1.23	1.18	26	29	66	1.17	1.14	99	26	139
22	0.8	2	0.32	0.30	−15	10	46	1.19	1.15	32	28	78	1.12	1.10	106	26	152
22	0.9	2	0.31	0.30	−19	11	58	1.15	1.11	39	27	102	1.12	1.10	111	28	174
22	1.0	2	0.31	0.30	−21	10	48	1.13	1.09	41	27	96	1.12	1.10	107	27	162
27	0.2	2	0.40	0.38	−26	11	45	1.15	1.12	26	28	83	0.44	0.43	1	12	58
27	0.3	2	0.38	0.36	−28	9	24	0.90	0.87	7	22	45	0.46	0.45	−2	11	36
27	0.4	2	0.35	0.33	−21	9	17	1.05	1.02	2	27	26	0.48	0.47	−4	11	11
LL regression with Epanechnikov kernel and LOO-CV																	
15	0.1	2	0.45	0.43	−49	10	40	1.22	1.18	−100	22	−26	0.78	0.77	−104	11	−30
15	0.2	2	0.36	0.34	−34	8	13	1.59	1.53	−145	40	−112	0.60	0.58	−54	11	−21
15	0.3	2	0.32	0.31	−36	7	17	1.91	1.85	134	48	173	0.60	0.58	−36	11	3
15	0.4	2	0.34	0.33	−40	8	33	1.83	1.76	−164	42	−106	0.43	0.42	−49	6	9
15	0.5	2	0.33	0.31	−40	8	34	2.20	2.12	−219	53	−160	0.41	0.41	−45	6	15
15	0.6	2	0.30	0.29	−33	7	29	0.94	0.91	8	19	56	0.33	0.32	−28	5	21
15	0.7	2	0.31	0.30	−40	7	23	0.94	0.91	−13	19	36	0.36	0.35	−40	5	8
15	0.8	2	0.29	0.28	−38	5	8	0.86	0.83	4	19	36	0.32	0.32	−29	5	3
22	0.1	2	731.51	699.39	2738	85,172	479,612	1564.87	1511.98	−111,628	127,410	365,231	492.49	482.11	−19,404	76,575	457,455
22	0.2	2	0.34	0.33	−34	8	0	0.83	0.80	−15	21	4	0.42	0.41	−25	8	−5
22	0.3	2	98.03	93.73	14,396	148	−250	101.69	98.25	15,174	147	513	100.00	97.89	15,028	100	367
22	0.4	2	98.05	93.75	14,399	147	−248	113.99	110.14	13,158	495	−1503	100.00	97.89	15,028	100	367
22	0.5	2	100.00	95.61	14,685	100	38	118.95	114.93	14,984	651	323	100.00	97.89	15,028	100	367
22	0.6	2	99.72	95.34	14,644	106	−3	100.59	97.19	15,004	120	343	100.00	97.89	15,028	100	367
22	0.7	2	100.00	95.61	14,685	100	38	100.00	96.62	14,922	100	261	100.00	97.89	15,028	100	367
22	0.8	2	0.29	0.28	−39	5	9	152.43	147.27	22,622	4264	22,655	0.31	0.30	−35	5	−2
LC regression with uniform kernel and LOO-CV																	
16	0.1	2	0.75	0.71	−56	18	46	1.53	1.48	−52	32	36	0.73	0.72	−59	15	29
16	0.5	2	1.22	1.17	−78	29	16	2.60	2.51	301	82	381	10.45	10.23	1419	242	1498
27	0.1	2	0.64	0.61	−38	16	31	1.30	1.26	13	32	68	0.59	0.58	−2	15	53
27	0.5	2	0.35	0.34	−16	12	53	1.34	1.30	25	33	79	1.40	1.37	117	32	171
16	0.1	4	0.71	0.68	−33	17	47	1.27	1.23	−1	31	65	0.67	0.65	−23	15	43
16	0.5	4	1.85	1.76	−139	39	50	2.29	2.22	18	51	193	7.09	6.94	769	157	943
27	0.1	4	0.66	0.63	−38	15	32	1.32	1.27	7	32	63	0.58	0.57	−15	14	40
27	0.5	4	0.39	0.37	−13	13	67	1.26	1.21	16	31	82	0.52	0.51	−10	13	56
16	0.1	6	1.83	1.75	−165	38	100	1.95	1.88	−178	29	72	1.55	1.51	−190	24	60
16	0.5	6	1.83	1.75	−6	56	271	1.08	1.04	80	65	344	1.66	1.63	225	74	488

References

Akaike, Hirotogu. 1973. Information theory and an extension of the maximum likelihood principle. In *International Symposium on Information Theory*, 2nd ed. Budapest: Akadémiai Kiadó.

Bauer, Daniel, and Hongjun Ha. 2015. A least-squares Monte Carlo approach to the calculation of capital requirements. Paper presented at the World Risk and Insurance Economics Congress, Munich, Germany, August 2–6. Available online: https://danielbaueracademic.files.wordpress.com/2018/02/habauer_lsm.png (accessed on 10 June 2018).

Bauer, Daniel, Andreas Reuss, and Daniela Singer. 2012. On the calculation of the solvency capital requirement based on nested simulations. *The Journal of the International Actuarial Association* 42: 453–99.

Bettels, Christian, Johannes Fabrega, and Christian Weiß. 2014. Anwendung von Least Squares Monte Carlo (LSMC) im Solvency-II-Kontext-Teil 1. *Der Aktuar* 2: 85–91.

Born, Rudolf. 2018. Künstliche Neuronale Netze im Risikomanagement. Master's thesis, Universität zu Köln, Köln, Germany.

Breusch, Trevor S., and Adrian R. Pagan. 1979. A simple test for heteroscedasticity and random coefficient variation. *Econometrica* 47: 1287–94. [CrossRef]

Burnham, Kenneth P., and David R. Anderson. 2002. *Model Selection and Multimodel Inference: A Practical Information-Theoretic Approach*, 2nd ed. New York: Springer-Verlag.

Castellani, Gilberto, Ugo Fiore, Zelda Marino, Luca Passalacqua, Francesca Perla, Salvatore Scognamiglio, and Paolo Zanetti. 2018. An Investigation of Machine Learning Approaches in the Solvency Ii Valuation Framework. Available online: http://dx.doi.org/10.2139/ssrn.3303296 (accessed on 14 August 2019).

Craven, Peter, and Grace Wahba. 1979. Smoothing noisy data with spline functions. *Numerische Mathematik* 31: 377–403. [CrossRef]

Dahlquist, Germund, and Åke Björck. 1974. *Numerical Methods*. Englewood Cliffs: Prentice-Hall.

Dobson, Annette J. 2002. *An Introduction to Statistical Modelling*, 2nd ed. Boca Raton, London, New York, and Washington: Chapman & Hall/CRC.

Drucker, Harris, Chris J.C. Burges, Linda Kaufman, Alex Smola, and Vladimir Vapnik. 1997. Support vector regression machines. In *Advances in Neural Information Processing Systems 9*. Denver: MIT Press, pp. 155–61.

Duchon, Jean. 1977. Splines minimizing rotation-invariant semi-norms in solobev spaces. In *Constructive Theory of Functions of Several Variables*. Edited by W. Schempp, and K. Zeller. Berlin: Springer, pp. 85–100.

Dutang, Christophe. 2017. *Some Explanations about the IWLS Algorithm to Fit Generalized Linear Models*. hal-01577698. France: HAL.

Eilers, Paul H.C., and Brian D. Marx. 1996. Flexible smoothing with b-splines and penalties. *Statistical Science* 11: 89–121. [CrossRef]

European Parliament, and European Council. 2009. *Directive 2009/138/EC on the Taking-Up and Pursuit of the Business of Insurance and Reinsurance (Solvency II)*. Directive. Brussels: European Council. pp. 112–127.

Friedman, Jerome H. 1991. Multivariate adaptive regression splines (with discussion). *The Annals of Statistics* 19: 1–141. [CrossRef]

Friedman, Jerome H. 1993. Fast MARS. In *Technical Report 110*. Stanford: Stanford University Department of Statistics.

Friedman, Jerome H., and Werner Stuetzle. 1981. Projection pursuit regression. *Journal of the American Statistical Association* 76: 817–23. [CrossRef]

Gay, David M. 1990. Usage summary for selected optimization routines. In *Computing Science Technical Report 153*. Murray Hill: AT&T Bell Laboratories.

Gordy, Michael B., and Sandeep Juneja. 2010. Nested simulations in portfolio risk measurement. *Management Science* 56: 1833–48. [CrossRef]

Green, P. J. 1984. Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society, Series B* 46: 149–92. [CrossRef]

Hartmann, Stefanie. 2015. Verallgemeinerte lineare Modelle im Kontext des Least Squares Monte Carlo Verfahrens. Master's thesis, Katholische Universität Eichstätt-Ingolstadt, Eichstätt, Germany.

Harvey, Andrew C. 1976. Estimating regression models with multiplicative heteroscedasticity. *Econometrica* 44: 461–65. [CrossRef]

Hastie, Trevor, and Daryl Pregibon. 1992. *Chapter 6 'Generalized Linear Models' in Statistical Models in S*. Boca Raton, London, New York, and Washington: Wadsworth & Brooks/Cole.

Hastie, Trevor, and Robert Tibshirani. 1986. Generalized additive models. *Statistical Science* 1: 297–318. [CrossRef]

Hastie, Trevor, and Robert Tibshirani. 1990. *Generalized Additive Models*. London: Chapman & Hall.

Hastie, Trevor, Robert Tibshirani, and Jerome H. Friedman. 2017. *The Elements of Statistical Learning*, 2nd ed. New York: Springer Series in Statistics.

Hayashi, Fumio. 2000. *Econometrics*. Princeton: Princeton University Press.

Hejazi, Seyed A., and Kenneth R. Jackson. 2017. Efficient valuation of scr via a neural network approach. *Journal of Computational and Applied Mathematics* 313: 427–39. [CrossRef]

Hocking, R. R. 1976. The analysis and selection of variables in linear regression. *Biometrics* 32: 1–49. [CrossRef]

Hurvich, Clifford M., Jeffrey S. Simonoff, and Chih-Ling Tsai. 1998. Smoothing parameter selection in nonparametric regression using an improved Akaike information criterion. *Journal of the Royal Statistical Society, Series B* 60: 271–93. [CrossRef]

Kandasamy, Kirthevasan, and Yaoliang Yu. 2016. Additive approximations in high dimensional nonparametric regression via the SALSA. Paper presented at the 33rd International Conference on Machine Learning, New York, NY, USA, June 19–24. pp. 69–78.

Kazimov, Nurlan. 2018. Least Squares Monte Carlo modeling based on radial basis functions. Master's thesis, Universität Ulm, Ulm, Germany.

Kopczyk, Dawid. 2018. Proxy Modeling in Life Insurance Companies With the Use of Machine Learning Algorithms. Working Paper. Available online: http://dx.doi.org/10.2139/ssrn.3396481 (accessed on 29 July 2019).

Krah, Anne-Sophie. 2015. Suitable information criteria and regression methods for the polynomial fitting process in the lsmc model. Master's thesis, Julius-Maximilians-Universität Würzburg, Würzburg, Germany.

Krah, Anne-Sophie, Zoran Nikolić, and Ralf Korn. 2018. A least-squares Monte Carlo framework in proxy modeling of life insurance companies. *Risks* 6: 62. [CrossRef]

Li, Qi, and Jeff Racine. 2004. Cross-validated local linear nonparametric regression. *Statistica Sinica* 14: 485–512.

Magnus, Jan R. 1978. Maximum likelihood estimation of the GLS model with unknown parameters in the disturbance covariance matrix. *Journal of Econometrics* 7: 281–312. [CrossRef]

Marra, Giampiero, and Simon N. Wood. 2012. Coverage properties of confidence intervals for generalized additive model components. *Scandinavian Journal of Statistics* 39: 53–74. [CrossRef]

Marx, Brian D., and Paul H.C. Eilers. 1998. Direct generalized additive modeling with penalized likelihood. *Computational Statistics & Data Analysis* 28: 193–209.

McCullagh, Peter, and John A. Nelder. 1989. *Generalized Linear Models*, 2nd ed. London and New York: Chapman & Hall.

McLean, Douglas. 2014. *Orthogonality in Proxy Generator*. Presentation, Insurance-ERS. Legendre Polynomial/QR Decomposition Equivalence in Multiple Polynomial Regression. New York City: Moody's Analytics.

Milborrow, Stephen. 2018. *Earth: Multivariate Adaptive Regression Splines*. Derived from mda:mars by Trevor Hastie and Rob Tibshirani. Uses Alan Miller's Fortran Utilities with Thomas Lumley's Leaps Wrapper. R Package Version 4.6.3. Available online: https://mran.microsoft.com/snapshot/2018-06-07/web/packages/earth/index.html (accessed on 29 June 2018).

Mourik, Teus. 2003. Market risk of insurance companies. In Discussion Paper IAA Insurer Solvency Assessment Working Party. Amsterdam, The Netherlands. Available online: http://www.actuaires.org/AFIR/colloquia/Maastricht/Mourik.png (accessed on 12 August 2019).

Nadaraya, Elizbar A. 1964. On estimating regression. *Theory of Probability and Its Applications* 9: 141–42. [CrossRef]

Nelder, John A., and Robert W. M. Wedderburn. 1972. Generalized linear models. *Journal of the Royal Statistical Society, Series A* 135: 370–84. [CrossRef]

Nikolić, Zoran, Christian Jonen, and Chengjia Zhu. 2017. Robust regression technique in lsmc proxy modeling. *Der Aktuar* 1: 8–16.

Nychka, Douglas. 1988. Bayesian confidence intervals for smoothing splines. *Journal of the American Statistical Association* 83: 1134–43. [CrossRef]

Pindyck, Robert S., and Daniel L. Rubinfeld. 1998. *Econometric Models and Economic Forecasts*. Ann Arbor: University of Michigan. Irwin: McGraw-Hill.

R Core Team. 2018. *stats: R Statistical Functions*. R package version 3.2.0. Vienna: R Foundation for Statistical Computing.

Racine, Jeffrey S., and Tristen Hayfield. 2018. np: Nonparametric Kernel Smoothing Methods for Mixed Data Types. R package version 0.60-8. Available online: https://github.com/JeffreyRacine/R-Package-np (accessed on 29 June 2018).

Runge, Carl. 1901. Über empirische Funktionen und die Interpolation zwischen äquidistanten Ordinaten. *Zeitschrift für Mathematik und Physik* 46: 224–43.

Schelthoff, Tom. 2019. Machine Learning Methods as Alternatives to the Least Squares Monte Carlo Model for Calculating the Solvency Capital Requirement of Life and Health Insurance Companies. Master's thesis, Universität zu Köln, Cologne, Germany.

Schoenenwald, Johannes J. 2019. Modelli Proxy per la Determinazione dei Requisiti di Capitale Secondo Solvency II. Master's thesis, Universitá degli Studi di Trieste, Trieste, Italy.

Sell, Robin. 2019. Nicht-Parametrische Regression im Risikomanagement. Bachelor's thesis, Universität zu Köln, Cologne, Germany.

Suykens, Johan A.K., and Joos Vandewalle. 1999. Least squares support vector machine classifiers. *Neural Processing Letters* 9: 293–300. [CrossRef]

Teuguia, Oberlain N., Jiaen Ren, and Frédéric Planchet. 2014. *Internal Model in Life Insurance: Application of Least Squares Monte Carlo in Risk Assessment*. Technical Report. Lyon: Laboratoire de Sciences Actuarielle et Financière.

Tibshirani, Robert. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B* 58: 267–88. [CrossRef]

Watson, Geoffrey S. 1964. On estimating regression. *Sankhya: The Indian Journal of Statistics, Series A* 26: 359–72.

Weiß, Christian, and Zoran Nikolić. 2019. An aspect of optimal regression design for LSMC. *Monte Carlo Methods and Applications* 25: 283–90. [CrossRef]

Wood, Simon N. 2000. Modelling and smoothing parameter estimation with multiple quadratic penalties. *Journal of the Royal Statistical Society, Series B* 62: 413–28. [CrossRef]

Wood, Simon N. 2003. Thin plate regression splines. *Journal of the Royal Statistical Society, Series B* 65: 95–114. [CrossRef]

Wood, Simon N. 2006. Generalized additive models. In *Lecture Notes, School of Mathematics*. Bristol: University of Bristol.

Wood, Simon N. 2017. *Generalized Additive Models: An Introduction with R*, 2nd ed. Boca Raton: CRC Press.

Wood, Simon N. 2018. mgcv: Mixed GAM Computation Vehicle with Automatic Smoothness Estimation. R package version 1.8–24. Available online: https://rdrr.io/cran/mgcv/ (accessed on 29 June 2018).

Wood, Simon N., Yannig Goude, and Simon Shaw. 2015. Generalized additive models for large data sets. *Journal of the Royal Statistical Society, Series C* 64: 139–55. [CrossRef]

Wood, Simon N., Zheyuan Li, Gavin Shaddick, and Nicole H. Augustin. 2017. Generalized additive models for gigadata: Modeling the u.k. black smoke network daily data. *Journal of the American Statistical Association* 112: 1199–210. [CrossRef]

Zuur, Alain F., Elena N. Ieno, Neil J. Walker, Anatoly A. Saveliev, and Graham M. Smith. 2009. *Mixed Effects Models and Extensions in Ecology with R*. Chapter GLM and GAM for Count Data. New York: Springer, pp. 209–43.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).

risks

MDPI

Article

Prediction of Claims in Export Credit Finance: A Comparison of Four Machine Learning Techniques

Mathias Bärtl [1] and Simone Krummaker [2,*]

[1] Hochschule für Technik, Wirtschaft und Medien Offenburg, 77652 Offenburg, Germany; mathias.baertl@hs-offenburg.de
[2] Faculty of Actuarial Science and Insurance, Cass Business School, University of London, London EC1Y8TZ, UK
* Correspondence: simone.krummaker@city.ac.uk

Received: 23 December 2019; Accepted: 20 February 2020; Published: 1 March 2020

Abstract: This study evaluates four machine learning (ML) techniques (Decision Trees (DT), Random Forests (RF), Neural Networks (NN) and Probabilistic Neural Networks (PNN)) on their ability to accurately predict export credit insurance claims. Additionally, we compare the performance of the ML techniques against a simple benchmark (BM) heuristic. The analysis is based on the utilisation of a dataset provided by the Berne Union, which is the most comprehensive collection of export credit insurance data and has been used in only two scientific studies so far. All ML techniques performed relatively well in predicting whether or not claims would be incurred, and, with limitations, in predicting the order of magnitude of the claims. No satisfactory results were achieved predicting actual claim ratios. RF performed significantly better than DT, NN and PNN against all prediction tasks, and most reliably carried their validation performance forward to test performance.

Keywords: machine learning; claims prediction; export credit insurance

1. Introduction

Predicting claims is a critical challenge for insurers and has significant implications for their managerial, financial and underwriting decisions. Changes in (expected) claims do not only affect the capital of an insurer, but also the capacity to underwrite further business. Insurance companies can increase premium rates and adjust their underwriting policy to balance the effect of unexpected claims (van der Veer 2019), but this will consequently impact their business opportunities negatively. We are, therefore, investigating machine learning (ML) techniques for claims prediction using an international dataset on export credit insurance claims.

Export credit insurance is a tool for exporters in mitigating risks that arise from exporting to other countries. It covers companies against the risk of non-payment of their buyer due to commercial and political risks. The commercial risks include full or partial default on payments, as well as protracted default or insolvency of private buyers, while political risks, include non-payment of public buyers or due to political events, e.g., government-imposed moratoria on payments, inability to transfer currency, or force majeure (Berne Union 2019d). Export credit insurance is widely used by exporters to protect their cash flows and receivables. Consequently, it also protects the profits against unwanted volatility due to unsystematic risk. It can also cover lenders involved in the export transaction (usually by granting loans or letters of credit for the buyer) against the default of their credit due to the aforementioned reasons. Often lenders are only willing to grant financing if export credit insurance is provided. Therefore, export credit insurance is regularly a key requirement for the realisation of an export transaction (Krummaker 2020).

The Export Credit Insurance business is differentiated with respect to the tenure of the credit granted. Short-term (ST) credits are typically up to one year, while medium- and long-term (MLT) credit insurance offers insurance for credit terms up to 15 years. MLT is mainly offered by public Export Credit Agencies (ECAs), even though in recent years the private market has increased its MLT capacities (Berne Union 2019d). In our study, we focus on MLT insurance provided by ECAs, which is characterised by higher risk than in the ST business. Furthermore, for some, ECAs claims are a rare occurrence. However, as the claims frequency is very low, the severity of potential claims can be high and might also exhibit long-tail properties. In our article we address the challenge of insurers in making reliable and consistent predictions of future claims based on historical claims experiences by conducting a comparative analysis of ML approaches on a long-term dataset of export credit claims.

The aim of this study is to assess the performance of ML techniques in identifying the occurrence of claims in export credit insurance and their potential performance loss when tested under near-realistic forecasting conditions. We were able to access a unique dataset provided by the Berne Union to compare four ML techniques by exposing them to three increasingly challenging prediction tasks. Furthermore, we evaluate their performance against a simple benchmark (BM) technique, as ML approaches are complex and resource-intensive to set up but might not achieve significantly better results for claims prediction and reserving (England and Verrall 2002).

First, this article contributes to the gap in the literature on export credit insurance and claims. Second, the paper also contribute to the advancement of the literature on claims prediction by providing an evaluation of ML approaches, including a comparison against a simple BM. This, thirdly, also has practical implications for actual claims prediction and reserving for export credit insurers and ECAs.

In the following section, we provide more background to the study before introducing the dataset and a description of ML. After this, we describe the ML techniques used for this study, before discussing the results. The conclusion also includes an outlook for further research.

2. Background

Export credit insurance is offered by private sector insurance companies, public government backed ECAs and some multilateral organisations. Most developed countries, but also many emerging countries and more developing countries, have their own ECA or access to multilateral credit insurers. ECAs are official or quasi-official branches of their governments which offer export credit insurance, guarantees and financing. ECAs are highly regulated in many countries in terms of their product offerings and conditions as they are instruments of governments' trade and foreign aid. To minimise opportunities for hidden subsidies and state aids, ECAs are regulated by international agreements on several levels. The World Trade Organization (WTO) has an explicit framework for trade policies, and the OECD arrangement imposes further detailed rules on its members. The aim of these regulations is to create a level playing field in the global export environment and coherence between national export credit policies (OECD 2018). International competition of exporters is supposed to be based on price and quality, and not on the most favourable terms of exporters' ECAs (Drysdale 2015). Consequently, ECAs of OECD countries are restricted to offer credit insurance only for risks which are deemed non-marketable, i.e., for which the private insurance market is unwilling to provide cover. ECAs mainly cover transactions with credit payment periods of longer than two years and/or to high-risk countries, as private insurers usually do not cover credit risk with repayment terms of longer than two years and can retreat from covering countries with increasing commercial or political risk. These medium- and long-term business (MLT) are typically capital goods, such as industry or infrastructure projects.[1] A further aspect of OECD ECA regulation is the application of minimum premium rates (MPR) for credit risk.[2] Thus, ECAs have less discretion in setting premiums than private insurers, which limits opportunities for managing underwriting and rates, claims ratios and reserves.

[1] Krummaker (2020) provides an overview of export credit markets, governance and key forms of export credit insurance.

[2] The MPR is based on several factors, including country risk classification, the time at risk, the buyer risk category and the percentage of risk retention (OECD 2018).

ECAs act as insurers of last resort and are usually reinsured or backed-up by their respective governments. While private insurers are required to maintain certain levels of long-term and short-term solvency, ECAs often just need to break even and not hold technical provisions for the liabilities and potential claims they take on with underwriting export credit insurance (Moser et al. 2008; European Commission 2012).

ECAs play an important role in facilitating international trade as they provide critical and significant cover to international trade transactions. In 2018, ca. 13% of global trade was covered by MLT export credit insurance provided by ECAs (Berne Union members, Berne Union 2019a). Although, ECAs are underwriting mid-and long-term business in non-marketable, riskier countries, claims still might be an exception. Some ECAs might experience claims only irregularly, but if claims occur, they might be significant. Therefore, it is questionable how well previous claims experiences might be suited to predict future claims.

Prior research in the areas of export credit insurance and finance has only really intensified since the early 2000s. Various papers have established the importance of export credit insurance or ECAs for the support of economic growth, or the relationship between imports and insured trade credits (e.g., Abraham and Dewit 2000; Egger and Url 2006; Moser et al. 2008; van der Veer 2015; Felbermayr and Yalcin 2013). Another strand of literature focuses on the relationship between trading companies and the impact of trade credit, financial market conditions and international trade, as well as the implications of the financial crisis (e.g., Auboin 2009; Korinek et al. 2010; Morel 2011; Auboin and Engemann 2014).

A key challenge for insurers is that, while claims are arising irregularly as a stochastic process of two components, the uncertain number and amount of claims, premiums are not stochastic and they are paid upfront. Although, claims reserving is a critical process in insurance companies, little research has been done on claims in the area of export credit insurance. van der Veer (2019) has carried out the only research examining the impact of export credit insurance claims on price and quality of private export credit insurance. With our study, we address this gap in the literature and aim to provide insights into potential advancements of claims prediction methods.

The export credit insurance industry is currently facing a period of higher uncertainty, driven by the global economic and geo-political environment. Claims in 2018 have risen to historically high levels, with total indemnifications of USD 6.4 bn, 17% higher than 2009 during the financial crisis and 75% higher than the annual average for the past decade (Berne Union 2019b).

This volatile environment makes it challenging for insurers and ECAs to derive reliable predictions of expected claims based on historical data. While, private insurers face increasing financial and regulatory requirements, ECAs have to justify that their use of taxpayers' money is effective and efficient, and creates the desired economic and social impact. For both, private and public insurers, this means that it is increasingly important to deliver reliable estimates of claims, claims reserves and associated expenses. As ECAs are an instrument of their governments' economic and international policies, the portfolio and structure of their business and consequently of their claims reflect national industry and (geographical) export structures, thus, are specific to each country. Moreover, some ECAs do not experience claims regularly; in the MLT business particularly, no claim is the norm and (larger) claims are an exception. Predicting claims and estimating claims reserves as accurately as possible thus is key to ECAs management and underwriting decisions, and will help to allocate capital that is provided by the taxpayer more efficiently.

Insurers have been using a range of deterministic and stochastic methods, such as the Chain Ladder or Bornhuetter-Ferguson method, to predict claims and the related claims reserves (Baudry and Robert 2019). However, developments on regulatory level as well as increasing uncertainty in export credit risks increase the need for the application of more sophisticated methods (England and Verrall 2002; Verall et al. 2012). Prior work by Wüthrich (2018a, 2018b), as well as Thesmar et al. (2019) show that ML approaches have benefits for claims prediction purposes.[3] The algorithms are able to discover patterns in multidimensional datasets or can find new predictors and relationships in the data that have not been used in the traditional methods (Thesmar et al. 2019). Wüthrich (2018a) further argues that ML techniques in claims reserving are flexible and able to work structured, as well as unstructured data.

3. The Berne Union Data

The Berne Union (International Union of Credit and Investment Insurers) is the international trade association of the global export credit and political risk insurance industry. The 85 members are Export Credit Agencies, private insurers of credit and political risk as well as multilateral institutions from 73 countries (Berne Union 2019a). In 2018, Berne Union members covered 13% of all cross-border merchandise trade, with USD 2.5 trillion covered by credit, and political risk insurers about USD 6bn claims paid (Berne Union 2019b). From the new MLT business written in 2018, 83% was accounted for by public ECAs (Berne Union 2019c).

The Berne Union collects comprehensive data on their members' ST and MLT business twice a year. Their database is unique in that it covers transactional information of 33 of the most relevant ECAs, making it the most extensive collection of structured data on export credit insurance and finance, and the best overall proxy for trade credit in general (Auboin and Engemann 2014). Its main purpose is to serve as a mechanism for Berne Union members to share their business information amongst themselves; to date, the Berne Union data have been used in only two scientific studies, which analysed the impact of trade credit and trade finance availability on trade (Auboin and Engemann 2014; Korinek et al. 2010).

The Berne Union database on MLT ECA business is organised by ECA, destination country, activity (insurance or lending) and half-year, covering the years 2005 to 2018. Each record details the volume of new commitments by type (Sovereign, Other Public, Banks, Corporates and Projects), the volume of claims and recoveries (political, commercial, total), offers, reinsurance, exposure, staff, premium income, administrative costs and cash flow. In light of the aim of this study, it is important to note that the data reflect underwritten but not rejected contracts. Given that ECA transactions undergo a high level of scrutiny before signing, claims are an exception, not the norm.

For the purposes of this study, we focus on combined insurance and lending MLT business, and we enriched the data with ECA and destination summary information to indicate their size, general development, business diversification, and claim history. A detailed list of added attributes, including their rationale, is provided at Appendix A. All monetary variables were deflated using the 2010 based International Monetary Fund (IMF) Export-Import-Price-Index (XMPI) to obtain constant USD values (International Monetary Fund et al. 2009). Table 1 provides descriptive statistics of the 25,396 records available of the ML exercise on exposure, new commitments and claims.

[3] While Wüthrich (2018b) generates synthetic individual claims data, Wüthrich (2018a) uses liability claims data and the analysis by Thesmar et al. (2019) is based on healthcare claims data.

Table 1. Totals of exposure, new commitments and claims (mean and standard deviation (SD) of records by year, in constant USD million).

Year [1]	Number of Records	Exposure		New Commitments		Claims Paid	
		Mean	SD	Mean	SD	Mean	SD
2007	1983	254.37	785.74	72.24	350.92	0.58	4.13
2008	2028	248.34	804.19	73.51	334.62	0.49	4.48
2009	2094	278.96	927.32	91.29	528.51	1.44	27.37
2010	2063	284.32	873.14	82.59	343.57	0.82	6.37
2011	2072	288.09	876.39	86.18	364.00	1.07	10.15
2012	2078	303.35	897.73	79.91	324.73	1.02	11.36
2013	2061	320.35	939.42	71.49	275.30	1.08	9.69
2014	2150	296.78	883.90	70.46	356.33	0.93	10.03
2015	2194	301.25	901.09	64.78	347.95	1.38	24.78
2016	2189	308.82	971.67	58.51	330.23	1.34	13.06
2017	2239	306.62	985.34	57.85	374.20	1.18	9.42
2018	2245	301.31	1007.71	59.29	314.81	1.40	12.28

[1] Data was enriched to include simple trend estimates based on the current and two antecedent years (see Appendix A for details). Records from 2005 and 2006 could therefore not be used in support of the actual ML exercise.

4. Supervised Machine Learning

Supervised ML techniques aim to uncover potential relationships between independent and one or several dependent variables (Rokach and Maimon 2005), or more often, to simply find a function that allows a good prediction of a target attribute, based on available input attributes (Varian 2014). The scientific literature on the subject provides a wide range of ML applications, including Naïve Bayesian Classifiers, Bayesian Networks, Logistic Regression, Decision Trees (DT), Conditional Inference Trees, Random Forests (RF), Support Vector Machines, k-Nearest-Neighbour and Neuronal Networks (NN). The Least Absolute Shrinkage and Selection Operator (LASSO) algorithm is used occasionally in economic applications and is alleged to be most familiar to economists (Mullainathan and Spiess 2017). All these techniques are, in principle, suitable in supporting the prediction of claims as intended by this study.

Amongst other factors, it is the field of application (Singh et al. 2016), including the dependencies of its inherent variables, data structure, data quality, parameter tuning or the performance measure, that determines whether one algorithm performs better than others. To this date, there is no commonly accepted approach to link a particular problem to the most suitable ML technique to solve it (Kuhn and Johnson 2013; Wanke and Barros 2016). It has, therefore, become popular to apply several techniques to the same task and compare their performances (for example, Fauzan and Murfi 2018; Lorena et al. 2011; Mullainathan and Spiess 2017; Razi and Athappilli 2005; Singh et al. 2016; Weerasinghe and Wijegunasekara 2016).

We follow this methodological framework by comparatively investigating DT, RF, NN and PNN to predict claims in export credit insurance. Although, these techniques are well-understood and documented, we will provide brief descriptions and our rationale for employing them in this section. More in-depth explanations can be found in the references of the relevant paragraphs. For descriptions of the techniques not covered here, we refer to the works of Athey (2018), Mullainathan and Spiess (2017), Varian (2014) or Wanke and Barros (2016). Singh et al. (2016) provide a concise comparison of the advantages and disadvantages of the different techniques, and Charte et al. (2019) give an overview on non-standard ML problems.

4.1. Decision Trees

A DT is a recursive partition of a dataset into subsets which, ideally, amongst themselves are most heterogeneous with respect to a given target attribute. The DT model representation begins with a top node covering the entire dataset, characterized by the distribution of the target attribute. A DT algorithm seeks to select from all available input attributes the one attribute which, at an optimal split value, separates the data so that target attribute distributions of the subsets diverge as much as possible from the parent node, and are as pure as possible, meaning that each successor node contains mostly records of the same target attribute value. Options to measure the degree of purity include, but are not limited to, Gini impurity, Gini index, gain ratio and information gain (Rokach and Maimon 2005).

Let p_i denote the probability of a target attribute of domain i to be chosen at random. If the record was also labelled randomly according to the target attribute distribution, then the probability of the record being labelled incorrectly is $1 - p_i$. If $\left|\text{dom}(y)\right|$ denotes the cardinality of the target attribute domain, the Gini impurity of target attribute y of a given dataset S is defined as:

$$\text{Gini impurity } (y, S) = \sum_{i=1}^{|\text{dom}(y)|} p_i(1 - p_i) = \sum_{i=1}^{|\text{dom}(y)|} p_i - p_i^2 = 1 - \sum_{i=1}^{|\text{dom}(y)|} p_i^2.$$

In a perfectly pure data (sub)set, the probability of a record of type i to be chosen is 1, and its probability to be labelled incorrectly is 0, resulting in a Gini impurity of 0. The less pure the dataset, the larger the Gini impurity measure.

Let A denote the set of n input attributes $A = \left\{a_1, \ldots, a_j, \ldots, a_n\right\}$, c_j the domain of input attribute a_j, $\left|\text{dom}\left(a_j\right)\right|$ the cardinality of a_j's domain, and $\left|S_{c_j}\right|$ the cardinality of subset S_{c_j} of records of c_j, then the Gini index at split a_j is defined as:

$$\text{Gini index } \left(y, a_j\right) = \sum_{c_j=1}^{|\text{dom}(a_j)|} \frac{\left|S_{c_j}\right|}{|S|} \cdot \text{Gini impurity } \left(y, S_{c_j}\right).$$

The optimal split attribute a_j is the one which results in the maximum Gini gain (the difference between Gini impurity (y, S) and Gini index $\left(y, a_j\right)$ (Rokach and Maimon 2005), or simply the a_j which generates the minimum Gini index $\left(y, a_j\right)$.

In a DT representation, a split is signified by edges leading from the parent node to child nodes, typically displaying the target attribute distribution of the subsets which they represent. The algorithm continues to split child nodes in the aforementioned manner and stops when predefined criteria are met. Such criteria typically include a maximum number of splits, a minimum Gini gain threshold, or a minimum number of records per node. Nodes that are not further split are called leaves or terminal nodes.

If the DT is to classify new data, the value of the split attribute at each node determines which edge to follow until a terminal node is reached; this node infers the prediction for a given instance (Varian 2014). Figure 1 is an indicative example of a DT model representation with a dichotomized target attribute “CLAIMS (NO/YES)”, with its most relevant predictor being “EXPOSURE” at a split point of 50 million USD, and below the ≥50 million USD branch a second predictor of “DESTINATION CLAIM HISTORY” at a split point of 400 million USD.

Finding an optimal DT by brute force is, under normal circumstances, computationally infeasible, because the search space increases exponentially with the number of attributes and their values. However, a range of efficient so-called inducers such as C4.5, CART or CHAID have been developed to find reasonably accurate approximations (Rokach and Maimon 2005); some are limited to either, discrete or continuous problems, some can process both.

The key advantages of DT are a generally good performance with relatively little computational effort, and the output of intuitive, self-explanatory models (Singh et al. 2016), which can be communicated well to practitioners. The latter makes DT highly interesting for applied research problems, which is why we include them in this study.

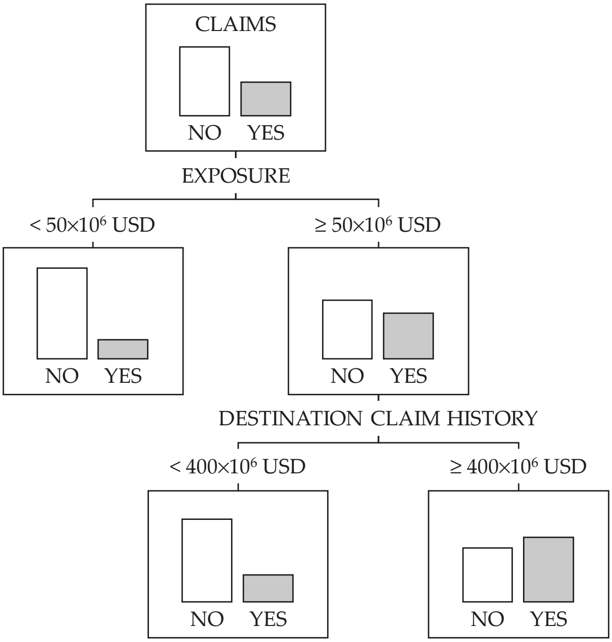


Figure 1. Indicative example of a decision trees (DT) representation.

4.2. Random Forests

DT can be sensitive to changes in the training sample, and are also likely to over-fit if training conditions are not carefully controlled (Singh et al. 2016; Varian 2014). The general idea behind RF is to train a multitude of DT, based on different bootstrap samples from the training data, and by sampling the input attributes that are available to the algorithm to choose from at each node (Breiman 2001; Fang et al. 2016; Varian 2014). As a result, RF algorithms generate a pre-defined number of DT, which may or may not come to different predictions when presented with new data. The overall prediction returned by an RF is the category chosen by the majority of DT (Lorena et al. 2011; Varian 2014), or the average result for continuous problems (Fang et al. 2016; Mullainathan and Spiess 2017).

RF often perform ahead of many other classifiers (Fang et al. 2016; Lorena et al. 2011; Singh et al. 2016) and are robust against overfitting (Fang et al. 2016; Liaw and Wiener 2002; Singh et al. 2016), which recommends them for inclusion in this study.

4.3. Neural Networks

NN consist of layers of so-called neurons (Claveria and Torra 2014). The number of neurons in the input layer equals the number n of input attributes. For a given record, each of the input neurons picks up the value of its associated input attribute $x_{\text{Input},j}$ and applies an activation function σ to calculate a signal value as output: $y_{\text{Input},j} = \sigma\big(x_{\text{Input},j}\big)$. Typically, sigmoid functions such as the hyperbolic tangent $\sigma(x) = \big(e^x - e^{-1}\big)/\big(e^x + e^{-1}\big)$ or a logistic function $\sigma(x) = 1/(1 + e^{-x})$ are used (LeCun et al. 2015). $y_{\text{Input},j}$ is forwarded to the neurons in the subsequent layer. One or several layers, known as hidden layers, collect and aggregate signals from preceding layers, and turn, them into new signals. Figure 2 is an illustration of an NN with just one hidden layer.

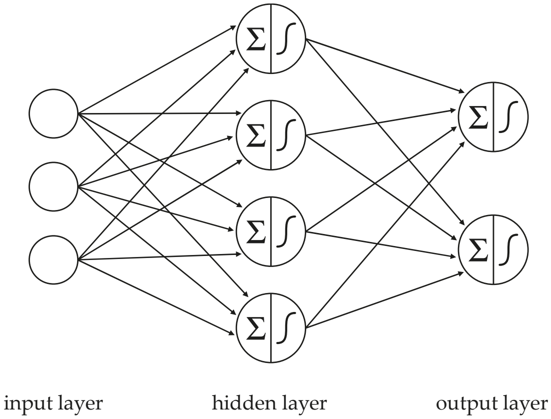


Figure 2. Illustration of a multilayer NN.

Let $y_{u,k}$ denote the signal that neuron l of layer v receives from neuron k of its preceding layer u, $w_{l,k}$ the weight that l applies to $y_{u,k}$, and $b_{v,l}$ a bias term, to calculate a weighted sum $z_{v,l}$,. l's signal $y_{v,l}$ is generated by applying an activation function σ to $z_{v,l}$:

$$z_{v,l} = \left(\sum_k w_{l,k} y_{u,k} \right) + b_{v,l};\ y_{v,l} = \sigma\left(z_{v,l}\right).$$

In classification problems, the number of neurons in the output layer equals the cardinality of the domain of the target attribute. During training, an objective function E measures for each record the (quadratic) error between the output signals $y_{\text{Output},i}$ of the output layer, and the actual target value y_i:

$$E = \sum_i \frac{1}{2}\left(y_{\text{Output},i} - y_i\right)^2;\ y_i = 1 \text{ for target domain } i,\ \text{otherwise } y_i = 0.$$

Given that the signals of each layer are functions of the weights, biases and signals of the preceding layers, E is ultimately a function of (averaged) weights and biases from all training records and all layers of the NN. The gradient of E indicates the sensitivity of the objective function to changes in these parameters:

$$\nabla E = \begin{bmatrix} \vdots \\ \frac{\partial E}{\partial w_{l,k}} \\ \frac{\partial E}{\partial b_{v,l}} \\ \vdots \end{bmatrix}.$$

The larger a partial derivative of E, the more the objective function benefits from its manipulation during the descend towards a minimum. Therefore, weights and biases are adjusted simultaneously in proportion of their negative partial derivative in every step of the training. This process is repeated until improvements in the cost function fall below a predefined threshold. When a trained NN is used for prediction, the learned rules are applied to new data, and the resulting output values are used as prediction values (LeCun et al. 2015).

We include NN in this study because they are thought to be better suited than DT to model complex, nonlinear relationships (Claveria and Torra 2014; Razi and Athappilli 2005; Singh et al. 2016). Although, Varian (2014) provides a case to the contrary. Given that it is possible for claims to be the result of nonlinear economic relationships, it is interesting to see whether NN perform better than DT in predicting claims.

4.4. Probabilistic Neural Networks

Specht (1990) proposed to modify NN by replacing the traditionally implemented sigmoid activation functions with statistically derived exponential functions (Iounousse et al. 2015). Specht named the class of such algorithms PNN and demonstrated that the introduced modification, under certain, but easy, to meet conditions, makes it possible to asymptotically approach the Bayes optimal decision surface of a classification problem (Specht 1990). PNN can map any input pattern to any number of classifications, are capable of handling erroneous, sparse or missing data well, and provide probability estimates in conjunction with their classification (Specht 1990). The feature of explicit probabilities allows for extended analyses, e.g., of classification errors, and provides opportunities to further improve prediction. In addition, PNN are more flexible than NN in handling different types of input variables, and it seems generally valuable to test a variation of NN alongside their original implementation, which is why we include PNN in our set of ML techniques.

5. Methodology

5.1. General Modelling Considerations

All ECAs exist to promote exports, but different national priorities have resulted in various designs and mandates under which they operate (Stephens and Smallridge 2002). Furthermore, an ECA's business is significantly impacted by its nation's economic size and export characteristics-profile. Similarly, the political, judicial and commercial structure and stability of a destination country are important factors of its risk profile. Classic econometric modelling requires such heterogeneity to be accounted for, for example, by introducing ECA or destination dummy variables, to reflect effects that are stable and specific to individual countries, and could, therefore, bias the model if omitted. The DT, RF and PNN techniques, and the NN technique with some limitations, are perfectly capable of recognizing ECA or destination names as input variables. However, in this study we deliberately prevented the ML algorithms from knowing the specific agents of a given transaction. The rationale is that if a certain attribute, such as ECA or destination name, is used during model training (see Section 5.4 below), the resulting model requires that information to be present for prediction purposes. Otherwise, when attempting to make a prediction for an ECA or destination, not observed during training, the model fails. This can create problems at the training-validation gateway. More importantly, it precludes the model from making predictions for "new" ECAs or destinations. However, these might be the most relevant cases for ML to be employed in export credit insurance claim prediction. To enable our ML models to deal with any agent, whether or not it contributed training data, we exclusively fed generic information such as export volumes, portfolio diversity etc. as inputs (see Appendix A for used attributes) to reflect different phenotypes of ECAs' and destinations' phenotypes. However, this approach bears some risks in introducing unobserved heterogeneity, which should be borne in mind when analysing prediction outcomes.

A second consideration is associated with the nature of the intended prediction. Claims gain most attention when they are exceptional, for example, when an ECA with traditionally low claims gets hit by a large number or sum of claims within a short period of time. Therefore, the identification of patterns preceding singular events of claims was considered as a potential aim of this study. However, during the explorative phase it showed that, across ECAs and destinations, the occurrence of claims is quite diverse. Although, most records in the database report no claims, ECA, destination or annual aggregates often do. There are some ECAs or destinations for which claims are actually rare. However, for some ECAs and destinations claims are a fairly regular feature, and some ECAs and destinations are somewhere in between. Given that this is the first time the Berne Union dataset is extensively analysed with a view towards claims, a decision was made to first explore the overall situation across all agents before focusing on subsets. Consistent with that, the study attempts a more general assessment of the adequacy of different ML techniques to be used in claim prediction.

5.2. Prediction Tasks

Predicting claims can take a variety of shapes. To compare the performance of the different ML techniques, we train models to solve prediction tasks with different degrees of difficulty (see Appendix A, section "Target attributes", for implementation details):

- "Claims YES/NO": At the simplest level, the technique is to predict whether or not a given export finance condition will incur claims as a dichotomous yes/no decision.
- "Claim ratio class": Claims can vary significantly in value, so that a yes/no prediction is a great simplification of the problem. Therefore, we also test ability of the techniques to predict the magnitude of claims, expressed as five classes of claims/exposure-ratios.
- "Claim ratio": Ultimately, we also want to evaluate how well ML techniques perform in predicting an actual claim ratio, measured in terms of claims/exposure.

5.3. Technical Implementation of ML Algorithms and Analysis

Today, a range of tools, such as Python, RapidMiner or R, are available to support comfortable implementations of ML workflows. For this study, we use the data analytics platform KNIME. KNIME is a free and open-source software for data retrieval, data blending, modelling, analysis and visualization. It includes a rich collection of ML and data mining components which can be assembled following a modular data pipelining concept (KNIME 2019). All data preparation, training and testing procedures were entirely designed and set up in KNIME; Table 2 shows a mapping of the KNIME ML nodes that were selected against the prediction problems. Details on the nodes are available via the KNIME node and workflow search engine (NodePit 2019).

Table 2. Mapping of ML techniques, prediction task and KNIME nodes.

Task	ML Technique			
	DT	RF	NN	PNN
Claims YES/NO	Decision Tree Learner Decision Tree Predictor	Random Forest Learner Random Forest Predictor	RProp MLP Learner MultiLayerPerceptron Predictor	PNN Learner (DDA) PNN Predictor
Claim ratio class	Decision Tree Learner Decision Tree Predictor	Random Forest Learner Random Forest Predictor	Not investigated [1]	PNN Learner (DDA) PNN Predictor
Claim ratio	Simple Regression Tree Learner Simple Regression Tree Predictor	Random Forest Learner (Regression) Random Forest Predictor (Regression)	RProp MLP Learner MultiLayerPerceptron Predictor	Not investigated [2]

[1] The only way to use the KNIME MLP node to obtain claim ratio classes is to calculate values first and classify them afterwards. This is assessed to add no value to the ML analysis and is therefore omitted. [2] During the exploratory study phase, KNIME PNN proved to be unduly computationally costly in solving problems with continuous target variables, and therefore were not further assessed against the "claim ratio" task.

5.4. Training, Validation and Test Data

It is well known that ML algorithms can over-fit, resulting in good in-sample but poor out-of-sample performance. Therefore, it is common to randomly split the data into a training and a validation set (Kuhn and Johnson 2013; Mullainathan and Spiess 2017), specify models based on training data and test them against the validation data. The objective function is to minimise deviations between predicted and actual target attribute values in the latter (Athey 2018). More advanced approaches divide the data into three types of data, including; training data to estimate models; validation data to choose a model, and; test data to assess its performance (Varian 2014).

Dividing the entire dataset into subsets for training, validation and testing by random sampling is a defence against overfitting. However, it might not be a valid strategy for obtaining reliable prediction models:

- Random sampling from the same population might, analogous to the law of large numbers or the Glivenko-Cantelli theorem, result in generally converging conditions in the subsets. A model which reflects the training data well without overfitting may, therefore, also be a good representation of the validation and test sample by sheer principles of statistics.
- In a practical setting, an insurer would have no choice but to use historic data to make forecasts about future data. Effectively, this implies a strictly chronological data separation, which is different from random sampling.

To test and counter these concerns, we exclude 2018 data from model development and validation, and only use them as test data later in the process. The records covering the period between 2007 and 2017 are used for training and validation. Figure 3 depicts the data separation and their use as part of the entire training, validation and testing procedures employed by this study.

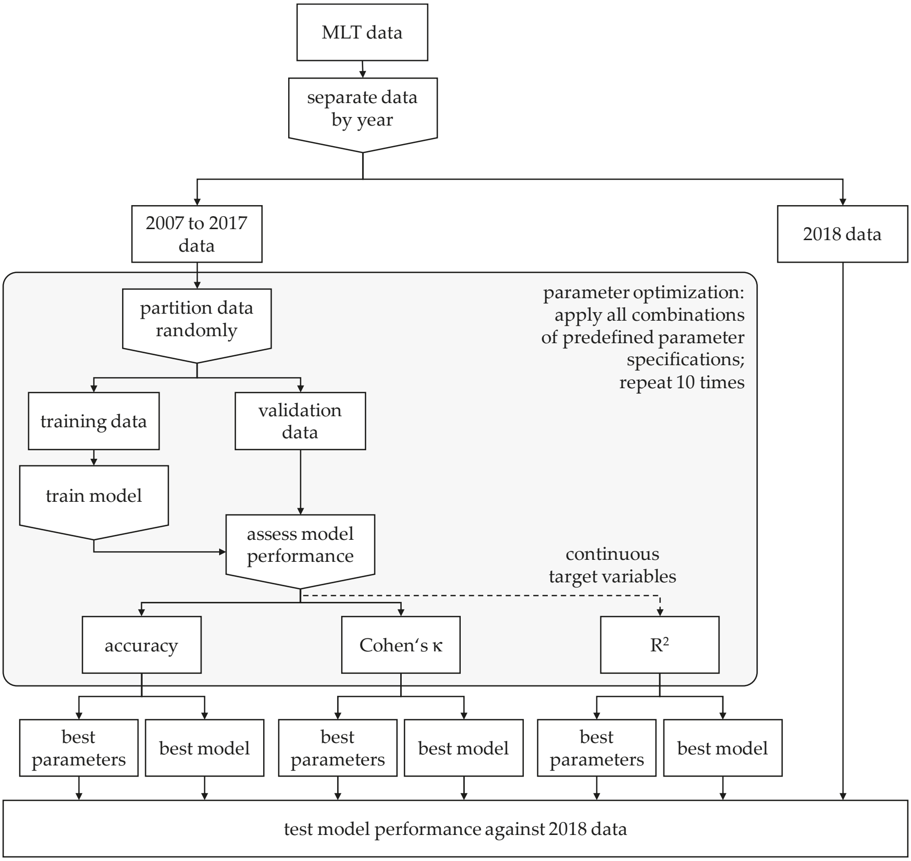


Figure 3. Model training, validation and testing process.

5.5. Parameter Optimisation

ML inducers optimise a specific objective function by tuning parameters that can be seen as internal to the algorithm. However, the performance of an algorithm is also affected by a range of parameters that require external intervention. Typical examples are the selection of the objective function itself or stopping criteria. Such parameters can neither be derived from the problem nor otherwise be independently calculated (Kuhn and Johnson 2013; Wanke and Barros 2016). External parameter optimisation is, therefore, integral to obtaining a powerful prediction model. Some of the externally determined parameters are specific to an algorithm, but also some more general conditions around data preparation and provision can play a role. Besides algorithm specific parameters, we investigate how the size of the training sample (relative to the validation sample) and the fraction of records with no claims affect model performance:

- The relation between the volume of the training and validation data addresses the simple question of whether training with a smaller and validation against a larger sample (which might protect against overfitting), or training with a larger and validation against a smaller sample yields better results.
- The rationale for reducing the number of records with no claims is their dominance in the Berne Union dataset (87.5% of the 2007–2017, and 86.2% of the 2018 records register a total of 0 claims paid). This imbalance will cause models to lean towards the prediction of no claims although it might be desirable to identify potential claims with precedence. A prioritized identification of a recessive value can be achieved by partial suppression of the dominant value during training.

An overview of the algorithm-specific and general external parameters, including applied variations, is provided at Appendix B. We explore all combinations of parameter variations by brute force.

5.6. Model Benchmark

ML techniques require extensive data preparation and can be computationally costly, raising the question of whether they actually perform better than simple heuristics (England and Verrall 2002).

The ML models of this study are generic and can be applied to any ECA and destination country, irrespective of whether or not the ECA has a history of providing cover for the destination. Although, no trivial method offer a fully equivalent capability, moving averages are a simple way for an ECA to predict claims for destinations that it engaged in business with previously. In such cases, an estimator for the claims ratio $r_{i,j,t} = \frac{c_{i,j,t}}{e_{i,j,t}}$ of ECA i and destination j in a given year t can be defined as ($e_{i,j}$ denotes the exposure of ECA i to destination j, and $c_{i,j}$ denotes the respective claims; l is the number of preceding years to be considered, also referred to as "window length" of the moving average):

$$\hat{r}_{i,j,t} = \frac{\sum_{\nu=1}^{l} c_{i,j,t-\nu}}{\sum_{\nu=1}^{l} e_{i,j,t-\nu}}.$$

The resulting estimator, or a transformation of it into a binary YES/NO variable or a claim ratio class, can be used as BM to help assess the benefit of instituting a more complex ML technique.

To avoid an arbitrary definition of the moving average's window length l, for each ECA i and destination j we determine the optimal window length $l_{i,j,\ opt}$ which minimizes:

$$\frac{1}{max\{1; t-2007\}} \cdot \sum_{t=2007}^{2017} \left| \frac{c_{i,j,t}}{e_{i,j,t}} - \frac{\sum_{\nu=max\{2007;t-l\}}^{t-1} c_{i,j,\nu}}{\sum_{\nu=max\{2007;t-l\}}^{t-1} e_{i,j,\nu}} \right|.$$

The data separation employed during the development of the ML models (see Section 5.4) is also applied to the BM, i.e., data from 2007 to 2017 are used to identify $l_{i,j,\ opt}$, and 2018 data serve to test the BM.

The execution of the BM optimisation yields that the optimal window length is mostly 1, meaning that, on average, the previous year's claims ratio often best predicts the current year's claim ratio. Table 3 provides an overview of the number of times each window length was determined to be optimal.

Table 3. Optimal window length for moving average BM.

Optimal Window Length	Number of ECA-Destination Combinations	%
1	2156	80.4
2	112	4.2
3	49	1.8
4	37	1.4
5	33	1.2
6	22	0.8
7	28	1.0
8	25	0.9
9	33	1.2
10	188	7.0

5.7. Assessment of Model Performance

The obvious measure to assess model performance is accuracy, the proportion of correctly classified records. However, it is useful to additionally consider Cohen's κ, originally designed to evaluate inter-rater reliability. Cohen's κ adjusts accuracy p_o by considering correct predictions that would occur at random,

$$\kappa = \frac{p_o - p_c}{1 - p_c},$$

where p_c is the proportion of records expected to be correctly classified by chance (Cohen 1960). A κ of 0 means that accuracy is equal to agreement at random, a κ of 1 indicates perfect agreement (Cohen 1960), equating to 100% correct model predictions. A further advantage of this prudent correction is that it penalises false predictions more evenly, irrespective of the predominance of individual values: As mentioned above, 86.2% of the 2018 records register 0 claims. Under these circumstances, a completely naïve model could achieve an accuracy of 0.862 by simply predicting "0 claims" 100% of the time. However, this would equal agreement by chance and result in $\kappa = 0$, which seems a more suitable evaluation of the worth of the model. For the assessment of continuous target variables, we use R^2.

It is possible for a model to perform well by chance during validation, preceding a much-reduced performance during testing. To account for that possibility, we repeat the parameter optimisation ten times. This approach is different from the more conventionally used cross-validation (Varian 2014), but should achieve a comparable level of model-validation; it greatly simplifies the implementation of the desired training/validation-sample-size ratio optimisation (see Section 5.5). The combination of parameters yielding the highest average performance are used to test the models against 2018 data. In addition, we collect the models with the highest performance overall for testing.

All performance measures are also applied to the BM by comparing the BM's prediction for year t with the actual value of year t. The BM's window length optimisation (see Section 5.6) does not involve any type of validation, which is why we apply the performance measures directly to the claims ratio predictions, generated during the optimisation stage. The test performance measures of the BM and the ML techniques are more comparable because, analogous to the ML model optimisation, the BM's window length optimisation is based on 2007 to 2017 data, with 2018 data reserved for testing.

6. Results

Table 4 shows the validation and test results for both, the "Claims YES/NO" and the "Claim ratio class" task in terms of accuracy. Cohen's κ results are shown in Table 5. Table 6 lists R^2 results, which we used as performance measure for the "Claim ratio" task. The BM performance measure is shown in the rightmost column (identical values are given against the "Best parameters" and "Best model" section per task, as no such distinction exists for the BM). The study observations include:

- Amongst the ML techniques, with only two exceptions RF generate the best performance.
- The accuracy achieved against the "Claim ratio class" task is not much different from the accuracy of the less challenging "Claims YES/NO" task. However, Cohen's κ is more reflective of performance differences, indicating that both, validation and test performance, deteriorate as the task becomes more difficult.
- None of the investigated ML techniques yield satisfactory results against the "Claim ratio" task; predictions of actual claim ratios turned out to be largely unreliable.
- The test performance is lower than validation performance (with only two exeptions), often by just a small margin. Performance losses are more pronounced when measured by Cohen's κ.
- No definitive conclusion can be made on whether validation should serve to identify optimal model parameters, or to actually generate the specific model for prediction (sometimes utilizing the best parameters, sometimes employing the best model yields better test performance; optimal parameters are provided at Appendix C).
- The accuracy of the ML techniques is sometimes better, but generally at similar levels as the BM's value.
- In terms of Cohen's κ, the BM performs better than any of the ML techniques. The reason is that some ECAs experience uninterrupted sequences of claims with certain destinations. Therefore, the simple rule "claims in $t-1$ indicate claims in t" employed by the BM (see Section 5.6) works well against the "Claims YES/NO" task, and also against the "Claim ratio class" task.
- Against the "Claim ratio" task, the ML techniques outperform the BM, although at a very low level.

Table 4. Best parameter and best model results: Accuracy (bold: best performing ML technique).

Task	Outcome	Dataset	DT	RF	NN	PNN	*BM*
Claims YES/NO	Best parameters	Validation	0.886	**0.900**	0.887	0.881	*0.901*
		Test	0.878	0.889	0.874	**0.897**	*0.896*
	Best model	Validation	0.900	**0.909**	0.900	0.898	*0.901*
		Test	0.878	**0.890**	0.848	0.864	*0.896*
Claim ratio class	Best parameters	Validation	0.881	**0.888**	–	0.877	*0.867*
		Test	0.861	0.869	–	**0.888**	*0.858*
	Best model	Validation	0.896	**0.903**	–	0.897	*0.867*
		Test	0.864	**0.870**	–	0.855	*0.858*

Tables 4–6 provide a "best performance" comparison, imitating outcomes of an actual insurer's claim prediction exercise. While, poorly performing models would normally be of little interest to practitioners, we collected all models from the parameter optimisation stage of this study, irrespective of their performance. This allows for more detailed analyses of the results which are provided in the following sections.

Table 5. Best parameter and best model results: Cohen's κ (bold: best performing ML technique).

Task	Outcome	Dataset	DT	RF	NN	PNN	*BM*
Claims YES/NO	Best parameters	Validation	0.352	**0.439**	0.357	0.292	*0.566*
		Test	0.322	**0.408**	0.340	0.275	*0.578*
	Best model	Validation	0.421	**0.489**	0.433	0.358	*0.566*
		Test	0.297	**0.423**	0.303	0.284	*0.578*
Claim ratio class	Best parameters	Validation	0.252	**0.336**	–	0.211	*0.446*
		Test	0.250	**0.320**	–	0.175	*0.458*
	Best model	Validation	0.276	**0.392**	–	0.272	*0.446*
		Test	0.240	**0.336**	–	0.170	*0.458*

Table 6. Best parameters and best model results: R^2 (bold figures: best performing ML technique).

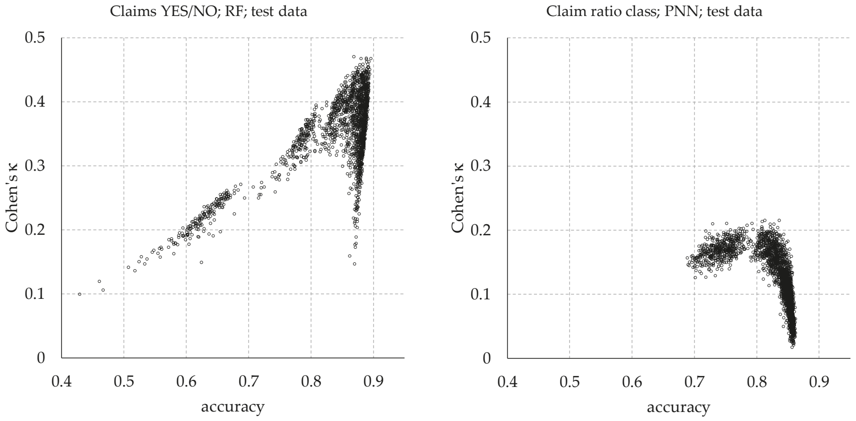
Task	Outcome	Dataset	DT	RF	NN	PNN	*BM*
Claim ratio	Best parameters	Validation	0.038	**0.071**	0.066	–	*0.000*
		Test	0.021	**0.053**	0.046	–	*0.011*
	Best model	Validation	0.081	**0.128**	0.126	–	*0.000*
		Test	0.037	**0.074**	0.027	–	*0.011*

6.1. Relationship between Accuracy and Cohen's κ

A comparison of Tables 4 and 5 indicates that Cohen's κ accentuates performance differences better than accuracy (parameter optimisation confirmed that Cohen's κ benefits from reducing the number of records with 0 claims down to 20 to 40% during training; highest accuracies were achieved with 80–100% of records with no claims; see Appendix C for parameter details). A high Cohen's κ might be associated with more correctly predicted claims (true positives) at the cost of less true negatives, thereby sacrificing some accuracy. We applied all models from the parameter optimisation stage to the test data, in order to understand the relationship between the two performance measures empirically, logged each model's accuracy and Cohen's κ and plotted them against each other. Figure 4 shows scatterplots of accuracy and Cohen's κ for RF and PNN models:

- For the RF models ("Claims YES/NO" task), shown on the left, accuracy and Cohen's κ increase together, peaking close to (0.89, 0.47). From the peak, there is a sharp drop of Cohen's κ, accompanied by a moderate reduction of accuracy.
- The PNN models ("Claim ratio class" task) on the right also show an initial joint increase of accuracy and Cohen's κ. Cohen's κ peaks at a value of 0.22, from which a further increase of accuracy is associated with a marked deterioration of Cohen's κ.

Scatterplots for all investigated ML techniques are provided at Appendix D, showing that against the "Claims YES/NO" task, DT, RF and NN generated models with high Cohen's κ while retaining high accuracy at the same time. Against the "Claims ratio class" task, only RF yielded models with both measures being high. In conjunction with its general advantages (see Section 5.7), Cohen's κ is assessed to be the more insightful measure for the purposes of this study. However, for other applications, accuracy might be more relevant.


Claims YES/NO; RF; test data

Claim ratio class; PNN; test data

Figure 4. Example scatterplots highlighting the relationship between accuracy and Cohen's κ (scatterplots for all ML techniques are provided at Appendix D).

6.2. Comparison of ML Technique Performance

With only two exceptions, RF consistently delivered the best performance (see Tables 4–6). We further compared the performance of all models by prediction task and ML technique via Kruskal-Wallis tests; the results are shown in Table 7. Appendix E provides boxplots to illustrate the performance of all models developed during the parameter optimisation exercise of this study (Figure A2; the left half of the table shows performance variations measured during validation, mirrored on the right by the corresponding performance of the same models applied to the test data). The tests confirm statistically significant differences between the performances of the techniques. Pairwise Wilcoxon-Mann-Whitney post-hoc tests were all significant with $p \simeq 0.0$, corroborating that RF are generally most successful in predicting claims under the conditions of this study (an interesting anomaly is that, against the "Claims YES/NO" task, NN are the worst performer in terms of accuracy, but the second-best performer in terms of Cohen's κ).

Table 7. Kruskal-Wallis tests on ML technique performance (test data; bold figures: highest median rank).

			Median Rank			
Task	**Measure**	***p*-Value**	**DT**	**RF**	**NN**	**PNN**
Claims Y/N	Accuracy	0.0	11,628.5	**19,613.5**	8972	10,504
	Cohens κ	0.0	11,134.5	**19,716.5**	13,003	5144
Claim ratio class	Accuracy	0.0	8365	**11,558.5**	–	4934
	Cohens κ	0.0	6526.5	**11,467.5**	–	4394
Claim ratio	R^2	0.0	2425.5	**11,310.5**	6115.5	–

6.3. Validation and Test Performance

Following the methodology outline of the study (see Section 5), we used the parameters and models that performed best during validation to make predictions for 2018 data, assuming that this approach is most likely to be adopted by practitioners. However, a model that performs well during validation might not be optimal when confronted with new data. In fact, a comparison of corresponding validation and test performance in Tables 4–6 shows a performance reduction in all but two cases.

Obviously, for an ML technique to be reliable it is important that its validation performance be a good indicator of its performance when used to make forecasts. To investigate this relationship, we calculated the correlation between validation and corresponding test performance, and estimated linear functions to describe their relationship; results are provided in Table 8 (standard errors of regression parameters are provided in brackets; all parameters are statistically significant with $p \simeq 0$):

- Against both the "Claims Y/N" and the "Claim ratio class" task, validation and test performance are generally highly correlated. An exception are NN, and also PNN, against the "Claims YES/NO" task, when performance is measured in terms of Cohen's κ. RF consistently exhibit the highest correlation for all tasks and measures, although sometimes by just a small margin.
- Validation-test correlations are much lower against the "Claim ratio" task, but RF, again, achieve the highest value.
- In conjunction with a validation-test-correlation close to 1, an intercept close to 0 and a slope close to 1 indicate greatest performance reliability. For Cohen's κ, which are considered the most insightful performance measure, and R^2 this is best achieved by RF.

Table 8. Correlation and relationship between validation and test performance.

Task	Measure	ML Technique	Validation-Test Correlation	Intercept (Std. Error)	Slope (Std. Error)
Claims Y/N	Accuracy	DT	0.981	−0.078 (0.003)	1.073 (0.004)
		RF	0.990	−0.097 (0.002)	1.098 (0.003)
		NN	0.952	−0.102 (0.003)	1.079 (0.004)
		PNN	0.990	−0.319 (0.002)	1.346 (0.002)
	Cohen's κ	DT	0.851	0.045 (0.002)	0.882 (0.009)
		RF	0.905	0.020 (0.003)	0.970 (0.008)
		NN	0.492	0.141 (0.003)	0.504 (0.010)
		PNN	0.688	0.090 (0.002)	0.625 (0.008)
Claim ratio class	Accuracy	DT	0.976	−0.108 (0.004)	1.107 (0.004)
		RF	0.979	−0.154 (0.004)	1.159 (0.004)
		PNN	0.978	−0.320 (0.003)	1.346 (0.004)
	Cohen's κ	DT	0.902	0.025 (0.001)	0.882 (0.007)
		RF	0.908	0.017 (0.002)	0.924 (0.007)
		PNN	0.882	0.017 (0.001)	0.830 (0.006)
Claim ratio	R^2	DT	0.214	0.011 (0.000)	0.168 (0.017)
		RF	0.706	0.013 (0.001)	0.812 (0.018)
		NN	0.611	0.007 (0.000)	0.487 (0.007)

6.4. Computational Complexity

The four investigated ML techniques exhibited very different properties in terms of run-time and model size. The DT algorithm consistently produced results much quicker than any of the other algorithms, whereas PNN proved to be most time consuming. Depending on the task, the PNN took, on average, up to 675 times as long as the DT to produce and validate one model. RF turned out to be the second quickest technique (between nine to 15 times DT run-time), followed by NN (45 to 50 times DT run-time).

On the other hand, RF models occupied significantly more storage than those produced by of any of the other techniques. To some extent, this is to be expected, given that one RF model consists of many DT (the RF models trained for the purposes of this study consisted of between 50 and 200 DT; see Appendix B for details of parameter settings). However, PNN models can also be relatively large. This is most certainly driven by their feature to provide probabilities against all possible classifications, rather than just a single classification. However, against the Claims Y/N task, this means three attributes (probability for class "NO", probability for class "YES", and prediction) instead of just one (prediction) and does not fully explain the size difference between NN and PNN models. DT and NN models were usually relatively small.

Neither, the run-time of the slowest, nor the model size of the most storage-consuming ML technique are of concern when a single model is being built. However, external parameter optimization, as undertaken as part of this study (see Section 5.5), can easily result in several thousands of models. In such instances, both, the time consumption of the PNN technique and the model size of the RF technique can easily push a regular office desktop to its limits.

Table 9 provides average run-times (in milliseconds (ms)) to train and validate one model based on 20,000 records for training and 3000 records for validation (64 Bit Windows machine, 2.11 GHz Intel® Core i7-8650U CPU, 16 GB RAM), and the average size of one model (in kilobyte (kB)) per task and ML technique.

Table 9. Comparison of ML algorithm run-times and model sizes.

Task	ML Technique	Average Time to Train and Validate One Model (ms)	Average Model Size (kB)
Claims Y/N	DT	342	5.1
	RF	4177	1686.5
	NN	15,533	12.3
	PNN	179,207	735.2
Claim ratio class	DT	272	5.2
	RF	4025	2124.0
	PNN	183,737	752.0
Claim ratio	DT	312	11.9
	RF	2821	5544.1
	NN	15,446	12.3

7. Conclusions and Outlook

The purpose of our study was to evaluate ML techniques as a means for the prediction of claims of export credit insurers. ML could be well-suited to provide more accurate claims predictions, as regulatory requirements require for more sophisticated approaches for predicting claims, as well as in calculating claims reserves, and the global environment of international trade might lead to more volatility in actual claims experience. While, insurers have been using deterministic or stochastic methods based on claims development triangles, more complex methods are based on stricter assumptions, which can lead to several issues in their application and interpretation. Insurers welcome automation and appreciate the increased speed of these methods, but it is still common to apply human judgement on the results. However, more advanced models are able provide additional information useful for the decision-making of the insurance company.

Therefore, we conducted a comparative study of four ML techniques and evaluated their ability to accurately predict claims based on a unique dataset of export credit insurance claims over the period of 2005 to 2018. Furthermore, we compared the ML techniques against the performance of a simple heuristic, based on moving averages of claims from destinations that the insurer has done business with previously.

Consistent with previous works (Fang et al. 2016; Lorena et al. 2011; Singh et al. 2016), RF provided the best results by a range of measures. Therefore, it seems advisable to include RF in any further research on the subject. However, RF can predict a target attribute value when provided with new data, but they do not readily reveal the logic underlying that prediction. The strength of traditional econometric approaches is that they help to extract relationships from masses of data by distilling compact equations. These equations can also be applied to new data for purposes of prediction, but more importantly, they can be analysed, in order to understand the relevance and inter-dependencies of the system defining variables. This benefit exists neither for RF nor NN, PNN or many other ML techniques, which is why they have been labelled "black boxes" by some (Olden and Jackson 2002). It is an interesting question to understand what place a technique that produces good predictions, but does not contribute to a better understanding of a subject, can have in academic research. An exception is the DT technique, because it generates human-readable rules which provide some insight into the most important predictors of the dependent variable. Therefore, we recommend to employ DT alongside with RF as a preparatory or augmenting step.

Several ML techniques have delivered satisfactory results against the "Claims Y/N" and "Claim ratio class" task, but the generally poor performance against the "Claim ratio" task is a serious shortcoming. While, it is unsurprising to find the lowest performance against the most challenging task, it is not obvious why predictions of claim ratios lag behind the two other tasks by such a large margin. A more detailed examination of the actual and predicted data indicates that model quality appears to be significantly hampered by singular events of high claims, suggesting that no model was capable of capturing the conditions preceding their occurrence. However, singular or exceptionally high claims, which were not a focal point of this paper, might be of particularly interesting to ECAs. Therefore, a follow-up study should investigate the prediction of claims of that type. This would require an exploration of the circumstances under which a claim is considered to be exceptional, and probably an addition of external economic data from sources such as OECD or similar.

It can also not be overlooked that the ML models in many respects performed no better, and often worse, than the simple heuristic "claims in $t-1$ indicate claims in t" as reflected by the BM (see Section 5.6). Unlike the ML models, the BM is limited to ECAs and destinations with already existing business relations. If such a business relationship does exist, the computationally much less complex BM rule must be seen as superior to the investigated ML techniques. In all other cases, ML might provide an alternative. Looking positively at the performance comparison between the BM and the ML techniques (Tables 4–6), it can be stated that ML is capable of predicting the virtue of a non-existing business relationship almost as well as if it would already exist. To help contain the effort of building ML models for practical applications, we provide the optimal model parameters as identified during this study at Appendix C.

Finally, there are two interesting topics for further research arising from the convergence of ML and traditional techniques employed in insurance economics. The first topic refers to a performance comparison between ML techniques and commonly used approaches such as Chain-Ladder or Bornhuetter-Ferguson methods (Wüthrich 2018a, 2018b). To allow for a direct and fair comparison, the requirement for ML models to be generic would have to be dropped, and individual claims data over a time period instead of aggregate claims would need to be analysed. In that context it should also be possible to better account for heterogeneity of ECAs and destinations, for example by following the approach proposed by Wüthrich (2018b). A second topic might evolve from the question whether classic problem-specific models, for example probability distributions for low-default portfolios (for example, Kiefer 2009), can or should be merged with ML techniques, and to what extent this could further improve prediction performance.

Author Contributions: Conceptualization, M.B. and S.K.; data curation, M.B.; formal analysis, M.B.; methodology, M.B.; project administration, S.K.; software, M.B.; validation, M.B.; visualization, M.B.; writing—original draft, M.B. and S.K.; writing—review editing, M.B. and S.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Acknowledgments: The authors thank the Berne Union and Vinco David for the provision of the dataset and particularly Paul Heaney for his support in understanding the specifics of the data. They also thank four anonymous reviewers for their comments and suggestions, which improved the article.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Data Enrichment

For the ML exercise, the year, the total of new commitments and the total exposure were used directly from the Berne Union database, and augmented with the following variables:

Target attributes (only one used at a time, depending on the prediction task):

- A dichotomous claims variable ("Claims YES/NO": "NO" if the total amount of claims paid equals 0, "YES" otherwise),
- Five classes of the claims/exposure ratio ("Claim ratio class"; classes are [0, 0], (0, 0.0033], (0.0033, 0.01], (0.01, 0.05], (0.05, ∞)),
- The claims/exposure ratio ("Claim ratio").

ECA summaries (annual values):

- Number of destination countries with exposure,
- Number of destination countries with exposure previous year (only used for NN),
- Number of destination countries with exposure two years ago (only used for NN),
- Number-of-destinations trend ("UP" for three consecutive years of increase, "DOWN" for three consecutive years of decrease, otherwise "AMBIGUOUS"), to indicate whether the ECA appears to generally expand or reduce the number of destinations in their portfolio (not used for NN),
- Destination exposure in % of the ECA's total exposure, to indicate the relevance of the destination for the ECA,
- Gini-coefficient of exposure, to indicate the ECA's exposure diversification across their destinations,
- Number of years with claims prior to the current year,
- % of years with claims prior to the current year,
- Total of new commitments in the current year,
- Total of new commitments in the previous year (only used for NN),
- Total of new commitments two years ago (only used for NN),
- Total of new commitments trend ("UP" for three consecutive years of increase, "DOWN" for three consecutive years of decrease, all other "AMBIGUOUS"), to indicate whether the ECA appears to generally expand or reduce the volume of their commitments (not used for NN).

Destination summaries (annual values):

- Number of ECAs with exposure,
- Number of ECAs with exposure previous year (only used for NN),
- Number of ECAs with exposure two years ago (only used for NN),
- Number-of-ECAs trend ("UP" for three consecutive years of increase, "DOWN" for three consecutive years of decrease, all other "AMBIGUOUS"), to indicate whether the destination appears to generally expand or reduce the number of ECAs it is doing business with (not used for NN),
- ECA exposure in % of the destination's total exposure, to indicate the relevance of the ECA for the destination,
- Gini-coefficient of exposure, to indicate the destination's exposure diversification across the ECAs it is doing business with,
- Number of years with claims prior to the current year,
- % of years with claims prior to the current year,
- Running total of claims until prior to the current year,
- Total of new commitments in the current year,
- Total of new commitments in the previous year (only used for NN),
- Total of new commitments two years ago (only used for NN),
- Total of new commitments trend ("UP" for three consecutive years of increase, "DOWN" for three consecutive years of decrease, all other "AMBIGUOUS"), to indicate whether the destination appears to generally attract an increasing or decreasing amount of commitments (not used for NN).

Appendix B. Summary of Externally Optimised and Fixed Parameters

Table A1 details algorithm-specific and general externally tested parameters, including their variation boundaries and increments.

Table A1. Parameter summary.

KNIME Node [ML Technique]	Parameter	Lower Limit	Upper Limit	Increment
Decision Tree Learner/ Simple Regression Tree Learner [DT]	Minimum number of records per node	30	90	20
	Quality measure	Gini index (fix)	-	n/a
	Pruning method	MDL (fix)	-	n/a
	Average split point	Yes (fix)	-	n/a
	Binary nominal splits	No (fix)	-	n/a
Random Forest Learner/ Random Forest Learner (Regression) [RF]	Number of models	50	200	50
	Split criterion	Information gain ratio (fix)	-	n/a
RProp MLP Learner [NN]	Number of hidden layers	1	3	1
	Number of neurons per layer	10	20	5
	Maximum number of iterations	100 (fix)	-	n/a
PNN Learner (DDA) [PNN]	Theta minus	0.1	0.35	0.05
	Theta plus	0.35	0.65	0.05
General	Training/validation partitioning fraction	0.1	0.9	0.1
	Fraction of records with 0 claims	0.1	1	0.1

Appendix C. Optimal Parameters

Appendix C lists the optimal parameters identified during the validation stage of this study by ML technique: Table A2 for DT, Table A3 for RF, Table A4 for NN and Table A5 for PNN.

Table A2. Results: Optimal DT Parameters.

Measure	Task	Outcome	Partitioning Fraction	0 Claim Fraction	Minimum Number of Records per Node
Accuracy	Claims YES/NO	Best parameters	0.9	1.0	30
		Best model	0.9	1.0	30
	Claim ratio class	Best parameters	0.9	0.8	30
		Best model	0.9	0.9	70
Cohen’s κ	Claims YES/NO	Best parameters	0.9	0.4	50
		Best model	0.9	0.5	30
	Claim ratio class	Best parameters	0.9	0.2	30
		Best model	0.8	0.2	50
R^2	Claim ratio	Best parameters	0.9	0.8	90
		Best model	0.9	0.8	70

Table A3. Results: Optimal RF Parameters.

Measure	Task	Outcome	Partitioning Fraction	0 Claim Fraction	Number of Models
Accuracy	Claims YES/NO	Best parameters	0.9	0.9	200
		Best model	0.9	0.9	200
	Claim ratio class	Best parameters	0.9	0.9	200
		Best model	0.9	0.9	200
Cohen's κ	Claims YES/NO	Best parameters	0.8	0.4	150
		Best model	0.9	0.3	200
	Claim ratio class	Best parameters	0.9	0.2	200
		Best model	0.9	0.2	200
R^2	Claim ratio	Best parameters	0.9	0.6	50
		Best model	0.9	0.8	200

Table A4. Results: Optimal NN Parameters.

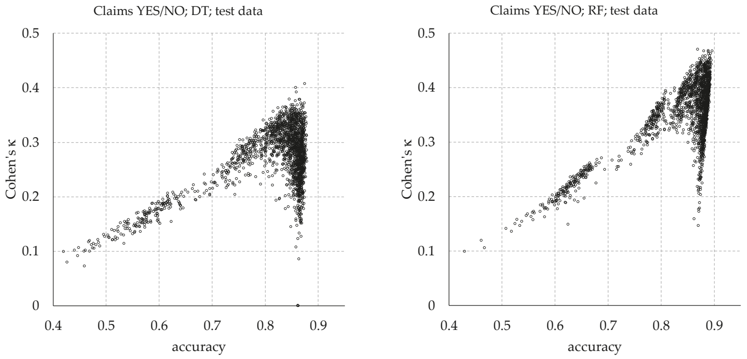
Measure	Task	Outcome	Partitioning Fraction	0 Claim Fraction	Layers	Neurons
Accuracy	Claims YES/NO	Best parameters	0.8	1.0	2	10
		Best model	0.9	0.9	2	10
Cohen's κ	Claims YES/NO	Best parameters	0.9	0.3	3	20
		Best model	0.9	0.4	2	20
R^2	Claim ratio	Best parameters	0.9	1.0	2	20
		Best model	0.9	1.0	2	20

Table A5. Results: Optimal PNN Parameters.

Measure	Task	Outcome	Partitioning Fraction	0 Claim Fraction	Theta Minus	Theta Plus
Accuracy	Claims YES/NO	Best parameters	0.9	1.0	0.30	0.65
		Best model	0.9	1.0	0.15	0.65
	Claim ratio class	Best parameters	0.9	1.0	0.30	0.55
		Best model	0.9	1.0	0.15	0.65
Cohen's κ	Claims YES/NO	Best parameters	0.9	0.4	0.20	0.60
		Best model	0.9	0.4	0.25	0.45
	Claim ratio class	Best parameters	0.9	0.2	0.30	0.40
		Best model	0.9	0.2	0.20	0.55

Appendix D. Accuracy—Cohen's κ Scatterplots

Figure A1 shows scatterplots to highlight the relationship between the performance measures "accuracy" and "Cohen's κ" for all investigated ML techniques and prediction tasks. The data results from applying all models developed during the parameter optimisation exercise to the test data.


Claims YES/NO; DT; test data

Claims YES/NO; RF; test data

Figure A1. *Cont.*

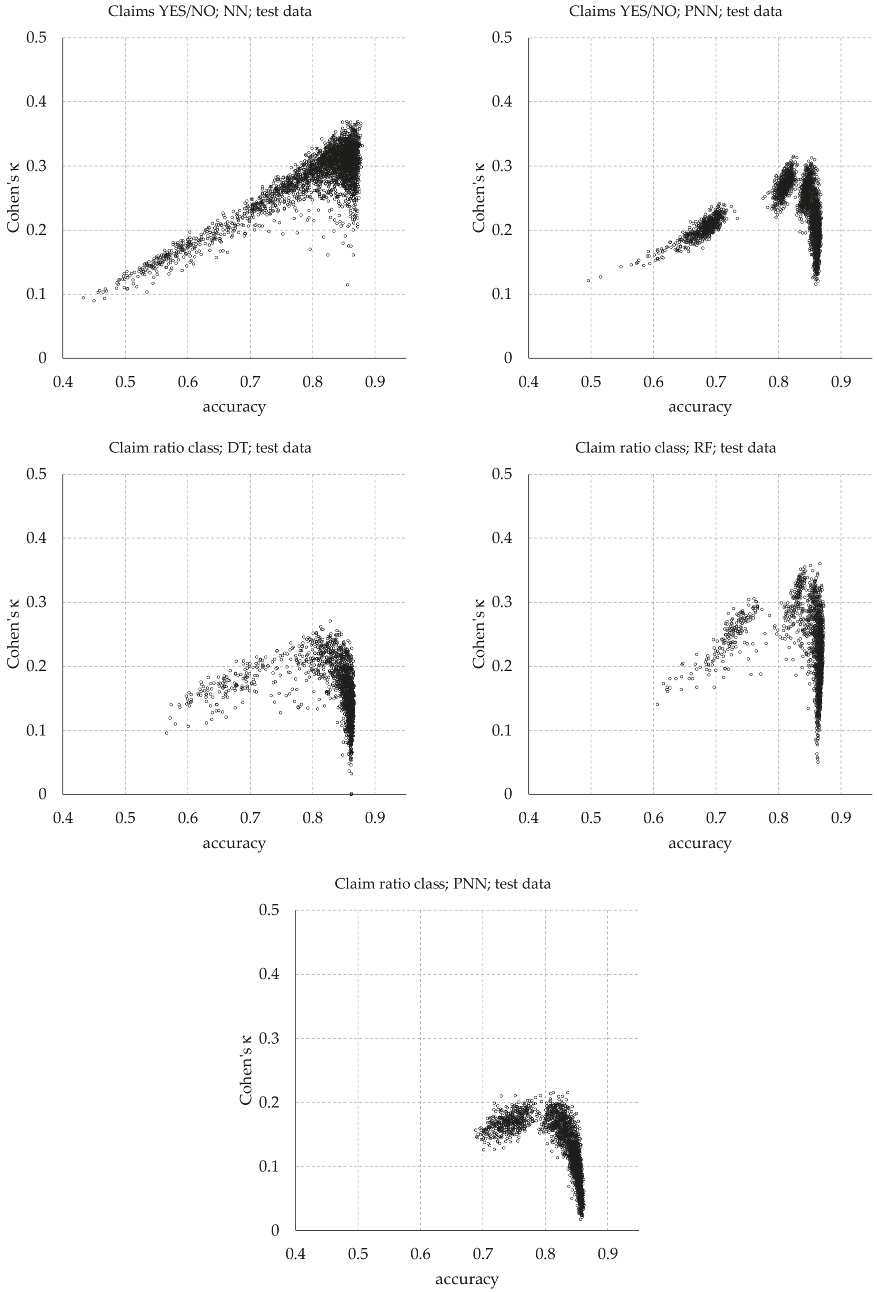


Figure A1. Scatterplots of Accuracy and Cohen's κ.

Appendix E. Boxplots on ML Technique Performance

The boxplots shown in Figure A2 illustrate the performance of all models developed during the parameter optimisation exercise of this study. The left side of the table shows performance variations measured during validation, mirrored on the right by the corresponding performance of the same models applied to the test data.

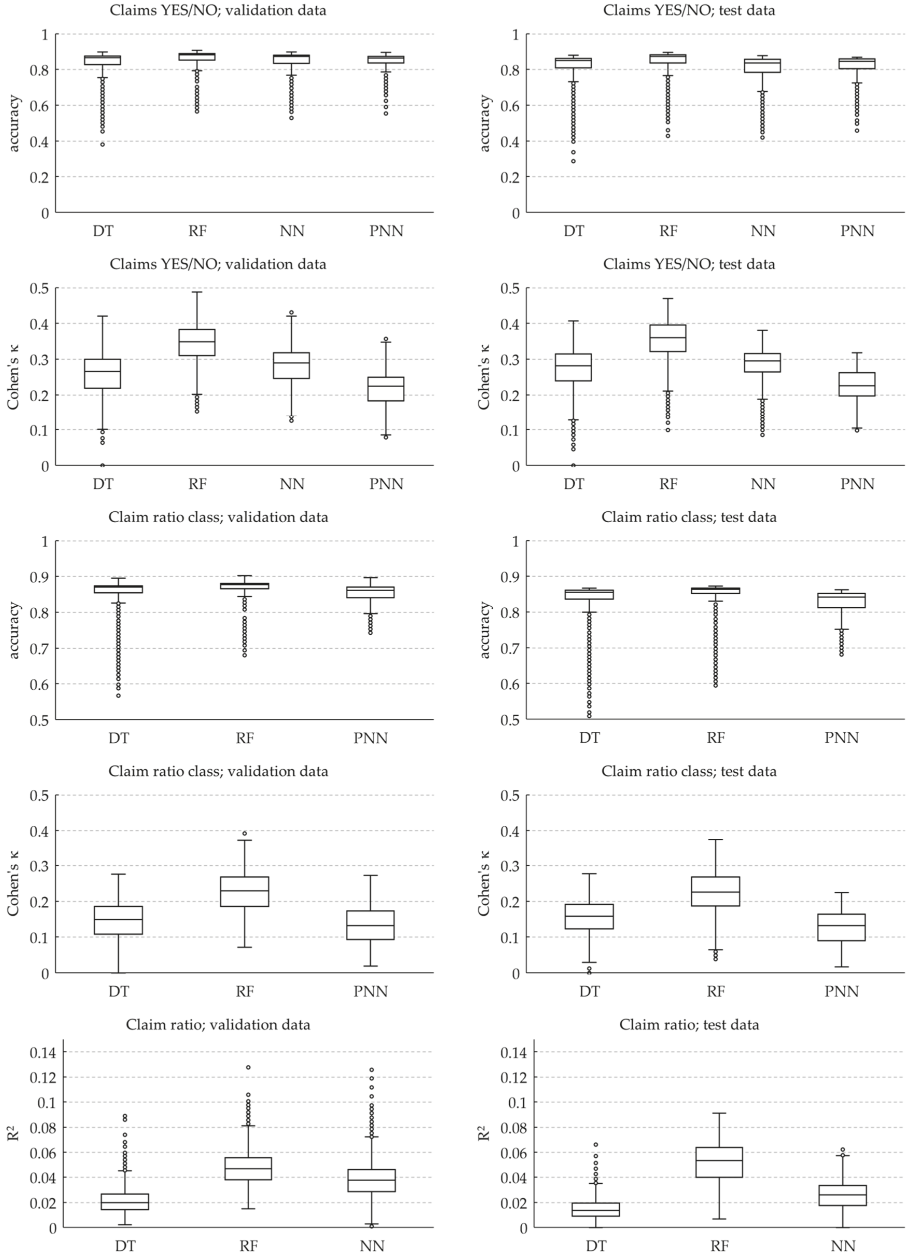


Figure A2. Boxplots: Comparison of ML techniques' performance (validation and test data).

References

Abraham, Filip, and Gerda Dewit. 2000. Export Promotion Via Official Export Insurance. *Open Economies Review* 1: 5–26. [CrossRef]

Athey, Susan. 2018. The impact of machine learning on economics. In *The Economics of Artificial Intelligence: An Agenda*. Chicago: University of Chicago Press, Available online: https://www.nber.org/chapters/c14009 (accessed on 26 February 2020).

Auboin, Marc. 2009. Restoring Trade Finance during a Period of Financial Crisis: Stock-Taking of Recent Initiatives. WTO Staff Working Paper ERSD-2009-16. Available online: https://www.wto-ilibrary.org/economic-research-and-trade-policy-analysis/restoring-trade-finance-during-a-period-of-financial-crisis_4de92d90-en (accessed on 16 July 2019).

Auboin, Marc, and Martina Engemann. 2014. Testing the trade credit and trade link: Evidence from data on export credit insurance. *Review of World Economics* 150: 715–43. [CrossRef]

Baudry, Maximillien, and Christian Y. Robert. 2019. A machine learning approach for individual claims reserving in insurance. *Applied Stochastic Models in Business and Industry* 35: 1127–55. [CrossRef]

Berne Union. 2019a. About the Berne Union. Available online: https://www.berneunion.org/Stub/Display/8 (accessed on 29 November 2019).

Berne Union. 2019b. Press Release 11/04/2019. Available online: http://cdn.berneunion.org/assets/Images/Berne%20Union%20Singapore%20SM%20Press%20Release.pdf (accessed on 29 November 2019).

Berne Union. 2019c. BU Spring Meeting Newsletter, Berne Union Statistics 2018 YE Commentary. Available online: http://cdn.berneunion.org/assets/Images/3923e9fd-215d-474e-80c9-6d10b984c302.zip (accessed on 29 November 2019).

Berne Union. 2019d. About Export Credit Insurance. Available online: https://www.berneunion.org/Stub/Display/17 (accessed on 23 December 2019).

Breiman, Leo. 2001. Random Forests. *Machine Learning* 45: 5–32. [CrossRef]

Charte, David, Francisco Charte, Salvador García, and Francisco Herrera. 2019. A snapshot on nonstandard supervised learning problems: Taxonomy, relationships, problem transformations and algorithm adaptations. *Progress in Artificial Intelligence* 8: 1–14. [CrossRef]

Claveria, Oscar, and Salvador Torra. 2014. Forecasting tourism demand to Catalonia: Neural networks vs. time series models. *Economic Modelling* 36: 220–28. [CrossRef]

Cohen, Jacob. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement* 20: 37–46. [CrossRef]

Drysdale, David. 2015. Why the OECD Arrangement Works (Even Though It Is Only Soft Law). In *The Future of Foreign Trade Support*. Edited by Andreas Klasen and Fiona Bannert. Durham: Wiley, pp. 5–7.

Egger, Peter, and Thomas Url. 2006. Public Export Credit Guarantees and Foreign Trade Structure: Evidence from Austria. *The World Economy* 29: 399–418. [CrossRef]

England, Peter D., and Richard J. Verrall. 2002. Stochastic claims reserving in general insurance. *Journal of the Institute of Actuaries* 129: 1–76. [CrossRef]

European Commission. 2012. *Study on Short-Term Trade Finance and Credit Insurance in the European Union*. Prepared by International Financial Consulting Ltd. for the European Commission. Available online: https://op.europa.eu/en/publication-detail/-/publication/a1ae8477-930c-44df-8c41-b38fb9ad94a4/language-en/format-PDF/source-112097066 (accessed on 27 January 2020).

Fang, Kuangnan, Yefei Jiang, and Malin Song. 2016. Customer profitability forecasting using Big Data analytics: A case study of the insurance industry. *Computers & Industrial Engineering* 101: 554–64. [CrossRef]

Fauzan, Muhammad A., and Hendri Murfi. 2018. The Accuracy of XGBoost for Insurance Claim Prediction. *International Journal of Advances in Soft Computing & Its Applications* 10: 159–71. Available online: http://home.ijasca.com/data/documents/11_IJASCA_The-accuracy-of-XGBoost_159-171.pdf (accessed on 27 February 2020).

Felbermayr, Gabriel J., and Erdal Yalcin. 2013. Export Credit Guarantees and Export Performance: An Empirical Analysis for Germany. *The World Economy* 36: 967–99. [CrossRef]

International Monetary Fund, International Labour Office, Organisation for Economic Co-Operation and Development, Statistical Office of the European Communities, United Nations, and World Bank. 2009. Export and Import Price Index Manual. Available online: https://www.imf.org/external/np/sta/xipim/pdf/xipim.pdf (accessed on 27 February 2020).

Iounousse, Jawad, Salah Er-Raki, Ahmed El Motassadeq, and Hassan Chehouani. 2015. Using an unsupervised approach of Probabilistic Neural Network (PNN) for land use classification from multitemporal satellite images. *Applied Soft Computing* 30: 1–13. [CrossRef]

Kiefer, Nicholas M. 2009. Default estimation for low-default portfolios. *Journal of Empirical Finance* 16: 164–73. [CrossRef]

KNIME. 2019. End to End Data Science. Available online: https://www.knime.com/ (accessed on 8 December 2019).

Korinek, Jane, Jean Le Cocguic, and Patricia Sourdin. 2010. The Availability and Cost of Short-Term Trade Finance and Its Impact on Trade OECD Trade Policy Papers No. 98. *OECD Trade Policy Papers*. Available online: https://www.oecd-ilibrary.org/trade/the-availability-and-cost-of-short-term-trade-finance-and-its-impact-on-trade_5kmdbg733c38-en (accessed on 27 February 2020).

Krummaker, Simone. 2020. Export Credit Insurance Markets and Demand. In *The Handbook of Global Trade Policy*. Edited by Andreas Klasen. Chichester: Wiley & Sons, pp. 536–54.

Kuhn, Max, and Kjell Johnson. 2013. *Applied Predictive Modeling*. Corrected at 5th Printing. New York, Heidelberg, Dordrecht and London: Springer, Available online: https://link.springer.com/content/pdf/10.1007/978-1-4614-6849-3.pdf (accessed on 26 February 2020).

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *Nature* 521: 436–44. [CrossRef] [PubMed]

Liaw, Andy, and Matthew Wiener. 2002. Classification and regression by random Forest. *R News* 2: 18–22.

Lorena, Ana C., Luis F. O. Jacintho, Marinez F. Siqueira, Renato de Giovanni, Lúcia G. Lohmann, André C.P.L.F. de Carvalho, and Missae Yamamoto. 2011. Comparing machine learning classifiers in potential distribution modelling. *Expert Systems with Applications* 38: 5268–75. [CrossRef]

Morel, Fabrice. 2011. Credit Insurance in Support of International Trade: Observations throughout the Crisis. In *Trade Finance during the Great Trade Collapse*. Edited by Jean-Pierre Chaffour and Mariem Malouche. Washington, DC: World Bank, pp. 337–56.

Moser, Christoph, Thorsten Nestmann, and Michael Wedow. 2008. Political Risk and Export Promotion: Evidence from Germany. *The World Economy* 31: 781–803. [CrossRef]

Mullainathan, Sendhil, and Jann Spiess. 2017. Machine Learning: An Applied Econometric Approach. *Journal of Economic Perspectives* 31: 87–106. [CrossRef]

NodePit. 2019. NodePit for KNIME. Available online: https://nodepit.com/nodepit-for-knime (accessed on 8 December 2019).

OECD. 2018. *Arrangement on Officially Supported Export Credits*. TAD/PG(2018)1. Available online: http://www.oecd.org/officialdocuments/publicdisplaydocumentpdf/?doclanguage=en&cote=tad/pg(2018)1 (accessed on 15 July 2019).

Olden, Julien D., and Donald A. Jackson. 2002. Illuminating the "black box": A randomization approach for understanding variable contributions in artificial neural networks. *Ecological Modelling* 154: 135–50. [CrossRef]

Razi, Muhammad A., and Kariakose Athappilli. 2005. A comparative predictive analysis of neural networks (NNs), nonlinear regression and classification and regression tree (CART) models. *Expert Systems with Applications* 29: 65–74. [CrossRef]

Rokach, Lior, and Oded Maimon. 2005. Top-Down Induction of Decision Trees Classifiers—A Survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)* 35: 476–87. [CrossRef]

Singh, Amanpreet, Narina Thakur, and Aakanksha Sharme. 2016. A Review of Supervised Machine Learning Algorithms. Paper presented at the 10th INDIACom; 2016 3rd International Conference on Computing for Sustainable Global Development, New Delhi, India, March 16–18; Edited by M. N. Hoda. Piscataway: IEEE, pp. 1310–15. Available online: https://ieeexplore.ieee.org/abstract/document/7724478 (accessed on 26 February 2020).

Specht, Donald F. 1990. Probabilistic neural networks. *Neural Networks* 3: 109–18. Available online: https://www.sciencedirect.com/science/article/pii/089360809090049Q (accessed on 26 February 2020). [CrossRef]

Stephens, Malcolm, and Diana Smallridge. 2002. A Study on the Activities of IFIs in the Area of Export Credit Insurance and Export Finance. INTAL-ITD-STA Occasional Paper 16. Inter-American Development Bank. Available online: https://publications.iadb.org/en/publication/study-activities-ifis-area-export-credit-insurance-and-export-finance (accessed on 26 February 2020).

Thesmar, David, David Sraer, Lisa Pinheiro, Nick Dadson, Razvan Veliche, and Paul Greenberg. 2019. Combining the Power of Artificial Intelligence with the Richness of Healthcare Claims Data: Opportunities and Challenges. *Pharmacoeconomics* 37: 745–52. [CrossRef] [PubMed]

van der Veer, Koen J. M. 2015. The Private Export Credit Insurance Effect on Trade. *Journal of Risk and Insurance* 82: 601–24. [CrossRef]

van der Veer, Koen J. M. 2019. Loss Shocks and the Quantity and Price of Private Export Credit Insurance: Evidence from a Global Insurance Group. *Journal of Risk and Insurance* 86: 73–102. [CrossRef]

Varian, Hal R. 2014. Big Data: New Tricks for Econometrics. *Journal of Economic Perspectives* 28: 3–28. [CrossRef]

Verall, Richard J., Ola Hossjer, and Susanna Bjorkwall. 2012. Modelling Claims Run-off with Reversible Jump Markov Chain Monte Carlo Methods. *ASTIN Bulletin* 42: 35–58.

Wanke, Peter, and Carlos P. Barros. 2016. Efficiency drivers in Brazilian insurance: A two-stage DEA meta frontier-data mining approach. *Economic Modelling* 53: 8–22. [CrossRef]

Weerasinghe, K. P. M. L. P., and M. C. Wijegunasekara. 2016. A Comparative Study of Data Mining Algorithms in the Prediction of Auto Insurance Claims. *European International Journal of Science and Technology* 5: 47–54. Available online: https://www.eijst.org.uk/images/frontImages/gallery/Vol._5_No._1/6._47-54.pdf (accessed on 26 February 2020).

Wüthrich, Mario V. 2018a. Machine learning in individual claims reserving. *Scandinavian Actuarial Journal* 6: 465–80. [CrossRef]

Wüthrich, Mario V. 2018b. Neural networks applied to chain–ladder reserving. *European Actuarial Journal* 8: 407–36. [CrossRef]

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).

risks

MDPI

Article

A Note on Combining Machine Learning with Statistical Modeling for Financial Data Analysis

José María Sarabia [1], Faustino Prieto [1], Vanesa Jordá [1] and Stefan Sperlich [2,*]

[1] Department of Economics, University of Cantabria, 39005 Santander, Spain; jose.sarabia@unican.es (J.M.S.); faustino.prieto@unican.es (F.P.); vanesa.jorda@unican.es (V.J.)

[2] Geneva School of Economics and Management, University of Geneva, 1211 Geneva, Switzerland

* Correspondence: stefan.sperlich@unige.ch; Tel.: +41-22-379-8223

Received: 19 December 2019; Accepted: 1 April 2020; Published: 3 April 2020

Abstract: This note revisits the ideas of the so-called semiparametric methods that we consider to be very useful when applying machine learning in insurance. To this aim, we first recall the main essence of semiparametrics like the mixing of global and local estimation and the combining of explicit modeling with purely data adaptive inference. Then, we discuss stepwise approaches with different ways of integrating machine learning. Furthermore, for the modeling of prior knowledge, we introduce classes of distribution families for financial data. The proposed procedures are illustrated with data on stock returns for five companies of the Spanish value-weighted index IBEX35.

Keywords: semiparametric modeling; machine learning; VaR estimation; analyzing financial data

JEL Classification: C14; C53; C58; G17; G22; C45

1. Introduction

The Editors of this Special Issue pointed out that machine learning (ML) has no unanimous definition. In fact, the term "ML", coined by Samuel (1959), is quite differently understood in the different communities. The general definition is that ML is concerned with the development of algorithms and techniques that allow computers to learn. The latter means a process of recognizing patterns in the data that are used to construct models; cf. "data mining" (Friedman 1998). These models are typically used for prediction. In this note, we speak about ML for data based prediction and/or estimation. In such a context, one may say that ML refers to algorithms that computer codes apply to perform estimation, prediction, or classification. As said, they rely on pattern recognition (Bishop 2006) for constructing purely data-driven models. The meaning of "model" is quite different here from what "model" or "modeling" means in the classic statistics literature; see Breiman (2001). He speaks of the data modeling culture (classic statistics) versus the algorithmic modeling coming from engineering and computer science. Many statisticians have been trying to reconcile these modeling paradigms; see Hastie et al. (2009). Even though the terminology comes from a different history and background, the outcome of this falls into the class of so-called semi-parametric methods; see Ruppert et al. (2003) or Härdle et al. (2004) for general reviews. According to that logic, ML would be a non-parametric estimation, whereas the explicit parametrization forms the modeling part from classic statistics.

Why should this be of interest? The Editors of this Special Issue also urge practitioners not to ignore what has already been learned about financial data when using presumably fully automatic ML methods. Regarding financial data for example, Buch-Larsen et al. (2005), Bolancé et al. (2012), Scholz et al. (2015, 2016), and Kyriakou et al. (2019) (among others) have shown the significant gains in estimation and prediction when including prior knowledge in nonparametric prediction. The first two showed how knowledge-driven data transformation improves nonparametric estimation of distribution and operational risk; the third paper used parametrically-guided ML for stock return prediction; the fourth imputed bond returns to improve stock return predictions; and the last proposed comparing different theory-driven benchmark models regarding their predictability. Grammig et al. (2020) combined the opposing modeling philosophies to predict stock risk premia. In this spirit, we will discuss the combination of purely data-driven methods with smart modeling, i.e., using prior knowledge. This will be exemplified along the analysis of the distributions (conditional and unconditional) of daily stock returns and calculations of their value-at-risk (VaR).

Notice finally that in the case of estimation, methods are desirable that permit practitioners to understand, maybe not perfectly, but quite well, what the method is doing to the data. This will facilitate the interpretation of results and any further inference. Admittedly, this is not always necessary and certainly also depends on the knowledge or imagination of the user. Yet, we believe it is often preferable to analyze data in a glass box than in a black box. This aspect is respected in our considerations.

Section 2 revisits the ideas of semiparametric statistics. Section 3 provides an intensive treatment of the distribution modeling followed by its combination with local smoothing. In Section 4, we give empirical illustrations. Section 5 concludes. In the Appendix are given additional details.

2. Preliminary Considerations and General Ideas

It is helpful to first distinguish between global and local estimation. Global means that the parameter or function applies at any point and to the whole sample. Local estimation applies only to a given neighborhood, like for kernel regression. It is clear that localizing renders a method much more flexible; however, the global part allows for an easy modeling, and its estimation can draw on the entire sample. Non-parametric estimators are not local by nature; for example, power series based estimators are not. Unless you want to estimate a function with discontinuities, local estimators are usually smoothing methods; see Härdle et al. (2004). This distinction holds also for complex methods (cf. the discussion about extensions of neural networks to those that recognize local features), which often turn out to be related to weighted nearest neighbor methods; see Lin and Jeon (2006) for random forests or Silverman (1984) for splines. The latter is already a situation where we face a mixture of global and local smoothing; another one is orthogonal wavelet series (Härdle et al. 1998).

Those mixtures are interesting because the global parts can borrow the strength from a larger sample and have a smoothing effect, while the local parts allow for the desired flexibility to detect local features. Power series can offer this only by including a (in practice unacceptable) huge number of parameters. This is actually a major problem of many complex methods, but mixtures allow substantially reducing this number. At the same time, they allow us to include prior knowledge about general features. For example, imposing shape restrictions is much simpler for mixtures (like splines) than it is for purely local smoothers; see Meyer (2008) and the references therein. Unless the number of parameters is pre-fixed, their selection happens via reduction through regularization, which can be implemented in many ways. Penalization methods like P-splines (Eilers et al. 2015) or LASSO (Tibshirani 1996) are popular. The corresponding problem for kernel, nearest neighbors, and related methods is the choice of the neighborhood size. In any case, one has to decide about the penalization criterion and a tuning parameter. The latter is until today an open question; presently, cross-validation-type methods are the most popular ones. For kernel based methods, see Heidenreich et al. (2013) and Köhler et al. (2014) for a review or Nielsen and Sperlich (2003) in the context of forecasting in finance.

The first question concerns the kind of prior information available, e.g., whether it is about the set of covariates, how they enter (linearly, additively, with interactions), the shape (skewness, monotonicity, number of modes, fat tails), or more generally, about smoothness. This is immediately followed by the question of how this can be included; in some cases, this is obvious (like if knowing the set of variables to be included); in some others, it is more involved (like including parameter information via Bayesian modeling). Knowledge about smoothness is typically supposed in order to justify a particular estimator and/or the selection method for the smoothing parameter. Information about the shape or how covariates enter the model comprises the typical ingredients of semiparametric modeling (Horowitz 1998) to improve nonparametric estimation (Glad 1998).

Consider the problem of estimating a distribution, starting with the unconditional case. In many situations, you are more interested in those regions for which data are hardly available. If you used then a standard local density estimator, you would try to estimate interesting parameters like VaR from only very few observations, maybe one to five, which is obviously not a good idea. Buch-Larsen et al. (2005) proposed to apply a parametric transformation using prior knowledge. Combining this way a local (kernel density) estimator with such a global one, however, allowed them to borrow strength from the model and from data that were further away. Similarly, consider conditional distributions. Locally, around a given value of the conditioning variable, you may have too few observations to estimate a distribution nonparametrically. Then, you may impose on this neighborhood the same probability law up to some moments, as we will do in our example below.[1]

Certainly, a good mixture of global and local fitting is problem-adapted. Then, the question falls into two parts: which is the appropriate parametric modeling, and how to integrate it with the flexible local estimator. For the former, you have to resort to expertise in the particular field. For the latter, we will discuss some popular approaches. All this will be exemplified with the challenges of modeling stock returns for five big Spanish companies and predicting their VaR.

In our example, the first step is to construct a parametric guide for the distribution of stock returns Y. To this aim, we introduce the class of generalized beta-generated (BG) distributions (going back, among others, to Eugene et al. (2002) and Jones (2004)), as this distribution class allows modeling skewed distributions with potentially long or fat tails. While this is not a completely new approach, we present it with an explicit focus on the above outlined objectives including the calculation of VaRs and combining it with nonparametric estimation and/or validation. Our validation is more related to model selection and testing, today well understood and established, and therefore kept short. The former, i.e., the combination with nonparametric estimation, is discussed for the problem of analyzing conditional distributions and can be extended to the combination with methods for estimating in high dimensions. For this example, in which the prior knowledge enters via a distribution class, we discuss two approaches: one is based on the method of moments, the other one on maximum likelihood. The latter is popular due to Rigby and Stasinopoulos (2005) and Severini and Staniswalis (1994). Rigby and Stasinopoulos (2005) considered a fully parametrized model in which each distribution parameter (potentially transformed with a known link) is written as an additive function of covariates, typically including a series of random effects. They proposed a backfitting algorithm (implemented in the R library GAMLSS) to maximize a penalized likelihood corresponding to a posterior mode estimation using empirical Bayesian arguments. Severini and Staniswalis (1994) started out with the parametric likelihood, but localized it by kernels. This is maximized then for some given values of the covariates.

3. A Practical Example

We now discuss the technical steps for the announced practical example. While this section focuses on the technical part, the empirical exercise is done in the next section.

3.1. Distribution Modeling

Often, the Student t distribution was used in financial econometrics and risk management to model the conditional asset returns, going back to Bollerslev (1987). However, it is well known that it does not describe very well the empirical features of most financial data. Therefore, several proposals have been made of skewed Student t distributions; see Theodossiou (1998) and Zhu and Galbraith (2010) for the

[1] Further advantages are that semiparametric modeling can help to overcome the curse of dimensionality and that semiparametric models are more robust to the choice of smoothing parameters.

context of finance or Jones and Faddy (2003) and Azzalini and Capitanio (2003) in (applied) statistics. For a more general discussion and compendium, see Rigby et al. (2019).

Let us consider two classes of skewed t distributions. Both are derived from the (generalized) BG classes; see Appendix A. These allow for the generation of many flexible distribution classes. The fist version depends on two parameters, whereas the second one depends on three. One may directly go for the second one. Remember, however, later on, that this is just a parametric guide for the semiparametric estimator. There is certainly a trade-off between the gain of flexibility, on the one hand, and the loss of its regularization, on the other hand. Moreover, each additional parameter in the global part may raise the costs of implementation and computation to an unacceptable degree. Both skewed t distributions are generated by taking a standard Student t as the baseline distribution in (A1) and (A2), respectively. Specifically, plugging in $F_Y(y;a,b) = \frac{1}{2}\left(1 + y/\sqrt{a+b+y^2}\right)$, a so-called scaled Student t_2, into (A1), gives our skewed t of Type 1 with the probability density function (pdf):

$$g_{T_1}(y;a,b) = k_1\left(1 + \frac{y}{\sqrt{a+b+y^2}}\right)^{a+1/2}\left(1 - \frac{y}{\sqrt{a+b+y^2}}\right)^{b+1/2}, \tag{1}$$

where $k_1^{-1} = B(a,b)\sqrt{a+b}2^{a+b-1}$ and $y \in \mathbb{R}$. We write $Y \sim T_1(a,b)$; see Jones and Faddy (2003). Plugging the baseline cumulative density (cdf) into (A2) gives our skewed t of Type 2 with the pdf:

$$g_{T_2}(y;a,b,c) = k_2\frac{1}{(a+b+y^2)^{3/2}}\left(1 + \frac{y}{\sqrt{a+b+y^2}}\right)^{ac-1}\left(1 - \frac{1}{2^c}\left(1 + \frac{y}{\sqrt{a+b+y^2}}\right)^c\right)^{b-1}, \tag{2}$$

where $k_2 = \frac{c(a+b)}{B(a,b)2^{ac}}$ and $y \in \mathbb{R}$. We write $Y \sim T_2(a,b,c)$; see Alexander and Sarabia (2010) and Alexander et al. (2012). The densities g_{T_1}, g_{T_2} are illustrated in Figure 1.

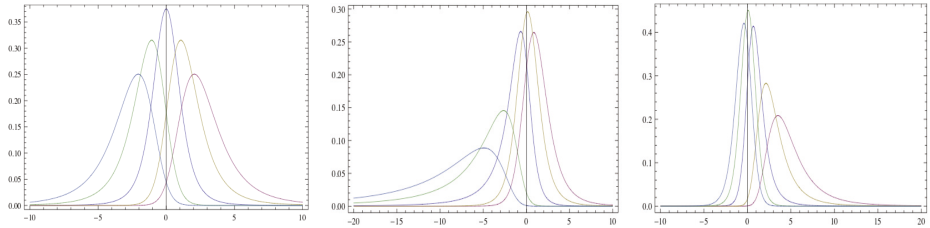


Figure 1. Graphics of the probability density function: left for the skewed $t1$ (1) when $(a,b) =$ (2,2), (5,2), (8,2), (2,5), and (2,8); center for the skewed $t2$ (2) with $(a,b,c) =$ (2,2,0.5), (8,2,0.5), (5,2,0.5), (2,5,0.5), and (2,8,0.5); and on the right, (2,2,2), (8,2,2), (5,2,2), (2,5,2), and (2,8,2).

For implementation, estimation, model selection, and in particular, for the possible integration of ML, we take a closer look at the basic properties of these two distributions. Note that for $a = b$ in (1), one obtains the classical Student t distribution with $2a$ degrees of freedom. The same is true for the three-parameter case (2) when setting $a = b$ with $c = 1$. Furthermore, the cdfs are given by:

$$G_{T_1}(y;a,b) = I\left(\frac{1}{2}\left\{1 + \frac{y}{\sqrt{a+b+y^2}}\right\};a,b\right), \tag{3}$$

and:

$$G_{T_2}(y;a,b,c) = I(F_Y^c(y;a,b);a,b), \tag{4}$$

respectively. Having explicit expressions for the pdf, one may think that likelihood based methods are straight-forward, including those for tests and model selection. Note, however, that for a sample of size m, you need to know the joint distribution of $(y_1, \ldots, y_m)$, which are not independent. In practice, many work with $\log \prod_{i=1}^{m} g_{T_k}(y_i;\theta)$, ($k = 1, 2$ with $\theta = (a, b)$ for $k = 1$, $\theta = (a, b, c)$ else) as a likelihood approximate, ignoring potential dependencies. This weakens both the efficiency of estimation and the validity of likelihood based inference. Therefore, it is interesting to look at alternatives. For estimation, recall the moment based approach; then, we need to to express parameters a, b, and if applicable c, in terms of estimable moments. For the case of the skewed t distribution of the first kind, we have:

$$E(Y^r) = \frac{(a+b)^{r/2}}{B(a,b)} \sum_{j=0}^{r} \binom{r}{j} 2^{-j}(-1)^j B\left(a - \frac{r}{2} - j, b - \frac{r}{2}\right) \tag{5}$$

if $a, b > r/2$. An advantage of this expression is that we do not need to estimate the centered, standardized moments. This is even more an advantage when we try to estimate these moments nonparametrically. Similarly, for the skewed t distribution of the second kind:

$$E(Y^r) = \frac{(a+b)^{r/2}}{B(a,b)} \sum_{j=0}^{r} (-1)^j \binom{r}{j} 2^{-j} \sum_{k=0}^{\infty} \binom{-r/2}{k} (-1)^k B\left(a - \frac{r/2+j-k}{c}, b\right) . \tag{6}$$

Clearly, this reveals a problem because we cannot solve this (easily) for a, b, and c.

3.2. Financial Risk Measures

The parametric part provides us with explicit formulas that we can use to derive closed expressions for the risk measures, namely the VaR and tail moments. In fact, it can be shown that:

$$\begin{aligned} \mathrm{VaR}_{T_1}[p; a, b] &= \frac{\sqrt{a+b}(2\mathrm{VaR}_B[p; a, b] - 1)}{2\sqrt{\mathrm{VaR}_B[p; a, b](1 - \mathrm{VaR}_B[p; a, b])}}, \quad \text{and} \quad (7) \\ \mathrm{VaR}_{T_2}[p; a, b, c] &= \frac{\sqrt{a+b}(2\mathrm{VaR}_B^{1/c}[p; a, b] - 1)}{2\sqrt{\mathrm{VaR}_B^{1/c}[p; a, b](1 - \mathrm{VaR}_B^{1/c}[p; a, b])}}, \quad (8) \end{aligned}$$

with $0 \le p \le 1$, where $\mathrm{VaR}_B[p; a, b]$ denotes the VaR of a classical $\mathcal{B}e(a, b)$ distribution. Furthermore, if $Y \sim T_1(a, b)$, then the for a given y_p, its corresponding tail moments can we written as:

$$\begin{aligned} E\{Y^k | Y \le y_p\} &= E\left\{ \frac{(a+b)^{k/2}(2B-1)^k}{2^k B^{k/2}(1-B)^{k/2}} \mid \frac{\sqrt{(a+b)}(2B-1)}{2\sqrt{B(1-B)}} \le y_p \right\} \\ &= \frac{(a+b)^{k/2}}{2^k} \sum_{j=0}^{k} a_j E\{B^{k/2-j}(1-B)^{k/2} \mid B \le h(x_p)\}, \end{aligned}$$

where $a_j = (-1)^j \binom{k}{j} 2^{k-j}$ and $h(z) = \{a + b + z^2 + \sqrt{(a+b)z^2 + z^4}\}/\{2(a + b + z^2)\}$, which is increasing in z, with $a, b \in \mathbb{R}^+$. For $k = 1$, we get the T(ail)VaR. If $B \sim \mathcal{B}e(a, b)$, then one has:

$$E\{B^{k/2-j}(1-B)^{k/2} \mid B \le h(x_p)\} = \frac{1 - F_{\tilde{B}}(h(y_p))}{F_B(h(y_p))} \frac{B(k/2 + a - j, b - k/2)}{B(a,b)}, \tag{9}$$

where $\tilde{B} \sim \mathcal{B}e(k/2 + a - j, b - k/2)$, $b > k/2$, and $h(z)$ as above. In sum, for the Type 1 class, we obtain explicit expressions for the tail moments that do not change when (a, b) are functions of covariates.

3.3. Combining The Prior with Nonparametric Estimation

First, let us briefly consider the case of being only interested in the estimation of the unconditional distribution $g(\cdot)$, say, in that of stock returns Y. Once a proper parametric choice (say G_θ) is found and the parameters θ are estimated, we know that $\tilde{Y} := G_{\hat{\theta}}(Y)$ has a pretty smooth density $\tilde{g}$, which is not too far from the uniform $[0, 1]$ distribution. Then, as $g(y) = \tilde{g}\{G_{\hat{\theta}}(y)\}$, you obtain the final estimate.

A next step to proceed could be to apply a nonparametric estimator like, for example, a kernel density for $\tilde{g}$. The prior estimation served here to stretch data where we had many observations, and contract them where we had only a few. You may say that this could also be done either by taking the empirical cdf for transformation or by taking the local bandwidth that always includes the same number of neighbors. This would be purely nonparametric and therefore renounce the use of prior information. In practice, these alternatives suffer from various problems like giving wiggly results, a much harder bandwidth choice, etc. For details and applications in actuarial sciences, consult Bolancé et al. (2012) and Martínez Miranda et al. (2009).

We turn now to the slightly more challenging problem of considering conditional distributions. For example, what is the stock return distribution of Telefónica and its VaR given a certain value of the Spanish IBEX35? Having huge datasets, you may try to estimate this with a fully nonparametric estimator. However, if you doubt that stationarity holds over a long period, you may want to restrict your dataset to not include more than twelve months (as an example). In such a case, you better resort to a parametric guide like above and turn a, b (and c if applicable) into flexible functions of the conditioning covariate X. Again, a likelihood based approach may now look at $\log prod_{i=1}^{m} g_{T_k}(y_i; \theta(x_i))$ ($k = 1, 2$ with $\theta(x_i) = (a(x_i), b(x_i))$ for $k = 1$, etc.) ignoring potential dependencies. Rigby and Stasinopoulos (2005) specified the elements of θ as additive, parametrized functions of x_i with some random coefficients, and maximized a penalized version of this. In contrast, Severini and Staniswalis (1994) did not parametrize the elements of θ, but maximized (along θ) the smoothed version, i.e., $sum_{i=1}^{m} \log g_{T_k}(y_i; \theta(x)) K_h(x - x_i)$ for any x. Here, $K_h(v) = h^{-1} K(v/h)$ for a kernel function $K(\cdot)$ with bandwidth h. That is, they obtained for given x an estimate of the value that θ takes at x. Notice that the latter method is usually implemented for θ containing only one element (typically the mean).

As said before, an interesting alternative is to estimate nonparametrically the moments of Y and then derive the elements of θ. Thanks to Formula (5), we can do this for the Type 1 skewed t distributions. More specifically, the algorithm would look as follows: For sample (x_i, y_i), $i = 1, 2, \ldots, m$ with y_i being the return of the stock and x_i the conditioning variable (in our application, the market return) conduct:

Step 1: Estimate the conditional first two moments μ_1 and μ_2 by:

$$\mu_1(x_i) = E(y|x_i), \quad \text{and} \quad \log \mu_2(x_i) = E(y^2|x_i) \, ,$$

where the nonparametric functions can be estimated, e.g., by kernel regression, splines, etc.

Step 2: You now may either take a grid over the range of X, with M grid points x_j, or you may calculate the estimates for each observation x_i. Let us call them $\hat{\mu}_{1j}$ and $\hat{\mu}_{2j}$, $j = 1, 2, \ldots, M$ for either case.

Step 3: Calculate estimates $(\hat{a}_j, \hat{b}_j)$ by solving in (a_j, b_j), $j = 1, 2, \ldots, M$, the non-linear system:

$$\hat{\mu}_{1j} = \frac{(a_j - b_j)\sqrt{a_j + b_j}}{2} \cdot \frac{\Gamma(a_j - \frac{1}{2})\Gamma(b_j - \frac{1}{2})}{\Gamma(a_j)\Gamma(b_j)}, \tag{10}$$

$$\hat{\mu}_{2j} = \frac{a_j + b_j}{4} \cdot \frac{(a_j - b_j)^2 + a_j - 1 + b_j - 1}{(a_j - 1)(b_j - 1)}. \tag{11}$$

Step 4: Then, the estimate of the conditional distribution is of the form:

$$\hat{g}(y|x_j) = g_F(y|\hat{a}_j, \hat{b}_j) \, .$$

Following Nielsen and Sperlich (2003), Kyriakou et al. (2019), and Mammen et al. (2019), you could use local linear regression in Step 1 for both functions, combined with the validated R^2 for the bandwidth choice. Obviously, any method known as ML, including LASSO variable selection, can be applied in this step. However, it is less evident how these methods could be combined with the aforementioned likelihood based approaches. Note further that the conditional distribution obtained by this strategy can also be used for semiparametric prediction of the unconditional distribution:

$$\hat{g}(y) = \frac{1}{m}\sum_{i=1}^{m} g_F(y|\hat{a}_i, \hat{b}_i) \, . \tag{12}$$

This shows that with $\hat{\mu}_1$, $\hat{\mu}_2$, you can predict the marginal distribution of Y for scenarios in which the distribution of X is changing (Dai et al. 2016). For example, we could predict the unconditional distribution of stocks for different distributions of the IBEX35. Note finally that Step 3 cannot easily be applied to the skewed t distribution of Type 2; recall (6). In such a case, you could only try to apply the idea of Rigby and Stasinopoulos (2005), but with procedures and algorithms that are still to be developed.

4. Empirical Illustration

Consider daily stock returns from 1 January 2015 to 31.12.2015 for five companies of the Spanish value-weighted index IBEX35; namely Amadeus (IT solutions for tourist industry), BBVA (global financial services), Mapfre (insurance market), Repsol (energy sector), and Telefónica (information and communications technology services); see Table 1. The returns are negatively skewed in four of the five companies considered, but positively skewed in the last one.

Table 1. Summary statistics. The sample size is $m = 261$ for all sets.

Stocks	Amadeus	BBVA	Mapfre	Repsol	Telefónica
Maximum daily return	0.046286	0.040975	0.050847	0.073466	0.062264
Minimum daily return	−0.097367	−0.060703	−0.067901	−0.0877323	−0.051563
Mean	0.000900	−0.000452	−0.000623	−0.001416	−0.000408
Standard deviation	0.014601	0.016249	0.015942	0.021349	0.016301
Skewness	−1.163797	−0.465779	−0.723655	−0.166165	0.130372
Kurtosis	10.292160	3.824688	4.873980	5.435928	4.422885

We first fit the data by the maximum likelihood (ignoring dependence) to both distribution classes, working with standardized data. Tables 2 and 3 show the parameter estimates with standard errors.

Table 2. Maximum likelihood estimates for the skewed t model of Type 1, standardized data. Standard errors are in parenthesis.

Stocks	Amadeus	BBVA	Mapfre	Repsol	Telefónica
$\hat{a}$	6.194309	10.773980	7.271484	5.009976	7.083988
	(2.378890)	(8.474473)	(3.684818)	(1.980271)	(3.810294)
$\hat{b}$	6.171897	10.76088	7.250156	5.005015	7.086958
	(2.378415)	(8.477441)	(3.686769)	(1.980433)	(3.810390)

Table 3. Maximum likelihood estimates for the skewed t model of Type 2, standardized data. Standard errors are in parenthesis.

Stocks	Amadeus	BBVA	Mapfre	Repsol	Telefónica
$\hat{a}$	1.050617	0.935678	0.8685684	0.808572	1.120804
	(0.443072)	(0.407211)	(0.329048)	(0.363157)	(0.650834)
$\hat{b}$	5.126098	7.144007	6.217545	2.998354	3.497796
	(2.091549)	(5.104088)	(3.184219)	(0.917090)	(1.110353)
$\hat{c}$	2.973896	3.653026	3.617017	2.721519	2.331579
	(0.761879)	(0.733523)	(0.668503)	(0.698227)	(0.774010)

To choose between these two models, one may use the Bayesian information criterion (BIC). However, as our (working) likelihood neglects the dependence structure, it might not be reliable. While the three-parameter model presents the largest values for BIC (not shown), the gain, however, is always close to, or smaller than, 1%.

An alternative is to apply ML methods comparing the parametric estimates with purely nonparametric ones. This is not recommendable any longer when switching to conditional distributions due to the moderate sample sizes. In Figure 2, you see how our models (in red) adapted to the empirical cdf (blue) for the stocks of Amadeus and BBVA. As expected, the three-parameter model gave slightly better fits. In practice, the interesting thing to see is where improvements occurred, if any. The practitioner has to judge then what is of interest for his/her problem; ML cannot do this for him/her. However, ML can offer specification tests; see Gonzales-Manteiga and Crujeiras (2013) for a review. For example, a test that formalizes our graphical analysis is the Kolmogorov–Smirnov (KS) test:

$$KS = \sup |F_m(y_i) - F(y_i;\hat{\theta})|, \; i = 1, 2, \ldots, m,$$

where $F_m(y_i)$ is the empirical cdf and $F(y;\hat{\theta})$ is the cdf of the particular model class with $\hat{\theta}$ from Tables 2 and 3. To calculate the p-value, we can use the parametric bootstrap:

Step 1: For the observed sample, find the maximum likelihood estimator, $F(y;\widehat{\boldsymbol{\theta}})$, $\hat{F}_m(y)$, and KS.

Step 2: Generate J bootstrap samples $y_1^{(j)}, \ldots, y_m^{(j)} \sim F(y;\widehat{\boldsymbol{\theta}})$ under H_0 (the data follow model F); fit them; and compute $F(y;\widehat{\boldsymbol{\theta}}^{(j)})$, $\hat{F}_m^{(j)}(y)$, and $KS^{(i)}$ for each bootstrap sample $j = 1, 2, \ldots, J$.

Step 3: Calculate the p-value as the fraction of synthetic bootstrap samples with a KS statistic greater than the empirical KS statistic obtained from the original data.

To obtain an approximate accuracy of the p-value for ϵ =0.01, we generated $J = \frac{1}{4}\epsilon^{-2} = 2500$ bootstrap samples. Table 4 shows the results for both distribution classes and all datasets. It can be seen that with all p-values larger than 0.499, both models could not be rejected at any reasonable significance level in any of the considered datasets.

Finally, let us see how different the VaR are, when calculated on the base of one model compared to the other; recall Equations (7) and (8). They were calculated at the 95% confidence level for all datasets; see Table 5. The T_1 model with two parameters provided slightly higher VaR values than the T_2 model. The difference again seemed to be somewhat marginal, except maybe for Amadeus.

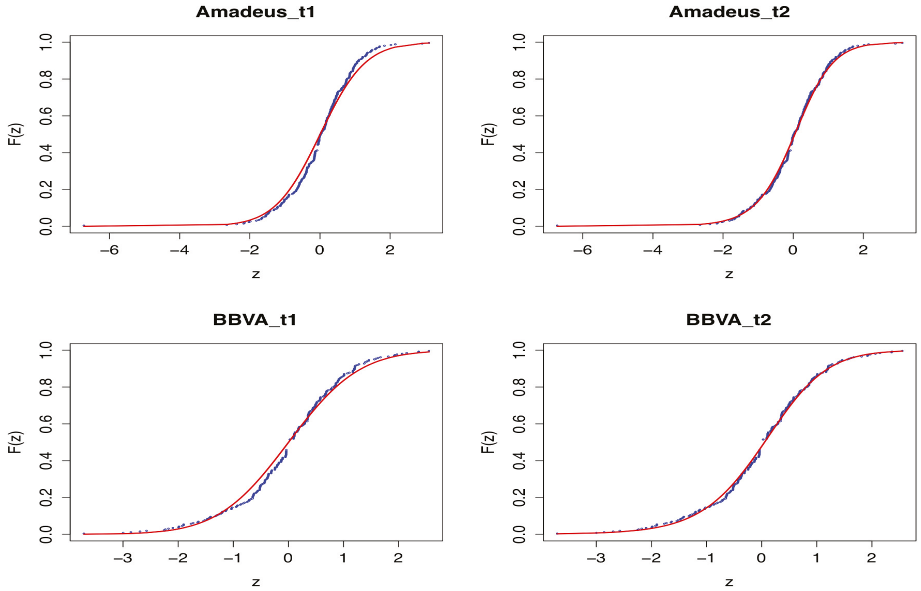


Figure 2. Plots of the theoretical cdfs of the skewed t models (LEFT: T_1 model; RIGHT: T_2 model) and the empirical cdf. Stocks: Amadeus; BBVA.

Table 4. Bootstrap p-values for both models and all five datasets.

Stocks	Amadeus	BBVA	Mapfre	Repsol	Telefónica
Skewed t T_1	0.593	0.676	0.499	0.761	0.829
Skewed t T_2	0.732	0.908	0.733	0.732	0.915

Table 5. Values at risk $\mathrm{VaR}_{T1}[0.05; a, b)]$ and $\mathrm{VaR}_{T2}[0.05; a, b, c]$ for the five stocks considered.

Stocks	Amadeus	BBVA	Mapfre	Repsol	Telefónica
VaR_{T1}	−0.024941	−0.028328	−0.028521	−0.040059	−0.029110
VaR_{T2}	−0.023089	−.02817	−0.027794	−0.038330	−0.028029

Let us turn to the estimation of the conditional distribution. Here, the integration of the ML happens by incorporating the covariates nonparametrically. For the sake of presentation and brevity, we restricted ourselves here to the moment based approach; for more details on the likelihood based one, we refer (besides the above cited literature) to the recent compendium of Rigby et al. (2019) and the references therein. Limiting ourselves to the moment based method automatically limited us to the skewed $t1$ class; recall Equation (6).[2] Furthermore, for the sake of illustration, we limited the exercise to the estimation of all distribution parameters as nonparametric functions of one given covariate X, namely the IBEX35. It was obvious that estimates for μ_1, μ_2 could equally well be the result of a complex multivariate regression or a variable selection procedure like LASSO. We estimated μ_1(IBEX35) and μ_2(IBEX35) using different methods provided by standard software; the presented results were obtained from penalized (cubic) spline regression with data-driven penalization. For details,

[2] You may develop numerical approximations working with (6), but this is clearly beyond the scope of this note. However, the above studies insinuate that the gain by using the more complex Type 2 class is rather marginal. Those advantages get easily compensated by the local estimator.

consult Ruppert et al. (2003) and the SemiPar project. Figure 3 gives an example of how this performed for the BBVA stocks.

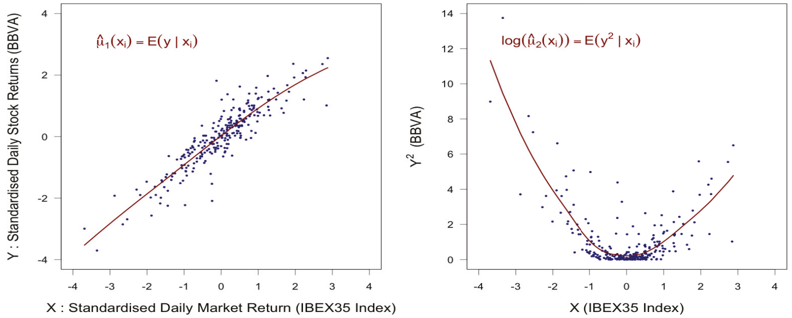


Figure 3. Estimates $\hat{\mu_1}$, $\widehat{\log \mu_2}$ for BBVA stock returns as functions of IBEX35.

Table 6 gives for the IBEX35 quantiles $Q_1 = -0.00768$, $Median = Q_2 = 0.00079$, $Q_3 = 0.00687$ the corresponding moment estimates of the different stock returns; Table 7 the corresponding (a_j, b_j) for Formulas (10) and (11). Table 8 gives the corresponding conditional VaRs obtained from Formula (7).

Table 6. First two moments of stock returns for given IBEX35 values (looking at its quantiles).

IBEX35	Amadeus		BBVA		Mapfre		Repsol		Telefónica	
Quartile	$\hat{\mu}_{1j}$	$\hat{\mu}_{2j}$	$\hat{\mu}_{1j}$	$\hat{\mu}_{2j}$	$\hat{\mu}_{1j}$	$\hat{\mu}_{2j}$	$\hat{\mu}_{2j}$	$\hat{\mu}_{1j}$	$\hat{\mu}_{2j}$	$\hat{\mu}_{1j}$
Q_1	−0.323500	2.336970	−0.493668	1.489340	−0.481228	2.215213	−0.430198	1.654110	−0.509551	1.941169
Q_2	0.025464	2.385443	0.094427	1.240531	0.060678	1.487417	0.056462	1.474321	0.034497	1.253373
Q_3	0.280358	2.492818	0.503731	1.523203	0.439138	1.623399	0.405799	1.698132	0.433284	1.421627

Table 7. Parameter (a_j, b_j) of the conditional stock return distributions for given IBEX35 values.

IBEX35	Amadeus		BBVA		Mapfre		Repsol		Telefónica	
Quartile	$\hat{a}_j$	$\hat{b}_j$	$\hat{a}_j$	$\hat{b}_j$	$\hat{a}_j$	$\hat{b}_j$	$\hat{a}_j$	$\hat{b}_j$	$\hat{a}_j$	$\hat{b}_j$
Q_1	1.742906	2.136783	5.397247	6.904428	1.990996	2.690554	3.143186	4.048275	2.490487	3.398563
Q_2	1.736629	1.709150	5.492494	5.226327	3.133551	3.019128	3.184576	3.076590	5.016808	4.924298
Q_3	1.953575	1.639172	6.478997	5.008818	4.327494	3.356619	3.640170	2.851049	6.809153	5.483973

Table 8. The conditional value at risk $VaR_{T1}(IBEX35)$ for given IBEX35 values.

IBEX35	Amadeus	BBVA	Mapfre	Repsol	Telefónica
Q_1	−0.037728	−0.038910	−0.044802	−0.053801	−0.043939
Q_2	−0.031199	−0.027981	−0.030278	−0.041117	−0.029327
Q_3	−0.026029	−0.020886	−0.022744	−0.032601	−0.021913

In Figure 4, you can see the entire conditional distributions for the three IBEX35 quantiles. Finally, in Figure 5, we plotted the resulting unconditional distributions when you integrate over the observed IBEX35 values; recall Equation (12). They reflect quite nicely the asymmetries and some fat tails.

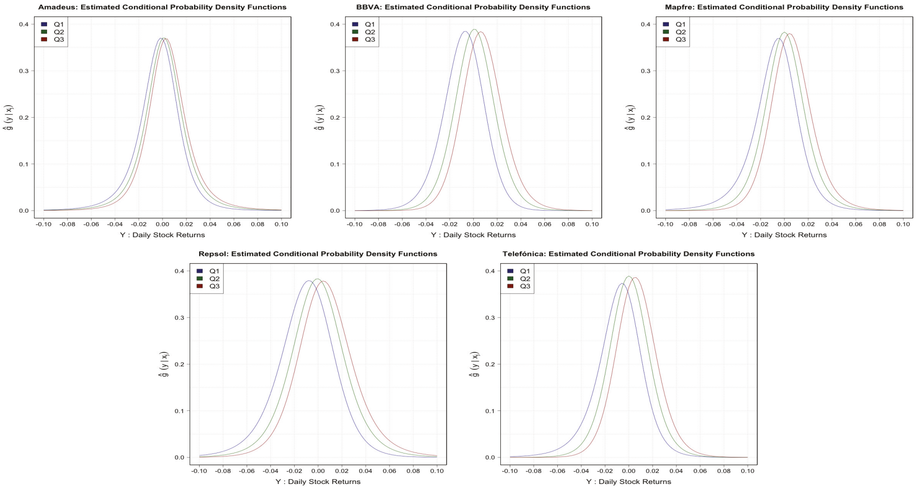


Figure 4. Conditional densities of stock returns at the quantiles of IBEX35 for Amadeus (**upper left**), BBVA (**upper center**), Mapfre (**upper right**), Repsol (**lower left**), and Telefónica (**lower right**).

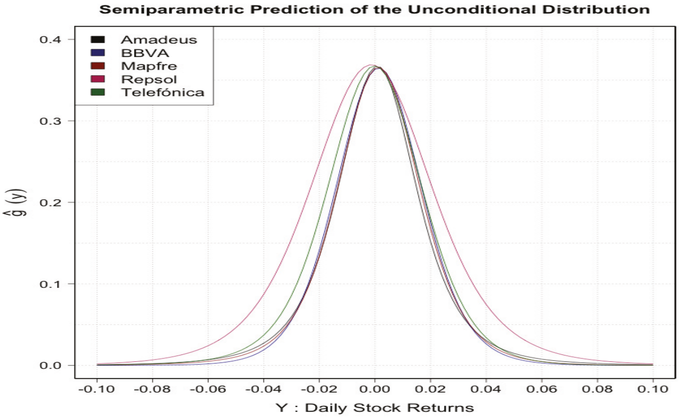


Figure 5. Unconditional densities of stock returns obtained from integrating the conditional ones over all observed IBEX35 values.

5. Discussion and Conclusions

In this note, we revisited the ideas of semiparametric modeling to propose for ML what one could call glass box modeling. It integrates ML in a mixture of a global and a local part in which the global one is as a parametric guide for the nonparametric estimate. We discussed different advantages of such a kind of smart modeling and the steps to be performed. In our illustration (analyzing financial data), we proposed as a parametric guide some (generalized) beta-generated distributions. In particular, we considered two classes of skewed t distributions. This allowed us to work with analytical expressions for the pdf, cdf, moments, and quantile functions, including the VaR, even on a local level. An empirical application with five datasets of stock returns was performed for illustration.

Author Contributions: All authors (J.M.S., F.P., V.J., S.P.) contributed equally to this work. All authors have read and agreed to the published version of the manuscript.

Funding: The authors thank the Institute and Faculty of Actuaries in the U.K. for funding their research through the grant "Minimizing Longevity and Investment Risk while Optimizing Future Pension Plans" and the Spanish Ministerio de Economía y Competitividad, Project ECO2016-76203-C2-1-P, for partial support of this work.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Some Classes of Beta-Generated Distributions

This is to recall the class of BG distributions and to present some basic properties of this class. Let us call pdf $f(y)$ the baseline probability function with $F(y)$ being the corresponding cdf. The class of BG distributions is defined in terms of the pdf by:

$$g_F(y;a,b) = \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} f(y)F(y)^{a-1}[1-F(y)]^{b-1}, \tag{A1}$$

where $\Gamma(y)$ denotes the gamma function and $a,b>0$ some real numbers. If for m being the sample size, we set $a=i$ and $b=m-i+1$ in (A1), one obtains the pdf of the ith order statistic from F (Jones 2004). For $a \neq b$, we obtain a family of skewed distributions, for $a=m$ and $b=1$ the distribution of the maximum, for $a=1$ and $b=m$ the one of the minimum, and for $a=b=1$ obviously $g_F=f$. In our context, the first property is the most interesting one. The parameters a and b control the tailweight of the distribution, where a controls the left-hand and b the right-hand tailweight. Consequently, for $a=b$, one obtains a symmetric sub-family, but still with $a=b$ controlling the tailweight. The BG distribution accommodates several kinds of tails like potential and exponential ones (Jones 2004).

For the moment method and in order to express the VaR in terms of moments or (a,b), we need to relate a,b to (directly) estimable moments. Let us now denote the cdf associated with (A1) by $G_F(y;a,b) = I(F(y);a,b)$ with $I(F(y);\cdot,\cdot)$ denoting the incomplete beta-function ratio:

$$B_x(a,b)/B_1(a,b)\text{ , where } B_y(a,b) = \int_0^y t^{a-1}(1-t)^{b-1}dt$$

where $0 \le y \le 1$, such that $B_1(a,b) = (\Gamma(a)\Gamma(b)/\Gamma(a+b)$. For a random variable $B \sim \mathcal{B}e(a,b)$, i.e., following the classical beta distribution, a simple stochastic representation of (A1) is $Y = F^{-1}(B)$. This allows for a direct simulation of the values of a random variable with pdf (A1). The raw (i.e., not centered, not normalized) moments of a BG distribution can be obtained by:

$$E[Y^r] = E[\{F^{-1}(B)\}^r] \qquad \text{for integers } r>0\,.$$

Recently, some extensions have been proposed, e.g., by Alexander and Sarabia (2010), Alexander et al. (2012) and Cordeiro and de Castro (2011), of which we consider the one towards three parameters: The generalized BG (GBG) distribution is defined for $a,b,c>0$ by the pdf:

$$g_F(y;a,b,c) = \frac{c\Gamma(a+b)}{\Gamma(a)\Gamma(b)} f(y)F(y)^{ac-1}[1-F(y)^c]^{b-1}. \tag{A2}$$

For $c=1$, we get the BG distribution; for $a=c=1$, one obtains the so-called proportional hazard model; and setting $a=1$ yields the so-called Kumaraswamy generated distribution.

References

Alexander, Carol, and José-María Sarabia. 2010. Generalized Beta-Generated Distributions. In *ICMA Centre Discussion Papers in Finance DP2010-09*. Reading: ICMA Centre.

Alexander, Carol, Gauss M. Cordeiro, Edwin M. M. Ortega, and José-María Sarabia. 2012. Generalized beta-generated distributions. *Computational Statistics & Data Analysis* 56: 1880–97.

Azzalini, Adelchi, and Antonella Capitanio. 2003. Distributions generated by perturbation of symmetry with emphasis on a multivariate skew t-distribution. *Journal of the Royal Statistical Society. Series B* 65: 367–89. [CrossRef]

Bishop, Christopher M. 2006. *Pattern Recognition and Machine Learning*. Heidelberg: Springer.

Bolancé, Catalina, Montserrat Guillén, Jim Gustafsson, and Jens Perch Nielsen. 2012. *Quantitative operational Risk Models*. New York: Chapman & Hall/CRC Finance.

Bollerslev, Tim. 1987. A Conditionally Heteroskedastic Time Series Model for Speculative Prices and Rates of Return. *The Review of Economics and Statistics* 69: 542–47. [CrossRef]

Breiman, Leo. 2001. Statistical Modeling: The Two Cultures. *Statistical Science* 16: 199–231. [CrossRef]

Buch-Larsen, Tine, Jens Perch Nielsen, Montserrat Guillén, and Catalina Bolancé. 2005. Kernel density estimation for heavy-tailed distributions using the Champernowne transformation. *Statistics* 39: 503–18. [CrossRef]

Cordeiro, Gauss M., and Mário de Castro. 2011. A new family of generalized distributions. *Journal of Statistical Computation and Simulation* 81: 883–93. [CrossRef]

Dai, Jing, Sefan Sperlich, and Walter Zucchini. 2016. A simple method for predicting distributions by means of covariates with examples from welfare and health economics. *Swiss Journal of Economics and Statistics* 152: 49–80. [CrossRef]

Eilers, Paul H. C., Brian D. Marx, and Maria Durbán. 2015. Twenty years of P-splines. *Statistics and Operation Research Transactions* 39: 149–86.

Eugene, Nicholas, Carl Lee, and Felix Famoye. 2002. The beta-normal distribution and its applications. *Communications in Statistics: Theory Methods* 31: 497–512. [CrossRef]

Friedman, Jerome H. 1998. Data Mining and Statistics: What's the connection? *Computing Science and Statistics* 29: 3–9.

Glad, Ingrid K. 1998. Parametrically guided non-parametric regression. *Scandinavian Journal of Statistics* 25: 649–68. [CrossRef]

Gonzales-Manteiga, Wenceslao, and Rosa M. Crujeiras. 2013. An updated review of goodness-of-fit tests for regression models. *Test* 22: 361–411. [CrossRef] [PubMed]

Grammig, Joachim, Constantin Hanenberg, Christian Schalg, and Jantje Sönksen. 2020. *Diverging Roads: Theory-Based vs. Machine Learning-Implied Stockrisk Premia*. Tübingen, Germany: University of Tübingen Working Papers in Business and Economics, No 130, University of Tübingen.

Härdle, Wolfgang, Gérard Kerkyacharian, Dominique Picard, and Alexander Tsybakov. 1998. *Wavelets, Approximation, and Statistical Applications*. Heidelberg: Springer.

Härdle, Wolfgang, Marlene Müller, Stefan Sperlich, and Alexander Werwatz. 2004. *Nonparametric and Semiparametric Models*. Heidelberg: Springer. [CrossRef]

Hastie, Trevor, Robert Tibshirani, and Jerome Friedman. 2009. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Heidelberg: Springer.

Heidenreich, Niels-Bastian, Anja Schindler, and Stefan Sperlich. 2013. Bandwidth Selection Methods for Kernel Density Estimation: A review of fully automatic selectors. *AStA Advances in Statistical Analysis* 97: 403–33.

Horowitz, Joel L. 1998. *Semiparametric Methods in Econometrics.* Heidelberg: Springer.

Jones, M. Chris, and M.J. Faddy. 2003. A Skew Extension of the t-Distribution, with Applications. *Journal of the Royal Statistical Society. Series B* 65: 159–74. [CrossRef]

Jones, M. Chris. 2004. Families of distributions arising from distributions of order statistics. *Test* 13: 1–43.

Köhler, Max, Anja Schindler, and Stefan Sperlich. 2014. A Review and Comparison of Bandwidth Selection Methods for Kernel Regression. *International Statistical Review* 82: 243–74. [CrossRef]

Kyriakou, Ioannis, Parastoo Mousavi, Jens Perch Nielsen, and Michael Scholz. 2019. Forecasting benchmarks of long-term stock returns via machine learning. *Annals of Operations Research* doi:10.1007/s10479-019-03338-4. [CrossRef]

Lin, Yi, and Yongho Jeon. 2006. Random Forests and Adaptive Nearest Neighbors. *Journal of the American Statistical Association* 101: 578–90. [CrossRef]

Mammen, Enno, Jens Perch Nielsen, Michael Scholz, and Stefan Sperlich. 2019. Conditional Variance Forecasts for Long-Term Stock Returns. *Risks* 7: 113. [CrossRef]

Martínez Miranda, M. Dolores, Jens Perch Nielsen, and Stefan Sperlich. 2009. One Sided Cross Validation for Density Estimation. In *Operational Risk Towards Basel III: Best Practices and Issues in Modeling, Management and Regulation*. Edited by Greg N. Gregoriou. Hoboken: John Wiley and Sons, pp. 177–96.

Meyer, Mary C. 2008. Inference using shape-restricted Regression Splines. *Annals of Applied Statistics* 2: 1013–33. [CrossRef]

Nielsen, Jens Perch, and Stefan Sperlich. 2003. Prediction of stock returns: A new way to look at it. *ASTIN Bulletin* 33: 399–417. [CrossRef]

Rigby, Robert A., and D. Mikis Stasinopoulos. 2006. Generalized additive models for location, scale and shape. *Journal of the Royal Statistical Society. Series C* 54: 507–54. [CrossRef]

Rigby, Robert A., Mikis D. Stasinopoulos, Gillian Z. Heller, and Fernanda De Bastiani. 2019. *Distributions for Modeling Location, Scale, and Shape: Using GAMLSS in R*. New York: Chapman & Hall/CRC Finance.

Ruppert, David, Matt P. Wand, and Raymond J. Carroll. 2003. *Semiparametric Regression*. Cambridge: Cambridge University Press.

Samuel, Arthur. 2006. Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development* 3: 210–29. [CrossRef]

Scholz, Michael, Jens Perch Nielsen, and Stefan Sperlich. 2015. Nonparametric prediction of stock returns based on yearly data: The long-term view. *Insurance: Mathematics and Economics* 65: 143–55. [CrossRef]

Scholz, Michael, Stefan Sperlich, and Jens Perch Nielsen. 2016. Nonparametric long term prediction of stock returns with generated bond yields. *Insurance: Mathematics and Economics* 69: 82–96. [CrossRef]

Severini, Thomas A., and Joan G. Staniswalis. 1994. Quasi-likelihood estimation in semiparametric models. *Journal of the American Statistical Association* 89: 501–11. [CrossRef]

Silverman, Bernard W. 1984. Spline Smoothing: The Equivalent Variable Kernel Method. *Annals of Statistics* 12: 898–916. [CrossRef]

Theodossiou, Panayiotis. 1998. Financial Data and the Skewed Generalized T Distribution. *Management Science* 44: 1650–61. [CrossRef]

Tibshirani, Robert. 1996. Regression Shrinkage and Selection via the lasso. *Journal of the Royal Statistical Society. Series B* 58: 267–88. [CrossRef]

Zhu, Dongming, and John Galbraith. 2010. A generalized asymmetric Student-t distribution with application to financial econometrics. *Journal of Econometrics* 157: 297–305.

© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http://creativecommons.org/licenses/by/4.0/).

MDPI
St. Alban-Anlage 66
4052 Basel
Switzerland
Tel. +41 61 683 77 34
Fax +41 61 302 89 18
www.mdpi.com

Risks Editorial Office
E-mail: risks@mdpi.com
www.mdpi.com/journal/risks

MDPI
St. Alban-Anlage 66
4052 Basel
Switzerland

Tel: +41 61 683 77 34
Fax: +41 61 302 89 18

www.mdpi.com

ISBN 978-3-03936-448-0