algorithms

MDPI

Article

Supervised Dynamic Correlated Topic Model for Classifying Categorical Time Series

Namitha Pais [1,†], Nalini Ravishanker [1,†] and Sanguthevar Rajasekaran [2,*,†]

[1] Department of Statistics, University of Connecticut, Storrs, CT 06269, USA; namitha.pais@uconn.edu (N.P.); nalini.ravishanker@uconn.edu (N.R.)
[2] Department of Computer Science and Engineering, University of Connecticut, Storrs, CT 06269, USA
* Correspondence: sanguthevar.rajasekaran@uconn.edu
† These authors contributed equally to this work.

Abstract: In this paper, we describe the supervised dynamic correlated topic model (sDCTM) for classifying categorical time series. This model extends the correlated topic model used for analyzing textual documents to a supervised framework that features dynamic modeling of latent topics. sDCTM treats each time series as a document and each categorical value in the time series as a word in the document. We assume that the observed time series is generated by an underlying latent stochastic process. We develop a state-space framework to model the dynamic evolution of the latent process, i.e., the hidden thematic structure of the time series. Our model provides a Bayesian supervised learning (classification) framework using a variational Kalman filter EM algorithm. The E-step and M-step, respectively, approximate the posterior distribution of the latent variables and estimate the model parameters. The fitted model is then used for the classification of new time series and for information retrieval that is useful for practitioners. We assess our method using simulated data. As an illustration to real data, we apply our method to promoter sequence identification data to classify *E. coli* DNA sub-sequences by uncovering hidden patterns or motifs that can serve as markers for promoter presence.

Keywords: classification; promoter sequence identification; time series; topic model

Citation: Pais, N.; Ravishanker, N.; Rajasekaran, S. Supervised Dynamic Correlated Topic Model for Classifying Categorical Time Series. *Algorithms* **2024**, *17*, 275. https://doi.org/10.3390/a17070275

Academic Editors: Grigorios Beligiannis, Efstratios F. Georgopoulos, Spiridon D. Likothanassis, Isidoros Perikos and Ioannis X. Tassopoulos

Received: 30 April 2024
Revised: 11 June 2024
Accepted: 20 June 2024
Published: 22 June 2024

1. Introduction

Data on multiple time series or sequences are ubiquitous in various domains, and their classification finds applications in numerous areas, such as human motion classification [1], earthquake prediction [2], and heart attack detection [3]. The classification of multiple real-valued time series have been well studied in the statistical literature (see the detailed review in [4]). However, for analyzing categorical time series, most statistical methods are primarily focused on examining a single time series. A few methods include the Markov chain model [5], the link function approach [6], and the likelihood-based method [7]. In the computer science literature, a number of sequence classification methods have been developed that are black-box in nature and may be difficult to interpret. These include the minimum edit distance classifier with sequence alignment [8,9] and Markov chain-based classifiers [10]. Overall, the classification of multiple categorical time series has not received much attention. Recent work has discussed a novel approach to the classification of categorical time series in the supervised learning framework, using a spectral envelope and optimal scalings [11,12]. We present an alternative approach for classifying multiple categorical time series in the topic modeling framework.

Topic modeling algorithms generally examine a set of documents referred to as a corpus in the natural language processing (NLP) domain. Often, the sets of words observed in documents appear to represent a coherent theme or topic. Topic models analyze the words in the documents in order to uncover themes that run through them. Many topic

modeling algorithms have been developed over time, including non-negative matrix factorization [13], latent Dirichlet allocation (LDA) [14], and structural topic models [15]. Topic modeling in the dynamic framework has been extensively studied to model the dynamic evolution of document collections over time [16,17]. However, the dynamic evolution of words over time within each document has not been addressed. The topic modeling literature typically assumes that words are interchangeable. We relax this assumption to model word time series (documents). We build a family of probabilistic dynamic topic models by extending the correlated topic model (CTM) in a supervised framework and modeling the dynamic evolution of the underlying latent stochastic process that reflects the thematic structure of the time series collection, and we classify the documents. Topic models like latent Dirichlet allocation, correlated topic models, and dynamic topic models are unsupervised, as only documents are used to identify topics by maximizing the likelihood (or the posterior probability) of the collection. In such modeling frameworks, we hope that the topics will be useful for categorization and are useful when no response is available and we want to infer useful information from the observed documents. However, when the main goal is prediction, a supervised topic modeling framework is beneficial, as jointly modeling the documents and the responses can identify latent topics that can predict the response variables for future unlabeled documents. The sDCTM framework is attractive because it provides a supervised dynamic topic modeling framework to (i) estimate the evolution of the latent process that captures the dynamic evolution of words in the document; and (ii) classify the time series.

We apply the sDCTM method for promoter sequence identification in *E. coli* DNA sub-sequences. A promoter is a region of DNA where RNA polymerase begins to transcribe a gene, and it is usually located upstream or at the $5'$ end of the transcription initiation site in DNA. DNA promoters are proven to be the primary cause of numerous human diseases, including diabetes [18] and Huntington's disease [19]. Thus, the identification of DNA sequences containing promoters has gained significant attention from researchers in the field of bioinformatics. Several computational methods and tools have been developed to analyze DNA sequences and predict potential promoter regions. These include the classification of promoter DNA sequences using a robust deep learning model, DeePromoter [20], deep learning, and the combination of continuous FastText N-grams [21] and the position-correlation scoring matrix (PCSM) algorithm [22]. We use the sDCTM model to analyze the *E. coli* DNA sub-sequences by treating each DNA sub-sequence as a document and the presence/absence of promoter regions as the associated class label.

The format of this paper is as follows: In Section 2, we describe the framework of the supervised dynamic correlated topic model, along with details on inference and parameter estimation. Section 3 discusses the simulation study conducted to assess our method. In Section 4, we present the results of applying the sDCTM method for promoter sequence identification in *E. coli* DNA subsequences and compare our method with the various classification techniques.

2. Supervised Dynamic Correlated Topic Model

Topic models are traditionally developed by treating words as interchangeable to identify semantic themes within each document. Our method aims to develop a family of probabilistic time series models to analyze the evolution of words over time within document collections. We assume that each document, represented by a categorical time series of words, arises from a generative process that includes latent variables. sDCTM provides a supervised framework for time series classification, allowing it to model class labels associated with each word time series. Alternatively, by removing the response component of the generative process, we can derive an unsupervised dynamic correlated topic model (DCTM).

Suppose we have a corpus $\mathcal{C}$ consisting of M documents. We represent the dth document as $\mathcal{D}_d = (\boldsymbol{w}_{d,1}, \boldsymbol{w}_{d,2}, \ldots, \boldsymbol{w}_{d,T})$, where $\boldsymbol{w}_{d,t}$ corresponds to a word observed at

time point t on the dth document for times $t = 1, 2, \ldots, T$. Each word is represented by a V-dimensional unit (basis) vector, i.e., $\boldsymbol{w}_{d,t} = (w_{d,t}^1, \ldots, w_{d,t}^V)'$ such that,

$$w_d^u = \begin{cases} 1 & \text{if } u = v, \\ 0 & u \neq v, \end{cases} \tag{1}$$

when $\boldsymbol{w}_{d,t}$ corresponds to the vth word from the vocabulary for $t = 1, \ldots, T$ and V denote the number of levels associated with the categorical word time series (document). Additionally, let $\boldsymbol{y}_d$ be a multinomial response associated with dth document for $d = 1, 2, \ldots, M$. Each $\boldsymbol{y}_d$ is represented by a $C \times 1$ unit (basis) vector, i.e., $\boldsymbol{y}_d = (y_{d,1}, \ldots, y_{d,C})'$ such that,

$$y_d^g = \begin{cases} 1 & \text{if } g = c, \\ 0 & g \neq c, \end{cases} \tag{2}$$

where $\boldsymbol{y}_d$ corresponds to the cth class for $c \in \{1, 2, \ldots C\}$, where C denotes the number of levels associated with the response variable. Suppose J is the number of assumed latent topics for capturing the hidden thematic structure in the corpus; then, $\boldsymbol{\delta}_{d,t}$ is a J dimensional vector corresponding to the latent topic structure for the dth document at time t.

In the promoter identification data, we treat each DNA sub-sequence to be a word time series (document) with $V = 4$ levels (indicating nucleotides a, g, c, and t). The response variable $\boldsymbol{y}_d$ for $d = 1, 2, \ldots, M$ indicates the presence/absence of a promoter region with $C = 2$ levels. The number of assumed topics J in the sDCTM model capture the hidden patterns within the observed DNA sub-sequences. Under the sDCTM framework, the dth document (DNA sub-sequence) $\mathcal{D}_d$ and its associated class label (presence/absence of a promoter) $\boldsymbol{y}_d$ for $d = 1, 2, \ldots, M$ arises from the following generative process. For $d = 1, 2, \ldots, M$,

- For $t = 1, 2, \ldots, T$

  Choose $\boldsymbol{\delta}_{d,t} \mid \boldsymbol{\delta}_{d,t-1} \sim N(\boldsymbol{\Phi}\boldsymbol{\delta}_{d,t-1}, \boldsymbol{\Sigma})$.
  Choose a topic $\boldsymbol{Z}_{\{d,t\}} \mid \boldsymbol{\delta}_{d,t} \sim Mult(1, f(\boldsymbol{\delta}_{d,t}))$, where
  $f(\delta_{d,t,j}) = \dfrac{\exp(\delta_{d,t,j})}{\sum_{l=1}^{J} \exp(\delta_{d,t,l})}$ for $j = 1, 2, \ldots, J$.
  Choose a word $\boldsymbol{w}_{d,t} \mid \{\boldsymbol{Z}_{\{d,t\}}, \boldsymbol{\beta}\} \sim Mult(1, \boldsymbol{\beta}_{z_{\{d,t\}}})$.

- Draw class label $\boldsymbol{y}_d \mid \boldsymbol{Z}_{d,1:T}, \boldsymbol{\eta} \sim softmax(\bar{\boldsymbol{z}}_d, \boldsymbol{\eta})$, where

$$\bar{\boldsymbol{z}}_d = \frac{1}{T} \sum_{t=1}^{T} \boldsymbol{Z}_{\{d,t\}},$$

is the topic frequencies for the dth document. The per-word topic indicator for each word $\boldsymbol{w}_{d,t}$ is represented by $\boldsymbol{Z}_{d,t}$ for $t = 1, \ldots, T$ and $d = 1, \ldots, M$. The model parameters include the latent VAR(1) parameter, the $J \times J$ matrices $\boldsymbol{\Phi}$ and $\boldsymbol{\Sigma}$, the $J \times V$ matrix of word probabilities $\boldsymbol{\beta}$, and the $J \times C$ matrix of regression coefficients $\boldsymbol{\eta}$. In Section 2.1, we discuss approximate inference techniques based on variational methods and construct the lower bound (Constructing the Lower Bound) to estimate the variational parameters, which can be used to approximate the posterior. In Section 2.2, we discuss the variational EM algorithm to estimate the model parameters. We also discuss simulated annealing (Section 2.2.1), a probabilistic technique employed in optimizing problems to estimate certain model parameters, along with details on selecting initial values (Section 2.2.2) for using this optimization technique.

2.1. Approximate Inference

Given the dth document $\mathcal{D}_d$ and the associated response $\boldsymbol{y}_d$ for $d = 1, 2, \ldots, M$, the posterior distribution of the latent variables $(\boldsymbol{\delta}_{d,1:T}, \boldsymbol{Z}_{d,1:T})$ is

$$p(\delta_{d,1:T}, \mathbf{Z}_{d,1:T}|\mathcal{D}_d, \mathbf{y}_d, \boldsymbol{\Phi}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{\eta}) = \frac{p(\delta_d, \mathbf{Z}_d, \mathcal{D}_d, \mathbf{y}_d \mid \boldsymbol{\Phi}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{\eta})}{p(\mathcal{D}_d, \mathbf{y}_d \mid \boldsymbol{\Phi}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{\eta})}, \quad (3)$$

for $d = 1, 2, \ldots, M$. The numerator term in Equation (3), $p(\delta_d, \mathbf{Z}_d, \mathcal{D}_d, \mathbf{y}_d \mid \boldsymbol{\Phi}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{\eta})$, corresponds to the joint distribution of the latent variables $(\delta_{d,1:T}, \mathbf{Z}_{d,1:T})$ and the observed document–response pair $(\mathcal{D}_d, \mathbf{y}_d)$ and is given by

$$p(\delta_d, \mathbf{Z}_d, \mathcal{D}_d, \mathbf{y}_d \mid \boldsymbol{\Phi}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{\eta}) = \prod_{t=1}^{T} \Big(p(\boldsymbol{w}_{d,t}|\mathbf{Z}_{d,t}, \boldsymbol{\beta}) \times p(\mathbf{Z}_{d,t}|\delta_{d,t}) \times p(\delta_{d,t} \mid \delta_{d,t-1}, \boldsymbol{\Phi}, \boldsymbol{\Sigma}) \Big) \times p(\mathbf{y}_d \mid \mathbf{Z}_{d,1:T}, \boldsymbol{\eta}),$$

for $d = 1, 2, \ldots, M$. This joint distribution of the latent variables and the document–response pair can be factorized due to the sDCTM framework. The denominator term in Equation (3), i.e., $p(\mathcal{D}_d, \mathbf{y}_d \mid \boldsymbol{\Phi}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{\eta})$, represents the joint distribution of the dth document–response pair $(\mathcal{D}_d, \mathbf{y}_d)$ and is given by

$$p(\mathcal{D}_d, \mathbf{y}_d \mid \boldsymbol{\Phi}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{\eta}) = \int \Big(\sum_{\mathbf{Z}_{d,1:T}} \Big(\prod_{t=1}^{T} p(\boldsymbol{w}_{d,t}|\mathbf{Z}_{d,t}, \boldsymbol{\beta}) \times p(\mathbf{Z}_{d,t}|\delta_{d,t}) p(\delta_{d,t} \mid \delta_{d,t-1}, \boldsymbol{\Phi}, \boldsymbol{\Sigma}) \Big) \times p(\mathbf{y}_d \mid \mathbf{Z}_{d,1:T}, \boldsymbol{\eta}) \Big) d\delta_{d,1:T},$$

for $d = 1, 2, \ldots, M$.

Computing the posterior distribution in Equation (3) is not analytically tractable. Hence, we use variational methods that consider a simple family of distributions over the latent variables, indexed by free variational parameters. The variational parameters are estimated to minimize the Kullback–Leibler (KL) divergence [23] between the variational distribution and the true posterior distribution. The latent variables in the sDCTM framework include the per-word topic structure $\delta_{d,t}$ and the per-word topic assignment $\mathbf{Z}_{d,t}$ for $t = 1, 2, \ldots T$ and $d = 1, 2, \ldots M$. We define the approximate variational posterior for the dth document as

$$q(\delta_{d,1:T}, \mathbf{Z}_{d,1:T}) = q(\delta_{d,1}, \delta_{d,2}, \ldots, \delta_{d,T} \mid \hat{\delta}_{d,1}, \hat{\delta}_{d,2}, \ldots, \hat{\delta}_{d,T}, \hat{\sigma}^2) \times \prod_{t=1}^{T} q(\mathbf{Z}_{d,t} \mid \gamma_{d,t}), \quad (4)$$

for $d = 1, 2, \ldots, M$, where $\mathbf{Z}_{d,t} \mid \gamma_{d,t}$ is $Mult(1, \gamma_{d,t})$, and we obtain $q(\delta_{d,1:T} \mid \hat{\delta}_{d,1:T})$ using a dynamic model with Gaussian "variational observations" $\{\hat{\delta}_{d,1}, \hat{\delta}_{d,2}, \ldots, \hat{\delta}_{d,T}\}$. This dynamic model is defined using a variational Kalman filter, where the variational parameters $\hat{\delta}_{d,1:T}$ are treated as observations. Using the variational distribution, we form a variational state space model as follows. For $t = 1, 2, \ldots T$,

Observation Equation:

$$\hat{\delta}_{d,t} \mid \delta_{d,t} \sim N(\delta_{d,t}, \hat{\sigma}^2 \boldsymbol{I}). \quad (5)$$

State Equation:

$$\delta_{d,t} \mid \delta_{d,t-1} \sim N(\boldsymbol{\Phi}\delta_{d,t-1}, \boldsymbol{\Sigma}). \quad (6)$$

The Gaussian variational forward filtering distribution $p(\delta_{d,t} \mid \hat{\delta}_{d,1:t})$ using standard Kalman filter calculations is characterized as follows. For $t = 1, 2, \ldots, T$,

$$\begin{aligned} \delta_{d,t} \mid \hat{\delta}_{d,1:t} &\sim N_J(\boldsymbol{m}_{d,t}, \boldsymbol{V}_{d,t}), \\ \boldsymbol{m}_{d,t} &= (\boldsymbol{I} - \boldsymbol{K}_{d,t})\boldsymbol{\Phi}\boldsymbol{m}_{d,t-1} + \boldsymbol{K}_{d,t}\hat{\delta}_{d,t}, \\ \boldsymbol{V}_{d,t} &= (\boldsymbol{I} - \boldsymbol{K}_{d,t})[\boldsymbol{\Phi}\boldsymbol{V}_{d,t-1}\boldsymbol{\Phi}' + \boldsymbol{\Sigma}], \end{aligned}$$

where

$$\boldsymbol{K}_{d,t} = [\boldsymbol{\Phi}\boldsymbol{V}_{d,t-1}\boldsymbol{\Phi}' + \boldsymbol{\Sigma}](\boldsymbol{\Phi}\boldsymbol{V}_{d,t-1}\boldsymbol{\Phi}' + \boldsymbol{\Sigma} + \hat{\sigma}^2\boldsymbol{I})^{-1},$$

is the Kalman gain matrix and the initial conditions are specified by $\boldsymbol{m}_{d,0}$ and $\boldsymbol{V}_{d,0}$.

Similarly, the Gaussian variational backward smoothing distribution $p(\boldsymbol{\delta}_{d,t-1} \mid \hat{\boldsymbol{\delta}}_{d,1:T})$ using standard Kalman smoothing calculations is characterized as follows. For $t = T$, $T-1, \ldots, 1$,

$$\begin{aligned} \boldsymbol{\delta}_{d,t-1} \mid \hat{\boldsymbol{\delta}}_{d,1:T} &\sim N_J(\tilde{\boldsymbol{m}}_{d,t-1}, \tilde{\boldsymbol{V}}_{d,t-1}), \\ \tilde{\boldsymbol{m}}_{d,t-1} &= (\boldsymbol{I} - \boldsymbol{J}_{d,t-1}\boldsymbol{\Phi})\boldsymbol{m}_{d,t-1} + \boldsymbol{J}_{d,t-1}\boldsymbol{m}_{d,t}, \\ \tilde{\boldsymbol{V}}_{d,t-1} &= \boldsymbol{V}_{d,t-1} + \boldsymbol{J}_{d,t-1}[\tilde{\boldsymbol{V}}_{d,t} - \boldsymbol{\Phi}\boldsymbol{V}_{d,t-1}\boldsymbol{\Phi}' - \boldsymbol{\Sigma}]\boldsymbol{J}'_{d,t-1}, \end{aligned}$$

where

$$\boldsymbol{J}_{d,t-1} = \boldsymbol{V}_{d,t-1}\boldsymbol{\Phi}'[\boldsymbol{\Phi}\boldsymbol{V}_{d,t-1}\boldsymbol{\Phi}' + \boldsymbol{\Sigma}]^{-1},$$

and the initial conditions are specified by $\boldsymbol{m}_{d,T} = \tilde{\boldsymbol{m}}_{d,T}$ and $\boldsymbol{V}_{d,T} = \tilde{\boldsymbol{V}}_{d,T}$.

Using the Kalman filter equations, the approximate variational posterior for the dth document is given by

$$q(\boldsymbol{\delta}_{d,1:T}, \boldsymbol{Z}_{d,1:T}) = q(\boldsymbol{\delta}_{d,t} \mid \hat{\boldsymbol{\delta}}_{d,1:T}, \hat{\sigma}^2) \times \prod_{t=1}^{T} q(\boldsymbol{Z}_{d,t} \mid \gamma_{d,t}), \tag{7}$$

where $\boldsymbol{Z}_{d,t} \mid \gamma_{d,t}$ is $Mult(1, \gamma_{d,t})$ and $\boldsymbol{\delta}_{d,t} \mid \hat{\boldsymbol{\delta}}_{d,1:T}, \hat{\sigma}^2 \sim N(\tilde{\boldsymbol{m}}_{d,t}, \tilde{\boldsymbol{V}}_{d,t})$ for $t = 1, 2, \ldots, T$ and $d = 1, 2, \ldots, M$.

Constructing the Lower Bound

Given the variational distribution defined in Equation (4), our goal is to estimate the variational parameters $\gamma_{d,t}$, $\hat{\boldsymbol{\delta}}_{d,t}$, and $\hat{\sigma}^2$ for $t = 1, 2, \ldots, T$ and $d = 1, 2, \ldots, M$ in order to minimize the KL divergence between the variational distribution $q(.)$ defined in Equation (4) and the true posterior $p(.)$ defined in Equation (3). This optimization problem of minimizing the KL divergence is equivalent to maximizing the evidence lower bound (ELBO) [24] defined below.

$$\begin{aligned} \log p(\mathcal{D}_d, \boldsymbol{y}_d | \boldsymbol{\Phi}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{\eta}) &= \log \int \sum_{\boldsymbol{Z}_{d,1:T}} p(\boldsymbol{Z}_{d,1:T}, \boldsymbol{\delta}_{d,1:T}, \mathcal{D}_d, \boldsymbol{y}_d \mid \boldsymbol{\Phi}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{\eta}) d\boldsymbol{\delta}_{d,1:T}, \\ &= \log \int \sum_{\boldsymbol{Z}_{d,1:T}} \frac{p(\boldsymbol{Z}_{d,1:T}, \boldsymbol{\delta}_{d,1:T}, \mathcal{D}_d, \boldsymbol{y}_d \mid \boldsymbol{\Phi}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{\eta})}{q(\boldsymbol{\delta}_{d,1:T}, \boldsymbol{Z}_{d,1:T})} \times \\ &\qquad q(\boldsymbol{\delta}_{d,1:T}, \boldsymbol{Z}_{d,1:T}) d\boldsymbol{\delta}_{d,1:T} \\ &\geq E_q\Big[\log p(\boldsymbol{Z}_{d,1:T}, \boldsymbol{\delta}_{d,1:T}, \mathcal{D}_d, \boldsymbol{y}_d \mid \boldsymbol{\Phi}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{\eta})\Big] - \\ &\qquad E_q\Big[\log q(\boldsymbol{\delta}_{d,1:T}, \boldsymbol{Z}_{d,1:T})\Big], \\ &= L(\hat{\boldsymbol{\delta}}_{d,1:T}, \hat{\sigma}^2, \gamma_d; \boldsymbol{\Phi}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{\eta}, \mathcal{D}_d, \boldsymbol{y}_d). \end{aligned}$$

We estimate the variational parameters $\gamma_{d,t}$, $\hat{\boldsymbol{\delta}}_{d,t}$, and $\hat{\sigma}^2$ for $t = 1, 2, \ldots, T$ and $d = 1, 2, \ldots, M$ by maximizing the ELBO for approximate inference. In Appendix A, we derive the ELBO and provide the update equations in order to estimate the variational parameters. We estimate the variational parameter $\gamma_{d,t}$ using a fixed-point update, $\hat{\sigma}^2$ using a constrained optimization using the DBCPOL routine available in the IMSL library [25] for Fortran 77, and $\hat{\boldsymbol{\delta}}_{d,1:T}$ using simulated annealing, which we describe in Section 2.2.1.

2.2. Parameter Estimation

In this section, we present a method for parameter estimation. In particular, given the corpus $\mathcal{C}$ with M documents and their associated responses denoted by $(\mathcal{D}, \boldsymbol{y})_{1:M}$, we wish

to estimate the model parameters $\boldsymbol{\Phi}^{J\times J}, \boldsymbol{\Sigma}^{J\times J}, \boldsymbol{\beta}^{J\times V}$, and $\boldsymbol{\eta}^{J\times C}$ in order to maximize the log likelihood of the data given by

$$l(\boldsymbol{\Phi}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{\eta} \mid (\mathcal{D}, \boldsymbol{y})_{1:M}) = \sum_{d=1}^{M} \log p(\mathcal{D}_d, \boldsymbol{y}_d | \boldsymbol{\Phi}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{\eta}). \tag{8}$$

Because the log likelihood is analytically intractable, we consider a tractable lower bound on the log likelihood by summing the ELBO for $\log p(\mathcal{D}_d, \boldsymbol{y}_d | \boldsymbol{\Phi}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{\eta})$, defined in Section 2.1 over M documents. We estimate the model parameters via an alternating variational EM (VEM) procedure [26] described below:

E-Step: For each document, we find the optimizing values of the variational parameters described in Section 2.1.
M-Step: Maximize the resulting lower bound on the log likelihood with respect to the model parameters.

In Appendix A, we derive the lower bound on the log likelihood defined in Equation (8) and provide the updated equations in order to estimate the model parameters. We estimate model parameter $\boldsymbol{\beta}$ using a fixed-point update. The model parameters $\boldsymbol{\Phi}^{J\times J}$ (with the stationarity condition), $\boldsymbol{\Sigma}$ (with the positive definite condition), and $\boldsymbol{\eta}$ are estimated using a randomized search optimization technique, simulated annealing, which we describe in the next section.

For parameter estimation, we use frequentist methods to estimate the unknown parameters of the sDCTM model. As an alternative, one can set up a Bayesian model for parameters estimation in the sDCTM model by employing the Minnesota prior on $\boldsymbol{\Phi}$ [27], inverse Wishart or Lewandowski–Kurowicka–Joe (LKJ) prior on $\boldsymbol{\Sigma}$, Dirichlet prior on $\boldsymbol{\beta}$, and multivariate normal prior on $\boldsymbol{\eta}$.

2.2.1. Simulated Annealing

Simulated annealing (SA) is a probabilistic technique employed for optimizing problems to find a good approximation to the global optimum of a given function [28,29]. It is often used when the search space is discrete and is inspired by the annealing technique in metallurgy, which involves the heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. The algorithm starts with a randomly generated solution and iteratively explores its neighboring states to find better solutions. The acceptance probability mechanism considers solutions that are worse than the current one in order to avoid getting stuck in a local optimum. Algorithm 1 presents a structured pseudocode to execute the simulated annealing technique for optimization problems, and Table 1 describes the initial parameters.

Table 1. Initial parameters for simulated annealing.

Parameter	Description
$\boldsymbol{x}$	n dimensional initial vector.
$\boldsymbol{v}$	n dimensional initial step length vector.
f	Objective function to minimize.
ϵ	Convergence criteria.
T	Temperature
r_T	Temperature reduction factor.
N	Maximum number of function evaluations.
N_T	Number of function evaluations before T adjustment.
N_v	Number of function evaluations before $\boldsymbol{v}$ adjustment.
count	Calculate the number of acceptances.
c	Controls how fast $\boldsymbol{v}$ adjusts.

Algorithm 1 Simulated Annealing for Optimization

procedure SIMULATEDANNEALING($\boldsymbol{x}, f$)
  $\boldsymbol{x}_{\text{opt}} = \boldsymbol{x}$ and $f_{\text{opt}} = f$, $count = 0$
  repeat
    for N_T times **do**
      for N_v times **do**
        $x_i' = x_i + r \times v_i$, $r_i \sim U(-1, 1)$ for $i = 1, 2, \ldots, n$
        $f' = \text{CALCULATE}f'(\boldsymbol{x}')$
        if $f' < f$ **then**
          $\boldsymbol{x} = \boldsymbol{x}'$ and $f = f'$
          $count = count + 1$
        end if
        if $f' < f_{\text{opt}}$ **then**
          $\boldsymbol{x} = \boldsymbol{x}'$ and $f = f'$
          $\boldsymbol{x}_{\text{opt}} = \boldsymbol{x}'$ and $f_{\text{opt}} = f'$
        end if
        if $f' \geq f$ **then**
          $p = \exp\left(\frac{f - f'}{T}\right)$ and $p' \sim U(0, 1)$
          if $p > p'$ **then**
            $\boldsymbol{x} = \boldsymbol{x}'$ and $f = f'$
          end if
        end if
        Adjust $\boldsymbol{v}$ such that 50% of the moves are accepted
        Set $ratio = \frac{count}{N_v}$
        if $ratio \geq 0.6$ **then**
          $v_i = v_i \times \left(1 + \frac{c \times (ratio - 0.6)}{0.4}\right)$ for $i = 1, 2, \ldots, n$
        end if
        if $ratio \leq 0.4$ **then**
          $v_i = \frac{v_i}{\left(1 + \frac{c \times (0.4 - ratio)}{0.4}\right)}$ for $i = 1, 2, \ldots, n$
        end if
      end for
      if $|f - f'| < \epsilon$ and $|f_{\text{opt}}^{\text{old}} - f_{\text{opt}}| < \epsilon$ **then**
        REPORT $\boldsymbol{x}_{\text{opt}}$, f_{opt}
      else
        Set $\boldsymbol{x} = \boldsymbol{x}_{\text{opt}}$ and $T = r_T \times T$
      end if
    end for
  until convergence or N times
end procedure

2.2.2. Selecting Initial Values of the Model Parameters

We use the SA technique, one of the top methods for non-derivative-based random search optimization, to estimate the model parameters $\boldsymbol{\eta}$, $\boldsymbol{\Phi}$, and $\boldsymbol{\Sigma}$ in the sDCTM framework. To ensure that the optimization works effectively, it is important to select appropriate initial values. We describe the steps to select initial values for $\boldsymbol{\eta}$, $\boldsymbol{\Phi}$, and $\boldsymbol{\Sigma}$ below.

Step 1. We select an initial search space within $\mathbb{R}^n$, where n represents the number of variables involved in the estimation (for instance, $n = C \times J$ for $\boldsymbol{\eta}$ and $n = J \times J$ for $\boldsymbol{\Phi}$ and $\boldsymbol{\Sigma}$). The initial search space is chosen based on our belief of where the estimates are likely to be found. Then, we divide this search space into 2^n smaller subspaces.
Step 2. For $\boldsymbol{\Phi}$ and $\boldsymbol{\Sigma}$, we evaluate the optimization function at the midpoint within each of the subspaces and choose the subspace with the highest (or lowest) value of the function based on the optimization problem as our new search space. In cases where multiple subspaces optimize the function, we consider each of them in the next step for further

exploration. In the sDCTM framework, for $\boldsymbol{\eta}$, we consider the training accuracy as our optimization (maximization) function, as $\boldsymbol{\eta}$ plays a crucial role in predicting the response variable. For model parameters $\boldsymbol{\Phi}$ and $\boldsymbol{\Sigma}$, we consider the lower bound on the log likelihood as the optimization (maximization) function.

Step 3. We repeat steps 1 and 2 until no significant improvement in the optimization function value is seen.

Step 4. If a single subspace is chosen as a search space to select the initial values for optimization, we choose our starting values from this subspace and estimate the parameters using the SA technique. If there are multiple subspaces selected as the initial search space, we choose the initial values from each of these subspaces and run the SA to obtain the parameter estimates. We choose the final parameter estimates from the search space yielding the highest (or lowest) value of the function based on the optimization problem. The sDCTM model is implemented using a standalone code in Fortran 77 [30]. The code for model estimation is posted on https://github.com/NamithaVionaPais/sDTCM (accessed on 10 June 2024).

3. Simulation Study

To evaluate the performance of our model, we conducted a simulation study. We generated a dataset consisting of $M = 120$ documents, where each document is represented by a word time series of length $T = 100$. The vocabulary size was set to $V = 6$, and the documents were divided into $C = 2$ classes. Each word time series was governed by an underlying latent topic structure with $J = 3$ topics. Our model was evaluated using a six-fold cross-validation with $M_{Tr} = 100$ train and $M_{Tst} = 20$ test documents. We conducted our analysis in Fortran 77. Given the observed word time series from M documents, we fitted the sDCTM model to estimate the model parameters, $\boldsymbol{\beta}$, $\boldsymbol{\Sigma}$, $\boldsymbol{\Phi}$, and $\boldsymbol{\eta}$. As the model parameters are matrices, we assessed the estimation error using the squared Frobenius norm. The squared Frobenius norm lies between two matrices, such as $\boldsymbol{M}$ and $\boldsymbol{N}$, and is defined as

$$d(\boldsymbol{M}, \boldsymbol{N}) = \text{tr}\{(\boldsymbol{M} - \boldsymbol{N})'(\boldsymbol{M} - \boldsymbol{N})\} \tag{9}$$

The average squared Frobenius norm (over the six-fold cross-validation) for each of the model parameters is shown in Table 2. The low values of the squared Frobenius norm on the model parameters suggest that we are able to estimate the matrices well.

Table 2. Average squared Frobenius norm between the true and estimated model parameters using the sDCTM model on the simulated dataset.

Parameter	Squared Frobenius Norm
$\boldsymbol{\Phi}$	0.5270
$\boldsymbol{\Sigma}$	0.3697
$\boldsymbol{\beta}$	0.1622
$\boldsymbol{\eta}$	0.2385

We used the estimated model parameter $\boldsymbol{\eta}$ and the variational parameter $\gamma_{d,t}$ for $t = 1, 2, \ldots, T$ and $d = 1, 2, \ldots, M_{Tr}$ to predict the response associated with the training data containing $M_{Tr} = 100$ documents as follows.

$$c_d = \underset{c \in \{1, \ldots C\}}{\text{argmax}}\, \boldsymbol{\eta}_c^T \bar{\gamma}_d, \tag{10}$$

for $d = 1, 2, \ldots, M_{Tr}$ and $\bar{\gamma}_d = \dfrac{T}{\sum_{t=1}^{T} \gamma_{d,t}} \gamma_{d,t}$. Table 3 shows the confusion matrix obtained on train data for one of the cross-validations. We were able to achieve a training accuracy of 100%, and the average training accuracy across all $k = 6$ cross-validation datasets was also 100%.

Table 3. Confusion matrix obtained from the sDCTM model on the test data of $M = 20$ documents based on a simulation.

	Actual	
Predicted	**Class 1**	**Class 2**
Class 1	80	0
Class 2	0	20

While classifying the test data, we estimated the variational parameters on the test documents given the estimated model parameters using variational inference. However, because we assume that we do not know the true label in the test documents, we replace the terms in the likelihood function associated with the response variable by the pseudo-label. The pseudo-label assigned to a test document is the class label associated with the nearest train document. We used the hamming distance to calculate the distance between the test and train sequence. Given the word time series, the associated pseudo-label, and the model parameters estimated using the train data, we estimated the variational parameters for each test document. We then obtained the test predictions using Equation (10). Table 4 shows the confusion matrix obtained on the test data for one of the cross-validation test datasets with a test accuracy of 80%. The average test accuracy across all $k = 6$ cross-validation datasets was 70.83%.

Table 4. Confusion matrix obtained from the sDCTM model on the test data in a simulation.

	Actual	
Predicted	**Class 1**	**Class 2**
Class 1	9	0
Class 2	4	7

The model parameter, denoted as β, provides an estimate of how words are distributed across a set of J latent topics. Figure 1 illustrates the proportions of words across each of the $J = 3$ latent topics. We observed that Word 1 was highly probable within Topic 1, Words 3, 4, and 6 were highly probable within Topic 2, and Word 4 was highly probable within Topic 3.

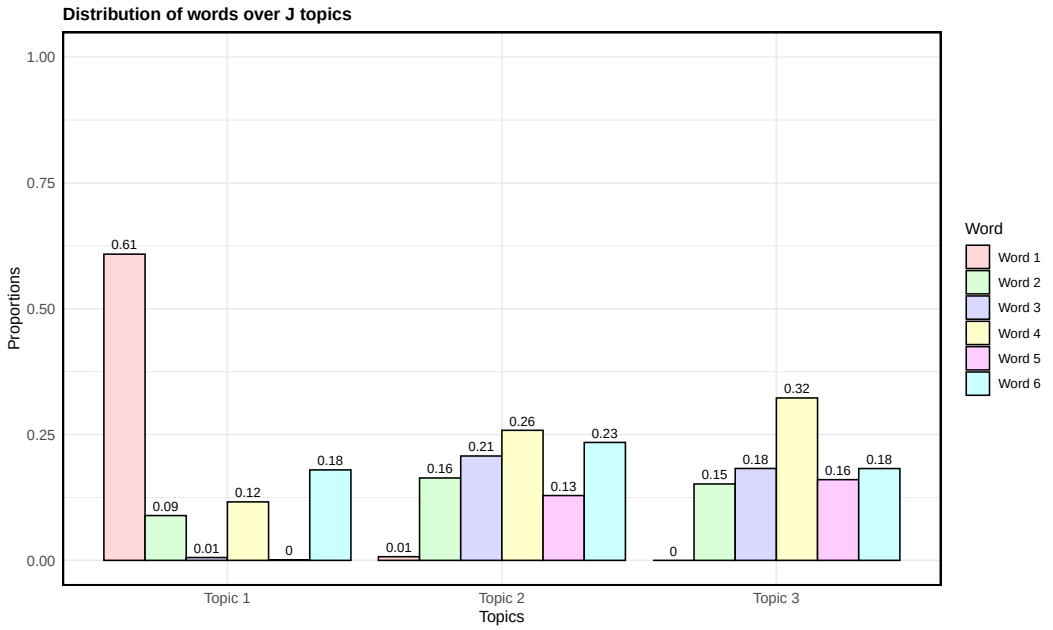

Figure 1. Estimated distribution of words over each of the $J = 3$ latent topics on the simulated data.

4. Application for Promoter Sequence Identification in *E. coli* DNA Sub-Sequences

Proteins are one of the most important classes of biological molecules, being the carriers of the message contained in the DNA. An important process involved in the synthesis of proteins is called transcription. During transcription, a single-stranded RNA molecule, called messenger RNA, is synthesized (using the complementarity of the bases) from one of the strands of DNA corresponding to a gene (a gene is a segment of the DNA that codes for a type of protein). This process begins with the binding of an enzyme called RNA polymerase to a certain location on the DNA molecule. This exact site, which determines which of the two strands of DNA will be a transcript and in which direction, is recognized by the RNA polymerase due to the existence of certain regions of DNA placed near the transcription start site of a gene, called promoters. Because determining the promoter region in the DNA is an important step in the process of detecting genes, the problem of promoter identification is of major importance within the field of bioinformatics.

We considered a dataset of 106 *E. coli* DNA sub-sequences that is publicly available at https://archive.ics.uci.edu/dataset/67/molecular+biology+promoter+gene+sequences (accessed on 12 March 2023), among which 53 DNA sub-sequences contain promoters and the remaining 53 DNA sub-sequences do not contain promoters. Each sub-sequence consists of 57 nucleotides, represented by the four nucleotide bases: adenine (a), guanine (g), cytosine (c), and thymine (t). Given the two sets of DNA sub-sequences, with and without the presence of promoter regions, we used sDCTM to uncover the hidden patterns or motifs that could serve as markers for promoter presence to classify the sub-sequences. A detailed data description is provided in [31].

We present the results of the sDCTM model applied to analyze the E. Coli DNA sub-sequences. We treated each DNA sub-sequence as a document and the presence/absence of promoter regions as the associated class label. Each word in the document was represented by the nucleotide observed at that position in the sub-sequence. We divided the data into an 80–20 train–test split in order to assess the model performance. We trained the model on data with $M_{tr} = 86$ (number of train DNA sub-sequences) and set $T = 57$ (length of each DNA sub-sequence), $V = 4$ (number of unique nucleotide levels), and $C = 2$ (indicating the levels of the response variable). To choose the number of topics J, we ran the model for different values of J and chose J based on the highest test accuracy. We set the number of latent topics as $J = 3$. The confusion matrices obtained on the train and test data with $M_{Tst} = 20$ (number of test DNA sub-sequences) using the sDCTM model are shown in Tables 5 and 6. We were able to achieve a training accuracy of 100% and a test accuracy of 90%.

Table 5. Confusion matrix obtained from the sDCTM model on the promoter identification train data with $M = 86$ DNA sub-sequences.

	Actual	
Predicted	**Promoter Presence**	**Promoter Absence**
Promoter presence	43	0
Promoter absence	0	43

Table 6. Confusion matrix obtained from the sDCTM model on the promoter identification test data with $M = 20$ DNA sub-sequences.

	Actual	
Predicted	**Promoter Presence**	**Promoter Absence**
Promoter presence	10	0
Promoter absence	2	8

The distribution of nucleotides accross the three topics identified by the sDCTM model is shown in Figure 2. We observed that nucleotide levels a, c, and g were highly probable within Topic 1, nucleotide level t was highly probable within Topic 2, and nucleotide level a was highly probable within Topic 3. In addition, based on the model predictions, we also constructed a sequence logo plot [32] to identify the diversity of sequences between the two predicted groups. The sequence logo plot based on the train model predictions are shown in Figure 3. We can see that that the promoter presence plot demonstrated significantly higher conservation at various positions compared with the promoter absence plot. The higher bits values in the promoter presence plot highlight critical motifs essential for promoter activity. In contrast, the promoter absence plot showed much smaller variability, suggesting fewer conserved elements.

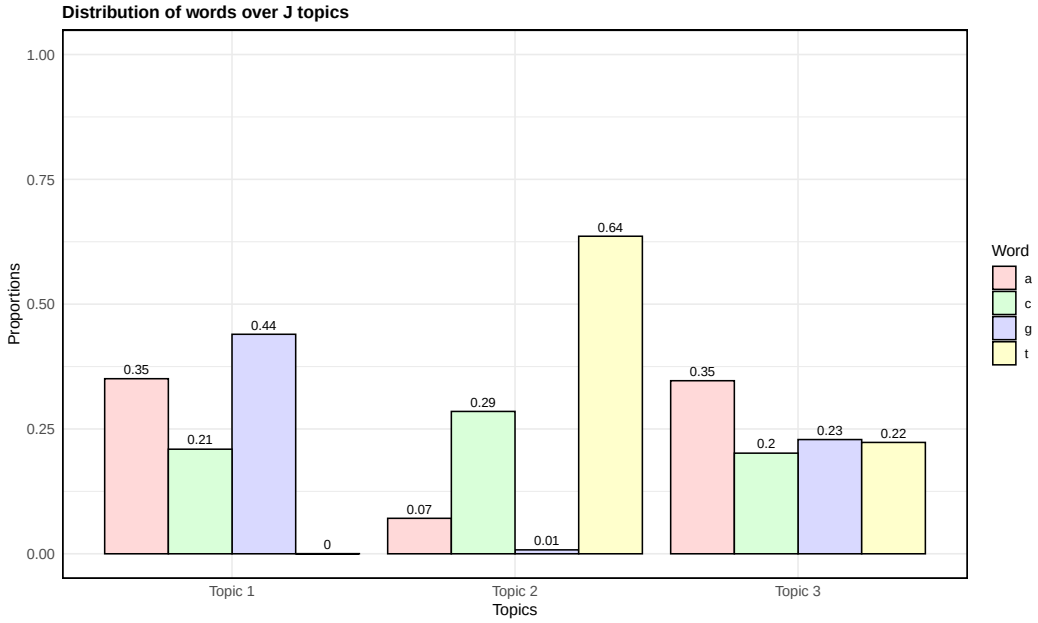

Figure 2. Estimated distribution of $V = 4$ nucleotides over each of the $J = 3$ latent topics.

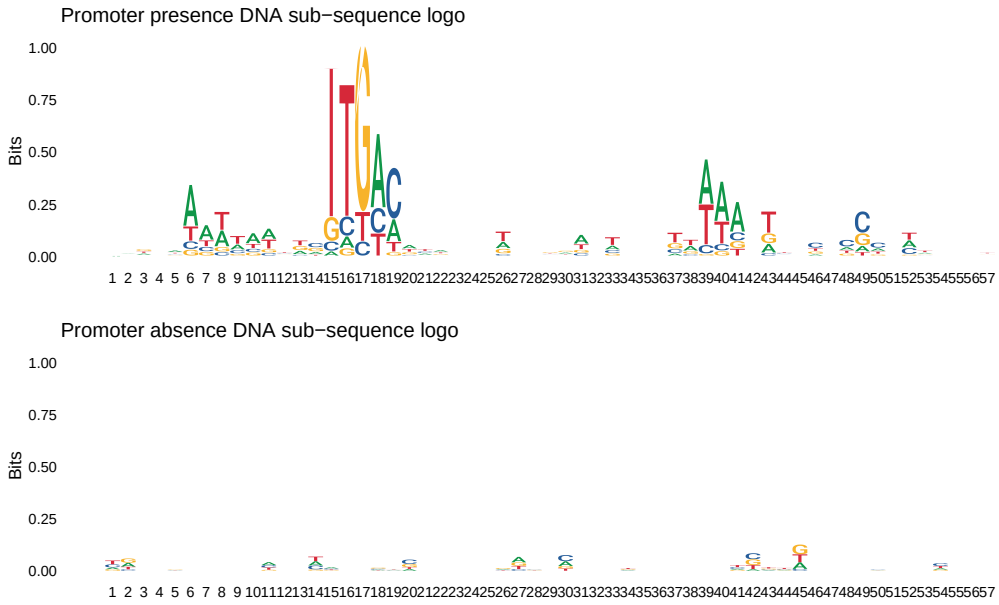

Figure 3. Sequence logo plot based on model predictions on the train data.

Comparison with Other Methods

We compared our sDCTM model with existing classification techniques, such as SVM, k-NN, and classification trees [33]. We used the k-NN classification technique, which is a popular approach used for sequence classification [10] on the observed data. We compared our approach to popular machine learning techniques like SVM and classification trees on the extracted features from the DNA sequence. We used the Haar wavelet transform and the simplest discrete wavelet transform (DWT) to extract features from the observed time series and build a classification model based on these features. We implemented the k-NN classification technique using the R package *class* [34], which identifies $K = 5$ nearest neighbors (in Euclidean distance) to classify the train and test data. We used the R package *wavelets* [35] to extract the DWT (with Haar filter) coefficients. The classification tree was implemented using the R package *party* [36], and the SVM was implemented using the R package *e1071* [37]. We ran each of these methods using an 80–20 train–test split to assess model performance. The train and test accuracies are shown in Table 7. In comparison with the the other classification techniques, sDCTM performed better on both the train and test data.

Table 7. Comparing accuracy using the sDCTM, k-NN, classification tree, and SVM methods on the train and test data.

Method	Train Accuracy	Test Accuracy
sDCTM	100%	90%
k-NN	84.8%	85%
Classification tree	87.2%	85%
SVM	100%	55%

5. Discussion

In this paper, we introduced the sDCTM model framework, which aims to classify categorical time series by dynamically modeling the underlying latent topics that capture the hidden thematic structure of the time series collection. We demonstrated the applicability of our method to real data by applying it to the promoter identification data to classify *E. coli* DNA sub-sequences based on promoter presence. Using sDCTM, we estimated the dynamic latent topic structure that serves as markers for promoter presence in a DNA sub-sequence and classify the sub-sequences. Among the $J = 3$ latent topics obtained using sDCTM, we observed that nucleotide levels *a*, *c*, and *g* were highly probable within Topic 1, nucleotide level *t* was highly probable within Topic 2, and nucleotide level *a* was highly probable within Topic 3. In addition, the logoplot based on the model predictions showed higher conservation at various positions in DNA sub-sequences with predicted promoter presence in comparison with DNA sub-sequences with predicted promoter absence. We compared our method with the k-NN, SVM, and classification tree method on a train–test data setup. Our comparative study results indicated that the sDCTM model performed better than the other classification techniques in both training and test datasets. This indicates that the estimated underlying latent topic structure captured by the sDCTM model is able to identify the promoter presence/absence in a DNA sub-sequence better in comparison with the other classification approaches. An extension of the sDCTM model accommodating word time series of varying lengths can be derived as a part of future work.

Author Contributions: Conceptualization, N.P., N.R. and S.R.; methodology, N.P., N.R. and S.R.; formal analysis, N.P., N.R. and S.R.; investigation, N.P., N.R. and S.R.; data curation, N.P.; writing—original draft preparation, N.P., N.R. and S.R.; writing—review and editing, N.P., N.R. and S.R.; visualization, N.P., N.R. and S.R.; supervision, N.R. and S.R.; project administration, N.R. and S.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data and analysis code are available at https://github.com/NamithaVionaPais/sDTCM (accessed on 10 June 2024), Github link.

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

Appendix A. Approximate Inference for sDCTM

We derive the evidence lower bound (ELBO) based on our choice of variational distribution defined in (4). Then, we derive the variational inference algorithm for the sDCTM model by maximizing the ELBO to estimate the variational parameters.

Appendix A.1. ELBO

$$
\begin{aligned}
L(\hat{\boldsymbol{\delta}}_{d,1:T}, \hat{\sigma}^2, \gamma_d; \boldsymbol{\Phi}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{\eta}) &= E_q\Big[\log p(\boldsymbol{Z}_{d,1:T}, \boldsymbol{\delta}_{d,1:T}, \mathcal{D}_d, \boldsymbol{y}_d \mid \boldsymbol{\Phi}, \boldsymbol{\Sigma}, \boldsymbol{\beta}, \boldsymbol{\eta})\Big] - \\
&\quad E_q\Big[\log q(\boldsymbol{\delta}_{d,1:T}, \boldsymbol{Z}_{d,1:T})\Big] \\
&= \sum_{t=1}^{T} E_q\Big[\log p(\boldsymbol{w}_{d,t} \mid \boldsymbol{Z}_{d,t}, \boldsymbol{\beta})\Big] + \\
&\quad \sum_{t=1}^{T} E_q\Big[\log p(\boldsymbol{Z}_{d,t} \mid \boldsymbol{\delta}_{d,t})\Big] \\
&\quad + \sum_{t=1}^{T} E_q\Big[\log p(\boldsymbol{\delta}_{d,t} \mid \boldsymbol{\delta}_{d,t-1}, \boldsymbol{\Phi}, \boldsymbol{\Sigma})\Big] + \\
&\quad E_q\Big[\log p(\boldsymbol{y}_d = \mathbf{1}_c \mid \boldsymbol{Z}_{d,1:T}, \boldsymbol{\eta})\Big] \\
&\quad - \sum_{t=1}^{T} E_q\Big[\log q(\boldsymbol{\delta}_{d,t} \mid \hat{\boldsymbol{\delta}}_{d,1:T}, \hat{\sigma}^2)\Big] - \sum_{t=1}^{T} E_q\Big[\log q(\boldsymbol{Z}_{d,t} \mid \gamma_{d,t})\Big] \\
&\geq \sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{i=1}^{V} \gamma_{dtj} w_{dt}^{i} \log \beta_{ji} \\
&\quad + \sum_{t=1}^{T}\sum_{j=1}^{J} \gamma_{dtj}\left[\tilde{m}_{dtj} - \log\Big(\sum_{l=1}^{J} \exp[\tilde{m}_{dtl} + \frac{\tilde{v}_{dtl}}{2}]\Big)\right] \\
&\quad - \frac{TJ}{2}\log(2\pi) - \frac{T}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{t=1}^{T} tr\Big[\boldsymbol{\Sigma}^{-1}(\tilde{\boldsymbol{V}}_{dt} + \boldsymbol{\Phi}\tilde{\boldsymbol{V}}_{dt}\boldsymbol{\Phi}')\Big] \\
&\quad - \frac{1}{2}\sum_{t=1}^{T}(\tilde{\boldsymbol{m}}_{dt} - \boldsymbol{\Phi}\tilde{\boldsymbol{m}}_{dt-1})'\boldsymbol{\Sigma}^{-1}(\tilde{\boldsymbol{m}}_{dt} - \boldsymbol{\Phi}\tilde{\boldsymbol{m}}_{dt-1}) \\
&\quad + \frac{1}{T}\boldsymbol{\eta}_c^T\gamma_{dt} - \log\left(\sum_{l=1}^{C}\prod_{t=1}^{T}\Big(\sum_{j=1}^{J}\gamma_{dtj}\exp(\frac{1}{T}\eta_{lj})\Big)\right) \\
&\quad + \frac{TJ}{2}\log(2\pi) + \frac{1}{2}\sum_{t=1}^{T}\log|\tilde{\boldsymbol{V}}_{d,t}| + \frac{TJ}{2} - \sum_{t=1}^{T}\sum_{j=1}^{J}\gamma_{dtj}log(\gamma_{dtj}) \\
&\geq \sum_{t=1}^{T}\gamma_{dt}'\log(\boldsymbol{\beta})\boldsymbol{w}_{dt} + \sum_{t=1}^{T}\gamma_{dt}'\left[\tilde{\boldsymbol{m}}_{dt} - \log\Big(\boldsymbol{J}\exp[\tilde{\boldsymbol{m}}_{dtl} + diag\frac{\tilde{\boldsymbol{V}}_{dt}}{2}]\Big)\right] \\
&\quad - \frac{T}{2}\log|\boldsymbol{\Sigma}| - \frac{1}{2}\sum_{t=1}^{T} tr\Big[\boldsymbol{\Sigma}^{-1}(\tilde{\boldsymbol{V}}_{dt} + \boldsymbol{\Phi}\tilde{\boldsymbol{V}}_{dt}\boldsymbol{\Phi}')\Big] \\
&\quad - \frac{1}{2}\sum_{t=1}^{T}(\tilde{\boldsymbol{m}}_{dt} - \boldsymbol{\Phi}\tilde{\boldsymbol{m}}_{dt-1})'\boldsymbol{\Sigma}^{-1}(\tilde{\boldsymbol{m}}_{dt} - \boldsymbol{\Phi}\tilde{\boldsymbol{m}}_{dt-1}) \\
&\quad + \frac{1}{T}\boldsymbol{\eta}_c^T\gamma_{dt} - \log\left(\sum_{l=1}^{C}\prod_{t=1}^{T}\Big(\exp\Big[\frac{\boldsymbol{\eta}_l}{T}\Big]\Big)'\gamma_{dt}\right) \\
&\quad + \frac{1}{2}\sum_{t=1}^{T}\log|\tilde{\boldsymbol{V}}_{d,t}| + \frac{TJ}{2} - \sum_{t=1}^{T}\gamma_{dt}'\log(\gamma_{dt})
\end{aligned}
$$

Appendix A.2. Variational Multinomial

$$\begin{aligned}
L[\gamma_{dtj}] &\geq \sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{i=1}^{V}\gamma_{dtj}w_{dt}^{i}\log\beta_{ji} \\
&\quad + \sum_{t=1}^{T}\sum_{j=1}^{J}\gamma_{dtj}\left[\tilde{m}_{dtj} - \log\Big(\sum_{l=1}^{J}\exp[\tilde{m}_{dtl}+\frac{\tilde{v}_{dtl}}{2}]\Big)\right] \\
&\quad + \frac{1}{T}\boldsymbol{\eta}_c^T\boldsymbol{\gamma}_{dt} - \log\left(\sum_{l=1}^{C}\prod_{t=1}^{T}\Big(\sum_{j=1}^{J}\gamma_{dtj}\exp(\frac{1}{T}\eta_{lj})\Big)\right) \\
&\quad - \gamma_{dtj}\log(\gamma_{dtj}) \\
&= \sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{i=1}^{V}\gamma_{dtj}w_{dt}^{i}\log\beta_{ji} \\
&\quad + \sum_{t=1}^{T}\sum_{j=1}^{J}\gamma_{dtj}\left[\tilde{m}_{dtj} - \log\Big(\sum_{l=1}^{J}\exp[\tilde{m}_{dtl}+\frac{\tilde{v}_{dtl}}{2}]\Big)\right] \\
&\quad + \frac{1}{T}\eta_{cj}\gamma_{dtj} - \log(\boldsymbol{h}^T\boldsymbol{\gamma}_{dt}) \\
&\quad - \gamma_{dtj}\log(\gamma_{dtj}) \\
&\geq \sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{i=1}^{V}\gamma_{dtj}w_{dt}^{i}\log\beta_{ji} \\
&\quad + \sum_{t=1}^{T}\sum_{j=1}^{J}\gamma_{dtj}\left[\tilde{m}_{dtj} - \log\Big(\sum_{l=1}^{J}\exp[\tilde{m}_{dtl}+\frac{\tilde{v}_{dtl}}{2}]\Big)\right] \\
&\quad + \frac{1}{T}\eta_{cj}\gamma_{dtj} - (\boldsymbol{h}^T\boldsymbol{\gamma}_t^{old})^{-1}\boldsymbol{h}^T\boldsymbol{\gamma}_t \\
&\quad - \log(\boldsymbol{h}^T\boldsymbol{\gamma}_t^{old}) + 1 - \gamma_{dtj}\log(\gamma_{dtj}) \\
&\quad + \Lambda_t(\sum_{j=1}^{J}\gamma_{tj} - 1).
\end{aligned}$$

Result used: We know that

$$\log(x) \leq \zeta^{-1}x + \log(\zeta) - 1, \forall\, x > 0, \zeta > 0$$

with equality holding iff $x = \zeta$. Then,

$$\begin{aligned}
\frac{\delta L[\gamma_{dtj}]}{\gamma_{dtj}} &= \log\beta_{jv} + \tilde{m}_{dtj} - \log\Big(\sum_{l=1}^{J}\exp[\tilde{m}_{dtl}+\frac{\tilde{v}_{dtl}}{2}]\Big) + \frac{1}{T}\eta_{cj} - \\
&\quad (\boldsymbol{h}^T\boldsymbol{\gamma}_{dt}^{old})^{-1}h_j - \log(\gamma_{dtj}) - 1 + \Lambda_t.
\end{aligned}$$

Equating to zero, we have a fixed-point estimate:

$$\gamma_{dtj} \propto \beta_{jv}\exp[\tilde{m}_{dtj} + \frac{1}{T}\eta_{cj} - (\boldsymbol{h}^T\boldsymbol{\gamma}_{dt}^{old})^{-1}h_j].$$

We can then normalize γ_{dtj} so that $\sum_j \gamma_{dtj} = 1$.

Appendix A.3. Variational Gaussian

$$\begin{aligned} L[\hat{\delta}_{dtj}] \quad \geq \quad & \sum_{\tau=t-1}^{T}\sum_{j=1}^{J}\gamma_{d\tau j}\left[\tilde{m}_{d\tau j}-\log\Big(\sum_{l=1}^{J}\exp[\tilde{m}_{d\tau l}+\frac{\tilde{v}_{d\tau l}}{2}]\Big)\right] \\ & -\frac{1}{2}\sum_{\tau=t-1}^{T}(\tilde{\boldsymbol{m}}_{d\tau}-\boldsymbol{\Phi}\tilde{\boldsymbol{m}}_{d\tau-1})'\boldsymbol{\Sigma}^{-1}(\tilde{\boldsymbol{m}}_{d\tau}-\boldsymbol{\Phi}\tilde{\boldsymbol{m}}_{d\tau-1}). \end{aligned}$$

We estimate $\hat{\delta}_{dtj}$ using simulated annealing.

$$\begin{aligned} L[\hat{\sigma}^2] \quad \geq \quad & \sum_{t=1}^{T}\sum_{j=1}^{J}\gamma_{dtj}\left[\tilde{m}_{dtj}-\log\Big(\sum_{l=1}^{J}\exp[\tilde{m}_{dtl}+\frac{\tilde{v}_{dtl}}{2}]\Big)\right] \\ & -\frac{1}{2}\sum_{t=1}^{T}tr\Big[\boldsymbol{\Sigma}^{-1}(\tilde{\boldsymbol{V}}_{dt}+\boldsymbol{\Phi}\tilde{\boldsymbol{V}}_{dt}\boldsymbol{\Phi}')\Big] \\ & -\frac{1}{2}\sum_{t=1}^{T}(\tilde{\boldsymbol{m}}_{dt}-\boldsymbol{\Phi}\tilde{\boldsymbol{m}}_{dt-1})'\boldsymbol{\Sigma}^{-1}(\tilde{\boldsymbol{m}}_{dt}-\boldsymbol{\Phi}\tilde{\boldsymbol{m}}_{dt-1}) \\ & +\frac{1}{2}\sum_{t=1}^{T}\log|\tilde{\boldsymbol{V}}_{d,t}|. \end{aligned}$$

We obtain $\hat{\sigma}^2$ using DBCPOL in Fortran, which minimizes a function using a direct search complex algorithm.

Appendix A.4. Estimation for sDCTM

In this section, we provide update equations to estimate the model parameters based on the lower bound on the log likelihood of the data.

$$l(\boldsymbol{\Phi},\boldsymbol{\Sigma},\boldsymbol{\beta},\boldsymbol{\eta}) \quad = \quad \sum_{d=1}^{M}\log p(\mathcal{D}_d,\boldsymbol{y}_d|\boldsymbol{\Phi},\boldsymbol{\Sigma},\boldsymbol{\beta},\boldsymbol{\eta}).$$

Then,

$$\begin{aligned} l(\boldsymbol{\Phi},\boldsymbol{\Sigma},\boldsymbol{\beta},\boldsymbol{\eta}) \quad \geq \quad & \sum_{d=1}^{M}\sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{i=1}^{V}\gamma_{dtj}w_{dt}^{i}\log\beta_{ji} \\ & +\sum_{d=1}^{M}\sum_{t=1}^{T}\sum_{j=1}^{J}\gamma_{dtj}\left[\tilde{m}_{dtj}-\log\Big(\sum_{l=1}^{J}\exp[\tilde{m}_{dtl}+\frac{\tilde{v}_{dtl}}{2}]\Big)\right] \\ & -\frac{MTJ}{2}\log(2\pi)-\frac{MT}{2}\log|\boldsymbol{\Sigma}|-\frac{1}{2}\sum_{d=1}^{M}\sum_{t=1}^{T}tr\Big[\boldsymbol{\Sigma}^{-1}(\tilde{\boldsymbol{V}}_{dt}+\boldsymbol{\Phi}\tilde{\boldsymbol{V}}_{dt}\boldsymbol{\Phi}')\Big] \\ & -\frac{1}{2}\sum_{d=1}^{M}\sum_{t=1}^{T}(\tilde{\boldsymbol{m}}_{dt}-\boldsymbol{\Phi}\tilde{\boldsymbol{m}}_{dt-1})'\boldsymbol{\Sigma}^{-1}(\tilde{\boldsymbol{m}}_{dt}-\boldsymbol{\Phi}\tilde{\boldsymbol{m}}_{dt-1}) \\ & +\sum_{d=1}^{M}\left[\boldsymbol{\eta}_{c_d}^{T}\bar{\gamma}_d-\log\left(\sum_{l=1}^{C}\prod_{t=1}^{T}\Big(\sum_{j=1}^{J}\gamma_{dtj}\exp(\frac{1}{T}\eta_{lj})\Big)\right)\right] \\ & +\frac{MTJ}{2}\log(2\pi)+\frac{1}{2}\sum_{d=1}^{M}\sum_{t=1}^{T}\log|\tilde{\boldsymbol{V}}_{d,t}|+\frac{MTJ}{2}-\sum_{d=1}^{M}\sum_{t=1}^{T}\sum_{j=1}^{J}\gamma_{dtj}\log(\gamma_{dtj}). \end{aligned}$$

Appendix A.5. Conditional Gaussian

$$\begin{aligned} L_{[\mathbf{\Phi}]} \quad \geq \quad & \sum_{d=1}^{M}\sum_{t=1}^{T}\sum_{j=1}^{J}\gamma_{dtj}\left[\tilde{m}_{dtj} - \log\Big(\sum_{l=1}^{J}\exp[\tilde{m}_{dtl} + \frac{\tilde{v}_{dtl}}{2}]\Big)\right] \\ & -\frac{1}{2}\sum_{d=1}^{M}\sum_{t=1}^{T} tr\Big[\mathbf{\Sigma}^{-1}(\tilde{\mathbf{V}}_{dt} + \mathbf{\Phi}\tilde{\mathbf{V}}_{dt}\mathbf{\Phi}')\Big] \\ & -\frac{1}{2}\sum_{d=1}^{M}\sum_{t=1}^{T}(\tilde{\boldsymbol{m}}_{dt} - \mathbf{\Phi}\tilde{\boldsymbol{m}}_{dt-1})'\mathbf{\Sigma}^{-1}(\tilde{\boldsymbol{m}}_{dt} - \mathbf{\Phi}\tilde{\boldsymbol{m}}_{dt-1}) \\ & +\frac{1}{2}\sum_{d=1}^{M}\sum_{t=1}^{T}\log|\tilde{\mathbf{V}}_{d,t}|. \end{aligned}$$

We estimate $\mathbf{\Phi}$ using simulated annealing.

$$\begin{aligned} L_{[\mathbf{\Sigma}]} \quad \geq \quad & \sum_{d=1}^{M}\sum_{t=1}^{T}\sum_{j=1}^{J}\gamma_{dtj}\left[\tilde{m}_{dtj} - \log\Big(\sum_{l=1}^{J}\exp[\tilde{m}_{dtl} + \frac{\tilde{v}_{dtl}}{2}]\Big)\right] \\ & -\frac{MT}{2}\log|\mathbf{\Sigma}| - \frac{1}{2}\sum_{d=1}^{M}\sum_{t=1}^{T} tr\Big[\mathbf{\Sigma}^{-1}(\tilde{\mathbf{V}}_{dt} + \mathbf{\Phi}\tilde{\mathbf{V}}_{dt}\mathbf{\Phi}')\Big] \\ & -\frac{1}{2}\sum_{d=1}^{M}\sum_{t=1}^{T}(\tilde{\boldsymbol{m}}_{dt} - \mathbf{\Phi}\tilde{\boldsymbol{m}}_{dt-1})'\mathbf{\Sigma}^{-1}(\tilde{\boldsymbol{m}}_{dt} - \mathbf{\Phi}\tilde{\boldsymbol{m}}_{dt-1}) \\ & +\frac{1}{2}\sum_{d=1}^{M}\sum_{t=1}^{T}\log|\tilde{\mathbf{V}}_{d,t}|. \end{aligned}$$

We estimate $\mathbf{\Sigma}$ using simulated annealing.

Appendix A.6. Conditional Multinomial

$$\begin{aligned} L_{[\beta_{ji}]} \quad &= \quad \sum_{d=1}^{M}\sum_{t=1}^{T}\sum_{j=1}^{J}\sum_{i=1}^{V}\gamma_{dtj}w_t^i\log\beta_{ji} + \sum_{j=1}^{J}\Lambda_j(\sum_{i=1}^{V}\beta_{ji} - 1) \\ \frac{\partial L}{\partial\beta_{ji}} \quad &= \quad \sum_{d=1}^{M}\sum_{t=1}^{T}\gamma_{dtj}w_t^i\frac{1}{\beta_{ji}} + \Lambda_j. \end{aligned}$$

This leads to a fixed-point update:

$$\beta_{ji} \quad \propto \quad \sum_{d=1}^{M}\sum_{t=1}^{T}\gamma_{dtj}w_t^i$$

We can then normalize $\boldsymbol{\beta}_j$.

Appendix A.7. Softmax Regression

$$L_{[\eta_{1:C}]} \quad = \quad \sum_{d=1}^{M}\left[\boldsymbol{\eta}_{c_d}^T\bar{\boldsymbol{\gamma}}_d - \log\left(\sum_{l=1}^{C}\prod_{t=1}^{T}\Big(\sum_{j=1}^{J}\gamma_{dtj}\exp(\frac{1}{T}\eta_{lj})\Big)\right)\right].$$

We estimate the $\boldsymbol{\eta}$ using simulated annealing. We add an $L2$ regularization term to prevent overfitting and prevent the inflation of parameter values.

References

1. Le Nguyen, T.; Gsponer, S.; Ilie, I.; O'reilly, M.; Ifrim, G. Interpretable Time Series Classification using Linear Models and Multi-resolution Multi-domain Symbolic Representations. *Data Min. Knowl. Discov.* **2019**, *33*, 1183–1222. [CrossRef]
2. Neuhauser, D.S.; Allen, R.M.; Zuzlewski, S. Northern California Earthquake Data Center: Data Sets and Data Services. In *AGU Fall Meeting Abstracts*; American Geophysical Union: Washington, DC, USA, 2015; Volume 2015, p. S53A-2774.
3. Olszewski, R.T. *Generalized Feature Extraction for Structural Pattern Recognition in Time-Series Data*; Carnegie Mellon University: Pittsburgh, PA, USA, 2001.
4. Bagnall, A.; Lines, J.; Bostrom, A.; Large, J.; Keogh, E. The Great Time Series Classification Bake Off: A Review and Experimental Evaluation of Recent Algorithmic Advances. *Data Min. Knowl. Discov.* **2017**, *31*, 606–660. [CrossRef]
5. Billingsley, P. Statistical Methods in Markov Chains. In *The Annals of Mathematical Statistics*; Institute of Mathematical Statistics: Hayward, CA, USA, 1961; pp. 12–40.
6. Fahrmeir, L.; Kaufmann, H. Regression Models for Non-Stationary Categorical Time Series. *J. Time Ser. Anal.* **1987**, *8*, 147–160. [CrossRef]
7. Fokianos, K.; Kedem, B. Prediction and Classification of Non-Stationary Categorical Time Series. *J. Multivar. Anal.* **1998**, *67*, 277–296. [CrossRef]
8. Navarro, G. A Guided Tour to Approximate String Matching. *ACM Comput. Surv. (CSUR)* **2001**, *33*, 31–88. [CrossRef]
9. Jurafsky, D.; Martin, J.H. Naïve Bayes Classifier Approach to Word Sense Disambiguation. In *Computational Lexical Semantics*; University of Groningen: Groningen, The Netherlands, 2009.
10. Deshpande, M.; Karypis, G. Evaluation of Techniques for Classifying Biological Sequences. In Proceedings of the Pacific-Asia Conference on Knowledge Discovery and Data Mining, Taipei, Taiwan, 6–8 May 2002; Springer: Berlin/Heidelberg, Germany, 2002; pp. 417–431.
11. Stoffer, D.S.; Tyler, D.E.; Wendt, D.A. The Spectral Envelope and its Applications. In *Statistical Science*; Institute of Mathematical Statistics: Hayward, CA, USA, 2000; pp. 224–253.
12. Li, Z.; Bruce, S.A.; Cai, T. Classification of Categorical Time Series Using the Spectral Envelope and Optimal Scalings. *arXiv* **2021**, arXiv:2102.02794.
13. Yan, X.; Guo, J.; Liu, S.; Cheng, X.; Wang, Y. Learning Topics in short texts by Non-Negative Matrix Factorization on Term Correlation Matrix. In Proceedings of the 2013 SIAM International Conference on Data Mining. SIAM, Austin, TX, USA, 2–4 May 2013; pp. 749–757.
14. Blei, D.M.; Ng, A.Y.; Jordan, M.I. Latent Dirichlet Allocation. *J. Mach. Learn. Res.* **2003**, *3*, 993–1022.
15. Roberts, M.E.; Stewart, B.M.; Tingley, D.; Airoldi, E.M. The Structural Topic Model and Applied Social Science. In *Advances in Neural Information Processing Systems Workshop on Topic Models: Computation, Application, and Evaluation. Harrahs and Harveys, Lake Tahoe*; Harvard University: Cambridge, MA, USA, 2013; Volume 4, pp. 1–20.
16. Blei, D.M.; Lafferty, J.D. Dynamic Topic Models. In Proceedings of the 23rd International Conference on Machine Learning, Pittsburgh, PA, USA, 25–29 June 2006; pp. 113–120.
17. Wang, C.; Blei, D.; Heckerman, D. Continuous Time Dynamic Topic Models. *arXiv* **2012**, arXiv:1206.3298.
18. Ionescu-Tîrgovişte, C.; Gagniuc, P.A.; Guja, C. Structural properties of gene promoters highlight more than two phenotypes of diabetes. *PLoS ONE* **2015**, *10*, e0137950. [CrossRef] [PubMed]
19. Coles, R.; Caswell, R.; Rubinsztein, D.C. Functional analysis of the Huntington's disease (HD) gene promoter. *Hum. Mol. Genet.* **1998**, *7*, 791–800. [CrossRef]
20. Oubounyt, M.; Louadi, Z.; Tayara, H.; Chong, K.T. DeePromoter: Robust Promoter Predictor using Deep Learning. *Front. Genet.* **2019**, *10*, 286. [CrossRef] [PubMed]
21. Le, N.Q.K.; Yapp, E.K.Y.; Nagasundaram, N.; Yeh, H.Y. Classifying Promoters by Interpreting the Hidden Information of DNA Sequences via Deep Learning and Combination of Continuous FastText N-Grams. *Front. BIoeng. BIotechnol.* **2019**, *7*, 305. [CrossRef] [PubMed]
22. Li, Q.Z.; Lin, H. The Recognition and Prediction of σ70 Promoters in Escherichia Coli K-12. *J. Theor. Biol.* **2006**, *242*, 135–141. [CrossRef] [PubMed]
23. Kullback, S.; Leibler, R.A. On Information and Sufficiency. *Ann. Math. Stat.* **1951**, *22*, 79–86. [CrossRef]
24. Blei, D.M.; Kucukelbir, A.; McAuliffe, J.D. Variational Inference: A Review for Statisticians. *J. Am. Stat. Assoc.* **2017**, *112*, 859–877. [CrossRef]
25. Visual Numerics. *IMSL® Fortran Numerical Math Library*; Visual Numerics Inc.: Houston, TX, USA, 2007.
26. Dempster, A.P.; Laird, N.M.; Rubin, D.B. Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. R. Stat. Soc. Ser. B (Methodol.)* **1977**, *39*, 1–22. [CrossRef]
27. Lütkepohl, H. *New Introduction to Multiple Time Series Analysis*; Springer Science & Business Media: New York, NY, USA, 2005.
28. Goffe, W.L.; Ferrier, G.D.; Rogers, J. Global Optimization of Statistical Functions with Simulated Annealing. *J. Econom.* **1994**, *60*, 65–99. [CrossRef]
29. Rajasekaran, S. On Simulated Annealing and Nested Annealing. *J. Glob. Optim.* **2000**, *16*, 43–56. [CrossRef]
30. Brainerd, W. Fortran 77. *Commun. ACM* **1978**, *21*, 806–820. [CrossRef]
31. Czibula, G.; Bocicor, M.I.; Czibula, I.G. Promoter sequences prediction using relational association rule mining. *Evol. Bioinform.* **2012**, *8*, EBO-S9376. [CrossRef] [PubMed]

32. Schneider, T.D.; Stephens, R.M. Sequence logos: A new way to display consensus sequences. *Nucleic Acids Res.* **1990**, *18*, 6097–6100. [CrossRef] [PubMed]
33. Zhao, Y. *R and Data Mining: Examples and Case Studies*; Academic Press: New York, NY, USA, 2012.
34. Venables, W.N.; Ripley, B.D. *R Package Class Modern Applied Statistics with S*, 4th ed.; Springer: New York, NY, USA, 2002; ISBN 0-387-95457-0.
35. Percival, D.B.; Walden, A.T. *Wavelet Methods for Time Series Analysis*; Cambridge University Press: Cambridge, UK, 2000; Volume 4.
36. Strobl, C.; Malley, J.; Tutz, G. An introduction to recursive partitioning: Rationale, application, and characteristics of classification and regression trees, bagging, and random forests. *Psychol. Methods* **2009**, *14*, 323. [CrossRef] [PubMed]
37. Meyer, D.; Wien, F. Support Vector Machines. *R News* **2001**, *1*, 23–26.