electronics

MDPI

Article

A Survey of One Class E-Commerce Recommendation System Techniques

Mohamed Khoali [1,*], Yassin Laaziz [1], Abdelhak Tali [1] and Habeeb Salaudeen [2]

[1] LabTIC, National School of Applied Sciences, Abdelmalek Essaadi University, Campus Ziaten, Tangier 90000, Morocco; ylaaziz@uae.ma (Y.L.); sakom@hotmail.com (A.T.)
[2] Department of Computer Engineering, Faculty of Engineering, Cyprus International University, 99010 Nicosia, Cyprus; hksalaudeen@gmail.com
* Correspondence: khoali.med@gmail.com

Abstract: Although several recommendation algorithms are widely used by both commercial and non-commercial platforms, they face unique challenges such as sparse data sets and the absence of negative or "neutral" feedback. One-class algorithms attempt to overcome the data sparsity problem by using the implicit feedback inherent in user's clicks and purchases, which are deduced from both positive and negative feedback. Existing literature uses several heuristic strategies to derive the negative samples needed for training, such as using random sampling or utilizing user-item interaction. However, these assumptions do not always reflect reality. In addition, with the explosive increase in the availability of big data for training recommendation systems, these methods might not adequately encapsulate the representations of the latent vectors. In this paper, we address the common issues of one-class recommendation and provide a survey on approaches that have been used to mitigate the existing challenges. To tackle the identified problems, we propose a neural network-based Bayesian Personalized Ranking (BPR) for item recommendation and personalized ranking from implicit feedback. BPR provides an optimization criterion derived from Bayesian analysis of a problem to develop an optimized model for such a problem. We conduct several experiments on two varieties of MovieLens datasets to illustrate the performance of the proposed approach. Our approach shows an impressive result in mitigating the issues of one-class recommendation when compared with the complexity of the state-of-the-art methods.

Keywords: one class recommendation; Bayesian Personalized Ranking (BPR); collaborative filtering; matrix factorization

check for updates

Citation: Khoali, M.; Laaziz, Y.; Tali, A.; Salaudeen, H. A Survey of One Class E-Commerce Recommendation System Techniques. *Electronics* **2022**, *11*, 878. https://doi.org/10.3390/electronics11060878

Academic Editor: Ricardo Santos

Received: 28 January 2022
Accepted: 7 March 2022
Published: 10 March 2022

1. Introduction

Many companies have realized the importance of predictive analysis methods facilitated with machine learning, especially the remarkable results from deep learning algorithms. This creates a justification for the wide adoption and growing interest in the use of recommender systems by e-commerce companies to boost sales. Ko et al. [1] identified the application domains of recommender systems as streaming service, social network service, tourism service, e-commerce service, healthcare service, education service, and academic information service.

The recommender system aims to predict user intents and provide product recommendation that is likely to be of interest to the customers; therefore, they are employed to help users discover new products and services. Every time you shop, a recommendation system is guiding you towards the most likely product you might purchase using your interaction history.

Recommender systems have evolved and are typically classified according to their application domains [1] or whether they are generally applicable to different domains. Recent recommender systems are further classified by Beheshti et al. [2] into sequential

recommenders such as Markov Chain-based methods, factorization-based methods, neural network-based models, content-aware recommenders, auxiliary information-based recommenders, and intelligent recommenders.

Recommendation systems rely on the type of input data that is used and can be grouped into two categories: explicit and implicit feedback. Explicit feedback, in this case, means that a user directly gave a report about a specific product. Implicit feedback is mostly driven by user behavior, such as eye-tracking [3] and browser event tracking [3,4], which records his preferences and other indirect relationships with the item. Collaborative Filtering methods rely on explicit data. In addition, aesthetic preference [5] has been used to provide recommendations in some studies.

Collaborative Filtering (CF) presents two entities that a model is based on, the user and the item, and two techniques have been identified in the literature to facilitate collaborative filtering. They are the neighborhood approach [6–8] and latent factor models [9–12]. The neighborhood approach models the relationships between the items or the users.

Latent factor models for recommender systems, mostly achieved by matrix factorization, transform the agents (users and items variables) into a single latent feature space of similar sizes, then employ an inner product that will model the relationship between a user and an item. In better terms, the user's factors are used to encode the preferences of the user; the item's factor will encode the properties of the item; the inner product applied to both factors encodes the distance in terms of attraction of the item to the user's preferences.

One-class recommendation [8] is exploited in scenarios where explicit numerical ratings are not available. This technique relies on modeling implicit feedback from users. Implicit feedback can be browsing history, media preferences, social activities, etc.

Most systems use implicit features to model preferences because they are easier to obtain, it also poses a challenge where data sparsity exists that makes it hard to identify representative negative examples (negative examples are mixed with missing positive examples). Therefore, the one-class recommendation uses only positive examples. Pan et al. [13] identified the different strategies to solve the one-class problem. They include:

1. Labeling negative examples to convert the data into a collaborative filtering problem.
2. Treating all the missing data as negative examples.
3. Treating all the missing data as unknown (ignored and not used for training).

Many existing methods for handling one-class problems based on these strategies are complicated and based on mere heuristic approaches. These methods are not always effective when tackling state-of-the-art recommendation tasks involving the vast amount of data that is now available in e-commerce. Therefore, we have identified some misconceptions and limitations to the one-class recommendation in this study.

In addition, we have used matrix factorization in collaboration with BPR to construct a deep neural network that preserves the simplicity of the solution while also increasing its ability to deal with non-linearity in the data. Using the Bayesian Personalized Ranking (BPR) method, the one-class problem of recommending items is transformed into a ranking problem. User-specific rankings of sets of items are established through a generative process, in which users and items are connected through missing samples. The probability that a user will consider an item is modeled by first modeling the odds that the user will consider an item and then eliciting a probability that the item will be viewed. Item pairs are used for training data to help optimize the ranking item pairs through prediction, which is achieved by approximating the matrix obtained from the dot product of user and item matrices using low-rank matrices.

This paper is structured as follows. Sections 2 and 3 present the background on Collaborative filtering and One class recommendation thereby introducing the existing challenges of the techniques. Section 4 presents a review of related works on solving one class recommendation problems. Section 5 provides information about the method proposed by the authors while Section 6 gives information about the dataset and the detailed experiment done. Section 7 concludes the work and presents future works.

2. Collaborative Filtering

We can consider a scenario where we have ratings for u users of an e-commerce service and i items/products for which the users gave a rating. Let the rating be denoted by r where rating between a user and an item is denoted as r_ui. The rating ranges from one to five stars; five is the highest number indicating a stronger preference by the user for the item he/she rated, while one is the lowest number indicating a weak preference or dislike by the user for the item that was rated.

Each user u is associated with a given vector p_u and every item is also associated with a vector $q\_(i)$ which belongs to the same latent space as the user vector. Therefore, given an item i, the elements of its vector q_i measure the extent to which the item possesses the stated factors (either negative or positive) and for a given user u, the elements of its vector $p\_(u)$ measure the extent of the attraction the user has for items that are the desired factors (either negative or positive). Thereafter, the dot product of both vectors (user and items) models the relationship between the user u and the item i (the total interest of the user in respect to an item). The rating is, thus, predicted by adding the results of the dot product, the overall average rating in the set, and the biases of both the user and the item.

Matrix Factorization–Problems and Solutions

Numerous works of literature have proposed variations of the Matrix factorization techniques, which are based on both implicit and explicit feedback.

Maximum Margin Matrix Factorization (X) [6–9,13–21] employed the use of norms of the U and V in place of dimensionality where U, the coefficient matrix, is identified as the product of the user's matrix (U) and the preference model (K), while V represents that of the item's matrix (I). By doing this, only important factors are modeled such that the very strong factors corresponding to the factors are given preferences above less important features. Weimer et al. [18] extended this technique to structured ranking losses. The extensions identified are the efficient computation of the gradient in multi-class ordinal regression, the introduction of bias for users and items, automated adaptive regularization scheme, and graph kernel to capture the similarities between users and items.

It is important to identify the successful algorithms when it comes to recommending items to users. First, most algorithms do not scale well with a huge dataset and, secondly, the algorithms do not do well when users have few ratings (a case of data sparsity). A common practice is data cleaning, which most research has generalized to. It involves removing entries that have few user ratings and not use them while training the model. However, Probabilistic Matrix Factorization [14,22–24] provides a way to avoid these challenges.

Probabilistic Matrix Factorization [14] models the user preference matrix (U) as a product of two lower-rank user and item matrices (I).

R_{ij} represents the rating of user I for item j, and V is the latent user and item feature matrices of the representation; U_i and V_j in the matrix represents the user-specific and item-specific latent feature vectors, respectively. A probabilistic linear model is developed using Gaussian observation noise with its probability density function as a normal distribution and the conditional distribution over the observed ratings is the probability density function of the Gaussian distribution with the mean and variance. An indicator function is equal to 1 if $i > j$ or if otherwise. Zero-mean spherical Gaussian priors are also placed on the user and item feature vectors and log of the posterior distribution determined.

Therefore, once the model is fitted, the users in the set that has few ratings will have feature vectors that are similar to the prior mean or the average user so that the predicted ratings of such users will be similar to the item average ratings.

Salakhutdinov and Mnih [14] emphasized that collaborative filtering methods are prone to overfitting because MAP estimate is mostly used to fit the data. This problem can only be avoided if the regularization parameters are well-tuned. This proposed method, Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo [15], intended to achieve efficient training by using Bayesian Probabilistic Matrix Factorization

(PMF) [25], which automatically controls the model capacity through the model parameters and hyper parameters.

Bayesian PMF [25] represents the prior distributions over the user and item feature vectors as Gaussian and placed Gaussian-Wish art priors on the hyper parameters of the user and item feature vectors. To determine the prediction distribution of the user (u) and item (i), marginalization of the model parameters and hyper parameters is employed. Using variation methods, evaluation of the prediction distribution could be achieved such that deterministic approximation is made for the posteriors.

This method with the Bayesian model boost produces a predictive distribution instead of a single number of other models tend to generate; therefore, it allows the confidence in the predicted values to be quantifiable.

Latent factor models have also been optimized by employing ranking loss as presented in the following literature.

Item recommendation provides a user-specific ranking for a set of items in a system. In addition, a generic learning algorithm, *LEARNBPR* [26,27] is proposed which is based on stochastic gradient descent which shows superiority to normal gradient descent algorithms. Rendle et al. [21] investigates scenarios where implicit data of the user (based on past purchases) is used to provide a ranking for item recommendation. However, implicit feedback systems only record positive observations; therefore, the problem of one-class recommendation is introduced [13,20,21].

The Bayesian method for finding the right personalized ranking for each item in the item set is aims to maximize the posterior probability. Prediction can be achieved by estimating the matrix obtained from the dot product of the user matrix and the item matrix and then approximates the result by the matrix product of low low-rank matrices. For ranking, *BPR-OPT* criterion is optimized by using *LEAARNBPR*.

The ranking recommendation which makes use of user feedbacks is widely used, but it has been noted that the effectiveness of these methods tends to decrease as the sparsity of the data increases. FISM [16] proposed a method that can alleviate this challenge by employing an item-based method which is used to generate ranking top-N recommendations learned from the item-item similarity matrix (the product of low dimensional latent factor matrices). Collaborative Less-is-More Filtering (CLIMF) [17] is developed off Collaborative Filtering where model parameters are learned through direct maximization (optimization) of the Mean Reciprocal Rank which is used to measure the performance of top-k recommendations.

Weston et al. [19] described how complex a user preference is and that it cannot be described by a single user-item representation. The authors proposed the usage of multiple interests in a way that a user set of functions is modeled using a set of latent vectors where each vector captures the user's latent interests.

3. One Class Recommendation

Weston et al. [19] also classified the state-of-the-art in one-class recommendation based on the method used as either point-wise method or pairwise method. The point-wise method attaches a numerical value to each evaluated item. Positive feedbacks are modeled as high preference scores while negative feedbacks are modeled as low preference scores. The pairwise method, on the other hand, models the ranking/order of feedbacks. Using implicit feedback, the relationship shows that users have a higher preference for positive feedback than on negative feedback items.

Pan et al. [13] proposed two methods to tackle the challenge of one-class recommendation, weighted low-rank approximation and negative example sampling. The one-class problem can be regarded as a class imbalance problem at the data level. By sampling, the data can be re-balanced. In the first proposed approach, which uses weighting, sampling strategies, such as all missing data being identified as weak negative and some missing data being identified as negative, are used so as to provide a balance between strategies of

missing as negative and missing as unknown. The second approach samples some missing values as negative examples by using sampling strategies.

Pappas and Popescu-Belis [20] focused on user comments which do not have explicit rating labels; therefore, the problem is modeled as a one-class problem. A sentiment-aware nearest neighbor model is then proposed. Additional user ratings were inferred by performing sentiment analysis of user comments (negative and positive) and integrating its output in a nearest-neighbor model as the prediction function.

The limitations and misconceptions regarding one-class recommendation that were deduced from the study of numerous state-of-the-art are identified below:

1. Data level strategies, such as data sampling, are based on several assumptions which do not robustly represent the underlining problem of one class recommendation. For example, missing data do not explicitly represent a negative example and ignoring this might also lead to leaving out important data that can be crucial during training.
2. Synthesized data might not be appropriate for production recommendation systems since they do not represent the real challenges faced by these systems.
3. Implementing the proposed models may be difficult, and the models themselves may not always be appropriate for, or practically applicable to, the real-world domain.
4. Ranking recommendations, which make use of user feedback, are widely used, but it has been noted that the effectiveness of these methods tend to decrease as the sparsity of the datasets increases.
5. It can be substantiated by the complexity of human minds, which make users' opinions unstable. The mind's fragility may lead to an individual's viewing an item's properties in different ways at once. Therefore, it seems better to view each dimension of a user's preference as a personalized projection of an item or items properties. The derived preference model should be able to capture complex relationships between items' properties and users' preferences.

This study focuses on the application of matrix factorization to one class recommendation problem in the context of simplicity and the ability to solve the problem at hand. We have reviewed several variations of matrix factorization techniques that have been proposed to solve one or more problems of collaborative filtering. In addition, we have also reviewed the one class recommendation problems and evaluated different proposed techniques to mitigate its challenges.

4. Related Works

This study examines how matrix factorization methods can be applied to implicit feedback in the one-class recommendation problem. The main contribution of our work is to apply a deep neural network to enhance already existing algorithms for better results. In this section, we review state-of-the-art on the one-class recommendation methods and machine learning techniques that have been used in these algorithms.

Chen et al. [28] classified state-of-the-art approaches for solving one-class recommendation system problems into methods based on matrix factorization, transfer learning approaches, and deep learning. However, the methods based on matrix factorization have proven to be more popular due to their simplicity. The artificial neural network has proven to be a generic algorithm that can be applied effectively to different application domains and produce remarkable results. Therefore, existing methods are adapting neural networks to produce better results.

Factorization-based methods are further classified by Chen et al. [28], based on the strategies employed into four classes: sampling strategy, weighting strategy [29], introducing constraint strategy, and varying loss function strategy. Transfer learning methods are also classified into adaptive transfer learning, integrative transfer learning, and collective transfer learning. Several deep learning methods have been proposed including Multilayer perceptrons, Attention-based networks, recurrent networks, reinforcement learning, and MN.

Personalized ranking based on Bayesian probability computations assumes that a given user will prefer an item he/she has interacted with to one he/she has not interacted with. This forms the basis for most ranking strategies and subsequent works have extended this assumption to propose different weighting strategies, like Adaptive Bayesian Personalized Ranking (ABPR) [30], which uses auxiliary feedback to learn a confidence score. Potential Preference-based BPR (BPR+) [31], however, extends the initial assumption by using potential scores between users and items in the system, as well as measures for the difference between unobserved items by the users.

Sampling strategies identify a negative feedback subset from the training data that can be used to train the classifier. A view-enhanced BPR sampler [32], for example, employs auxiliary feedback to facilitate sampling: BPR assumptions are used to model users' preferences, but each user item is represented by three parts, target feedback, auxiliary feedback, and no feedback. Three predefined probabilities are used to select an item pair from these three parts; the selected item pair is then used as a training instance. An improved sampling strategy was proposed in MF-BPR [33] that uses only un-iterated items as negative feedback in one class recommendation. Three types of strategies of sampling negative items were further proposed: (i) uniform sampler, (ii) over-sampling of the popularly interacted items from the perspective of popularity skewness problem, and (iii) multi-channel sampler that employed both popularity of the items and the level of the feedback.

Loss function strategies for one-class recommendations involve the adoption of various loss functions to optimize a learning function. While loss functions for one-class recommendations are usually classified into point-wise and pairwise, newer methods have been proposed. Ding et al. [32] employed a different, but similar (same assumptions), loss with BPR. The difference is in the incorporation of a new loss term for the introduced auxiliary feedback. An additional parameter is used to capture the preferences that exist between the user, the item, and the auxiliary feedback. Ding et al. [33] derived intermediate feedbacks from items that have auxiliary feedback using point-wise view-enhanced element-wise alternating least squares. The loss function is derived from the pairwise ranking of the users, items, and auxiliary feedback. Square loss is used within the function to optimize the preference that exists between the user and the item including the auxiliary feedback. It also features a regularization hyper-parameter to overcome overfitting, margin-based loss, and a tunable parameter for the interactions between the item and the user.

Several deep learning methods have been proposed, including MLP [34–36], Attention-based networks, recurrent networks, reinforcement learning [37], and MN, to tackle one-class recommendation systems.

Multi-branch MLP [34–36] has been used to model the interaction functions for auxiliary feedbacks. It captures the different types of feedback which are later concatenated within a single architecture in order to process different types of user behaviors. Gao et al. [35] critiqued the approach go using the inner product of MF for modeling the different types of user behaviors due to its over simplicity. The authors, therefore, model each of the behaviors with a different prediction function, such as MLP.

Auto-encoder application [36,38] to one-class recommendation is rooted in FISM where the latent representation of a user is deducted from his/her interaction with items that are used with the inner product of the item vectors to predict the preference. Few works have been conducted regarding the Auto-encoder application to the one-class recommendation. Wu et al. [36] proposed Queryable Variational Auto-encoder (Q-VAE) as a means to model the relationships between the users and the items. Conditional variational auto-encoders (CVAEs) [38] have also been proposed to exploit the tendency of users with similar characteristics to have the tendency of having an interest in the same items. The method learns label verification signals to ensure an exclusive latent mean factor for users of similar characteristics.

5. Materials and Methods

Bayesian Personalized Ranking (BPR) [21] has been proposed to mitigate the challenges of real-world scenarios, where explicit feedback is not readily available and implicit feedback seems more logical. BPR enables models to differentiate between positive and negative observations using missing values, negative samples, or user-item pairs that have not been observed.

BPR is a pairwise approach to personalized ranking such that it considers a pair of items for each user so that it can optimally approximate the ordering for the pair of items and the user.

As a pairwise personalized ranking loss, it is derived from the maximum posterior estimator. To train a BPR model, the training data must consist of positive and negative pairs of the items. The negative pair can be referred to as the missing values also. However, an assumption that a given user will prefer the positive item to all other items he did not interact with is made.

BPR represents the training data as triplet tuples in the form (user u, positive item I, negative item j) where the user u has a preference for item i over item j. The objective of the BPR loss is to maximize the posterior probability and can be represented as:

$$p(\theta|>_u) \propto \; p(>_u|\theta\,)p(\theta), \tag{1}$$

The parameters of the recommendation model are represented as θ, and $>_u$ represents the desired personalized total ranking of all the items for a given user u. To derive the optimization function for personalized ranking, maximum posterior estimator is used.

$$\begin{aligned} \text{BPR(loss)} \; &= \; In\; p(\theta \mid >_u) \\ &\propto \; In\; p(>_u|\,\theta)p(\theta), \\ = In \prod_{(u,i,j\,\in D)} &\sigma\big(y_{ui} - y_{uj}\big)p(\theta), \end{aligned} \tag{2}$$

$$\begin{aligned} &\sum_{(u,i,j\,\in\,D)} In\;\sigma\big(y_{ui} - y_{uj}\big) + In\; p(\theta) \\ = &\sum_{(u,i,j\,\in\,D)} In\;\sigma\big(y_{ui} - y_{uj}\big) - \lambda_\theta||\theta||^2, \end{aligned}$$

The likelihood function is only a part of the Bayesian modeling technique. A general prior density was additional introduced. Therefore, the generic optimization model is formulated as seen in Equation (2). λ_θ represents the parameters used for regularization, and $y_{ui} - y_{uj}$ captures the important relationship between the user and the pair of items facilitated by the model class (which can be any model such as matrix factorisation).

D is the training set that consists of triplet tuples, I denoted all items in the set such that a user u can like an item and other items are not negative samples. y_{ui} and y_{uj} denote the scores predicted by the model for the user u to the items i and j. Prior $p(\theta)$ denotes the normal distribution with zero mean and variance covariance matrix. Figure 1 highlights the training set D where it is not feasible to learn directly from the set since it consists of missing data; therefore, the missing values are replaced by 0. This poses a problem as the learning model ranks all future ranking predictions from what it has learned from the negative feedback used during training so it only predicts 0s and cannot rank. Figure 2 shows an optimization employed by BPR to create user specific pairwise preference between pair of items with–denoting a user preferring an item j over another item i. Therefore, item pairs are used as training data so ranking item pairs are used instead of just scoring items as negative or positive or replacing missing values with 0.

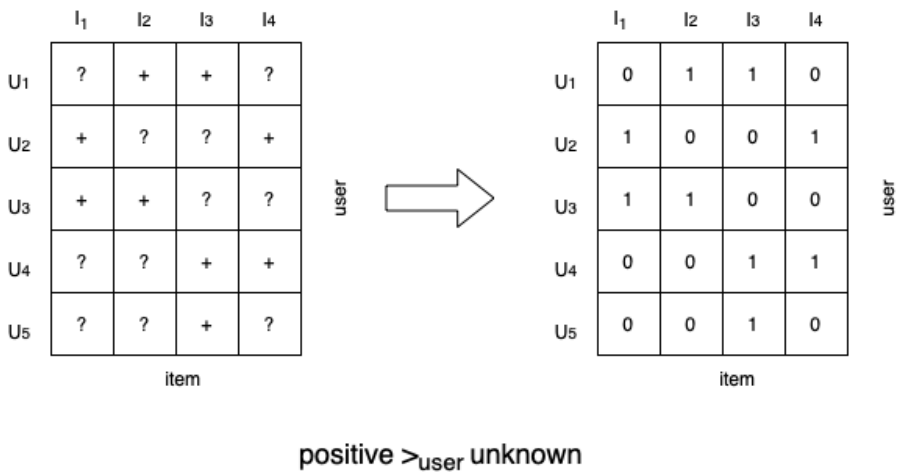

Figure 1. Generation of negative samples for items the user did not interact with [25].

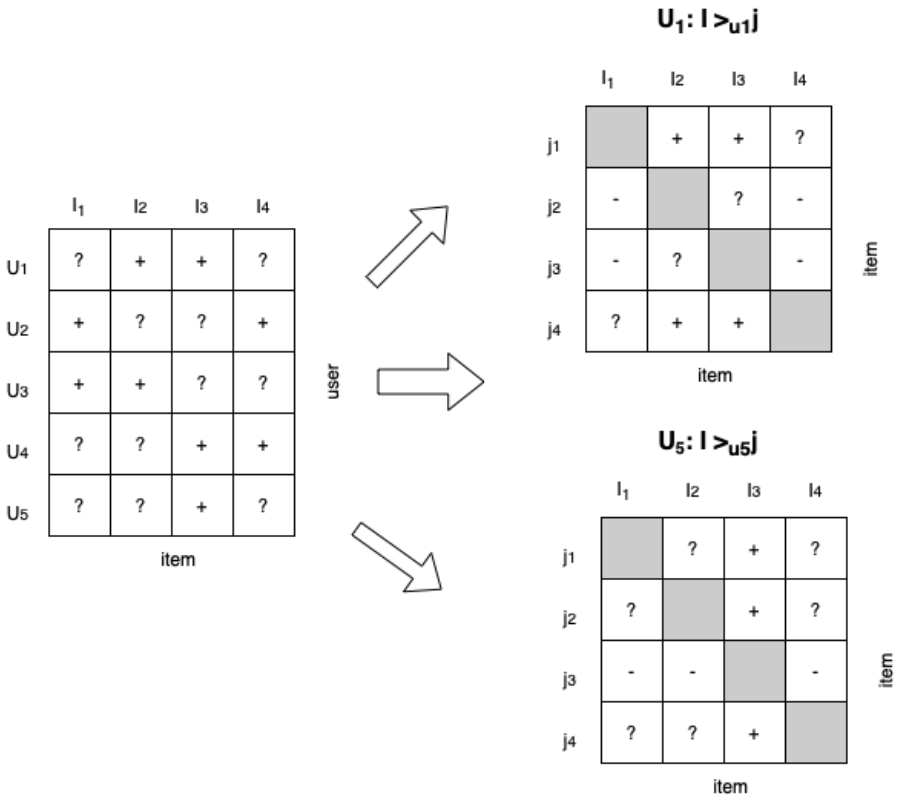

Figure 2. The creation of user specific pairwise preferences between pair of items showing the preference of the user for item *i* over *j*.

6. Experiments

6.1. Dataset

The Movie Lens dataset was collected by GroupLens, a research group that studies the effects of collaborative filtering on recommendation accuracy. The dataset consists of users, movies, and movie ratings for the selected users. Ratings are integer values from one to five, with five being the highest rating and one being the lowest rating.

The model was trained on two datasets. The first dataset contained 1,000,000 entries and the second contained 100,000 entries. While the first dataset initially worked well for training the model, challenges due to server computing capability occurred when a high number of trainable parameters were used. Therefore, the second dataset was more widely used for training and evaluation purposes.

The dataset contains a user ID, movie ID, ratings given by a user to the movie ranging from one to five, and the timestamp.

Experiment I-Bayesian Personalized Ranking. This experiment involves some perquisite steps which are enumerated below:

We created a sorted list of unique movies and unique users, then calculated the number of unique users for each. A dictionary containing each unique user ID was created, and another dictionary for each unique movie ID and a total number of unique movies was created. Next, we added the movie IDs from each unique user to the dictionary, which distinguished each entry's positive reviews; this is the test subset that was not used during training, but these movies should appear in your top recommendations accordingly. Therefore, entries with greater than four ratings were used to create the subtest. The condition existed such that if there exists a user in the subtest with more than 20 reviewed movies and the movies had greater than four ratings was used to create the test set.

From the test set, the ground truth is created, which is a collection of records with each unique user id and the corresponding array of movie ids. The training set is created by combining the test set and the dataset. The ground truth training set is created with ratings greater than three and consists of records with each unique user id and the corresponding array of movie ids.

6.2. Building Triplets

Bayesian personalized Ranking requires for training a triplet of the user, positive item, and negative item. A pair of each positive ranked movie with ratings greater than three and all negative movies which has ratings less than three are created for each user.

6.3. Neural Network

The prediction is achieved by multiplication of the user vector by the item matrix to produce a list of scores.

Loss function-BPR triplet loss (BPR-OPT) and Identity loss were employed.

The input to the network is the triplet from user input, positive item input and negative item input. The output is the triplet loss function.

6.4. Configuration

The network structure is defined by the hyper-parameters provided. The following are the configurations of the network: latent dimension = 350, batch size = 128, learning rate = 0.001, and validation Set = 33% of dataset. The latent dimension signifies the number of user and item latent factors. The batch size of 128 was selected because of the capacity of the machine used for training. Higher batch size could have been used if the capacity of the GPU is much higher. For deep learning models, the learning rate is an important hyper-parameter. Since stochastic gradient descent algorithms allow the setting of learning rate, it is important to have a learning rate that provides suitable training. Due to inability to perform extensive experimentation, the learning rate was selected by first trying out different values until 0.001 was selected to produce the best loss value at a good training speed.

Movie Lens Dataset 1—total number of parameters is 3,411,100 which are all trainable parameters. The training data have lengths of 84, 516, and 129.

The main idea is to extract some movies for users who have large amounts of positive reviews into the test subtest. Two movies are extracted for each user who has more than 20. Movie Lens Dataset 2—total number of parameters is 3,616,900 which are all trainable parameters. The training data have a length of 12,689,903. In total, 4,187,668 records were used for evaluating the performance of the model. Table 1 shows the parameters available at each layer of the network, and the output shape of each of the layers.

Table 1. Network layers.

Layer (Type)	Output Shape	Params	Connected to
Positive_item_input (inputLayer)	[(None,1)]	0	
Negative_item_input (InputLayer)	[(None, 1)]	0	
User_input (InputLayer)	[(None,1)]	0	
Item_embedding (Embedding)	(None, 1, 350)	3,403,400	Positive_item_input[0][0] Negative_item_input[0][0]
User_embedding (Embedding)	(None, 1, 350)	213,500	User_input[0][0]
Flatten_1 (Flatten)	(None, 350)	0	Item_embedding[0][0]
Flatten_2 (Flatten)	(None, 350)	0	Item_embedding[1][0]
Flatten_3 (Flatten)	(None, 350)	0	User_embedding[0][0]
Lambda_4 (Lambda)	(None,1)	0	Flatten_1[0][0] Flatten_2[0][0] Flatten_3[0][0]

Several challenges were experienced during training due to computing power constraints. Due to the high numbers of trainable parameters in the network, powerful GPUs are required to get reasonable training duration, which will facilitate iterative training to get better results. For the experiment, Google Colab, a free cloud-based infrastructure that provides access to computational resources such as processing power and storage for developing machine learning solutions, could not help when computing the triplets' algorithm; therefore, we could not use it for training. A MacBook Air 2020 with the M1 chip was used to train the model. While the CPU was able to compute the triplets in a reasonable time, the GPUs could not be utilized for training.

6.5. Evaluation Metrics

The ROC curve is a measure of the confidence a binary model has in its decisions. It is used to evaluate the probability of a positive classification for a binary prediction model. Area under curve (AUC) is used to measure how well a model can distinguish between two classes. Mean average precision (MAP) quantifies how effective the model is at performing its tasks. A good way to validate the ROC curve is to plot the training and validation loss together, and make sure they are of similar magnitude. The plot in Figure 3 shows that they have related values, thereby validating the effectiveness of the model.

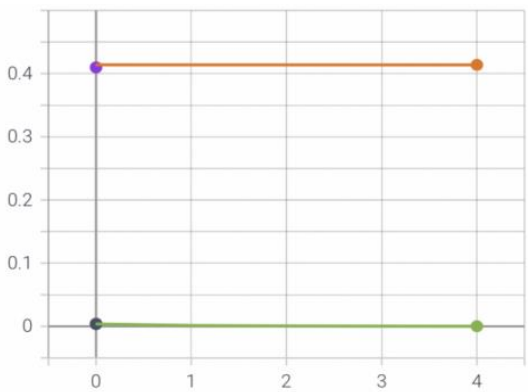

Figure 3. 1–train loss vs. validation loss (orange line represents the validation loss while the green line is the training loss).

6.6. Result

To measure the effectiveness of the method, we used AUC and MAE metrics. AUC is used to measure the area under the ROC curve and presents the probability of the model to rank a random positive example higher than another random negative example. The higher the value (between 0 and 1), the better the prediction correctness of the model. MAE also represents the effectiveness of the model in prediction. There is the mean of average precision for all executed queries. Results are presented in Table 2, which shows that the method performed reasonably well on both datasets with some improvement recorded between datasets. With MovieLens (1,000,000) dataset, AUC and MAP on training data is 0.95 and 0.30, respectively, while on test data, AUC of 0.79 and MaP of 0.12 were recorded. MovieLens (100,000) recorded AUC of 0.86 and MaP of 0.33 on set.

Table 2. Results.

Dataset	Area Under Curve		Mean Average Precision	
MovieLens (1,000,000)	Train	0.95	Train	0.30
	Test	0.79	Test	0.12
MovieLens (100,000)	Train	0.86	Train	0.33

7. Conclusions

In this paper, we have addressed several research issues with the design of recommendation systems. We have identified several techniques that have been used to resolving these issues. These techniques include Maximum Margin Matrix Factorization, Probabilistic Matrix Factorization, and Bayesian Probabilistic Matrix Factorization. By identifying weaknesses in collaborative filtering, we showed the various techniques to overcome those challenges—namely, Bayesian Probabilistic Matrix Factorization.

We have presented the scenario and problems of one-class recommendation. We also surveyed methods for mitigating these problems including FISM and CLIMF.

These methods employed strategies such as weighted low-rank approximation, negative data sampling to re-balance the data and generate new training data and modeling the probability that a user will interact with an item. Bayesian Personalized Ranking (BRP), however, converted the one-class recommendation problem into a ranking problem and solved the problem by proposing a maximum posterior estimator derived from Bayesian analysis. We have converted this solution to be based on a deep neural network structure to further improve the performance of the technique and demonstrated how the proposed deep network based BRP, based on matrix factorization, can be applied to state-of-the-art recommendation systems. The proposed method is applied to the Movie-lens datasets using pairs of movies in the dataset; BPR provides optimization criteria to deduce personalized ranking for each user in the dataset. Instead of treating missing data as negative values, item pairs are used for training the model, thereby eliminating the need to automatically score missing values as negative. The method was tested on the datasets and reasonable result was achieved.

Due to the restricted computing resources, we could not extensively develop or test the algorithm on large datasets. With the availability of better computing resources, we plan to extend this research and explore deep neural networks in greater detail.

In future works, we will be experimenting on real data collected from our e-commerce platform and presenting the results in further published studies.

Author Contributions: Conceptualization, Y.L. and A.T.; methodology, M.K.; software, M.K. and H.S.; validation, Y.L. and A.T.; formal analysis, M.K.; investigation, M.K. and H.S.; resources, M.K.; data curation, M.K. and H.S.; writing—original draft preparation, M.K. and H.S.; writing—review and editing, M.K. and H.S.; visualization, M.K.; supervision, Y.L. and A.T.; project administration,

Y.L. and A.T.; funding acquisition, M.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

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

References

1. Ko, H.; Lee, S.; Park, Y.; Choi, A. A Survey of Recommendation Models, Techniques, and Application Fields. *Electronics* **2022**, *11*, 141. [CrossRef]
2. Beheshti, A.; Yakhchi, S.; Mousaeirad, S.; Ghafari, S.M.; Goluguri, S.R.; Edrisi, M.A. Towards Cognitive Recommender Systems. *Algorithms* **2020**, *13*, 176. [CrossRef]
3. Sulikowski, P.; Zdziebko, T.; Coussement, K.; Dyczkowski, K.; Kluza, K.; Sachpazidu-Wójcicka, K. Gaze and Event Tracking for Evaluation of Recommendation-Driven Purchase. *Sensors* **2021**, *21*, 1381. [CrossRef] [PubMed]
4. Sulikowski, P.; Zdziebko, T.; Hussain, O.; Wilbik, A. Fuzzy Approach to Purchase Intent Modeling Based on User Tracking For E-commerce Recommenders. In Proceedings of the 2021 IEEE International Conference on Fuzzy Systems, Luxembourg, 11–14 July 2021. [CrossRef]
5. Xiao, L.; Mao, H.; Wang, S. Research on Mobile Marketing Recommendation Method Incorporating Layout Aesthetic Preference for Sustainable m-Commerce. *Sustainability* **2020**, *12*, 2496. [CrossRef]
6. Chae, D.K.; Lee, S.C.; Lee, S.Y.; Kim, S.W. On identifying k-nearest neighbours in neighbourhood models for efficient and effective collaborative filtering. *Neurocomputing* **2018**, *278*, 134–143. [CrossRef]
7. Duong, T.N.; Than, V.D.; Tran, T.H.; Dang, Q.H.; Nguyen, D.M.; Pham, H.M. An Effective Similarity Measure for Neighborhood-based Collaborative Filtering. In Proceedings of the 2018 5th NAFOSTED Conference on Information and Computer Science (NICS), Ho Chi Minh City, Vietnam, 23–24 November 2018; pp. 250–254. [CrossRef]
8. Cai, W.; Zheng, J.; Pan, W.; Lin, J.; Li, L.; Chen, L.; Ming, Z. Neighbourhood-enhanced transfer learning for one-class collaborative filtering. *Neurocomputing* **2019**, *341*, 80–87. [CrossRef]
9. Rennie, J.D.M.; Srebro, N. Fast maximum margin matrix factorization for collaborative prediction. In Proceedings of the 22nd International Conference on Machine Learning, Bonn, Germany, 7–11 August 2005; pp. 713–719. [CrossRef]
10. Mongia, A.; Jhamb, N.; Chouzenoux, E.; Majumdar, A. Deep latent factor model for collaborative filtering. *Signal Process.* **2019**, *169*, 107366. [CrossRef]
11. Yu, X.; Jiang, F.; Du, J.; Gong, D. A cross-domain collaborative filtering al-gorithm with expanding user and item features via the latent factor space of auxiliary domains. *Pattern Recognit.* **2020**, *94*, 96–109. [CrossRef]
12. Neve, J.; Palomares, I. Latent factor models and aggregation operators for collaborative filtering in reciprocal recommender systems. In Proceedings of the 13th ACM Conference on Recommender Systems, Copenhagen, Denmark, 16–20 September 2019; pp. 219–227.
13. Pan, R.; Zhou, Y.; Cao, B.; Liu, N.N.; Lukose, R.; Scholz, M.; Yang, Q. One-Class Collaborative Filtering. In Proceedings of the 2008 Eighth IEEE International Conference on Data Mining, Pisa, Italy, 15–19 December 2008; pp. 502–511.
14. Salakhutdinov, R.; Mnih, A. Probabilistic Matrix Factorization. In Proceedings of the 25th International Conference on Machine Learning, Helsinki, Finland, 5–9 July 2008; pp. 880–887.
15. Salakhutdinov, R.; Mnih, A. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In Proceedings of the 25th International Conference on World Wide Web, Montreal, QC, Canada, 11–15 April 2016; pp. 880–887.
16. Kabbur, X.N.S.; Karypis, G. FISM: Factored Item Similarity Models for Top-N Recommender Systems. In Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–14 August 2008; pp. 659–667.
17. Shi, Y.; Karatzoglou, A.; Baltrunas, L. CLiMF: Learning to Maximize Reciprocal Rank with Collaborative Less-is-More Filtering. In Proceedings of the Sixth ACM Conference on Recommender Systems, Dublin, Ireland, 9–13 September 2012; pp. 139–146.
18. Weimer, M.; Karatzoglou, A.; Smola, A. Improving Maximum Margin Matrix Factorization. *Mach. Learn.* **2008**, *72*, 263–276. [CrossRef]
19. Weston, J.; Weiss, R.J.; Yee, H. Nonlinear latent factorization by embedding multiple user interests. In Proceedings of the 7th ACM Conference on Recommender Systems, Hong Kong, China, 12–16 October 2013; pp. 65–68.
20. Pappas, N.; Popescu-Belis, A. Sentiment analysis of user comments for one-class collaborative filtering over ted talks. In Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval, Dublin, Ireland, 28 July–1 August 2013; pp. 773–776. [CrossRef]
21. Rendle, S.; Freudenthaler, C.; Gantner, Z.; Lars, S.T. *BPR: Bayesian Personalized Ranking from Implicit Feedback*; UAI'09: Arlington, VA, USA, 2009; pp. 452–461.
22. Ren, X.; Song, M.; Haiphong, E.; Song, J. Context-aware probabilistic matrix factorisation modeling for point-of-interest recommendation. *Neurocomputing* **2017**, *241*, 38–55. [CrossRef]
23. Huang, L.; Tan, W.; Sun, Y. Collaborative recommendation algorithm based on probabilistic matrix factorisation in probabilistic latent semantic analysis. *Multimed Tools Appl.* **2019**, *78*, 8711–8722. [CrossRef]

24. Li, K.; Zhou, X.; Lin, F.; Zeng, W.; Alterovitz, G. Deep Probabilistic Matrix Factorisation Framework for Online Collaborative Filtering. *IEEE Access* **2019**, *7*, 56117–56128. [CrossRef]
25. Aa, T.V.; Chakroun, I.; Haber, T. Distributed Bayesian Probabilistic Matrix Factorization. *Procedia Comput. Sci.* **2017**, *108*, 1030–1039. [CrossRef]
26. Isinkaye, F.O.; Folajimi, Y.O.; Adeyemo, A.B. On collaborative filtering model optimised with multi-item attribute information space for enhanced recommendation accuracy. *Int. J. Intell. Syst. Technol. Appl.* **2020**, *19*, 3.
27. Cohen, D.; Aharon, M.; Koren, Y.; Somekh, O.; Nissim, R. Expediting Exploration by Attribute-to-Feature Mapping for Cold-Start Recommendations. In Proceedings of the Eleventh ACM Conference on Recommender Systems, Como, Italy, 27–31 August 2017; pp. 184–192.
28. Chen, X.; Li, L.; Pan, W.; Ming, Z. A Survey on Heterogeneous One-class Collaborative Filtering. *ACM Trans. Inf. Syst.* **2020**, *38*, 1–54. [CrossRef]
29. He, X.; Tang, J.; Du, X.; Hong, R.; Ren, T.; Chua, T.-S. Fast Matrix Factorization with Nonuniform Weights on Missing Data. *IEEE Trans. Neural Netw. Learn. Syst.* **2019**, *31*, 2791–2804. [CrossRef]
30. Liu, B.; Su, Y.; Zha, D.; Gao, N.; Xiang, J. CARec: Content-Aware Point-of-Interest Recommendation via Adaptive Bayesian Personalised Ranking. *Aust. J. Intell. Inf. Processing Syst.* **2020**, *15*, 61–68.
31. Zhang, Q.; Ren, F. Potential Preference Based Bayesian Personalised Ranking for One-Class Collaborative Filtering. *Int. J. Adv. Intell.* **2021**, *12*, 113–120.
32. Ding, J.; Yu, G.; He, X.; Feng, F.; Li, Y.; Jin, D. Sampler Design for Bayesian Personalised Ranking by Leveraging View Data. *IEEE Trans. Knowl. Data Eng.* **2021**, *33*, 667–681.
33. Ding, J.; Yu, G.; Li, Y.; He, X.; Jin, D. Improving Implicit Recommender Systems with Auxiliary Data. *ACM Trans. Inf. Syst.* **2020**, *38*, 1–27. [CrossRef]
34. Wen, H.; Liu, X.; Yan, C.; Jiang, L.; Sun, Y.; Zhang, J.; Yin, H. Leveraging multiple implicit feedback for personalised recommendation with neural network. In Proceedings of the 1st International Conference on Artificial Intelligence and Advanced Manufacturing, Dublin, Ireland, 17–19 October 2019; pp. 1–6.
35. Gao, C.; He, X.; Gan, D.; Chen, X.; Feng, F.; Li, Y.; Chua, T.-S.; Yao, L.; Song, Y.; Jin, D. Learning to Recommend with Multiple Cascading Behaviors. *IEEE Trans. Knowl. Data Eng.* **2019**, *33*, 2588–2601. [CrossRef]
36. Wu, G.; Bouadjenek, M.R.; Sanner, S. One-Class Collaborative Filtering with the Queryable Variational Autoencoder. In Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval, Paris, France, 21–25 July 2020; pp. 921–924.
37. Zou, L.; Xia, L.; Du, P.; Zhang, Z.; Bai, T.; Liu, W.; Nie, J.; Yin, D. Pseudo Dyna-Q: A reinforcement learning framework for interactive recommendation. In Proceedings of the 13th International Conference on Web Search and Data Mining, Houston, TX, USA, 3–7 January 2020; pp. 816–824.
38. Pang, B.; Yang, M.; Wang, C. A Novel Top-N Recommendation Approach Based on Conditional Variational Au-to-Encoder. Advances in Knowledge Discovery and Data Mining. *Lect. Notes Comput. Sci.* **2019**, *11440*, 357–368.