

## Research Article

# Neural Personalized Ranking via Poisson Factor Model for Item Recommendation

Yonghong Yu <sup>1</sup>, Li Zhang,<sup>2</sup> Can Wang,<sup>3</sup> Rong Gao,<sup>4</sup> Weibin Zhao,<sup>1</sup> and Jing Jiang<sup>1</sup>

<sup>1</sup>College of Tongda, Nanjing University of Posts and Telecommunications, China

<sup>2</sup>Department of Computer and Information Sciences, Northumbria University, UK

<sup>3</sup>School of Information and Communication Technology, Griffith University, Australia

<sup>4</sup>School of Computer Science, Hubei University of Technology, China

Correspondence should be addressed to Yonghong Yu; yuyh@njupt.edu.cn

Received 22 September 2018; Revised 16 December 2018; Accepted 18 December 2018; Published 3 January 2019

Academic Editor: Rongqing Zhang

Copyright © 2019 Yonghong Yu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Recommender systems have become indispensable for online services since they alleviate the information overload problem for users. Some work has been proposed to support the personalized recommendation by utilizing collaborative filtering to learn the latent user and item representations from implicit interactions between users and items. However, most of existing methods simplify the implicit frequency feedback to binary values, which make collaborative filtering unable to accurately learn the latent user and item features. Moreover, the traditional collaborating filtering methods generally use the linear functions to model the interactions between latent features. The expressiveness of linear functions may not be sufficient to capture the complex structure of users' interactions and degrades the performance of those recommender systems. In this paper, we propose a neural personalized ranking model for collaborative filtering with the implicit frequency feedback. The proposed method integrates the ranking-based poisson factor model into the neural networks. Specifically, we firstly develop a ranking-based poisson factor model, which combines the poisson factor model and the Bayesian personalized ranking. This model adopts a pair-wise learning method to learn the rankings of users' preferences between items. After that, we propose a neural personalized ranking model on top of the ranking-based poisson factor model, named NRPFM, to capture the complex structure of user-item interactions. NRPFM applies the ranking-based poisson factor model on neural networks, which endows the linear ranking-based poisson factor model with a high level of nonlinearities. Experimental results on two real-world datasets show that our proposed method compares favorably with the state-of-the-art recommendation algorithms.

## 1. Introduction

Recommender systems [1] have become an indispensable component in E-commerce, online news and social media sites. These systems alleviate the information overload problem for users, by discovering the users' hidden preferences and providing users with the personalized information, products, or services. With such attractive features, recommender systems are widely employed in many online applications, including Amazon, Youtube, and Netflix.

As one of the most widely used techniques for recommender systems, collaborative filtering (CF) [2] has achieved great success in E-commerce. CF methods, which are independent of specific domains, make recommendations by

analyzing the past activities of users. The main idea of CF is to learn the latent user preferences and the item characteristics by modelling the user-item interaction behaviors. Among a variety of CF methods, matrix factorization (MF) [3, 4] has drawn a large amount of attentions, due to its effectiveness and efficiency for coping with large datasets. MF method assumes that only a few latent factors contribute to the preferences of users and the characteristics of items. Matrix factorization approach simultaneously embeds both the user and item feature vectors into a low-dimensional latent factor space.

Most of the traditional CF methods work on explicit feedback, i.e., the ratings on items given by users. They generally apply the point-wise regression methods to predict

the ratings for unobserved items. However, explicit feedback may not always be available, since it is comparatively difficult to collect. As a result, using the implicit feedback (e.g., clicks, bookmarks and purchases) to express users' preferences is more common in the practical recommender systems. In CF with implicit feedback, only the positive instances are observed, while the negative instances and the missing values are mixed together, which make the personalized recommendation with implicit feedback more challenging.

Collaborative filtering with implicit feedback is referred to as the One-Class Collaborative Filtering (OCCF) problem [5, 6]. In order to solve the OCCF problem, Pan et al. [5] and Hu et al. [6] proposed a Weighted Regularized Matrix Factorization (WRMF) method. Rendle et al. [7] formulated recommendation as a ranking problem and proposed the Bayesian Personalized Ranking (BPR). Zhao et al. [8] proposed the Social Bayesian Personalized Ranking (SBPR) model, which integrates social connections with the users' implicit feedback to estimate users' rankings of items. However, most of the existing methods for OCCF simplify the implicit interactions between users and items. Regardless of how many times a user interacts with an item, they use the binary implicit feedback to indicate whether a user has clicked or viewed an item [5–8]. In other words, the existing methods generally use the matrix factorization models to quantify the binary implicit interactions between users and items. Intuitively, the number of implicit interactions reflects the degree of the user's preferences for items. The larger the number of interactions, the more preferred. Hence, such simplified schemes make the CF methods unable to accurately capture the users' preferences for items.

In addition, as reported in [9], matrix factorization based models use a linear function (i.e., inner product) to model the interactions between the user latent features and the item latent features. The expressiveness of the linear function is too limited to capture the complex structure of users' interactions, which hinders the performance of recommender systems. Hence, He et al. [9] proposed a general framework, named neural collaborative filtering (NCF) for recommender systems and suggested applying neural networks to learn the nonlinear interaction function from data. The nonlinear interaction function learned from the interaction data endows recommender systems with a high level of nonlinearities. Similar to the traditional recommendation methods that work on the explicit feedback, however, the point-wise learning method in NCF degrades the recommendation performance due to the data sparsity issue.

To tackle the aforementioned issues, in this paper, we propose a neural personalized ranking model for collaborative filtering with implicit frequency feedback, which integrates the ranking-based poisson factor model with the neural networks. Specifically, we basically adopt the poisson factor model (PFM) [10, 11], instead of the classical matrix factorization techniques, to model the implicit interactions between users and items. The poisson factor model replaces the usual Gaussian likelihood in the probabilistic matrix factorization with the Poisson likelihood, which guarantees the nonnegativity of latent factors. Moreover, as pointed

out by Ma et al. [10], the poisson factor model is better at modelling the frequency data than the traditional matrix factorization models. However, the data sparsity of implicit frequency feedback limits the performance of the poisson factor model, because the observed feedback available is not sufficient for the poisson factor model to learn latent features. To solve the data sparsity issue, we develop a ranking-based poisson factor model, which combines the poisson factor model and the Bayesian personalized ranking. The ranking-based poisson factor model adopts a pairwise learning method to learn the rankings of preferences between items. In order to capture the complex structure of interactions, moreover, we propose a neural personalized ranking model on top of the ranking-based poisson factor model, called NRPFM. NRPFM integrates the ranking-based poisson factor model with the neural networks, which provides the linear ranking-based poisson factor model with a high level of nonlinearities. In our neural personalized ranking model, we use the multilayer perceptron (MLP) [12] to learn the nonlinear and nontrivial user-item interaction relationships. Hence, our proposed neural personalized ranking model unifies the strengths of the ranking-based poisson model in learning users' preferences ranking between items from implicit frequency feedback, and the neural networks in capturing the nonlinear user-item interaction relationships.

The key contributions of our work are summarized as follows:

- (i) We propose a ranking-based poisson factor model, which combines the poisson factor model and the Bayesian personalized ranking to tackle the data sparsity of implicit frequency feedback.
- (ii) We propose a neural personalized ranking model for collaborative filtering with implicit frequency feedback. This neural personalized ranking model integrates the ranking-based poisson factor model with the neural networks, which endows the linear ranking-based poisson factor model with a high level of nonlinearities.
- (iii) We perform extensive experiments to evaluate our proposed method on real-life datasets. The results show that our proposed method outperforms the state-of-the-art recommendation algorithms.

The rest of this paper is organized as follows. Section 2 briefly reviews the related work in recommender systems. Section 3 introduces some preliminary knowledge. Section 4 describes the details of our proposed item recommendation algorithm. Experiments are evaluated in Section 5. Finally, we conclude this paper and present some directions for future work in Section 6.

## 2. Related Work

In this section, we review the major related work for recommender systems, including the traditional collaborative filtering methods and the neural network-based recommendation methods.

*2.1. Collaborative Filtering.* Collaborative filtering (CF) [2] approaches are widely deployed in the modern E-commerce websites and have achieved a great success. CF approaches include two main categories [2]: memory-based algorithms and model-based algorithms, according to different ways of utilizing a user-item rating matrix.

Memory-based CF algorithms, also known as neighbor-based methods, use the entire user-item rating matrix to generate recommendations. Typical memory-based algorithms include user-based methods [2] and item-based methods [13, 14]. The underlying assumption of memory-based methods is that similar users share common interests, and users usually prefer similar items. The key issue of user-based and item-based methods is to adopt suitable similarity measures to calculate the pairwise similarity between users or between items. Typical similarity measures include the cosine similarity, the Pearson correlation coefficient, and the adjusted cosine similarity [13]. Model-based CF methods firstly learn a predictive model, which characterizes the rating behaviors of users, by exploiting the statistical and machine learning techniques. They use the predictive models to predict users' future behaviors. Typical model-based filtering approaches include Bayesian networks [2], clustering model [15, 16], latent semantic analysis [17, 18], and restricted Boltzmann machines [19].

As the most popular approaches among various CF methods, matrix factorization methods (MF) [3, 4] have attracted a lot of attentions due to their effectiveness and efficiency in dealing with a very large scale user-item rating matrix. The basic assumption of matrix factorization is that only a few latent factors contribute to the preferences of users and the characteristics of items. Therefore, matrix factorization approaches simultaneously embed both user and item feature vectors into a low-dimensional latent factor space, where the correlation between user's preference and item characteristics can be computed directly. Typical matrix factorization approaches include NMF [20], PMF [4], SVD++ [21], and MMMF [22].

These above matrix factorization based recommendation algorithms generally learn the latent feature vectors of users and items from users' explicit feedback (i.e., users' ratings on items). Explicit feedback however may not always be available since it is difficult to collect. So, it is more common for recommender systems to present users' preferences using implicit feedback (e.g., clicks, bookmarks and purchases) in real world, since implicit feedback is relatively easy to obtain. However, only positive instances are observed in implicit feedback, and negative instances and missing values are mixed together, which make the personalized recommendation with implicit feedback more challenging. Collaborative filtering with implicit feedback is referred as the One-Class Collaborative Filtering (OCCF) problem [5, 6]. To solve the OCCF problem, Pan et al. [5] and Hu et al. [6] proposed a Weighted Regularized Matrix Factorization (WRMF) method. WRMF treats all missing entries as negative instances and assigns varying confidence to positive and negative instances. Rendle et al. [7] modeled the rankings of feedback and proposed a Bayesian Personalized Ranking (BPR) criterion for recommendation

systems based on implicit feedback. Pan et al. [23] extended BPR and proposed the Group Bayesian Personalized Ranking (GBPR), via introducing the richer interactions among users. GBPR aggregates the features of similar users in groups to reduce sampling uncertainty. In [8], Zhao et al. proposed the Social Bayesian Personalized Ranking (SBPR) model, which integrates social connections with users' implicit feedback to estimate users' rankings of items. In [24], Zhao et al. utilized the cross-region community matching technique to generate personalized locally interest locations for users. In particular, they proposed the Bayesian probabilistic tensor factorization with social and location regularization (BPTF-SLR) framework to extract users' latent social dimensions from users' implicit feedback. It must be noted that the recommender systems with implicit feedback are more practical than those with explicit feedback. Therefore, the recent research directions on recommendation have been shifted towards learning users' hidden preferences from implicit feedback, rather than inferring users' tastes from explicit feedback.

## *2.2. Neural Networks-Based Recommendation Approaches.*

Recently, many research has employed the neural network technique to design recommendation algorithms, because neural network technique is able to effectively capture the non-linear and non-trivial user-item interaction relationships [25] and extract deep and abstract feature representations for users and items, leading to large improvements in recommendation quality. Representatives of neural network-based recommendation algorithms include Wide & Deep Learning [26], NCF [9], NFM [27], AutoRec [28], CDAE [29], and ConvMF [30].

Among deep neural network techniques, the multilayer perceptron (MLP) [12] is able to approximate measurable function and widely adopted in recommender systems. To provide the App recommendation in Google play, Cheng et al. [26] presented the Wide & Deep Learning approach, which consists of the wide learning model and the deep learning model. The wide learning component is a single layer perceptron and can effectively memorize feature interactions using the cross-product feature transformations. The deep learning component applies the MLP to generalize to the unobserved feature interactions through low-dimensional embeddings. In [9], He et al. proposed a general framework named neural collaborative filtering (NCF) for CF based on neural networks. Specifically, NCF leverages multilayer perceptron to learn the user-item interaction function, which endows NCF modelling with a high level of nonlinearities. Under the NCF framework, three instantiations of NCF are presented, i.e., GMF, MLP, and NeuMF. GMF employs a linear kernel to model the latent feature interactions, and MLP utilizes a nonlinear kernel to the user-item interaction function. Based on GMF and MLP, NeuMF unifies the linearity of GMF and the nonlinearity of MLP for modelling the complex interactions between users and items. Furthermore, Wang et al. [31] extended NCF to solve the cross-domain social recommendation problem (i.e., recommending the relevant items of information domains to the potential users

of social networks) and proposed a neural social collaborative ranking (NSCR) approach. NSCR enhances NCF by plugging a pairwise pooling operation on top of embedding vectors and utilizes the graph regularization technique to model the cross-domain social relations. In addition, in order to simultaneously model the low-order feature interactions and the high-order feature interaction, Guo et al. [32] proposed an end-to-end model, named deepFM, which seamlessly fuses the factorization machine (FM) [33] and MLP. Similar to deepFM, which employs FM and MLP for recommendation, He et al. [27] proposed the neural factorization machine (NFM) for prediction under sparse settings. Unlike other MLP-based methods, NFM introduces a Bi-Interaction pooling component on top of embeddings vectors, which captures the second-order feature interactions in the low-level and greatly facilitates the following hidden layers of NFM to learn the high-order feature interactions. An extension of NFM, called AFM, is also proposed in [34]. AFM takes the importance of different feature interactions into consideration and learns the importance of each feature interaction via a neural attention network.

As one of the core components of deep neural network, autoencoder [35] technique is able to reconstruct inputs in the output layer via a low-dimensional hidden space; some researchers have employed the autoencoder technique in recommender systems to improve the recommendation performance. For example, Sedhain et al. [28] utilized the autoencoder paradigm to make recommendation and proposed AutoRec. Specifically, AutoRec takes the user-partial observed vectors or the item-partial vectors as inputs and embeds them into a low-dimensional hidden space. Finally, AutoRec reconstructs inputs in the output layer by directly optimizing Root Mean Square Error (RMSE). According to the types of inputs, AutoRec includes two variants: U-AutoRec and I-AutoRec, which correspond to take the user-partial observed vectors and the item-partial vectors as inputs, respectively. Compared to the classical autoencoder technique, denoising autoencoder (DAE) [36] techniques are able to discover more robust representations and avoid learning an identity function. In order to take the advantages of DAE, several research work employs DAE techniques for CF [37, 38]. Li et al. [38] proposed the deep collaborative filtering framework (DCF), which unifies the deep learning models with MF based CF. The key deep learning model used in DCF is the marginalized denoising auto-encoders (mDA) [39], which is more computationally efficient and has a closed-form solution to learn model parameters. Moreover, Wu et al. [29] utilized the idea of DAE and proposed the collaborative denoising autoencoder (CDAE) for CF. The key difference between CDAE and the above DAE-based CF methods is that CDAE considers the personalized user factors by encoding a latent vector for each user, which greatly improve the recommendation performance. These above DAE-based recommendation methods [29, 37, 38] assume that the observed user-item interactions are a corrupted version of the user's full preferences and learn the latent representations of the corrupted user-item preferences, which can be used to reconstruct users' full preferences. In contrast, Wang et al. [40] proposed a hierarchical Bayesian

model, called CDL. CDL integrates the stacked denoising autoencoder (SDAE) [41] into PMF [4] and jointly uses SDAE to learn the deep representations for item content and utilizes PMF to perform collaborative filtering for ratings matrix. Note that CDL takes the item features as inputs for SDAE while other DAE-based methods (i.e., AutoRec, DCF, and CDAE) make user feedback as inputs. In other words, CDL utilizes the deep learning component to model the auxiliary information rather than model user behaviors.

Convolution neural network (CNN) [42] is a specialized kind of feedforward neural network for processing the data with grid-like topology. CNN-based recommendation methods usually utilize CNN to extract the deep and abstract feature representations [30, 43–45] from images, audio, and text information. Wang et al. [46] proposed a visual content enhanced point-of-interest (POI) recommendation method (VPOI). VPOI incorporates the visual features extracted via CNN into PMF for learning the latent features of users and POIs. He et al. [44] proposed a visual Bayesian personalized ranking (VBPR) algorithm that incorporates the visual features learned from the product images by CNN into MF. In [45], Oord et al. utilized CNN to extract the latent features from music audio for the music recommendation. Gong et al. [43] formulated the hashtag recommendation task as a multiclass classification problem and proposed an attention based CNN architecture for the hashtag recommendation. Zheng et al. [47] proposed the deep cooperative neural network (DeepCoNN). DeepCoNN consists of two parallel CNNs coupled together by a shared common layer to model the user behaviors and the item properties from reviews. Moreover, Kim et al. [30] integrated CNN into the probabilistic matrix factorization and proposed the convolutional matrix factorization for recommendation (ConvMF). ConvMF utilizes CNN to capture the contextual information of item content and further enhance the rating prediction accuracy. Other neural network-based approaches include the recurrent neural network (RNN) based methods [48–50], the restricted Boltzmann machine (RBM) based methods [19, 51], and the generative adversarial network based methods [52].

There are also other cross-domain multimodal research developments on recommendation systems that provide interesting and insightful discussions to guide future directions. As an example, Nie et al. [53] proposed a scheme to re-rank web images for complex queries from probabilistic perspective. Specifically, they first proposed a heuristic approach to detect noun-phrase based visual concepts from complex query. Then, they proposed a heterogeneous network to automatically estimate the relevance score of each image, which jointly integrates three layer relationships, spanning from semantic level to visual level. Nie et al. [54] focused on a challenging image search performance prediction problem. By analyzing Normalized Discounted Cumulative Gain (NDCG) and Average Precision (AP), they found that only the prediction of the images' relevance probabilities was required to compute their mathematical expectations. Therefore a query-adaptive graph-based learning approach was proposed to estimate the relevance probability of each image to a given query.

The works that are most related to ours are BPR [7] and NCF [9]. Comparing BPR with NRPFM, the major difference between BPR and NRPFM includes three aspects: (1) BPR is designed to learn latent user and item feature vectors from implicit binary feedback, while NRPFM is designed to learn latent user and item representations from implicit frequency feedback, which is more practical in recommendation scenarios; (2) BPR uses a linear function (i.e., inner product) to model the interactions between user latent features and item latent features. By contrast, NRPFM uses a nonlinear function learned from a multilayer perceptron to model the complex interactions between latent user features and latent item features, which endows NRPFM with a high level of nonlinearity; (3) essentially, BPR assumes that users’ implicit feedback follows the Gaussian distribution, which does not fit the heavily skewed frequency data well. But NRPFM assumes that users’ implicit feedback follows the Poisson distribution, which is more suitable for fitting the skewed frequency feedback. In addition, although both NCF and NRPFM leverage the multilayer perceptron to model the nonlinear interactions between latent user features and latent item features, the difference between NRPFM and NCF mainly lies in three aspects. (1) Similar to BPR, NCF is also designed for implicit binary feedback, which is a simplified form of implicit frequency feedback. In contrast, NRPFM directly learns users’ preferences and items’ characteristics from the implicit frequency feedback. (2) Moreover, NCF leverages a point-wise method to learn latent user and item feature vectors, while NRPFM utilizes a pair-wise method to learn latent feature vectors for users and items. For NCF, only the observed feedback has contributions to the learning of latent user and item feature vectors. In contrast, NRPFM makes the missing values also contribute to learning latent user and item features. Hence, NRPFM to some extent better alleviates the data sparsity problem. (3) NCF also assumes that users’ implicit feedback follows the Gaussian distribution, while NRPFM makes a different assumption that users’ implicit feedback follows the Poisson distribution.

Our proposed Neural Personalized Ranking via Poisson Factor Model is an important attempt of this direction. Moreover, we consider more practical recommendation scenarios, in which users represent their preferences using the implicit frequency feedback, rather than using the simplified implicit binary feedback. Furthermore, although it is direct to use deep learning techniques to extend personalized ranking recommendation models (i.e., BPR), this scheme is not able to infer latent user preferences and item characteristics from implicit frequency feedback because traditional ranking based personalized recommendation models essentially are designed to deal with implicit binary feedback. By contrast, our proposed deep learning based personalized ranking model is built on top of BPR, Poisson factor model, and deep learning technique and unify the strengths of these models to more accurately learn latent user and item features from implicit frequency feedback. In short, our contributions are three-fold. Specifically, (1) we use Poisson factor model to model users’ implicit frequency feedback. Subsequently, (2) we utilize the nonlinear function learned from the

deep learning algorithm to model the complex interactions between latent user feature vectors and latent item feature vectors. Finally, (3) in order to train the recommendation model, we sample the set of triplets with partial order from implicit frequency feedback, resulting in alleviating the data sparsity problem.

### 3. Preliminary Knowledge

In this section, we introduce the preliminary knowledge related to our proposed neural personalized ranking based recommendation algorithm. We firstly describe the recommendation problem in Section 3.1. Then, we briefly introduce the Poisson factor model in Section 3.2.

*3.1. Problem Description.* In the typical recommender systems with implicit frequency feedback, users’ implicit feedback is used to construct the user-item interaction matrix  $R \in \mathfrak{R}^{M \times N}$ , which is comprised of two entity sets: the set of  $M$  users  $U = \{u_1, u_2, \dots, u_M\}$  and the set of  $N$  items  $I = \{v_1, v_2, \dots, v_N\}$ . Each entry  $r_{ui}$  of  $R$  represents the number of interactions between user  $u$  and item  $i$ . The number of interactions reflects the users’ preferences for certain items. Note that, in the recommender systems with explicit feedback, the feedback  $r_{ui}$  usually indicates the value of the rating on item  $i$  given by user  $u$ . Ratings are usually integers and fall into  $[0, 5]$ , in which 0 indicates the missing value, since the user has not yet rated that item. In the recommender systems with implicit frequency feedback, however, the feedback  $r_{ui}$  has a larger range compared with ratings. For example, in shopping websites, user may click hundreds of times at some products, while user may click few times for other products. Moreover, 0 is the mixture of the missing value and the negative instance in implicit scenario, which indicates the user is not aware of the item, and the user does not like it, respectively. The set of items interacting with the user  $u$  is denoted as  $I_u$  ( $I_u \subseteq I$ ). In practice, the user-item interaction matrix  $R$  is generally very sparse with many unknown entries, since a typical user may have only interacted with a tiny percentage of items. This sparse nature of the user-item interaction matrix leads to the poor recommendation quality. In this paper, we use “feedback” and “interaction” interchangeably.

The task of recommender systems with implicit frequency feedback is to learn the users’ hidden preferences by utilizing users’ interaction history and provide them with the ranked lists of items that user may be interested in.

*3.2. Poisson Factor Model.* Poisson factor model (PFM) [10] is a generative probabilistic model, which assumes that each observed element  $r_{ui}$  follows the Poisson distribution with the expectation  $f_{ui}$ :  $r_{ui} \sim \text{Poisson}(f_{ui})$ . The expected value matrix  $\mathbf{F} \in \mathbb{R}^{M \times N}$  is factorized into the user latent feature matrix  $\mathbf{P} \in \mathbb{R}^{K \times M}$  and the item latent feature matrix  $\mathbf{Q} \in \mathbb{R}^{K \times N}$ :  $\mathbf{F} \sim \mathbf{P}^T \mathbf{Q}$ . Besides assuming that the Poisson distribution generates the observed elements, PFM places Gamma priors over  $p_{uk}$  and  $q_{ik}$ ,

$$\begin{aligned}
p(p_{uk} | \alpha_k, \beta_k) &= \frac{p_{uk}^{\alpha_k-1} \exp(-p_{uk}/\beta_k)}{\beta_k^{\alpha_k} \Gamma(\alpha_k)}, \\
p(q_{ik} | \alpha_k, \beta_k) &= \frac{q_{ik}^{\alpha_k-1} \exp(-q_{ik}/\beta_k)}{\beta_k^{\alpha_k} \Gamma(\alpha_k)},
\end{aligned} \tag{1}$$

where  $\Gamma(\cdot)$  is the Gamma function.  $\alpha_k$  and  $\beta_k$  are the shape and rate parameters of Gamma distribution, respectively.

The generative process of an observed element  $r_{ui}$  is as follows.

- (1) For each user  $u$ , generate each component of the user latent feature vector:  $p_{uk} \sim \text{Gamma}(\alpha_k, \beta_k)$ .
- (2) For each item  $i$ , generate each component of the item latent feature vector:  $q_{ik} \sim \text{Gamma}(\alpha_k, \beta_k)$ .
- (3) Generate  $r_{ui} \sim \text{Poisson}(\mathbf{p}_u^T \mathbf{q}_i)$ .

The posterior distribution of  $\mathbf{P}$  and  $\mathbf{Q}$  given the user-item interaction matrix  $\mathbf{R}$  is as follows:

$$\begin{aligned}
p(\mathbf{P}, \mathbf{Q} | \mathbf{R}, \boldsymbol{\alpha}, \boldsymbol{\beta}) \\
\propto p(\mathbf{R} | \mathbf{P}) p(\mathbf{P} | \boldsymbol{\alpha}, \boldsymbol{\beta}) p(\mathbf{Q} | \boldsymbol{\alpha}, \boldsymbol{\beta}).
\end{aligned} \tag{2}$$

where  $\boldsymbol{\alpha} = \{\alpha_1, \alpha_2, \dots, \alpha_K\}$ ,  $\boldsymbol{\beta} = \{\beta_1, \beta_2, \dots, \beta_K\}$ .

Maximizing the log-posterior distribution  $p(\mathbf{P}, \mathbf{Q} | \mathbf{R}, \boldsymbol{\alpha}, \boldsymbol{\beta})$  results in the following objective function:

$$\begin{aligned}
\mathcal{L}^{\text{PFM}} &= \min_{\mathbf{P}, \mathbf{Q}} \sum_{u=1}^M \sum_{k=1}^K \left( \frac{p_{uk}}{\beta_k} - (\alpha_k - 1) \ln \left( \frac{p_{uk}}{\beta_k} \right) \right) \\
&+ \sum_{i=1}^N \sum_{k=1}^K \left( \frac{q_{ik}}{\beta_k} - (\alpha_k - 1) \ln \left( \frac{q_{ik}}{\beta_k} \right) \right) \\
&+ \sum_{u=1}^M \sum_{i=1}^N (f_{ui} - r_{ui} \ln f_{ui}) + \text{const.}
\end{aligned} \tag{3}$$

PFM applies the stochastic gradient descent algorithm (SGD) technique to learn the user latent feature matrix  $\mathbf{P}$  and the item latent feature matrix  $\mathbf{Q}$ .

## 4. Our Approach

Our motivation is to learn user preferences and item characteristics from implicit frequency feedback, as well as model the complex structure of user interactions via deep learning. In order to implement this motivation, we do not directly fit the observed implicit frequency feedback, but fit the partial orders of user preferences for items embedded in the triplets, which are sampled from implicit frequency feedback according to our assumption. To endow the ranking-based Poisson factor model with a high level of nonlinearities, we use the multilayer perceptron to learn the nonlinear and nontrivial user-item interaction relationships. In the following sections, we elaborate our proposed neural personalized ranking model that integrates the neural networks into the ranking based poisson factor model for collaborative filtering with implicit frequency feedback.

**4.1. Ranking-Based Poisson Factor Model.** In practical recommender systems, the implicit interactions between users and items usually are displayed in the form of frequency data, which reflects the degree of the user's preferences for items. The larger the number of interactions, the more preferred. Traditional recommendation models generally simplify the implicit frequency feedback to be binary feedback, which to some extents leads to information loss. The Poisson factor model [10] replaces the usual Gaussian likelihood in probabilistic matrix factorization with the Poisson likelihood, which guarantees the nonnegativity of latent factors. Moreover, as reported in [10], Poisson factor model is suitable for modelling the frequency data, which displays similar property to implicit interactions. Hence, we basically adopt the poisson factor model to quantify users' interaction behaviors and learn the latent user features and the item features from implicit frequency feedback. Similar to classical matrix factorization models [3, 4], the Poisson factor model basically adopts the point-wise regression method to learn the latent representations for users and items from the observed feedback. In this sense, only the observed feedback has contributions to the learning of the latent user features and the item features. Due to the data sparsity issue that commonly exists in recommender systems, the available observed feedback is not sufficient for the poisson factor model to learn the latent features, resulting in degrading the performance of recommender systems.

The Bayesian personalized Ranking (BPR) [7] is a popular pairwise learning method for collaborative filtering with binary feedback and has been widely adopted in many recommendation models [8, 23]. BPR learns the latent user and item features by optimizing the Bayesian pairwise ranking criterion. Unlike the point-wise learning methods, BPR assumes that users prefer the observed items over the nonobserved items. In fact, BPR essentially makes the missing values contribute to the training of recommendation model. Hence, this pair-wise learning method somehow alleviates the data sparsity problem.

To tackle the sparsity of implicit frequency feedback, we propose a ranking based Poisson factor model, which combines the Poisson factor model and the Bayesian personalized ranking. The ranking based Poisson factor model adopts a pair-wise learning method to learn the rankings of preferences between items. Specifically, we assume that users' preferences for items increase as the numbers of interactions increase. This assumption implies three aspects: (1) the ranking of preferences for the observed item is higher than that of the preferences for nonobserved item; (2) if two items are observed, a user prefers the one with the larger number of interactions over the another one with the fewer number of interactions; (3) for two nonobserved items, we can not infer the order of their preferences. Let  $r_{ui}$  denote the number of interactions between the user  $u$  and the item  $i$ , and  $r_{uj}$  for the user  $u$  and the item  $j$ . If  $r_{ui} > 0$  and  $r_{uj} = 0$ , then the user  $u$  prefers the item  $i$  over the item  $j$ ; i.e.,  $i >_u j$ , where  $>_u$  indicates the preference relationships between user  $u$  and items. If  $r_{ui} - r_{uj} \geq \tau$  and  $r_{uj} > 0$ , then  $i >_u j$ , where  $\tau$  denotes a threshold parameter. In other words, if the difference between  $r_{ui}$  and  $r_{uj}$  surpasses the threshold  $\tau$ , we assume that the preference

on the item  $i$  is ranked higher than the preference on the item  $j$ . Formally, training set  $D_R$  (i.e., the set of triplet  $(u, i, j)$ ) is defined as follows:

$$D_R = \left\{ (u, i, j) \mid (r_{ui} > 0 \wedge (r_{uj} = 0)) \vee \left( (r_{ui} - r_{uj}) \geq \tau \wedge (r_{uj} > 0) \right) \right\}. \quad (4)$$

Given the preference relationships between the user  $u$  and items  $>_u$ , we maximize the posterior probability  $p(\Theta \mid >_u)$  to learn the latent user and item features.  $\Theta$  denotes model parameters; i.e.,  $\Theta = \{\mathbf{P}, \mathbf{Q}\}$ . Through the Bayesian inference, the posterior probability of  $\mathbf{P}$  and  $\mathbf{Q}$  can be obtained as follows:

$$p(\Theta \mid >_u) \propto p(>_u \mid \Theta) p(\Theta). \quad (5)$$

All users are presumed to be independent of each other, and the preference ranking of one (user,item) pair for a specific user is also assumed to be independent of the rankings of other (user,item) pairs. As a result, the likelihood function for all the users is formulated as

$$\prod_{u \in \mathcal{U}} p(>_u \mid \Theta) = \prod_{(u,i,j) \in \mathcal{U} \times \mathcal{I} \times \mathcal{I}} p(i >_u j \mid \Theta)^{I((u,i,j) \in D_R)} \times (1 - p(i >_u j \mid \Theta))^{I((u,i,j) \notin D_R)}, \quad (6)$$

where  $I(x)$  is the indication function. Based on the totality and antisymmetry properties of preferences relationship, (6) is rewritten as

$$\prod_{u \in \mathcal{U}} p(>_u \mid \Theta) = \prod_{(u,i,j) \in D_R} p(i >_u j \mid \Theta). \quad (7)$$

$p(i >_u j \mid \Theta)$  denotes the probability that the user  $u$  prefers the item  $i$  over the item  $j$  and is defined as

$$p(i >_u j \mid \Theta) = \sigma(\hat{r}_{uij}(\Theta)), \quad (8)$$

where  $\sigma(\cdot)$  is the logistic sigmoid function.  $\hat{r}_{uij}(\Theta)$  is a real value function of model parameters and captures the relationship between the user  $u$ , the item  $i$ , and the item  $j$  and is defined as follows:

$$\hat{r}_{uij}(\Theta) = \hat{r}_{ui}(\Theta) - \hat{r}_{uj}(\Theta) = \mathbf{p}_u^T \mathbf{q}_i - \mathbf{p}_u^T \mathbf{q}_j. \quad (9)$$

In addition, Gamma priors are assumed for the latent user and item feature vectors:

$$p(\Theta) = p(\mathbf{P} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}) p(\mathbf{Q} \mid \boldsymbol{\alpha}, \boldsymbol{\beta}) = \prod_{u=1}^M \prod_{k=1}^K \frac{p_{uk}^{\alpha_k - 1} \exp(-p_{uk}/\beta_k)}{\beta_k^{\alpha_k} \Gamma(\alpha_k)} \times \prod_{i=1}^N \prod_{k=1}^K \frac{q_{ik}^{\alpha_k - 1} \exp(-q_{ik}/\beta_k)}{\beta_k^{\alpha_k} \Gamma(\alpha_k)}. \quad (10)$$

Substitute the likelihood function defined in (7) and the model parameter priors defined in (10) into (5); then maximize the log of the posterior probability; we obtain the

objective function of the ranking-based poisson factor model as follows:

$$\begin{aligned} \mathcal{L}^{RPFM} = \min_{\mathbf{P}, \mathbf{Q}} & -\ln \sigma(\hat{r}_{uij}(\Theta)) \\ & + \sum_{u=1}^M \sum_{k=1}^K \left( \frac{p_{uk}}{\beta_k} - (\alpha_k - 1) \ln \left( \frac{p_{uk}}{\beta_k} \right) \right) \\ & + \sum_{i=1}^N \sum_{k=1}^K \left( \frac{q_{ik}}{\beta_k} - (\alpha_k - 1) \ln \left( \frac{q_{ik}}{\beta_k} \right) \right). \end{aligned} \quad (11)$$

It should be noted that both PFM and ranking-based PFM basically assume that each observed element  $r_{ui}$  in the user-item interaction matrix follows the Poisson distribution. Moreover, they both place Gamma priors over each entry of latent user feature matrix and item feature matrix. In addition, they basically are generative probabilistic models. The ranking-based PFM model is an extension of PFM. The difference between the ranking-based PFM and PFM models is that PFM is a point-wise recommendation model, whereas the ranking-based PFM adopts a pair-wise method to learn model parameters. For PFM, only observed feedback has contributions to learning recommendation model parameters. For ranking-based PFM, both observed feedback and missing values contribute to learning model parameters.

#### 4.2. The Architecture of Neural Personalized Ranking Model.

As shown in (9), the ranking based Poisson factor model uses a linear function (i.e., inner product) to model the interactions between the user latent features and the item latent features. As reported in [9], the expressiveness of the linear function is limited and may not be sufficient to capture the complex structure of users' interactions, which hinders the performance of the ranking based Poisson factor model. To capture the complex structure of interactions; therefore we develop a neural personalized ranking model on top of the ranking-based Poisson factor model, called NRPFM. This model integrates the neural networks with the ranking-based Poisson factor model, which endows the linear ranking-based Poisson factor model with a high level of nonlinearities. In our neural personalized ranking model, we use the multilayer perceptron (MLP) [12] to learn the nonlinear and nontrivial user-item interaction relationships. Figure 1 presents the architecture of our proposed model, which consists of two branches - The left branch is used to predict score for the positive (user,item) pair and the right branch for the negative pair. Each branch includes four layers: embedding layer, merge layer, hidden layers, and prediction layer.

**Embedding Layer.** Embedding layer is aimed at mapping users and items into a low-dimensional latent space and uses the compact and dense real value vectors, instead of the sparse and high-dimensional vectors, to represent users and items.

The input of our model is a triplet  $(u, i, j)$  that indicates the indexes of user  $u$  and the corresponding ranking item pair  $(i, j)$ . After one-hot encoding the user and item indexes, we obtain the sparse representations of users and items. Then, we

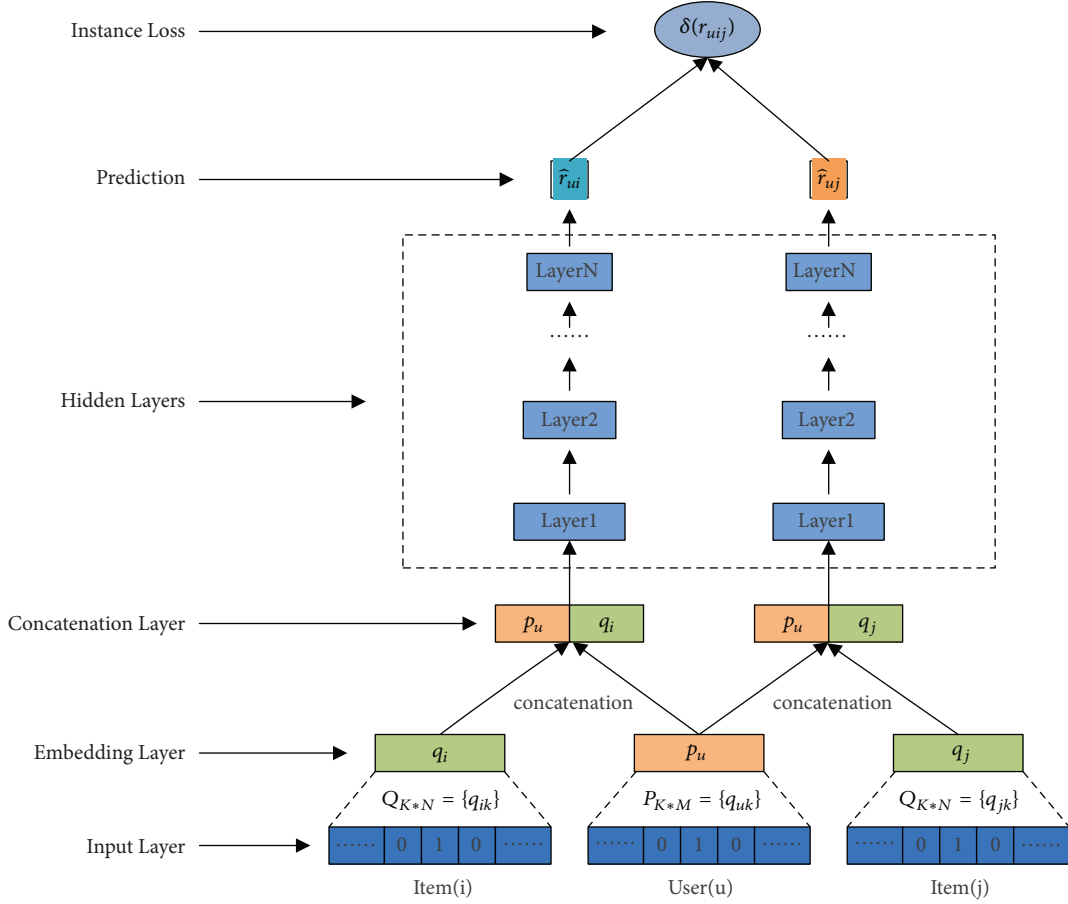


FIGURE 1: Neural personalized ranking architecture.

use the embedding table lookup to obtain the embeddings of user  $u$  and items  $i$  and  $j$ . Formally,

$$\begin{aligned} p_u &= \mathbf{P}.\text{onehot}(u), \\ q_i &= \mathbf{Q}.\text{onehot}(i), \\ q_j &= \mathbf{Q}.\text{onehot}(j), \end{aligned} \quad (12)$$

where  $\text{onehot}(\cdot)$  indicates the result of one-hot encoding for user or item.  $\mathbf{P} \in \mathbb{R}^{K \times M}$  and  $\mathbf{Q} \in \mathbb{R}^{K \times N}$  are the user embedding matrix and the item embedding matrix, respectively. The embedding matrices and the latent feature matrices derived from matrix factorization models have the same semantics. Hence,  $\mathbf{p}_u$  represents the latent feature vector of user  $u$  and  $\mathbf{q}_i$  indicates the latent feature vector of item  $i$ .

Above the embedding layer, we concatenate on the user embedding and the item embedding for each (user,item) pair in the merge layer. The concatenation of the embeddings of user and item jointly encodes the user preferences and the item characteristics. Then, we feed the concatenated embedding into the hidden layers. This design for the merge layer is widely adopted in the multilayer perceptron- (MLP-) based recommendation methods [9, 26].

**Hidden Layers and Prediction Layer.** Since the simple concatenation of embeddings does not account for any interactions between the user latent features and the item

latent features, we utilize a multilayer perceptron (MLP) to learn the user-item interaction relationships, which endows our model with a high level of nonlinearities. MLP stacks multiple fully connected hidden layers, where each hidden layer nonlinearly transforms the output of the previous hidden layer via the weight matrix and the activation function and feeds their output into the following hidden layer. The entire MLP adopts the tower structure, where the size of layer  $l$  is the half of the size of layer  $l - 1$ . Based on this structure, a higher hidden layers is able to learn more abstract features for users and items.

The prediction layer is connected with the last hidden layer and takes ReLU as the activation function to predict the scores on items. Formally, the prediction score on the item  $i$  given by the user  $u$  is defined as follows:

$$\begin{aligned} \hat{r}_{ui} &= \text{ReLU}\left(\mathbf{W}_p^{(i)T} \mathbf{z}_L^{(i)}\right), \\ \mathbf{z}_L^{(i)} &= a_L \left(\mathbf{W}_L^{(i)T} \mathbf{z}_{L-1}^{(i)} + \mathbf{b}_L^{(i)}\right), \\ &\vdots \\ \mathbf{z}_1^{(i)} &= a_1 \left(\mathbf{W}_1^{(i)T} \mathbf{x}^{(i)} + \mathbf{b}_1^{(i)}\right), \\ \mathbf{x}^{(i)} &= \begin{bmatrix} \mathbf{p}_u \\ \mathbf{q}_i \end{bmatrix}, \end{aligned} \quad (13)$$



where  $\mathbf{W}_p^{(i)}$  is the weight matrix of predication layer.  $\mathbf{W}_l^{(i)}$ ,  $\mathbf{b}_l^{(i)}$ , and  $a_l$  denote the weight matrix, the bias vector, and the activation function for the  $l$ th hidden layer, respectively.  $\mathbf{x}^{(i)}$  is the concatenation of the user embedding  $\mathbf{p}_u$  and the item embedding  $\mathbf{q}_i$ . The prediction score on the item  $j$  given by the user  $u$  is computed in the same way. We adopt ReLU as the activation functions for hidden layers, because other activation functions, such as sigmoid and tanh, suffer from the saturation and lead to overfitting.

**4.3. Model Learning.** The prediction scores reflect users' preferences for items. After obtaining the prediction scores for the positive (user, item) pair and the negative (user, item) pair, i.e.,  $\hat{r}_{ui}$  and  $\hat{r}_{uj}$ , the real value function  $\hat{r}_{uij}(\Theta)$  is rewritten as follows:

$$\begin{aligned} \hat{r}_{uij}(\Theta) &= \hat{r}_{ui}(\Theta) - \hat{r}_{uj}(\Theta) \\ &= \text{ReLU}\left(\mathbf{W}_p^{(i)T} \mathbf{Z}_L^{(i)}\right) - \text{ReLU}\left(\mathbf{W}_p^{(j)T} \mathbf{Z}_L^{(j)}\right). \end{aligned} \quad (14)$$

Integrating the above relationship function  $\hat{r}_{uij}(\Theta)$  with the ranking-based Poisson factor model, the objective function of our neural personalized ranking model is defines as follows:

$$\begin{aligned} \mathcal{L} &= \min_{\mathbf{P}, \mathbf{Q}} \sum_{(u,i,j) \in D_R} -\ln \sigma \\ &\cdot \left( \text{ReLU}\left(\mathbf{W}_p^{(i)T} \mathbf{Z}_L^{(i)}\right) - \text{ReLU}\left(\mathbf{W}_p^{(j)T} \mathbf{Z}_L^{(j)}\right) \right) \\ &+ \sum_{u=1}^M \sum_{k=1}^K \left( \frac{P_{uk}}{\beta_k} - (\alpha_k - 1) \ln \left( \frac{P_{uk}}{\beta_k} \right) \right) \\ &+ \sum_{i=1}^N \sum_{k=1}^K \left( \frac{q_{ik}}{\beta_k} - (\alpha_k - 1) \ln \left( \frac{q_{ik}}{\beta_k} \right) \right). \end{aligned} \quad (15)$$

We initialize the user embedding matrix  $\mathbf{P}$  and the item embedding matrix  $\mathbf{Q}$  with the Poisson distribution, which is parameterized by the shape parameters  $\alpha$  and the rate parameters  $\beta$ . At the training stage, according to the rules of the construction of training set  $D_R$ , we uniformly sample triplets  $(u, i, j)$  in each iteration and control the number of negative items for each positive instance. After shuffling these sampled triplets, a batch of triplets are fed into our neural personalized ranking model. For the optimization algorithm, we adopt Adam [55] to update gradients, since Adam tunes the learning rate based on the adaptive schemes and thus yields fast convergence.

## 5. Experiments

In this section, we conduct several experiments on real datasets to compare the performance of our proposed recommendation algorithm with the state-of-the-art methods.

**5.1. DataSet Description.** There are many datasets used to evaluate the performance of recommendation algorithms,

TABLE 1: Statistics of Foursquare and Gowalla.

Statistics	Foursquare	Gowalla
Num. of Check-ins	194,108	242,172
Num. of Users, $M$	2,321	5,000
Num. of Locations, $N$	5,596	23,997
Sparsity	99.18%	99.86%
Avg. Locations per User	45.57	32.13

such as MovieLens (<https://grouplens.org/datasets/movielens/>), Yelp (<https://www.yelp.com/dataset/challenge>), Epinions (<https://snap.stanford.edu/data/soc-Epinions1.html>), and Netflix (<https://www.netflixprize.com/>). However, most of them only consist of explicit feedback (i.e., ratings). In our experiment, we choose two publicly available datasets (<http://www.ntu.edu.sg/home/gaocong/datacode.htm>): Foursquare and Gowalla, to evaluate the performance of our proposed method, since they include implicit frequency feedback of users. Foursquare (<https://foursquare.com/>) and Gowalla (<http://gowalla.com/>) are two popular location-based social networks (LBSNs), which attract lots of attention from both the industry and the academia recently. In both Foursquare and Gowalla, users present their preferences in terms of the check-ins at locations (e.g., restaurants, tourists spots, and stores). In other words, users interact with locations via the check-ins. The number of check-ins to some extent indicates the degree of users' preferences for locations.

In the Foursquare dataset, all check-ins were collected within Singapore from Aug. 2010 to Jul. 2011. Check-ins of Gowalla dataset were made within California and Nevada from Feb. 2009 to Oct. 2010. In both datasets, users who have checked in fewer than 5 locations have been removed. Meanwhile, locations with less than 5 users have been filtered out. Foursquare dataset contains 194,108 check-in observations from 2,321 users at 5,596 locations. We randomly sample a subset from the original Gowalla dataset for evaluation. The sampled Gowalla dataset includes 242,172 check-in observations from 5,000 users at 23,997 locations. After aggregating the check-in records based on user and location identifiers, we obtain 105,764 entries in the user-location interaction matrix of Foursquare, and 160,689 entries in the user-location interaction matrix of Gowalla, respectively. The sparsity of Foursquare and Gowalla datasets is  $1 - 105764/(2321 \times 5596) = 99.19\%$  and  $1 - 160689/(5000 \times 23997) = 99.86\%$ , respectively. Hence, both the Foursquare and Gowalla datasets are very sparse. In Foursquare dataset, each user checked in 45.57 locations on average. And in Gowalla, each user checked in 32.13 locations on average.

The general statistics of Foursquare and Gowalla are summarized in Table 1.

**5.2. Evaluation Metrics.** We focus on the recommendation problem with implicit feedback, which is formulated as the item recommendation problem aimed at providing users with top- $k$  highest ranked items. Therefore, we employ two widely used rank metrics to evaluate the performance of different recommendation algorithms, i.e., Precision@ $k$  and Recall@ $k$ ,

where  $k$  is the length of ranked recommendation list. Given a user  $u$ , the precision and recall are defined as follows:

$$\begin{aligned} \text{Precision}_{u@k} &= \frac{|I_u^{\text{rec}} \cap I_u^{\text{test}}|}{k} \\ \text{Recall}_{u@k} &= \frac{|I_u^{\text{rec}} \cap I_u^{\text{test}}|}{|I_u^{\text{test}}|} \end{aligned} \quad (16)$$

where  $I_u^{\text{rec}}$  is the top- $k$  recommended item list for the user  $u$  and  $I_u^{\text{test}}$  is the visited items list by the user  $u$  in testing set. The final Precision@ $k$  and Recall@ $k$  of the entire recommendation algorithm are computed by averaging the precision and recall values over all the users, respectively. For both metrics, we set  $k = 3, 5, 10$  to evaluate the performance in our experiments.

**5.3. Compared Approaches.** In order to evaluate the effectiveness of our proposed method, we compare our method with the following state-of-the-art approaches:

- (1) PMF: this method was proposed by Mnih and Salakhutdinov [4] and can be viewed as a probabilistic extension of SVD [56] model. PMF represents the latent user and item feature vector by means of a probabilistic graphic model with Gaussian observation noise.
- (2) BPR: BPR adopts a Bayesian Personalized Ranking criterion [7] for item ranking. BPR is a pair-wise learning method for OCCF problem. In our experiments, we employ a uniform sampling strategy to sample the user-item pairs for model training.
- (3) MLP: MLP is an instantiation of NCF [9], which concatenates the user and item embeddings, and feeds the concatenation into neural networks to model the nonlinear user-item interactions.
- (4) NeuMF: NeuMF is another instantiation of NCF, which is a strong baseline that fuses the generalized matrix factorization and the multilayer perceptron to simultaneously model the linear and nonlinear interactions between the latent user features and the latent item features.
- (5) PFM: this method was proposed by Ma et al. [10]. PFM focuses on website recommendation and models users' implicit feedback using the Poisson distribution.
- (6) NRPFM: NRPFM is described in Section 4. NRPFM is a neural personalized ranking model for collaborative filtering with implicit frequency feedback, which integrates the ranking-based Poisson factor model with the neural networks.

**5.4. Experiment Settings.** In order to make a fair comparison, we set the parameters of each method, according to respective references or based on our experiments. Under these parameter settings, each method achieves its best performance. For PMF, we set  $\lambda_U$  and  $\lambda_V$  to be 0.001. For BPR,  $\lambda_{\ominus} = 0.001$ ,

we employ a uniform sampling strategy to sample the user-item pairs for model training. For PMF, BPR, and PFM, we set the learning rate involved in the gradient descent algorithm to be 0.0001. For MLP, we adopt the tower structure for neural networks with three hidden layers, where the sizes of each hidden layer are (32, 16, 8). The embedding size of users and items is equal to the size of the first hidden layer, i.e. 32. For NeuMF, we adopt the default parameters setting of the original paper: the number of hidden layers is set to be 3, and the sizes of each hidden layer are (32, 16, 8), respectively. The number of negative samples for each positive one is set to be 4, and the embedding size of the generalized matrix factorization is set to be 10. For PFM,  $\alpha_k = 2, \beta_k = 0.05, k = 1, \dots, K$ . For our proposed NRPFM, we initialize the model parameters using Gamma distribution with the shape parameter  $\alpha_k = 2$  and the rate parameter  $\beta_k = 0.2$ . We set the number of hidden layers to be 3 and the sizes of each hidden layer to be (32, 16, 8). Meanwhile, we set the number of negative samples per positive instance to be 8,  $\tau = 2$  and adopt Adam with the batch size of 256 as optimizer.

For each user, we randomly extract 70% of the visited locations as the training set and the remaining 30% of locations as the testing data. We conduct data splitting five times and report the average results on the test sets for each dataset.

**5.5. Recommendation Quality Comparisons.** Tables 2 and 3 report the recommendation quality of all the compared methods on Foursquare and Gowalla datasets.

From Tables 2 and 3, we have the following observations: (1) on both datasets, PMF performs the worst among all the compared methods. Besides the data sparsity issue, one possible reason is that PMF assumes a user's implicit feedback follows the Gaussian distribution, which is not suitable for modeling the implicit frequency feedback. (2) BPR achieves better performance than PMF. This is because BPR adopts the pair-wise learning method to infer the latent user and item feature vectors, making missing values contribute to the learning of model parameters. Hence, to some extent, this pair-wise learning method is able to alleviate the data sparsity problem. (3) Although MLP and NeuMF apply the point-wise methods to learning the latent representations of users and items, they are generally superior to BPR. This shows the strengths of the neural networks in capturing the complex structure of users' interactions. The performance improvements of MLP and NeuMF over BPR demonstrate that using the neural network to capture the nonlinear user-item interaction relationships is beneficial for collaborative filtering. (4) PFM outperforms PMF by a large margin and achieves comparable performance to BPR. This observation indicates that the Poisson factor model is more suitable for modeling users' implicit frequency feedback than probabilistic matrix factorization. Meanwhile, the overall performance of PFM is worse than BPR. This is because PFM only uses the observed feedback to learn the latent feature vectors, suffering from the data sparsity issue. (5) NRPFM consistently outperforms other methods, which demonstrates the effectiveness of our proposed model for collaborative filtering

TABLE 2: Performance comparison on Foursquare.

Metric	PMF	BPR	MLP	NeuMF	PFM	NRPFM
Precision@3	0.00530	0.10777	0.10843	0.11350	0.10492	<b>0.12075</b>
Precision@5	0.00497	0.09932	0.10053	0.10523	0.09294	<b>0.10760</b>
Precision@10	0.00467	0.08151	0.08173	0.08328	0.06450	<b>0.08535</b>
Recall@3	0.00111	0.02329	0.02388	0.02473	0.02306	<b>0.02700</b>
Recall@5	0.00184	0.03654	0.03730	0.03933	0.03574	<b>0.04138</b>
Recall@10	0.00356	0.06073	0.06128	0.06320	0.05082	<b>0.06630</b>

TABLE 3: Performance comparison on Gowalla.

Metric	PMF	BPR	MLP	NeuMF	PFM	NRPFM
Precision@3	0.00093	0.04442	0.04782	0.04883	0.04436	<b>0.05320</b>
Precision@5	0.00085	0.03656	0.03826	0.03848	0.03148	<b>0.04167</b>
Precision@10	0.00068	0.02796	0.02602	0.02673	0.01666	<b>0.02800</b>
Recall@3	0.00021	0.02115	0.02540	0.02693	0.02442	<b>0.02817</b>
Recall@5	0.00044	0.02753	0.03278	0.03343	0.02900	<b>0.03537</b>
Recall@10	0.00118	0.03950	0.04104	0.04265	0.03026	<b>0.04337</b>

TABLE 4: Performance of NBPFM with different hidden layers on Foursquare.

Metric	NBPFM-0	NBPFM-1	NBPFM-2	NBPFM-3	NBPFM-4
Precision@3	0.11656	0.11730	0.12378	0.12075	0.11885
Precision@5	0.10643	0.10764	0.11203	0.10760	0.10480
Precision@10	0.08530	0.08614	0.08968	0.08535	0.08475
Recall@3	0.02573	0.02550	0.02743	0.02700	0.02640
Recall@5	0.04003	0.04033	0.04183	0.04138	0.03910
Recall@10	0.06410	0.06485	0.06708	0.06630	0.06620

with implicit frequency feedback. Our proposed method improves the Precision@3 of NeuMF by 6.4% and 8.9% on Foursquare and Gowalla, respectively. In terms of Recall@3, the improvements of NBPFM over NeuMF are 9.2% and 4.6% on Foursquare and Gowalla, respectively. This observation confirms our assumption that integrating the strengths of the ranking-based poisson model in learning users’ preferences ranking between items and neural networks in capturing nonlinear user-item interaction relationships is able to boost the recommendation quality. (6) All the compared methods perform better on Foursquare than on Gowalla. The reason is that Gowalla is more sparse than Foursquare. With the dense user-item interactions, recommendation methods is more able to accurately learn the latent user and item feature vectors, resulting in better recommendation performance.

## 5.6. Sensitivity Analysis

**5.6.1. Impact of the Depth of Neural Networks.** In our proposed method, we use the neural networks, i.e, MLP, to learn the nonlinear interactions between the user and item feature vectors from users’ implicit feedback. The depth of neural networks is an important factor that affects the expressiveness of neural networks. In the section, we conduct a group of experiments to investigate the impact of the depth of neural networks on the recommendation quality. We fix the size

of the last hidden layer as 8 and vary the depth of neural networks from 1 to 4. For example, if the depth of neural networks is 4, then the structure of neural networks is  $8 \rightarrow 16 \rightarrow 32 \rightarrow 64$ , and the embedding size of user and item is 64. Other parameters keep the same settings as described in Section 5.4. We only present the experimental results on Foursquare in Table 4 and the experimental results on Gowalla show similar trends.

In Table 4, the NBPFM- $n$  denotes the NBPFM method with  $n$  hidden layers. As demonstrated in Table 4, NBPFM with different number of hidden layers consistently outperforms NeuMF. Moreover, it is not beneficial for NBPFM that stack more hidden layers in MLP to learn the nonlinear interaction functions between the latent user and item feature vectors. A possible reason is that a deeper neural network makes NBPFM have more trainable parameters, which are relatively difficult to learn with limited training data, resulting in the degradation of recommendation performance. In addition, for NBPFM without hidden layers, the performance of NBPFM-0 is even better than that of NeuMF. NBPFM-0 can be viewed as a variant of the ranking-based poisson factor model that utilizes the concatenation of user and item embeddings to predict scores. This observation somehow shows the effectiveness of our proposed ranking-based Poisson factor model for collaborative filtering with implicit frequency feedback.

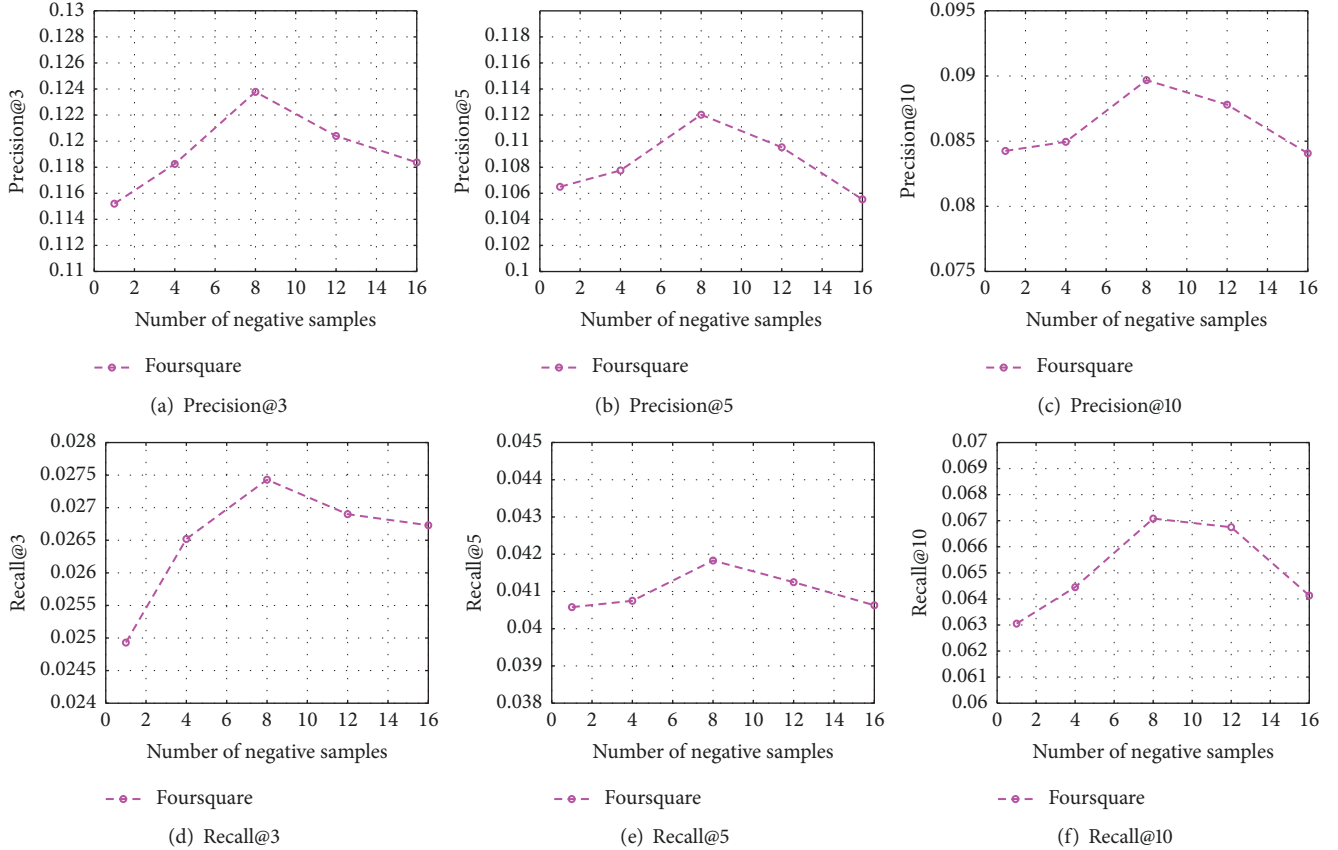


FIGURE 2: The impact of negative samples.

**5.6.2. Impact of Negative Samples.** In this section, we conduct another group of experiments to investigate the impact of negative samples on the recommendation quality. We vary the number of negative samples for each positive one from 1 to 16 and observe the changes of recommendation quality. We set the number of hidden layers to be 2 since NBPFM achieves better performance under this settings, which is shown in Table 4. And other parameters remain unchanged. The experimental results of NBPFM on Foursquare are shown in Figure 2.

As indicated in Figure 2, our proposed neural personalized model is sensitive to the number of negative samples. The recommendation quality firstly improves as the number of negative samples increases and then degrades as the number of negative samples further increases. This indicates that insufficient or too many negative samples may hurt the recommendation performance of NBPFM. NBPFM achieves the best performance when the number of negative samples is around 8.

**5.6.3. Impact of Parameters  $\alpha_k$  and  $\beta_k$ .** In our proposed method, parameters  $\alpha_k$  and  $\beta_k$  control the shapes and scales of the Gamma distributions, which are used to initialize the user embedding matrix and the item embedding matrix. We perform another group of experiments to evaluate the sensitivities of  $\alpha_k$  and  $\beta_k$ , by changing the values of  $\alpha_k$  from 1 to 5 given  $\beta_k = 0.2$ , or varying the values of  $\beta_k$  from 0.1 to

0.5 given  $\alpha_k = 2$ . The experimental results of NBPFM with respect to different  $\alpha_k$  and  $\beta_k$  on Foursquare are plotted in Figures 3 and 4, respectively.

As we can see, both  $\alpha_k$  and  $\beta_k$  significantly affect the performance of our proposed recommendation model. In the case of  $\alpha_k$  with the fixed value 2, NBPFM performs best when  $\beta_k$  is around 0.2, further reducing or increasing the value of  $\beta_k$  leads to worse performance. In the case of fixing the value of  $\beta_k$  to be 0.2, NBPFM shows the similar trends; i.e., all the evaluation metrics firstly move upwards and then begin to drop down, when  $\alpha_k$  surpasses a certain threshold. This observation indicates that NBPFM is also sensitive to the initializations of the user embedding matrix and the item embedding matrix, which initially encode the user preferences and the item characteristics, respectively.

## 6. Conclusion and Future Work

In this paper, we propose a neural personalized ranking model for collaborative filtering with implicit frequency feedback, which integrates the ranking-based poisson factor model with the neural networks. We firstly develop a ranking-based Poisson factor model, which combines the Poisson factor model and the Bayesian personalized ranking to model sparse implicit frequency feedback. The ranking-based Poisson factor model adopts a pair-wise learning method to learn the rankings of preferences between items. Then, we utilize

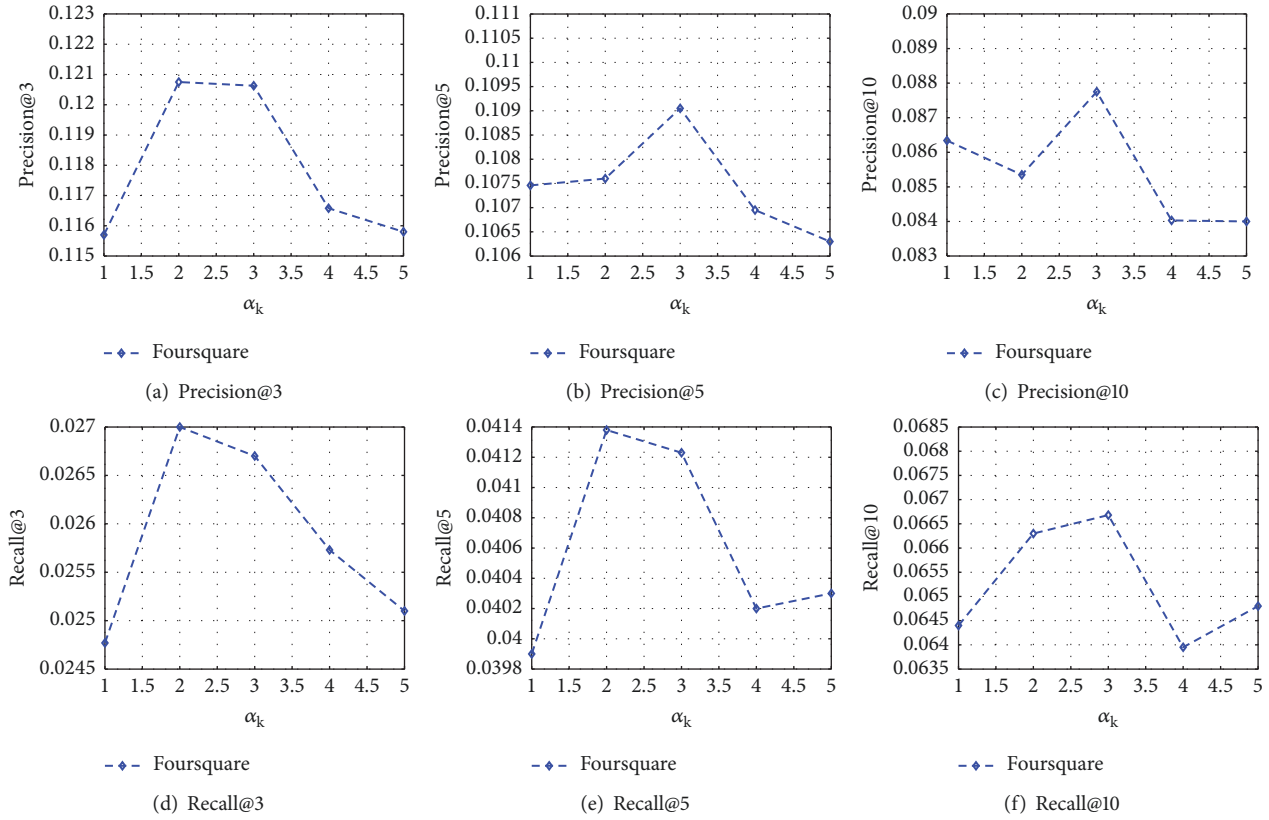


FIGURE 3: The impact of  $\alpha_k$ .

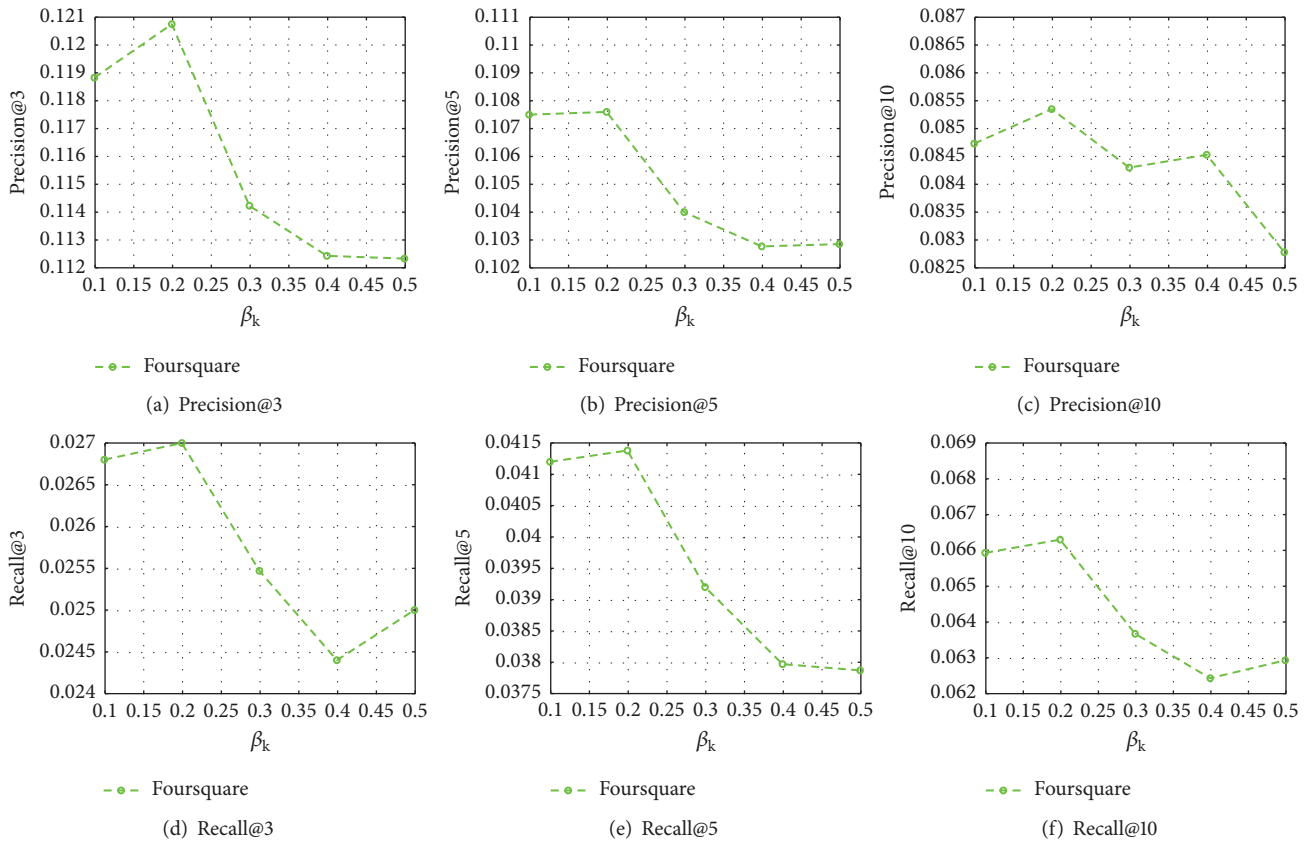


FIGURE 4: The impact of  $\beta_k$ .

the neural networks to further extend the ranking-based Poisson factor model, and propose a neural personalized ranking model to capture the complex structure of user-item interactions. The neural personalized ranking model leverages the multilayer perceptron to learn the nonlinear user-item interaction relationships and endows the linear ranking-based Poisson factor model with a high level of nonlinearities. Experimental results on two real-world datasets show that our proposed method outperforms the state-of-the-art recommendation algorithms.

We only infer users' preferences and items' characteristics from users' implicit feedback. Since auxiliary information, such as social relationships and user reviews, is beneficial to recommendation algorithms, we plan to integrate these auxiliary information into our proposed neural personalized ranking model to boost the recommendation performance. In addition, recent advances in deep learning, e.g., attention mechanism, graph convolutional neural network, and generative adversarial network, have shown great potential in the fields of natural language processing and computer vision. Hence, applying the above deep learning techniques to recommender systems would be an interesting direction.

## Data Availability

The ZIP data used to support the findings of this study are available at <http://www.ntu.edu.sg/home/gaocong/datacode.htm>. The ZIP data is publicly available and can be directly downloaded from <http://www.ntu.edu.sg/home/gaocong/data/poidata.zip>. The description of datasets is presented in Section 5.1 of our manuscript.

## Conflicts of Interest

The authors declare that they have no conflicts of interest.

## Acknowledgments

This work is supported in part by the Natural Science Foundation of the Higher Education Institutions of Jiangsu Province (Grant no. 17KJB520028), NUPTSF (Grant no. NY217114), Tongda College of Nanjing University of Posts and Telecommunications (Grant no. XK203XZ18002), and Qing Lan Project of Jiangsu Province.

## References

- [1] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 6, pp. 734–749, 2005.
- [2] J. S. Breese, D. Heckerman, and C. Kadie, *Empirical analysis of predictive algorithms for collaborative filtering*, UAI, Morgan Kaufmann Publishers Inc, pp. 43–52, 1998.
- [3] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *The Computer Journal*, vol. 42, no. 8, pp. 30–37, 2009.
- [4] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," *NIPS, NIPS'07*, pp. 1257–1264, 2007.
- [5] R. Pan, Y. Zhou, B. Cao et al., "One-class collaborative filtering," in *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM '08)*, pp. 502–511, Pisa, Italy, December 2008.
- [6] Y. Hu, C. Volinsky, and Y. Koren, "Collaborative filtering for implicit feedback datasets," in *Proceedings of the 8th IEEE International Conference on Data Mining (ICDM '08)*, pp. 263–272, ICDM, Pisa, Italy, December 2008.
- [7] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: bayesian personalized ranking from implicit feedback," in *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI '09)*, pp. 452–461, AUAI Press, Montreal, Canada, June 2009.
- [8] T. Zhao, J. McAuley, and I. King, "Leveraging social connections to improve personalized ranking for collaborative filtering," in *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, CIKM 2014*, pp. 261–270, ACM, China, November 2014.
- [9] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, "Neural Collaborative Filtering," in *WWW, International World Wide Web Conferences Steering Committee*, pp. 173–182, 2017.
- [10] H. Ma, C. Liu, I. King, and M. R. Lyu, "Probabilistic factor models for web site recommendation," in *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'11*, pp. 265–274, ACM, China, July 2011.
- [11] P. Gopalan, L. Charlin, and D. M. Blei, "Content-based recommendations with Poisson factorization," in *Proceedings of the 28th Annual Conference on Neural Information Processing Systems 2014, NIPS 2014*, pp. 3176–3184, Canada, December 2014.
- [12] K. Hornik, M. Stinchcombe, and H. White, "Multilayer feedforward networks are universal approximators," *Neural Networks*, vol. 2, no. 5, pp. 359–366, 1989.
- [13] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proceedings of the 10th International Conference on World Wide Web (WWW '01)*, pp. 285–295, ACM, 2001.
- [14] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, 2003.
- [15] G.-R. Xue, C. Lin, Q. Yang et al., "Scalable collaborative filtering using cluster-based smoothing," in *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '05)*, pp. 114–121, ACM, Salvador, Brazil, 2005.
- [16] Y. Yu, C. Wang, Y. Gao, L. Cao, and X. Chen, "A Coupled Clustering Approach for Items Recommendation," in *Advances in Knowledge Discovery and Data Mining*, vol. 7819, pp. 365–376, PAKDD, Springer, Berlin, Germany, 2013.
- [17] T. Hofmann, "Latent semantic models for collaborative filtering," *ACM Transactions on Information and System Security*, vol. 22, no. 1, pp. 89–115, 2004.
- [18] T. Hofmann, "Collaborative Filtering via Gaussian Probabilistic Latent Semantic Analysis," *SIGIR Forum (ACM Special Interest Group on Information Retrieval)*, pp. 259–266, 2003.
- [19] R. Salakhutdinov, A. Mnih, and G. Hinton, "Restricted Boltzmann machines for collaborative filtering," in *Proceedings of the 24th International Conference on Machine Learning (ICML '07)*, vol. 227, pp. 791–798, ACM, Corvallis, Ore, USA, June 2007.
- [20] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," *NIPS*, pp. 556–562, 2001.

- [21] Y. Koren, "Factorization meets the neighborhood: a multi-faceted collaborative filtering model," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*, pp. 426–434, ACM, New York, NY, USA, August 2008.
- [22] N. Srebro, J. Rennie, and T. S. Jaakkola, "Maximum-margin matrix factorization," *NIPS*, pp. 1329–1336, 2005.
- [23] W. Pan and L. Chen, "GBPR: Group preference based bayesian personalized ranking for one-class collaborative filtering," in *Proceedings of the 23rd International Joint Conference on Artificial Intelligence, IJCAI 2013*, pp. 2691–2697, AAAI Press, China, August 2013.
- [24] Y. Zhao, L. Nie, X. Wang, and T. Chua, "Personalized Recommendations of Locally Interesting Venues to Tourists via Cross-Region Community Matching," *ACM Transactions on Intelligent Systems and Technology*, vol. 5, no. 3, pp. 1–26, 2014.
- [25] S. Zhang, L. Yao, and A. Sun, *Deep Learning Based Recommender System: A Survey And New Perspectives*, <https://arxiv.org/abs/1707.07435>.
- [26] H.-T. Cheng, L. Koc, J. Harmsen et al., "Wide & deep learning for recommender systems," in *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, DLRS 2016*, pp. 7–10, ACM, USA.
- [27] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2017*, pp. 355–364, ACM, Japan, August 2017.
- [28] S. Sedhain, A. K. Menon, S. Sanner, and L. Xie, "Autorec: Autoencoders meet collaborative filtering," in *Proceedings of the the 24th International Conference*, pp. 111–112, WWW '15 Companion, Florence, Italy, May 2015.
- [29] Y. Wu, C. DuBois, A. X. Zheng, and M. Ester, "Collaborative denoising auto-encoders for top-N recommender systems," in *Proceedings of the 9th ACM International Conference on Web Search and Data Mining, WSDM 2016*, pp. 153–162, ACM, USA, February 2016.
- [30] D. Kim, C. Park, J. Oh, S. Lee, and H. Yu, "Convolutional matrix factorization for document context-aware recommendation," in *Proceedings of the 10th ACM Conference on Recommender Systems, RecSys 2016*, pp. 233–240, ACM, USA, September 2016.
- [31] X. Wang, X. He, L. Nie, and T.-S. Chua, "Item silk road: Recommending items from information domains to social users," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2017*, pp. 185–194, ACM, Japan, August 2017.
- [32] H. Guo, R. Tang, Y. Ye, Z. Li, and X. He, "DeepFM: A factorization-machine based neural network for CTR prediction," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI 2017*, pp. 1725–1731, AAAI Press, Australia, August 2017.
- [33] S. Rendle, "Factorization machines," in *Proceedings of the 10th IEEE International Conference on Data Mining, ICDM 2010*, pp. 995–1000, Australia, December 2010.
- [34] J. Xiao, H. Ye, X. He, H. Zhang, F. Wu, and T.-S. Chua, "Attentional factorization machines: Learning the weight of feature interactions via attention networks," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence, IJCAI 2017*, pp. 3119–3125, AAAI Press, Australia, August 2017.
- [35] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Proceedings of the 20th Annual Conference on Neural Information Processing Systems (NIPS '06)*, pp. 153–160, Mass, USA, December 2006.
- [36] P. Vincent, H. Larochelle, Y. Bengio, and P. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, pp. 1096–1103, ACM, July 2008.
- [37] F. Strub and J. Mary, "Collaborative filtering with stacked denoising autoencoders and sparse inputs in," *NIPS workshop on machine learning for eCommerce*, 2015.
- [38] S. Li, J. Kawale, and Y. Fu, "Deep collaborative filtering via marginalized denoising auto-encoder," in *Proceedings of the 24th ACM International Conference on Information and Knowledge Management, CIKM 2015*, pp. 811–820, ACM, Australia, October 2015.
- [39] M. Chen, Z. Xu, K. Q. Weinberger, and F. Sha, "Marginalized denoising autoencoders for domain adaptation," *ICML*, pp. 1627–1634, 2012.
- [40] H. Wang, N. Wang, and D. Yeung, "Collaborative Deep Learning for Recommender Systems," in *Proceedings of the 21th ACM SIGKDD International Conference*, pp. 1235–1244, Sydney, NSW, Australia, August 2015.
- [41] P. Vincent, H. Larochelle, I. Lajoie, and P. Manzagol, "Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion," *Journal of Machine Learning Research*, vol. 11, pp. 3371–3408, 2010.
- [42] Y. LeCun, K. Kavukcuoglu, C. Farabet et al., "Convolutional networks and applications in vision," in *Proceedings of the IEEE International Symposium on Circuits and Systems*, vol. 2010, pp. 253–256, IEEE, Paris, France, June 2010.
- [43] G. Yuyun and Z. Qi, "Hashtag recommendation using attention-based convolutional neural network," in *Proceedings of the 25th International Joint Conference on Artificial Intelligence, IJCAI 2016*, pp. 2782–2788, USA, July 2016.
- [44] R. He and J. McAuley, "VBPR: Visual Bayesian personalized ranking from implicit feedback," in *Proceedings of the 30th AAAI Conference on Artificial Intelligence, AAAI 2016*, pp. 144–150, USA, February 2016.
- [45] A. Van den Oord and S. B. Dieleman, "Deep content-based music recommendation," *NIPS*, pp. 2649–2651, 2013.
- [46] S. Wang, Y. Wang, J. Tang, K. Shu, S. Ranganath, and H. Liu, "What Your Images Reveal: Exploiting Visual Contents for Point-Of-Interest Recommendation," in *International World Wide Web Conferences Steering Committee (WWW)*, pp. 391–400, Perth, Australia, April 2017.
- [47] L. Zheng, V. Noroozi, and P. S. Yu, "Joint Deep Modeling of Users and Items Using Reviews for Recommendation," in *Proceedings of the the Tenth ACM International Conference*, pp. 425–434, WSDW, ACM, Cambridge, UK, February 2017.
- [48] B. Hidasi, A. Karatzoglou, L. Baltrunas, and D. Tikk, *Session-Based Recommendations with Recurrent Neural Networks*, ICLR, 2016.
- [49] S. Wu, W. Ren, C. Yu, G. Chen, D. Zhang, and J. Zhu, "Personal recommendation using deep recurrent neural networks in NetEase," in *Proceedings of the 32nd IEEE International Conference on Data Engineering, ICDE 2016*, pp. 1218–1229, IEEE, Finland, May 2016.
- [50] M. Quadrana, A. Karatzoglou, B. Hidasi, and P. Cremonesi, "Personalizing session-based recommendations with hierarchical recurrent neural networks," in *Proceedings of the 11th ACM Conference on Recommender Systems, RecSys 2017*, pp. 130–137, ACM, Italy, August 2017.

- [51] K. Georgiev and P. Nakov, "A non-IID framework for collaborative filtering with Restricted Boltzmann Machines," in *Proceedings of the 30th International Conference on Machine Learning, ICML 2013*, pp. 1148–1156, USA, June 2013.
- [52] J. Wang, L. Yu, W. Zhang et al., "IRGAN: A minimax game for unifying generative and discriminative information retrieval models," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2017*, pp. 515–524, ACM, Japan, August 2017.
- [53] L. Nie, S. Yan, M. Wang, R. Hong, and T.-S. Chua, "Harvesting visual concepts for image search with complex queries," in *Proceedings of the 20th ACM International Conference on Multimedia*, pp. 59–68, ACM, Japan, November 2012.
- [54] L. Nie, M. Wang, Z. Zha, and T. Chua, "Oracle in Image Search," *ACM Transactions on Information and System Security*, vol. 30, no. 2, pp. 1–23, 2012.
- [55] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *ICLR*, pp. 1–15, 2014.
- [56] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Application of Dimensionality Reduction in Recommender System - A Case Study," *WebKDD Workshop*, 2000.



