Collaborative Preference Embedding against Sparse Labels

Shilong Bao[1,2], Qianqian Xu[3], Ke Ma[1,2],
Zhiyong Yang[1,2], Xiaochun Cao[1,2],Qingming Huang[3,4,5*]
[1]State Key Laboratory of Information Security, Institute of Information Engineering, CAS, Beijing, China
[2]School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
[3]Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, CAS, Beijing, China
[4]School of Computer Science and Technology, University of Chinese Academy of Sciences, Beijing, China
[5]Key Laboratory of Big Data Mining and Knowledge Management, Chinese Academy of Sciences, Beijing, China
baoshilongcs@gmail.com,{make,yangzhiyong,caoxiaochun}@iie.ac.cn,xuqianqian@ict.ac.cn,qmhuang@ucas.ac.cn

ABSTRACT

Living in the era of the internet, we are now facing with a big bang of online information. As a consequence, we often find ourselves troubling with hundreds and thousands of options before making a decision. As a way to improve the quality of users' online experience, *Recommendation System* aims to facilitate personalized online decision making processes via predicting users' responses toward different options. However, the vast majority of the literature in the field merely focus on datasets with sufficient amount of samples. Different from the traditional methods, we propose a novel method named as *Collaborative Preference Embedding* (CPE) which directly deals with sparse and insufficient user preference information. Specifically, we represent the intrinsic pattern of users/items with a high dimensional embedding space. On top of this embedding space, we design two schemes specifically against the limited generalization ability in terms of sparse labels. On one hand, we construct a margin function which could indicate the consistency between the embedding space and the true user preference. From the margin theory point-of-view, we then propose a generalization enhancement scheme for sparse and insufficient labels via optimizing the margin distribution. On the other hand, regarding the embedding as a code for a user/item, we then improve the generalization ability from the coding point-of-view. Specifically, we leverage a compact embedding space by reducing the dependency across different dimensions of a code (embedding). Finally, extensive experiments on a number of real-world datasets demonstrate the superior generalization performance of the proposed algorithm.

KEYWORDS

Preference Embedding, Margin Distribution, Collaborative Filtering

ACM Reference Format:

Shilong Bao, Qianqian Xu, Ke Ma, Zhiyong Yang, Xiaochun Cao, Qingming Huang. 2019. Collaborative Preference Embedding against Sparse Labels. In *Proceedings of the 27th ACM International Conference on Multimedia (MM '19), Oct. 21–25, 2019, Nice, France.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3343031.3350915

*Corresponding author.

MM '19, October 21–25, 2019, Nice, France

ACM ISBN 978-1-4503-6889-6/19/10…$15.00
https://doi.org/10.1145/3343031.3350915

1 INTRODUCTION

With the explosive growth of online information, users today on the Internet are facing with the rapid expansion of multiple options (e.g., which movie to see, which book/article to read, etc.). Inferring user's preference over a set of candidates has thus become an urgent problem. Based on these inferences, we can provide the target user with a small part of objects that he/she is most likely to be interested in to simplify the tedious choosing procedure. In this way, preference learning must bear good fruits in the long run for improving the quality of user's experience. During the past decades, *Recommendation Systems* (RS) has emerged as one of the most universal and popular methods for users preference prediction [10, 14, 38]. More specifically, RS could help us identify the most interesting and relevant new objects for a target user based on his/her historical preference records. According to the existing literatures in this area, such preference records therein could be collected either from explicit feedback signals or implicit feedback signals [13]. The explicit feedback can directly reflect the preference of users toward objects. For example, we can collect users' star ratings for movies and collect their preference for articles by hitting thumbs-up/down buttons. However, the explicit feedback is not always available because sometimes users are reluctant to rate directly. Compared to explicit feedback, the implicit feedback can be tracked automatically and is much easier to collect, such as purchase history, watching habits, or even mouse movements. Over the past decades, a significant amount of RS techniques have come out to serve the purpose of preference learning. Among such efforts, *Collaborative Filtering* (CF) is known as the most popular and effective method realizing the long-desired goal of RS [29]. In one word, CF comes out from an intuitive idea that users sharing similar preferences tend to choose similar items and perform similar interaction with the items [23]. To do this, CF first determines which users are similar and then carries out recommendation according to their historical behaviours. Mathematically, the majority studies of CF and its variants adopt the Matrix Factorization (MF) framework due to its superior performance. In a nutshell, MF projects users and items into a shared latent low dimensional space, where users and items are represented as vectors in the embedding space. Then a user's preference on an item is represented as the dot product between their latent vectors. Representative variants along this line

include Bayesian Personalized Ranking [27], Weighted Regularized Matrix Factorization [13], SVD-based model [16], Visual Bayesian Personalized Ranking [8], etc.

Despite the effectiveness of MF for collaborative filtering, a remaining issue of MF lies in that it does not model the relationship between users and items explicitly. This is due to the fact dot product does not satisfy the triangle inequality [12, 26, 28], which might lead to sub-optimal performance. Therefore, many efforts have been made to address this problem recently. Noteworthy is the work presented in [6], where the dot product is replaced with non-linear neural networks. The basic idea is to apply non-linear activation functions over the dot product of user and item latent factors. As an improved variant of this framework [9] proposes a general framework named as *Neural network-based Collaborative Filtering* (NCF), which unifies the Generalized Matrix Factorization (GMF), Multi-Layer Perceptron (MLP) and Neural Matrix Factorization (NeuMF). Another orthogonal direction is to directly introduce a distance metric, such as Euclidean distance, which explicitly induces the triangle inequality. *Collaborative Metric Learning* (CML) [12] is one of the most representative methods along this direction, which applies metric learning to collaborative filtering problems. Motivated by metric learning algorithm, CML learns a joint metric space to capture not only users' preferences but also the user-user and item-item relationships in this space.

The recent advances of the relevant studies have revolutionarily prompted the representation ability of the traditional MF methods. Nonetheless, the existing issues, especially regarding the relevant datasets, have long been left behind this wave of revolution. A typical recommendation dataset usually contains a huge set of users and items, which makes a complete collection of the preference information almost impossible. More practically, we often observe sparse and insufficient supervision information over the users' preference. Unfortunately, it is well-known from the machine learning perspective that insufficient supervision often leads to the notorious *overfitting* problem, where a limited generalization ability is often observed. To the best of our knowledge, the mainstream of the relevant studies do not consider this issue explicitly.

In this paper, we propose a novel collaborative filtering based method with a specific focus on improving the generalization ability against sparse labels, called *Collaborative Preference Embedding* (CPE). More specifically, the main contributions of this work are as follows:

- We first construct a latent embedding space for each user and item, where we formulate the preference information with the distance between different embeddings. On top of this space, we propose a specific margin function for our underlying problem, where a higher value of the margin function indicates a better quality of the embedding.
- With the strength of the margin theory, a reasonable generalization ability is often achieved with a reasonable margin distribution. Motivated by this fact, we propose a new objective function based on our defined margin function where the margin distribution is explicitly optimized.
- Moreover, we also promote the generalization ability from the coding perspective, where we regard each embedding point as a specific code of a user/item. More precisely, we propose to improve the compactness of the embedding space via leveraging the independence across different dimensions of the codes (feature embeddings).

2 RELATED WORK

2.1 Collaborative Filtering

There have been many *Collaborative Filtering* (CF) techniques developed in the area of RS, which are mainly divided into two categories: memory-based [4, 34] and model-based [11, 17] methods. Among the memory-based CF algorithms, K-Nearest Neighbors (KNN) is known as the most extensively used implementation [2, 3]. Typically, the variants of KNN method include user-based and item-based [22] approaches, which directly return recommendations based on similar users and similar items. The model-based CF approaches aim to learn a model via partially observed user-item interactions, so that recommendation system can recognize more complex patterns and realize intelligent predictions. In the past few years, Matrix Factorization (MF) has been the most popular and well-known CF method because of its high efficiency and superior performance. MF can be regarded as a matrix-approximating process which projects the users and items to a unified latent factor space. Furthermore, it estimates the missing entries via their latent factors' dot product. More specifically, let R_{ij} denote the rating of item j marked by user i , MF methods learn user vector $\boldsymbol{u}_i \in \mathbb{R}^d$ and item vector $\boldsymbol{v}_j \in \mathbb{R}^d$, and use their dot product $\boldsymbol{u}_i^T \boldsymbol{v}_j$ to estimate R_{ij} [17]. In general, the original MF methods are often based on explicit feedback, such as users' ratings and reviews, etc. However, in many real-world applications, the explicit feedback is often highly biased and not always available, which might limit the performance of MF model. On the contrary, the implicit feedback (e.g., purchase, movie watched, click, etc.) is relatively abundant and easy to collect, which has drawn widespread attention [15, 20, 24] recently. Under the context of implicit feedback, we could only observe an item when at least one user interacts with it. Moreover, we cannot tell apart the unobserved positive pairs and the unobserved negative ones, which makes it improper to use traditional MF methods [12, 13, 24]. To address this problem, Weighted Regularized Matrix Factorization (WRMF) [13, 24] is proposed to express the implicit users' observations from two perspectives: a preference label and a confidence level. WRMF regards all user-item interactions as the positive preferences and treats unobserved pairs as negative items. Moreover, WRMF adopts a confidence level for each item, where positive items have higher weights than negative ones. Another direction is to consider a pairwise learning framework. Bayesian Personalized Ranking (BPR) [27] begins an early trail in this direction from a Bayesian AUC optimization perspective. More precisely, BPR uses a pairwise *log-sigmoid* loss to directly optimize the AUC ranking, which can be successfully applied to the model of MF (BPR-MF) and adaptive kNN (BPR-kNN).

2.2 Collaborative Metric Learning

Metric learning aims to learn a distance metric that can reflect the relationship among data samples. The key idea is to assign smaller distance between similar pairs, and larger distance between dissimilar ones. The concept of similarity is also important for RS.

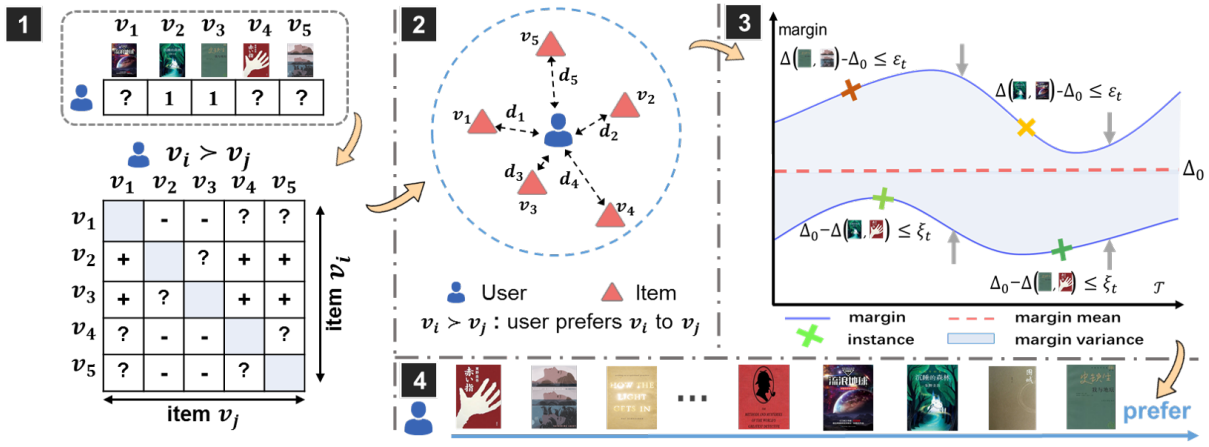

Figure 1: The framework of our method. We deploy the implicit feedback signals to form our dataset (as shown in **(1)**). For each row i and column j, if the user prefers item i to j, we denote this relationship as "+". If the opposite is the case, it is denoted as "-". However, we cannot determine the relationship among observed items and also among unobserved items, and we represent them as "?". We transform these implicit feedback into pairwise comparison and learn a metric space where each user/item is assigned with a numerical embedding in the space. Moreover, to deal with the sparse preference supervision, we propose a novel method CPE. Specifically, our proposed model enhances the generalization ability from both the coding point-of-view (leveraging a compact space as shown in **(2)**), and from the margin point-of-view (improving the margin distribution as shown in **(3)**). With the learned embedding space, we could then predict the users' preference for the unseen items in the dataset (as shown in **(4)**).

If we can represent users/items in a high dimensional Euclidean space, then we should also expect that similar items should have a closer distance than the dissimilar ones. Furthermore, another advantage of the metric learning framework is that it preserves the *triangle inequality* [12, 26, 28] while the MF based methods fail to do so. Consequently, metric learning has been widely adopted in the CF area [21, 25]. For the point-of-interest (POI) recommendation, [7] proposes a Metric Embedding method to model the sequential information and individual preference by projecting POI into a low dimensional Euclidean space. Moreover, Collaborative Metric Learning (CML) [12] proposed recently is one of the most competitive and effective methods for implicit feedback, following the idea of the largest margin nearest neighbour algorithm (LMNN) [31]. The goal of CML is to learn a joint user-item metric space and the preference of user-item pairs can be captured by the Euclidean distance between the vectors of users and items. Since the metric space obeys the triangle inequality, CML can also capture the user-user and item-item relationships besides the user-item relationships. Nevertheless, CML only optimizes the minimum margin, which is not sufficient to guarantee a reasonable generalization performance. Different from CML, our method directly optimizes the margin distribution which could further enhance the generalization ability according to the recent results from margin theory [35, 36].

3 METHODOLOGY

In this section, we first introduce the necessary notations and formulate our problem mathematically. Secondly, we project the users/items into an embedding space and define a margin function on this space which captures the preference consistency between the predicted labels and the true labels. Thirdly, we propose a generalization enhancement scheme in the pursuit of a reasonable margin distribution. Finally, we propose a regularization scheme to leverage a compact embedding space from the coding point-of-view. In a nutshell, Fig.1 illustrates the framework of our proposed model.

3.1 Preliminaries

In our recommendation system, assume that our dataset contains user-item interaction records for M users and N items. Mathematically we denote the set of all users as $\mathcal{U} = \{u_1, \ldots, u_M\}$ and the set of all items as $\mathcal{V} = \{v_1, v_2, ..., v_N\}$. In Sec.1 and Sec.2, we have mentioned that *implicit feedback* enjoys several advantages over *explicit feedback*. Motivated by this fact, we then develop our model with *implicit feedback* signals. In the context of *implicit feedback*, we believe that users tend to have higher preferences for the items that they interact with (e.g. we often have higher preference for the movies in our favourite list). Given the implicit feedback signals, our primary goal is to learn the users' preference for the items. Equipped with the learned preferences, we could then improve users' online experience quality in the long run via recommending unseen items for the users according to the preference.

Given the primary goal, our preliminaries finish with definition and formulation of users preferences over different items. More specifically, given a user u_i, we explore his/her preference regarding the items $\mathcal{V}$ in a pairwise manner [32, 33]. To this end, the preference of u_i is presented as a set of triplets

$$\mathcal{T}_i = \{(i, j, k) \mid i \in [M], u_i \in \mathcal{U},\ j, k \in [N],\ v_j, v_k \in \mathcal{V}\},$$

where (i, j, k) represents that user u_i prefers v_j to v_k, and $[M] = \{1, 2, \ldots, M\}$. Moreover, we denote the overall triplet set as $\mathcal{T}$, with $\mathcal{T} = \bigcup_{i=1}^{M} \mathcal{T}_i$. Equivalently, we denote such a triplet (i, j, k) as a relation: $v_j \succ_{u_i} v_k$. In addition, we express the supervision

information over a triplet (i, j, k) as a label $y_{jk}^{(i)}$:

$$y_{jk}^{(i)} = \begin{cases} 1, & v_j \succ_{u_i} v_k, \\ -1, & otherwise, \end{cases}$$

where $\mathcal{Y} = \{y_{jk}^{(i)}\}$ is the set of triplet labels.

3.2 Preserving Preference Consistency with a Margin Formulation

Now we turn to formulate the preference relation conveyed by $\mathcal{T}$ and $\mathcal{Y}$ on top of a high dimensional embedding space for the users and items. In a nutshell, we expect to learn an embedding for each user and item in a high dimensional space. Based on the embedding space, we then capture the preference comparisons in $\mathcal{Y}$ via comparing the relative distances on the space. More specifically, we learn an embedding $\mathbf{f}_{u_i} \in \mathbb{R}^d$ for each user u_i and learn $\mathbf{f}_{v_j} \in \mathbb{R}^d$ for each item v_j. We define the distance between the user u_i and item v_j as :

$$\mathbf{d}(i, j) = ||\mathbf{f}_{u_i} - \mathbf{f}_{v_j}||.$$

To reflect the preference relation, we expect to observe a small $\mathbf{d}(i, j)$ if u_i likes item v_j. If the opposite is the case, we then expect to observe a large $\mathbf{d}(i, j)$. Now we could further unfold the comparisons covered by $\mathcal{Y}$ and $\mathcal{T}_i$. If $v_j \succ_{u_i} v_k$ or equivalently $y_{jk}^{(i)} = 1$ holds, then we have u_i prefers v_j to v_k, which leads to the inequality $\mathbf{d}(i, j) < \mathbf{d}(i, k)$. If the opposite is the case, we have $\mathbf{d}(i, k) < \mathbf{d}(i, j)$. Putting the two cases together, we then expect the following relations:

$$\begin{cases} \mathbf{d}(i, j) < \mathbf{d}(i, k), & v_j \succ_{u_i} v_k \\ \mathbf{d}(i, j) > \mathbf{d}(i, k), & v_j \prec_{u_i} v_k \end{cases}, \forall (i, j, k) \in \mathcal{T}_i \tag{1}$$

where $i \in [M], j, k \in [N], \ j \neq k$.

To realize Eq.(1), we first measure the consistency between the learned embeddings $\mathbf{f}_{u_i}$,$\mathbf{f}_{v_j}$, $\mathbf{f}_{v_k}$ with the triplet labels $y_{jk}^{(i)}$. Borrowing the wisdom of the margin theory in SVM, we then construct a margin $\Delta_{jk}^{(i)}$ for our task as the indicator

$$\begin{aligned} \Delta_{jk}^{(i)} &= y_{jk}^{(i)} \cdot \left(\mathbf{d}(i, k)^2 - \mathbf{d}(i, j)^2\right) \\ &= y_{jk}^{(i)} \cdot \left(||\mathbf{f}_{u_i} - \mathbf{f}_{v_k}||^2 - ||\mathbf{f}_{u_i} - \mathbf{f}_{v_j}||^2\right) \end{aligned} \tag{2}$$

Now let's see how $\Delta_{jk}^{(i)}$ corresponds to the consistency. If the label $y_{jk}^{(i)} = 1$, we have that u_i prefers item v_j to v_k. In this case, we should have $\mathbf{d}(i, j) < \mathbf{d}(i, k)$, then a reasonable embedding should bear a large positive value of $\Delta_{jk}^{(i)}$. If by contrary, $y_{jk}^{(i)} = -1$, then user u_i prefers item v_k to v_j, we should also expect to observe a large positive $\Delta_{jk}^{(i)}$. *In this way, it becomes clear that enforcing a positive $\Delta_{jk}^{(i)}$ could guarantee the preference consistency.*

This inspires us to adopt an initial model with a hinge-loss-based objective function:

$$\underset{\mathbf{f}_u, \mathbf{f}_v}{\mathbf{argmin}} \ \frac{1}{|\mathcal{T}|} \cdot \sum_{(i,j,k) \in \mathcal{T}} \mathbf{max}\left(\hat{\Delta}_0 - \Delta_{jk}^{(i)}, 0\right) \tag{3}$$

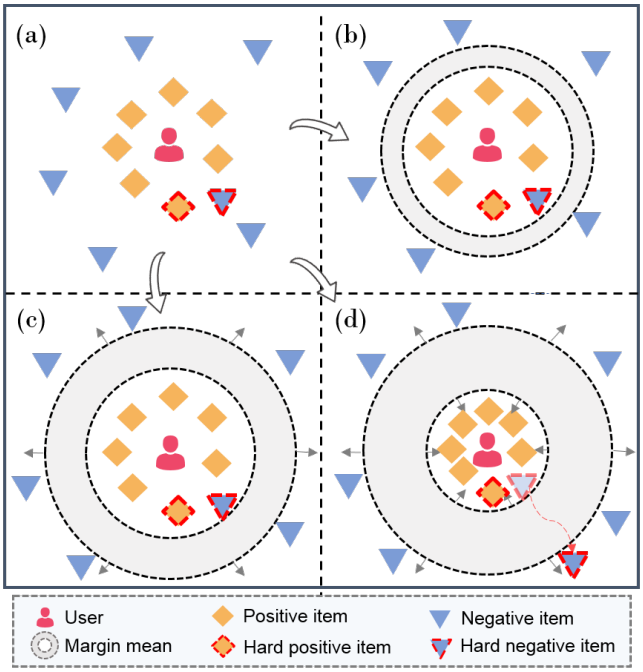

Figure 2: An illustration of why optimizing the margin distribution of our metric space can achieve better generalization performance. (a) Initial space. (b) The effects of small margin mean. (c) The effects of large margin mean and large margin variance. (d) The embedding space with large margin mean and small margin variance.

where $\hat{\Delta}_0 > 0$ is a predefined constant. Inspired by the *SVM formulation*, we can further convert (3) equivalently to the following optimization problem

$$\begin{aligned} \underset{\boldsymbol{\xi}, \mathbf{f}_u, \mathbf{f}_v}{\mathbf{argmin}} \ & \frac{1}{|\mathcal{T}|} \cdot \sum_{(i,j,k) \in \mathcal{T}} \xi_{jk}^{(i)} \\ s.t. \ & \Delta_{jk}^{(i)} \geq \hat{\Delta}_0 - \xi_{jk}^{(i)}, \ \ \xi_{jk}^{(i)} \geq 0, \\ & \forall (i, j, k) \in \mathcal{T} \end{aligned} \tag{4}$$

where $\boldsymbol{\xi} = \{\xi_{jk}^{(i)} \mid (i, j, k) \in \mathcal{T}\}$ is the slack variable.

According to the above problem, any comparison pair with a margin value lower than $\hat{\Delta}_0$ will enlarge the objective function with a positive $\xi_{jk}^{(i)}$. Then minimizing the above problem is equivalent to enforcing $\mathbf{min}_{i,j,k}(\Delta_{jk}^{(i)}) > \hat{\Delta}_0$ as much as possible. In an ideal case, all the margin values $\Delta_{jk}^{(i)}$ are no less than $\hat{\Delta}_0$, we then achieve the minimum of 0.

3.3 Generalization Performance Enhancement for Sparse Labels

So far, we have achieved an initial model based on the margin formulation of the preference consistency. *But can this model necessarily guarantee a reasonable generalization performance even if the preference labels are sparse?* Unfortunately, according to the recent studies of margin theory, the answer might be negative. As we have

pointed out in the preceding subsec., minimizing (4) only improves $\min_{i,j,k} \Delta_{jk}^{(i)}$, the minimum margin over the observed sample. The recent advance of margin theory [35, 36] pointed out that maximizing the minimum margin alone does not necessarily lead to promising generalization performance. The margin distribution, rather than the minimum margin, is more crucial to the overall generalization performance.

How can we seek out a margin distribution in order to guarantee the generalization performance? Let us answer this question from an intuitive perspective with Fig.2. Fig.2-(a) depicts the initial embedding space. Fig. 2(b)-(d) give three possible cases of the embedding space after the optimization process. The initial model does not optimize the margin mean directly. As shown in Fig.2-(b), after the optimization process, we might come to find an embedding space with a *small* margin mean (note that the thickness of the grey ring represents the magnitude of margin mean). If this is the case, then most of the samples are suffered from a small $\Delta_{jk}^{(i)}$ and thus a small fault tolerance. What is worse, with a sparse supervision, the training data distribution could not render a good coverage of the true data distribution. In this way, a small margin mean brings extra vulnerability toward erroneous results during the test phase, since the incoming test data is very likely to be inconsistent with the training data distribution. Moreover, Fig.2-(c) shows another case with a *large* margin mean and a *large* margin variance. In this case, though we have a large margin mean, the significant variance indicates there still exists hard instances suffering from small $\Delta_{jk}^{(i)}$. In other words, the optimization procedure fails to learn the intrinsic pattern contained in the hard instances. Hence, the generalization performance is still not ideal. Ultimately, in Fig.2-(d), we see that, with a *large* margin mean and a *small* margin variance, not only the relevant items become much closer to the target user than the irrelevant ones, but there are also much fewer hard and misplaced instances. *In this sense, we expect to simultaneously maximize the margin mean and minimize the margin variance to reach a reasonable generalization performance.*

Next we derive the loss function realizing this pursuit. The average of $\Delta_{jk}^{(i)}$ could be represented as:

$$\overline{\Delta}_0 = \frac{1}{|\mathcal{T}|} \cdot \sum_{(i,j,k)\in\mathcal{T}} \Delta_{jk}^{(i)}, \tag{5}$$

According to Eq.(1), we see that scaling the distance function $\mathbf{d}$ does not affect the prediction results. Then we can set the margin mean as a constant Δ_0 to simplify the loss function.

$$\Delta_0 = \frac{1}{|\mathcal{T}|} \cdot \sum_{(i,j,k)\in\mathcal{T}} \Delta_{jk}^{(i)}, \tag{6}$$

Meanwhile, to minimize the margin variance, we could restrict the margin deviation $\left|\Delta_{jk}^{(i)} - \Delta_0\right|$. To this end, we deploy a lower bound $\xi_{jk}^{(i)}$ and an upper bound $\epsilon_{jk}^{(i)}$ on the term $\left|\Delta_{jk}^{(i)} - \Delta_0\right|$. This induces the following two inequalities:

$$\Delta_{jk}^{(i)} \le \Delta_0 + \epsilon_{jk}^{(i)} \tag{7}$$

$$\Delta_{jk}^{(i)} \ge \Delta_0 - \xi_{jk}^{(i)} \tag{8}$$

where $\epsilon_{jk}^{(i)} > 0, \xi_{jk}^{(i)} > 0$ are soft-margin variables. Back to our initial model (4), we replace the constraints $\Delta_{jk}^{(i)} \ge \hat{\Delta}_0 - \xi_{jk}^{(i)}, \xi_{jk}^{(i)} \ge 0$ with the above-mentioned inequalities. The resulting loss function becomes:

$$\begin{aligned} \underset{\mathbf{f}_u, \mathbf{f}_v, \boldsymbol{\xi}, \varepsilon}{\textbf{argmin}} \ & \frac{1}{|\mathcal{T}|} \sum_{(i,j,k)\in\mathcal{T}} \xi_{jk}^{(i)} + \epsilon_{jk}^{(i)} \\ s.t. \ & \Delta_{jk}^{(i)} \ge \Delta_0 - \xi_{jk}^{(i)}, \ \Delta_{jk}^{(i)} \le \Delta_0 + \epsilon_{jk}^{(i)}, \\ & \xi_{jk}^{(i)} \ge 0, \ \epsilon_{jk}^{(i)} \ge 0, \ \forall (i,j,k) \in \mathcal{T}. \end{aligned} \tag{9}$$

This optimization problem can be further simplified with a hinge-loss formulation:

$$\underset{\mathbf{f}_u, \mathbf{f}_v}{\textbf{argmin}} \ \mathcal{L}_d = \frac{1}{|\mathcal{T}|} \sum_{(i,j,k)\in\mathcal{T}} \textbf{max}\left(\Delta_{jk}^{(i)} - \Delta_0, 0\right) + \textbf{max}\left(\Delta_0 - \Delta_{jk}^{(i)}, 0\right). \tag{10}$$

Minimizing the loss function will push all $\Delta_{jk}^{(i)}$ toward the normalized margin mean Δ_0. By setting a proper Δ_0, we can then simultaneously maximize the margin mean and minimize the margin variance and thus realize a reasonable margin distribution.

3.4 Defencing Overfitting with a Compact Embedding Space

Apart from the margin theory point-of-view, we know that reducing the dimensionality of a high dimensional space could also help us defense the overfitting issues caused by spare labels. In this way, we adopt two extra regularization terms to leverage a compact embedding space.

First of all, we restrict our embedding space with a bounded ℓ_2 norm, i.e.,

$$||\mathbf{f}_{u_i}||^2 \le l, \ \ ||\mathbf{f}_{v_j}||^2 \le l$$

where l can control the bound. This is a widely adopted scheme to restrict the model complexity.

Moreover, we could also enhance the compactness of our embedding space from the coding perspective. Specifically, we can regard a point $\mathbf{f} \in \mathbb{R}^d$ in our embedding space as a code for a given user/item in our dataset. To reduce the redundant information in the coding scheme, we need to guarantee that each dimension of $\mathbf{f}$ mostly contains unique information that is independent/irrelevant with the other dimensions (*i.e. improving the information rate of the code*). To do this, we first measure the correlation between different dimensions with the well-known *covariance matrix* in statistics. Specifically, we concatenate all the user/item embeddings by rows and obtain $\boldsymbol{F} = \left[\mathbf{f}_{u_1}, \cdots, \mathbf{f}_{u_M}, \mathbf{f}_{v_1}, \cdots, \mathbf{f}_{v_N}\right]^\top$. $\mathbf{f}_i$ contains i-th row of $\boldsymbol{F}$. Then the covariance matrix capturing the correlation across dimensions could be represented as $\boldsymbol{C} \in \mathbb{R}^{d\times d}$

$$\boldsymbol{C} = \frac{1}{N+M} \sum_{i=1}^{N+M} (\mathbf{f}_i - \overline{\mathbf{f}})^\top (\mathbf{f}_i - \overline{\mathbf{f}}) \tag{11}$$

where $\overline{\mathbf{f}} \in \mathbb{R}^{1\times d}$ is the average embedding for each dimension with $\overline{\mathbf{f}}_i = \frac{1}{N+M} \sum_{k=1}^{N+M} F_{ki}$. In the covariance matrix, the entry C_{ij} could be rewritten as:

$$C_{ij} = \frac{1}{N+M} \sum_{k=1}^{N+M} (F_{ki} - \overline{\mathbf{f}}_i) \cdot (F_{kj} - \overline{\mathbf{f}}_j),$$

Table 1: Statistics of the datasets. %Density is equal to $\frac{\#Ratings}{\#Users \times \#Items}$. The smaller the value of %Density is, the more sparsely labeled the dataset is.

Datasets	MovieLens-100K	CiteULike-T	Book-Crossing
Domain	Movie	Paper	Book
#Users	943	7,947	11,209
#Items	1,682	25,975	7,490
#Ratings	55,376	142,794	98,205
%Density	3.4912%	0.0692%	0.1170%

which represents the covariance between the i-th dimension and the j-th dimension. If $i = j$, then C_{ii} (or C_{jj}) becomes the sample variances of the i-th dimension. Moreover, if $C_{ij} = 0$, we then have that the i-th dimension is independent with the j-th dimension on the dataset. To reduce the redundant correlation across different dimension, we could then push all off-diagonal entries of $\boldsymbol{C}$ toward 0. To do this, we then push the covariance close to the identity matrix $\boldsymbol{I}$. Specifically, we adopt the *log-determinant divergence (LDD)* [19] to measure the "closeness" between $\boldsymbol{C}$ and $\boldsymbol{I}$. According to the derivation in [19], this is equivalent to minimizing the regularization term $\mathcal{R}_C$

$$\mathcal{R}_C = tr(\boldsymbol{C}) - \log(\det(\boldsymbol{C})) - d, \tag{12}$$

where $tr(\cdot)$ denotes matrix trace, d is the number of dimensions, and $det(\boldsymbol{C})$ denotes determinant of $\boldsymbol{C}$. Let $\lambda_1, \lambda_2, \ldots, \lambda_d$ be the eigenvalues of $\boldsymbol{C}$. Then according to the definition of determinant, we have $\log(det(\boldsymbol{C})) = \log\left(\prod_{i=1}^{d} \lambda_i\right)$. Unfortunately, if $\boldsymbol{C}$ is a singular matrix with zero eigenvalues, we have $\log(det(\boldsymbol{C})) = -\infty$. Note that $\boldsymbol{C}$ is a positive semi-definite matrix with $\lambda_i \geq 0, \forall i$. We can add a diagonal matrix $\delta \cdot \boldsymbol{I}$ ($\delta > 0$) to $\boldsymbol{C}$ to avoid such a numerical disaster. To do this, we replace the regularization term $\mathcal{R}_C$ with

$$\tilde{\mathcal{R}}_C = tr(\boldsymbol{C}) - \log(\det(\boldsymbol{C} + \delta \cdot \boldsymbol{I})) = tr(\boldsymbol{C}) - \sum_{i=1}^{d} \log(\lambda_i + \delta).$$

Ultimately, together with the margin distribution constraints and the compactness constraints, we then come to our final objective function as follows:

$$\begin{aligned} \underset{\mathbf{f}_u, \mathbf{f}_v, \boldsymbol{\xi}, \boldsymbol{\epsilon}}{\textbf{argmin}} \quad & \mathcal{L}_d + \mu \cdot \tilde{\mathcal{R}}_C \\ s.t. \quad & ||\mathbf{f}_{\boldsymbol{u_i}}||^2 \leq l, ||\mathbf{f}_{\boldsymbol{v_j}}||^2 \leq l, \end{aligned} \tag{13}$$

where μ is the regularization parameter.

3.5 Optimization and Complexity

Optimization. We adopt the *AdaGrad* optimizer [5] to solve (13), which updates the parameters with gradient evaluation from $\mathcal{L}_d$ and $\tilde{\mathcal{R}}_C$. Apparently, it is easy to calculate the gradient for $\mathcal{L}_d$, so we narrow our focus on the calculations about $\tilde{\mathcal{R}}_C$. According to (11), the covariance matrix can be rewritten as

$$\boldsymbol{C} = \frac{1}{k} \boldsymbol{F}^T \boldsymbol{H} \boldsymbol{F}$$

where k is the total number of users and items in a mini-batch, and $\boldsymbol{H}$ is a $k \times k$ squared matrix with its diagonal elements being $1 - \frac{1}{k}$ and off-diagonal elements being $-\frac{1}{k}$. With the Eq.(12), we have

$$\begin{aligned} \frac{\partial \mathcal{R}_c}{\partial \boldsymbol{F}} &= \frac{\partial tr(\boldsymbol{C})}{\partial \boldsymbol{F}} - \frac{\partial \log(\det(\boldsymbol{C}))}{\partial \boldsymbol{F}} \\ &= \frac{2}{k}\left(\boldsymbol{H}\boldsymbol{F} - \boldsymbol{H}\boldsymbol{F}(\boldsymbol{C} + \delta \boldsymbol{I})^{-1}\right) \\ &= \frac{2}{k}\boldsymbol{H}\boldsymbol{F}\left(\boldsymbol{I} - (\boldsymbol{F}^T\boldsymbol{H}\boldsymbol{F} + \delta \boldsymbol{I})^{-1}\right) \end{aligned} \tag{14}$$

Complexity. Accordingly, in order to solve the above optimization strategy, the complexity for gradient evaluation over an epoch is $O(b \cdot (t + d \cdot k^2 + d^3))$, where t is the batch size and b is the number of batch.

4 EXPERIMENTS

We conduct comprehensive experiments to demonstrate the superiority of CPE. Empirical results on three different benchmark datasets, including MovieLens-100K, CiteULike-T and BookCrossing, consistently show that our method can achieve reasonable generalization performance even when suffering sparse preference information. The details of three datasets are summarized in Table 1.

4.1 MovieLens-100K

Dataset. In MovieLens-100K[1], each user has at least 20 ratings, and the ratings range from 1 to 5. If user u_i provides an item v_j a rate no less than 4, we regard item v_j as a positive item for user u_i. Otherwise, if v_j has a rate lower than 4 or if v_j is not rated by u_i, we then regard it as a negative item. We use these positive and negative items to form the set of triplet $\mathcal{T}$ (see Sec. 3.1). However, under the context of implicit feedback, it is not necessary to sweep over the complete pairwise comparisons for each iteration of the training procedure. This motivates us to adopt the *negative example sampling* strategy to reduce the number of training samples [18, 24, 37]. For the sake of simplicity, in each iteration, we only select a subset of n negative items to update the parameters.

Evaluation Metrics. In most cases, users often focus on the top-K items in the recommended list, so we evaluate all methods with the following measures: *Precision@K*, *Recall@K*, *Normalized Discounted Cumulative Gain* (NDCG@K), *Mean Average Precision* (MAP) and *Area Under ROC Curve* (AUC).

Competitors. We evaluate CPE against 5 competitors, including:

- **BPR-MF [27]** is one of the classical matrix factorization methods, which optimizes the pairwise ranking between the positive and negative items.
- **WRMF [13]** is an effective MF model, which considers all user-item interactions as binary implicit feedback (positive or negative) and employs a confidence value to control the weight of unobserved interactions.
- **GMF [9]** can be regarded as a generalized MF method. It is one of the instantiation of *NCF*, which applies a linear kernel to model the latent user-item interactions.
- **NeuMF [9]** is a unified framework bridging GMF and multi-layered perceptron (MLP). NeuMF concatenates the output of MF and MLP, and converts the recommendation task into a regression problem.

[1] https://grouplens.org/datasets/movielens/

Table 2: Performance comparison on **MovieLens-100K** dataset. The best method is marked as red, and the second is marked as green.

Method	P@30 ↑	R@30 ↑	NDCG@30 ↑	P@50 ↑	R@50 ↑	NDCG@50 ↑	MAP ↑	AUC ↑
GMF[9]	0.3107	0.2393	0.3527	0.3556	0.3001	0.4249	0.2468	0.8815
NeuMF[9]	**0.3567**	**0.2731**	**0.4061**	**0.4116**	**0.3740**	**0.4778**	**0.3054**	**0.9053**
BPR-MF[27]	0.3557	0.2714	**0.4080**	0.4050	0.3674	0.4700	0.2882	0.9027
WRMF[13]	0.3433	0.2662	0.3734	0.3800	0.3469	0.4129	0.2822	0.8890
CML-PAIR[12]	0.3384	0.2476	0.3941	0.3809	0.3047	0.4365	0.2566	0.8475
CML-WARP[12]	0.3402	0.2485	0.3949	0.3835	0.3076	0.4354	0.2616	0.8596
CPE (ours)	**0.3633**	**0.2793**	0.4002	**0.4283**	**0.3910**	**0.4957**	**0.2938**	**0.9056**

- **CML [12]** is a competitive model, which bridges the effort of metric learning and collaborative filtering. We evaluate CML performance with two different losses: pairwise hinge loss (denoted as **CML-PAIR**) and Weighted Approximate-Rank Pairwise (WARP) loss (denoted as **CML-WARP**).

Implementation details. We implement our model with TensorFlow [1], and minimize the objective function (13) *AdaGrad* [5]. More precisely, we set the learning rate $\alpha = 0.05$ and the regularization parameter $\mu = 5 \times 10^{-3}$. The dimension of the embedding space d is set to 300, and we set the constant $\Delta_0 = 1.2$ and $l = 1$. For the negative sampling strategy, we set the size of the negative items' subset as $n = 30$. For P@K, R@K and NDCG@K, we set $K \in \{30, 50\}$ to test these algorithms in Top-K recommendation task. What is noteworthy is that for MAP and AUC, we calculate them over the entire items set $\mathcal{V}$.

Results and Discussions. Table 2 shows the performance of different methods with various metrics. Firstly, we can observe that, in most cases, our proposed method outperforms all the competitors. Specifically, our method outperforms the second best model by up to 1.67%, 1.70%, 1.79% in terms of Precision@50, Recall@50, and NDCG@50, respectively and there is a slight improvement on AUC. Secondly, our method outperforms BPR-MF, WRMF, and GMF by up to 2.57%, 8.28%, and 9.09% respectively (achieved at NDCG@50, NDCG@50, R@50, respectively). The superiority of our method over these MF models demonstrates the fact that our approach captures the preference relationships between users and items more explicitly, since Euclidean distance can directly induce the triangle inequality. Moreover, our method outperforms other collaborative metric learning based model CML-PAIR and CML-WARP, confirming that merely optimizing the minimum margin cannot guarantee a reasonable performance (depicted in Fig. 2). Furthermore, NeuMF is another competitive algorithm, with significant improvement over traditional MF models BPR-MF and WRMF, respectively. Normally, NeuMF can exhibit better performance than other traditional MF methods and distance metric based methods by virtue of introducing the non-linear transformations. In our method, we can not only learn the distance metric in the embedding space but also optimize the margin distribution to enhance the generalization. This induces a significant improvement over the NeuMF method.

4.2 CiteULike-T

Datasets. There are two versions of CiteULike in [30] including CiteULike-a and CiteULike-t. Both of them are collected from CiteULike and we use the **CiteULike-T** to test our model's performance.

Implementation details. We adopt the same training strategy as Sec. 4.1, except that we set d as 200 and the margin mean Δ_0 as 0.5.

Results and Discussions. Table 3 shows the complete comparison of these methods on this dataset. From the results we can draw the following conclusions:

(1) our method outperforms all competitors in terms of all the metrics. Moreover, we can often observe that the improvements are featured with a large margin. (2) Typically, in terms of NDCG@50, our method significantly outperforms WRMF algorithm by 7.54% and outperforms the BPR-MF model by 6.87%. This is a solid evidence showing that our model alleviates the inherent limitation of MF (i.e., its unreasonable representation ability) and enjoys better representations of the relations among users and items. (3) We also observe that CML is a very competitive baseline, this is supported by a sharp improvement over the MF-based methods: BPR-MF and WRMF. This again verifies that metric learning based models are more effective than MF models merely using dot product. (4) Despite the good performance of CML, our approach still outperforms the CML-PAIR and CML-WARP. The possible reason lies in that when confronting sparse and insufficient preference information, optimizing our proposed objective function explicitly can obtain a more compact embedding space and render a better coverage of the true data distribution. (5) Moreover, NCF framework based models achieve a much lower performance on this dataset. This is possibly due to the overfitting problem caused by an extremely large sparsity.

4.3 Book-Crossing

Datasets. Book-Crossing[2] is also an explicit feedback dataset with ratings ranging from 1 to 10. We abandon the books which are rated less than 10 times and abandon the users who have less than 5 ratings. Furthermore, we regard the items with a rating greater than 5 as the positive items, and regard the remaining items as negative ones.

Implementation details. We adopt the same strategy with Sec.4.1, except that d is set to 100, the margin mean Δ_0 is set to 1.8 and we set n as 20.

Results and Discussions. Table 4 shows the experimental comparison results. We find that our method outperforms all competitors consistently on all the metrics, with a large margin. Next, CML-WARP achieves the second best performance, which outperforms the MF-based competitors. As expected, metric-learning-based methods (including CPE and CML-*) outperform the MF-based methods. Furthermore, it is interesting to note that our

[2] http://www2.informatik.uni-freiburg.de/ cziegler/BX/

Table 3: Performance comparison on **CiteULike-T** dataset. The best method is marked as red, and the second is marked as green.

Method	P@30 ↑	R@30 ↑	NDCG@30 ↑	P@50 ↑	R@50 ↑	NDCG@50 ↑	MAP ↑	AUC ↑
GMF[9]	0.1000	0.0449	0.1046	0.1560	0.0464	0.1645	0.0333	0.7302
NeuMF[9]	0.0917	0.0452	0.0983	0.1480	0.0521	0.1553	0.0328	0.7127
BPR-MF[27]	0.1794	**0.1003**	0.2041	0.2009	0.1130	0.2274	**0.0929**	**0.8538**
WRMF[13]	0.1681	0.0930	0.1943	0.2048	0.1083	0.2207	0.0864	0.8284
CML-PAIR[12]	0.1656	0.0832	0.2006	0.2110	0.1051	0.2560	0.0709	0.8173
CML-WARP[12]	**0.1889**	0.0955	**0.2241**	**0.2311**	**0.1297**	**0.2851**	0.0838	0.8474
CPE (ours)	**0.2111**	**0.1118**	**0.2356**	**0.2525**	**0.1645**	**0.2961**	**0.1079**	**0.8699**

Table 4: Performance comparison on **Book-Crossing** dataset. The best method is marked as red, and the second is marked as green.

Method	P@30 ↑	R@30 ↑	NDCG@30 ↑	P@50 ↑	R@50 ↑	NDCG@50 ↑	MAP ↑	AUC ↑
GMF[9]	0.0833	0.0311	0.1092	0.1400	0.0412	0.1310	0.0421	0.6405
NeuMF[9]	0.0778	0.0312	0.0939	0.1501	0.0510	0.1618	0.0455	0.6385
BPR-MF[27]	0.1476	0.0792	0.1525	0.1833	0.1249	0.2092	0.0615	0.7278
WRMF[13]	0.1238	0.0681	0.1308	0.1767	0.1223	0.1929	0.0548	0.7109
CML-PAIR[12]	0.1734	0.0816	0.1890	0.2067	0.1525	0.2484	0.0561	0.7510
CML-WARP[12]	**0.1810**	**0.1055**	**0.1975**	**0.2400**	**0.1782**	**0.2689**	**0.0836**	**0.7605**
CPE (ours)	**0.2067**	**0.1188**	**0.2161**	**0.2869**	**0.1952**	**0.3247**	**0.1038**	**0.8359**

Table 5: Ablation experiments' performance on **CiteULike-T** dataset. The best method is marked as red.

Method	P@30 ↑	R@30 ↑	NDCG@30 ↑	P@50 ↑	R@50 ↑	NDCG@50 ↑	MAP ↑	AUC ↑
CPE-WO-MAR	0.1638	0.0791	0.1868	0.2033	0.0930	0.2507	0.0667	0.8185
CPE-WO-REG	0.2032	0.1022	0.2290	0.2327	0.1477	0.2697	0.0973	0.8536
CPE	**0.2111**	**0.1118**	**0.2356**	**0.2525**	**0.1645**	**0.2961**	**0.1079**	**0.8699**

Table 6: Ablation experiments' performance on **BookCrossing** dataset. The best method is marked as red.

Method	P@30 ↑	R@30 ↑	NDCG@30 ↑	P@50 ↑	R@50 ↑	NDCG@50 ↑	MAP ↑	AUC ↑
CPE-WO-MAR	0.1583	0.0663	0.1860	0.2120	0.1344	0.2778	0.0778	0.8031
CPE-WO-REG	0.1667	0.0716	0.2150	0.2600	0.1614	0.3117	0.0958	0.7848
CPE	**0.2067**	**0.1188**	**0.2161**	**0.2869**	**0.1952**	**0.3247**	**0.1038**	**0.8359**

method outperforms the second best method (CML-WARP) consistently over all involved metrics. Noteworthy are the significant improvements over AUC (7.54%), NDCG@50 (5.58%), and P@50 (4.69%). This again underpins the effectiveness of our proposed model. With these experimental results, we can conclude that our approach has promising representation capacity and achieves superior generalization performance.

4.4 Ablation Studies

In order to demonstrate the effectiveness of two schemes to the final improvement, we conduct ablation experiments on two datasets, i.e., CiteULike-T and Book-Crossing. Specifically, we report the performance CPE without regularization (**CPE-WO-REG**) and CPE without margin distribution (**CPE-WO-MAR**, only maximizing the minimum margin, see subsec.3.3) respectively, and then compare them with our **CPE** method. The experimental results are shown in Table 5 and Table 6. From the results, we can conclude that the proposed two schemes in this paper, i.e., generalization enhancement scheme and regularization scheme, are both necessary for our model, since the two variants (CPE-WO-REG and CPE-WO-MAR) fail to outperform CPE.

5 CONCLUSION

In this paper, we develop a novel collaborative filtering based method called CPE to effectively address the problem of sparse and insufficient preference supervision in *Recommendation Systems*. To alleviate the limited generalization ability in terms of sparse labels, we design a margin function and propose a generalization enhancement scheme via optimizing the margin distribution. In addition, we adopt a novel regularization strategy to leverage a compact embedding space, which can further enhance the generalization performance from the coding point-of-view. Extensive experiments demonstrate that our method can achieve superior generalization performance and outperform state-of-the-art methods in a wide range of recommendation datasets.

6 ACKNOWLEDGMENTS

This work was supported in part by National Natural Science Foundation of China (No. 61620106009, U1636214, 61861166002, 61672514), in part by Key Research Program of Frontier Sciences, CAS: QYZDJ-SSW-SYS013, in part by Beijing Natural Science Foundation (No. 4172068, 4182079), and in part by Youth Innovation Promotion Association CAS.

$$\underset{\mathbf{f}_u, \mathbf{f}_v, \xi, \epsilon}{\textbf{argmin}} \ \frac{1}{|\mathcal{T}|} \sum_{(i,j,k)\in\mathcal{T}} \textbf{max}\left(\Delta_{jk}^{(i)} - \Delta_0, 0\right) + \textbf{max}\left(\Delta_0 - \Delta_{jk}^{(i)}, 0\right) + \mu \cdot \left(tr(C) - \sum_{i=1}^{d} \log(\lambda_i + \delta)\right) \quad s.t. \ \ ||\mathbf{f}_{\boldsymbol{u_i}}||^2 \leq l, ||\mathbf{f}_{\boldsymbol{v_j}}||^2 \leq l \tag{15}$$

REFERENCES

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. Tensorflow: A system for large-scale machine learning. In *USENIX Symposium on Operating Systems Design and Implementation (OSDI)*. 265–283.

[2] Gediminas Adomavicius and Alexander Tuzhilin. 2005. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge & Data Engineering* 17, 6 (2005), 734–749.

[3] Jesus Bobadilla, Antonio Hernando, Fernando Ortega, and Jesus Bernal. 2011. A framework for collaborative filtering recommender systems. *Expert Systems with Applications* 38, 12 (2011), 14609–14623.

[4] JESUS Bobadilla, Francisco Serradilla, Antonio Hernando, et al. 2009. Collaborative filtering adapted to recommender systems of e-learning. *Knowledge-Based Systems* 22, 4 (2009), 261–265.

[5] John C. Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research* 12 (2011), 2121–2159.

[6] Gintare Karolina Dziugaite and Daniel M. Roy. 2015. Neural Network Matrix Factorization. *CoRR* abs/1511.06443 (2015).

[7] Shanshan Feng, Xutao Li, Yifeng Zeng, Gao Cong, Yeow Meng Chee, and Quan Yuan. 2015. Personalized Ranking Metric Embedding for Next New POI Recommendation. In *International Joint Conference on Artificial Intelligence*. 2069–2075.

[8] Ruining He and Julian McAuley. 2016. VBPR: Visual Bayesian Personalized Ranking from Implicit Feedback. In *AAAI Conference on Artificial Intelligence*. 144–150.

[9] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *International Conference on World Wide Web*. 173–182.

[10] Shintami Chusnul Hidayati, Cheng-Chun Hsu, Yu-Ting Chang, Kai-Lung Hua, Jianlong Fu, and Wen-Huang Cheng. 2018. What Dress Fits Me Best?: Fashion Recommendation on the Clothing Style for Personal Body Shape. In *ACM Multimedia Conference on Multimedia Conference*. 438–446.

[11] Thomas Hofmann. 2004. Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems* 22, 1 (2004), 89–115.

[12] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. 2017. Collaborative metric learning. In *International Conference on World Wide Web*. 193–201.

[13] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *IEEE International Conference on Data Mining*. 263–272.

[14] Yang Hu, Xi Yi, and Larry S. Davis. 2015. Collaborative Fashion Recommendation: A Functional Tensor Factorization Approach. In *ACM Conference on Multimedia Conference*. 129–138.

[15] Christopher C Johnson. 2014. Logistic matrix factorization for implicit feedback data. *Advances in Neural Information Processing Systems* 27 (2014).

[16] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *ACM International Conference on Knowledge Discovery and Data Mining*. 426–434.

[17] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *IEEE Computer* 42, 8 (2009), 30–37.

[18] Maciej Kula. 2015. Metadata Embeddings for User and Item Cold-start Recommendations. In *ACM Conference on Recommender Systems*. 14–21.

[19] Brian Kulis, Mátyás A Sustik, and Inderjit S Dhillon. 2009. Low-rank kernel learning with Bregman matrix divergences. *Journal of Machine Learning Research* 10, 2 (2009), 341–376.

[20] Joonseok Lee, Samy Bengio, Seungyeon Kim, Guy Lebanon, and Yoram Singer. 2014. Local collaborative ranking. In *International Conference on World Wide Web*. 85–96.

[21] Wentao Li, Min Gao, Wenge Rong, Junhao Wen, Qingyu Xiong, Ruixi Jia, and Tong Dou. 2017. Social recommendation using Euclidean embedding. In *International Joint Conference on Neural Networks*. 589–595.

[22] Greg Linden, Brent Smith, and Jeremy York. 2003. recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing* 7, 1 (2003), 76–80.

[23] Fernando Ortega, José-Luis SáNchez, JesúS Bobadilla, and Abraham GutiéRrez. 2013. Improving collaborative filtering-based recommender systems results using Pareto dominance. *Information Sciences* 239 (2013), 50–61.

[24] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. 2008. One-class collaborative filtering. In *IEEE International Conference on Data Mining*. 502–511.

[25] Chanyoung Park, Donghyun Kim, Xing Xie, and Hwanjo Yu. 2018. Collaborative Translational Metric Learning. In *IEEE International Conference on Data Mining*. 367–376.

[26] Parikshit Ram and Alexander G Gray. 2012. Maximum inner-product search using cone trees. In *ACM International Conference on Knowledge Discovery and Data Mining*. 931–939.

[27] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Conference on Uncertainty in Artificial Intelligence*. 452–461.

[28] Anshumali Shrivastava and Ping Li. 2014. Asymmetric LSH (ALSH) for sublinear time maximum inner product search (MIPS). In *Advances in Neural Information Processing Systems*. 2321–2329.

[29] Poonam B Thorat, RM Goudar, and Sunita Barve. 2015. Survey on collaborative filtering, content-based filtering and hybrid recommendation system. *International Journal of Computer Applications* 110, 4 (2015), 31–36.

[30] Hao Wang, Binyi Chen, and Wu-Jun Li. 2013. Collaborative Topic Regression with Social Regularization for Tag Recommendation. In *International Joint Conference on Artificial Intelligence*. 2719–2725.

[31] Kilian Q Weinberger and Lawrence K Saul. 2009. Distance metric learning for large margin nearest neighbor classification. *Journal of Machine Learning Research* 10, 2 (2009), 207–244.

[32] Qianqian Xu, Qingming Huang, and Yuan Yao. 2012. Online crowdsourcing subjective image quality assessment. In *ACM Multimedia Conference on Multimedia Conference*. 359–368.

[33] Qianqian Xu, Jiechao Xiong, Xinwei Sun, Zhiyong Yang, Xiaochun Cao, Qingming Huang, and Yuan Yao. 2018. A Margin-based MLE for Crowdsourced Partial Ranking. In *ACM Multimedia Conference on Multimedia Conference*. 591–599.

[34] Jin-Min Yang and Kin Fun Li. 2009. Recommendation based on rational inferences in collaborative filtering. *Knowledge-Based Systems* 22, 1 (2009), 105–114.

[35] Teng Zhang and Zhi-Hua Zhou. 2017. Multi-class optimal margin distribution machine. In *International Conference on Machine Learning*. 4063–4071.

[36] Teng Zhang and Zhi-Hua Zhou. 2018. Semi-Supervised Optimal Margin Distribution Machines.. In *International Joint Conference on Artificial Intelligence*. 3104–3110.

[37] Tong Zhao, Julian McAuley, and Irwin King. 2015. Improving latent factor models via personalized feature projection for one class recommendation. In *ACM International on Conference on Information and Knowledge Management*. 821–830.

[38] Zhengzhong Zhou, Xiu Di, Wei Zhou, and Liqing Zhang. 2018. Fashion Sensitive Clothing Recommendation Using Hierarchical Collocation Model. In *ACM Multimedia Conference on Multimedia Conference*. 1119–1127.