Constrained Preference Embedding for Item Recommendation

Xin Wang, Congfu Xu*, Yunhui Guo, Hui Qian
College of Computer Science and Technology, Zhejiang University, China
{cswangxinm, xucongfu, gyhui, qianhui}@zju.edu.cn

Abstract

To learn users' preference, their feedback information is commonly modeled as scalars and integrated into matrix factorization (MF) based algorithms. Based on MF techniques, the preference degree is computed by the product of user and item vectors, which is also represented by scalars. On the contrary, in this paper, we express users' feedback as constrained vectors, and call the idea constrained preference embedding (CPE); it means that we regard users, items and all users' behavior as vectors. We find that this viewpoint is more flexible and powerful than traditional MF for item recommendation. For example, by the proposed assumption, users' heterogeneous actions can be coherently mined because all entities and actions can be transferred to a space of the same dimension. In addition, CPE is able to model the feedback of uncertain preference degree. To test our assumption, we propose two models called CPE-s and CPE-ps based on CPE for item recommendation, and show that the popular pair-wise ranking model BPR-MF can be deduced by some restrictions and variations on CPE-s. In the experiments, we will test CPE and the proposed algorithms, and prove their effectiveness.

1 Introduction

How to represent customers' behavior is an important aspect for designing item recommendation algorithms. Unfortunately, there are no general and ideal solutions for different application scenarios so far. As for modeling users' explicit feedback such as rating scores, a successful assumption is to represent them as different integers. The primarily methods for leveraging them are matrix factorization (MF) techniques, according to which, users and items are represented by low-rank latent factors (i.e., numeric vectors), and preference degree is computed by the product of related vectors. However, absolutely correlating scalars with users' feedback may lead to some problems. For example, although the rating scores are uniformly distributed, the preference degree may not be liner. As we know, users' attitudes tend to be following a long-tail distribution, which means most users prefer giving 3, 4 and 5 stars, and hence the difference between 3 stars and 1 star should be more obvious than the difference between 5 stars and 3 stars.

What's more, in applications, users' behavior is not limited to rating scores, but can be heterogeneous actions. For example, a user may give some tags to a favorite book, add a pair of shoes into a shopping cart and visit some pages about children's clothing. Traditional MF based approaches may face two problems in this situation. First, it is difficult to ascertain the preference degree of those actions. Different from modeling rating scores, translating "giving some tags" and "adding an item into a shopping cart" into real numbers is a difficult task because we cannot exactly assign some values to the preference degree. The situation is even worse when we are not sure whether one type of behavior is more positive than another one. Most MF based algorithms ignore those problems and directly express all the heterogeneous feedback as integer 1 and the unobserved correlations as 0, and hence fail to capture differentiated information from each type of feedback. Second, for most MF based item recommendation algorithms, different kinds of preference information is finally translated into the same user and item latent space, hence the value in each dimension is hard to explain. Simply embedding heterogeneous information into two types of vectors is inadequate and inflexible when we import more and more kinds of information from e-commerce sites into recommendation algorithms.

To deal with the discussed problems, in this paper, we introduce a novel method called constrained preference embedding (CPE) to model users' behavior. For CPE, we no longer regard the behavior information as numerical values, but embed them in a high-dimensional space together with users and items. In other words, all entities and feedback are represented by d dimensional vectors, e.g., the rating scores from one to five are expressed as five vectors. Then, a modified *add approximation* is employed to model (user, item) correlations. This process is similar to some knowledge relationship mining methods for relationship discovery[Bordes *et al.*, 2013; Chen *et al.*, 2013], but we emphasize on fine grained actions with degree information. For CPE, the L_2-norm of the feedback vectors is used to represent preference degree in a relative way, and hence it avoids correlating them with explicit

*Corresponding author

numbers. Another advantage of the proposed idea is that, users' behavior can share an isomorphic structure and be congruently modeled by a unified method. Because we express preference degree in a relative way, each type of behavior will be assigned an auto-adjusted satisfaction value according with datasets rather than an absolute number. In addition, by embedding users' feedback, CPE can translate different kinds of behavior information to feedback vectors instead of restricting it in user and item space.

In section 2, we start by discussing matrix factorization (MF) techniques and talk about some related MF based item recommendation approaches. Then, we introduce our constrained preference embedding (CPE) method in section 3. Based on CPE, we propose two item recommendation algorithms CPE-s and CPE-ps in section 4. In section5, we test CPE's performance on real-world datasets. Finally, we draw conclusion in section 6.

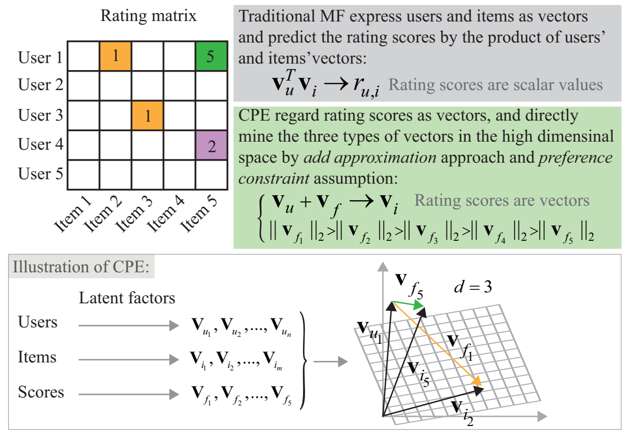

Figure 1: Illustration of CPE.

2 Background

2.1 Matrix Factorization

Based on the low-rank assumptions, the matrix factorization methods[Koren *et al.*, 2009; Koren, 2008] represent each user and item as a d dimensional vector. The preference of user u for item i is represented as a scalar $r_{u,i} \in \mathbb{R}$. If we denote the vector of u as $\mathbf{v}_u \in \mathbb{R}^{d,1}$ and the vector of i as $\mathbf{v}_i \in \mathbb{R}^{d,1}$, the preference of u for i can be predicted by $\mathbf{v}_u^T \mathbf{v}_i$. In order to learn the factors, a probabilistic version of MF (PMF)[Salakhutdinov and Mnih, 2008] assumes $r_{u,i}$ to be sampled from Gaussian distribution with the mode of $\mathbf{v}_u^T \mathbf{v}_i$, and tries to optimize the maximum posterior on all observed $(u, r_{u,i}, i)$ triplets.

2.2 Preference Ranking

To learn a ranked list of items, some related point-wise, pair-wise and list-wise preference ranking algorithms have been proposed based on matrix factorization assumptions. For example, iMF[Hu *et al.*, 2008; Lin *et al.*, 2014] and OCCF[Pan *et al.*, 2008] consider both observed and unobserved (user, item) correlations in a point-wise way and try to optimize the following loss function:

$$\sum_{(u,i) \in D^1 \cup D^0} (\mathbf{v}_u^T \mathbf{v}_i - r_{u,i})^2 + \Theta \tag{1}$$

where D^1 and D^0 are observed set and unobserved set, and Θ is the regularization term $\lambda_u ||\mathbf{v}_u||_2^2 + \lambda_i ||\mathbf{v}_i||_2^2$.

The MF based pair-wise methods are similar to iMF or OCCF, but they adopt different preference comparing structures. For example, the state-of-the-art pair-wise algorithm BPR-MF[Rendle *et al.*, 2009; Pan and Chen, 2013] assumes user u may prefer an observed item i than an unobserved item j, and directly optimizes their relationships by logistic regression with the parameter $(\mathbf{v}_u^T \mathbf{v}_i - \mathbf{v}_u^T \mathbf{v}_j)$. Finally, we maximize the following equation:

$$\sum_{(u,i) \in D^1, (u,j) \in D^0} \ln \sigma(\mathbf{v}_u^T \mathbf{v}_i - \mathbf{v}_u^T \mathbf{v}_j) + \Theta \tag{2}$$

For MF based list-wise algorithms, each action is compared with a list of actions. For example, CofiRank[Weimer *et al.*, 2007] is proposed to directly optimize NDCG[Valizadegan *et al.*, 2009] scores based on maximum margin matrix factorization models, and ListRank[Cao *et al.*, 2007; Shi *et al.*, 2010] tries to optimize a function of cross entropy of item lists based on PMF.

As we can see from the above loss functions, those MF based algorithms are powerful for modeling isomorphic feedback with assigned preference degree information, but some of them may be inflexible and ineffective when the following situations occur:

(1) It is difficult to determine the preference degree $p(f)$ of feedback f (e.g., $\{p(f_a) =?, p(f_b) =?, p(f_c) =?\}$).

(2) We are only given the relative positiveness of some types of feedback(e.g., $\{p(f_a) < p(f_b), p(f_c) < p(f_d)\}$).

(3) We try to leverage heterogeneous feedback and combine them in a single model (e.g., $\{p(f_a) = 1, p(f_b) = 2, p(f_c) =?, p(f_d) < p(f_e), p(f_g) =?, p(f_h) =?, ...\}$).

where, $p(\cdot)$ denotes preference degree. Note that, here f is a type of feedback, e.g., "giving 2 stars", "click" or "browse". For example, if we denote "giving two stars" as feedback b, then $p(f_b)$ may be 2, i.e., $p(f_b) = r_{i,j}$ if $r_{i,j} = 2$.

3 Constrained Preference Embedding

In this section, we introduce our proposed method constrained preference embedding (CPE) and discuss some of its characteristics. We denote a kind of feedback f as $\mathbf{v}_f \in \mathbb{R}^{d,1}$. Therefore, the rating scores $\{1, 2, 3, 4, 5\}$ are represented as $\{\mathbf{v}_{(1\ star)}, \mathbf{v}_{(2\ stars)}, \mathbf{v}_{(3\ stars)}, \mathbf{v}_{(4\ stars)}, \mathbf{v}_{(5\ stars)}\}$, and users' preference can be optimized by the following function:

$$\min \mathcal{L}_{u \in U, f \in F, i \in I}(\mathbf{v}_u, \mathbf{v}_f, \mathbf{v}_i) \tag{3}$$

where U, I and F are user, item and feedback set, $\mathbf{v}_u$ and $\mathbf{v}_i$ are user vector and item vector.

In this paper, we assume users, items and preference obey *add approximation* rule, which means $\mathbf{v}_u + \mathbf{v}_f$ should be close to $\mathbf{v}_i$ if u gives f to i.The idea is illustrated in Figure 1. Based on the rating matrix and *add approximation*, we should optimize $\mathbf{v}_{u_1} + \mathbf{v}_{1\ star} \rightarrow \mathbf{v}_{i_2}$, $\mathbf{v}_{u_1} + \mathbf{v}_{5\ stars} \rightarrow \mathbf{v}_{i_5}$,

$\mathbf{v}_{u_3} + \mathbf{v}_{1\ star} \rightarrow \mathbf{v}_{i_3}$ and $\mathbf{v}_{u_4} + \mathbf{v}_{2\ stars} \rightarrow \mathbf{v}_{i_5}$, where "$\rightarrow$" indicates "approximates".

For item recommendation, we also need to introduce collaborative filtering[Sarwar *et al.*, 2001] features and preference degree information by giving some constraints on users' feedback. We assume that if u has strong preference for i, $\mathbf{v}_u$ and $\mathbf{v}_i$ should be close; if u has similar preference for i_1 and i_2, $\mathbf{v}_{i_1}$ and $\mathbf{v}_{i_2}$ should be close; if u_1 and u_2 have similar preference for i, $\mathbf{v}_{u_1}$ and $\mathbf{v}_{u_2}$ should be close. To achieve this, we control the L_2-norm (i.e., vector norm) of $\mathbf{v}_f$ to make sure that it is smaller if f is more positive. Because the square root of L_2-norm of $\mathbf{v}_f$ (i.e., $||\mathbf{v}_f||_2$) is the Euclidean distance, based on the former rules, the stronger the preference of u for i, the shorter the distance between $\mathbf{v}_u$ and $\mathbf{v}_i$. It means that the correlated (user, item), (user, user) or (item, item) should have similar direction and length in terms of their vectors.

4 CPE for Item Recommendation

In this section, we propose two item recommendation methods called CPE-s and CPE-ps based on our CPE assumption. For CPE-s, we adopt *add approximation* rule and pair-wise preference comparison strategy, and optimize a loss function with soft constraints. CPE-ps is similar to CPE-s, but it is based on vector projection. In this paper, we primarily consider rating scores and unobserved (user, item) correlations (denoted as "unobserved feedback") for studying our models.

4.1 CPE-s

We assume that if user u has a strong preference for item i, she will give it a higher rating score. Therefore, our task is to minimize the difference between $\mathbf{v}_u + \mathbf{v}_f$ and $\mathbf{v}_i$ for each triplet (u, f, i). Given users' rating matrix, the loss function on the overall triplets is as follows:

$$\begin{aligned} \mathcal{L} &= \sum_{(u,f,i)\in D} d(\mathbf{v}_u + \mathbf{v}_f - \mathbf{v}_i)^2 + \Theta \\ s.t. &\quad \{||\mathbf{v}_{f_p}||_2^2 < ||\mathbf{v}_{f_q}||_2^2 \mid p > q\}, \quad f_p, f_q \in F \end{aligned} \tag{4}$$

where $(u, f, i) \in D$ denotes the observed and unobserved triplets; $d(\cdot)$ is Euclidean distance; Θ controls the norm of the parameters. For the rating scores, $\mathbf{v}_{f_k}$ is the vector of "giving k star(s)" action and comes from the set $\{\mathbf{v}_{f_k}|k = 1, 2, 3, 4, 5\}$, and the "unobserved feedback" is denoted as $\mathbf{v}_{f_0}$. For our model, f_p is assumed to be more positive than f_q for all $p > q$, which means the correlation of observed triplets are stronger than the unobserved ones.

All the vectors are randomly constructed. The possible $\mathbf{v}_u$, $\mathbf{v}_i$ and $\mathbf{v}_f$ in the above loss function are optimized with L_2-regularization terms. This constraint is important for CPE-s because it prevents the learning algorithm to trivially minimize the optimization function by artificially increasing the norms of $\mathbf{v}_u$, $\mathbf{v}_i$ and $\mathbf{v}_f$.

Instead of directly optimizing the loss function with constraints on preference, we convert them to the following soft unconstrained function:

$$\begin{aligned} \mathcal{L} = &\sum_{(u,f,i)\in D} d(\mathbf{v}_u + \mathbf{v}_f - \mathbf{v}_i)^2 + \Theta \\ &- \sum_{p>q} \frac{w_{f_p,f_q}}{e} \ln \mathcal{C}(||\mathbf{v}_{f_q}||_2^2 - ||\mathbf{v}_{f_p}||_2^2) \end{aligned} \tag{5}$$

where p and q belong to F; e is hyperparameter; w_{f_p,f_q} denotes the weight of the preference comparison between f_p and f_q; it is computed by the product of f_p's and f_q's frequency in the triplets, and guarantees that the feedback priority can be adjusted according to the datasets instead of some predetermined values. Here $\mathcal{C}(x)$ can be some monotonic increasing functions. We here adopt the sigmoid function.

4.2 CPE-ps

For CPE-s, if both u_1 and u_2 give i the same rating score, $\mathbf{v}_{u_1}$ and $\mathbf{v}_{u_2}$ should be very similar, which means CPE-s may suppress exploiting personalized information. The reason leading to the consequence is that for the elements in F (i.e., $\{f_k|k = 0, 1, 2, 3, 4, 5\}$), we keep the same user and item vectors. To overcome the shortcomings, we adopt a projection method inspired by [Wang *et al.*, 2014b] to model vector relationships. We create a plane P_f with the normal vector $\mathbf{w}_f$ for each type of feedback $\mathbf{v}_f$. Then, for each triplet $(\mathbf{v}_u, \mathbf{v}_f, \mathbf{v}_i)$, we project $\mathbf{v}_u$ and $\mathbf{v}_i$ to P_f, denoting the projected vectors as $\mathbf{v}_{\perp,u}$ and $\mathbf{v}_{\perp,i}$ respectively. Finally, we optimize $(\mathbf{v}_{\perp,u}, \mathbf{v}_f, \mathbf{v}_{\perp,i})$ similar to CPE-s, where $\mathbf{v}_{\perp,u}$ and $\mathbf{v}_{\perp,i}$ are computed according to the following equations:

$$\begin{aligned} \mathbf{v}_{\perp,u} &= \mathbf{v}_u - \mathbf{w}_f^T \mathbf{v}_u \mathbf{w}_f \\ \mathbf{v}_{\perp,i} &= \mathbf{v}_i - \mathbf{w}_f^T \mathbf{v}_i \mathbf{w}_f \end{aligned} \tag{6}$$

Therefore, the loss function can be described as $d(\mathbf{v}_{\perp,u} + \mathbf{v}_f - \mathbf{v}_{\perp,i})$ and is illustrated in Figure 2(a). The L_2-norm of $\mathbf{w}_f$ is restricted to 1 to control $\mathbf{v}_{\perp,u}$ and $\mathbf{v}_{\perp,i}$. Based on our CPE assumption, the loss function is

$$\begin{aligned} \mathcal{L} &= \sum_{(u,f,i)\in D} d(\mathbf{v}_{\perp,u} + \mathbf{v}_f - \mathbf{v}_{\perp,i})^2 \\ s.t. &\quad \begin{cases} \{||\mathbf{v}_{f_p}||_2^2 < ||\mathbf{v}_{f_q}||_2^2 \mid p > q\}, & f_p, f_q \in F \\ ||\mathbf{w}_f||_2^2 = 1, & f \in F \\ \mathbf{w}_f^T \mathbf{v}_f / ||\mathbf{v}_f||_2 \leq \epsilon, & f \in F \end{cases} \end{aligned} \tag{7}$$

where $d(\cdot)$ is the Euclidean distance; $\mathbf{w}_f^T \mathbf{v}_f / ||\mathbf{v}_f||_2 \leq \epsilon$ guarantees that $\mathbf{v}_f$ is in the translated plane. Similar to CPE-s, we do not directly optimize the above function but convert it to the following soft unconstrained loss function:

$$\begin{aligned} \mathcal{L} = &\sum_{(u,f,i)\in D} d(\mathbf{v}_{\perp,u} + \mathbf{v}_f - \mathbf{v}_{\perp,i})^2 + \frac{w_f}{e} \sum_{f\in F} \frac{(\mathbf{w}_f^T \mathbf{v}_f)^2}{||\mathbf{v}_f||_2^2} \\ &- \frac{w_{f_p,f_q}}{e} \sum_{p>q} \ln \mathcal{C}(||\mathbf{v}_{f_q}||_2^2 - ||\mathbf{v}_{f_p}||_2^2) + \Theta \end{aligned} \tag{8}$$

where $|||\mathbf{w}_f||_2^2$ is constrained to 1 in the learning procedure; w_f is the weight of f.

CPE-ps is better than CPE-s because it can help exploit more personalized information. For example, $\mathbf{v}_{u_1}$ and $\mathbf{v}_{u_2}$ can be different even if $\mathbf{v}_{\perp,u_1}$ equals to $\mathbf{v}_{\perp,u_2}$. That is, users' preference can be shared through projected correlations, while some personalized information can be remained.

4.3 Learning and Item Recommendation

The proposed algorithms are carried out by stochastic gradient descent (SGD) to optimize $\mathbf{v}_u, \mathbf{v}_f$ and $\mathbf{v}_i$. Specifically,

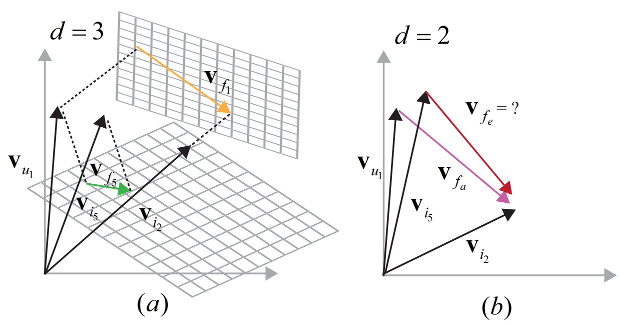

Figure 2: Illustration of CPE-ps and SCPE.

for preference embedding and positiveness learning, in each iteration, we randomly sample t_1 triplets according to the distribution of each feedback in D^1 and t_2 unobserved feedback, and optimize them according to E.q.(5) and E.q.(8) until they converge to a stable state. Note that for item recommendation in the experiments, we directly update the related $\mathbf{v}_u$ and $\mathbf{v}_i$ on $\ln \mathcal{C}(||\mathbf{v}_{f_q}||_2^2 - ||\mathbf{v}_{f_p}||_2^2)$ part for better learning efficiency and performance. The form can be expressed by E.q.(13), and we will discuss it later.

The u's preference for i is computed according to E.q.(9) for CPE-s, and E.q.(10) for CPE-ps, and the top k items with the greatest $\tilde{p}_{u,i}$ are recommended to u.

$$\tilde{p}_{u,i} = 1/(||\mathbf{v}_i - \mathbf{v}_u||_2^2 + 1) \tag{9}$$

$$\tilde{p}_{u,i} = 1/(||\mathbf{v}_{\perp,i} - \mathbf{v}_{\perp,u}||_2^2 + 1) \tag{10}$$

where p denotes preference degree.

4.4 Discussion

In this section, we discuss some features of CPE and explain why they are flexible for preference learning.

Semi-constrained preference embedding (SCPE). An advantage of CPE is that it can model users' heterogeneous actions in a unified way. In applications, although the ratings can be directly modeled by the scores, it is hard to specify some types of implicit feedback (e.g., "click" and "browse") to a certain preference degree, because we are not sure, for example, whether "click" and "browse" are more positive than "giving 2 stars". The MF based algorithms cannot directly model the latter information well; a compromise is to represent all types of users' behavior as implicit feedback, and correlate them with 1 (e.g., BPR-MF). Those approaches may have some shortcomings because the diversity of behavior information is lost. On the contrary, a modified CPE method called semi-constrained preference embedding (SCPE) may provide a possible fine grained solution. Specifically, for SCPE, all the elements in F are considered in the *add approximation* part, but the L_2-norm of the unknown feedback is not constrained in the learning process. For example, given the set $\{f_k | k = a, b, c, d, e\}$ and the experience that $p(f_a) > p(f_b)$, $p(f_c) > p(f_d)$, and $p(f_e) =?$, the loss function can be

$$\begin{aligned} \min \quad & \mathcal{L}(\mathbf{v}_u, \mathbf{v}_i, \mathbf{v}_{f_a}, \mathbf{v}_{f_b}, \mathbf{v}_{f_c}, \mathbf{v}_{f_d}, \mathbf{v}_{f_e}) \\ s.t. \quad & \begin{cases} \mathcal{C}_1(||\mathbf{v}_{f_b}||_2^2 - ||\mathbf{v}_{f_a}||_2^2) \\ \mathcal{C}_2(||\mathbf{v}_{f_d}||_2^2 - ||\mathbf{v}_{f_c}||_2^2) \end{cases} \end{aligned} \tag{11}$$

where f_e is modeled in $\mathcal{L}(\cdot)$ to help correlate (user, item) pairs and predict users' potential preference, which is illustrated in Figure 2(b). We also find that besides benefiting item recommendation, SCPE can also help learn preference degree of f_e by the help of $f_{i \in a,b,c,d}$. Therefore, besides for item recommendation, SCPE may also be used to study whether a user's behavior is positive or not according to some assistant behavior. We will study it in the next section.

Relation between CPE-s and BPR-MF. As we discussed above, BPR-MF is an effective pair-wise ranking algorithm for item recommendation, and its loss function is represented in E.q.(2). For our CPE-s, if we assume $\mathcal{C}(\cdot)$ in E.q.(5) as logistic regression, and denote f_p as positive feedback and f_n as negative feedback, the main part of the soft preference restriction can be expressed as

$$-\ln \sigma(||\mathbf{v}_{f_n}||^2 - ||\mathbf{v}_{f_p}||^2) \tag{12}$$

With pair-wise CPE assumption, for two triplets (u, f_p, i) and (u, f_n, j), the distance between $\mathbf{v}_u$ and $\mathbf{v}_i$ should be shorter than the distance between $\mathbf{v}_u$ and $\mathbf{v}_j$. Based on *add approximation*, we replace $\mathbf{v}_{f_p}$ with $\mathbf{v}_i - \mathbf{v}_u$ and $\mathbf{v}_{f_n}$ with $\mathbf{v}_j - \mathbf{v}_u$, and constrain L_2-norm of item vectors to a constant c. Hence, our task is to maximize the following function:

$$\begin{aligned} & \sum \ln \sigma(||\mathbf{v}_j - \mathbf{v}_u||_2^2 - ||\mathbf{v}_i - \mathbf{v}_u||_2^2) + \Theta \\ = & \sum \ln \sigma(2[\mathbf{v}_u^T \mathbf{v}_i - \mathbf{v}_u^T \mathbf{v}_j] + ||\mathbf{v}_i||_2^2 - ||\mathbf{v}_j||_2^2) + \Theta \\ = & \sum \ln \sigma(2[\mathbf{v}_u^T \mathbf{v}_i - \mathbf{v}_u^T \mathbf{v}_j]) + \Theta \\ & s.t. \quad ||\mathbf{v}_i||^2 = ||\mathbf{v}_j||^2 = c \end{aligned} \tag{13}$$

As we can see from E.q.(13), by some restrictions and variations, we can deduce BPR-MF with a constraint of item L_2-norm by pair-wise CPE. This reveals the potential relation between CPE-s and BPR-MF, and also shows why it is reasonable to express $p_{u,i}$ as $1/(||\mathbf{v}_i - \mathbf{v}_u||_2^2 + 1)$.

5 Experiments

5.1 Datasets and Evaluation Metrics

In this section, we study CPE on some real-world datasets in different domains and categories. The number of users, the number of items and the sparsity information is listed in Table 1. The first 5 datasets are from Amazon.com[McAuley and Leskovec, 2013]; the *movies*, *books* and *music* datasets are from DouBan.com[Wang *et al.*, 2014a]. Each dataset is subdivided into three parts; 80% of it is used for training, 10% is used for validation and the last 10% is left for test.

The evaluation metrics we used are NDCG[Valizadegan *et al.*, 2009],Precision, and F1[Wang *et al.*, 2014a] scores.

5.2 Baselines and Parameter Settings

To comprehensively study our proposed models, we compare them with the following state-of-the-art ranking methods:

Table 1: Datasets used in the experiments.

dataset	#users	#items	sparsity
beauty (Amazon)	167,725	29,004	0.99995
tools&games (Amazon)	283,514	51,004	0.99997
clothing&accessories(Amazon)	128,794	66,370	0.99993
shoes (Amazon)	73,590	48,410	0.99989
industrial&scientific (Amazon)	29,590	22,622	0.99980
movies (Douban)	5,664	10,013	0.97423
books (Douban)	10,024	10,115	0.99280
music (Douban)	10,208	11,200	0.99157

- point-wise: iMF *(explicit, implicit)*
- pair-wise: BPR *(implicit)*, GBPR *(implicit)*
- list-wise: ListRank *(explicit)*, CofiRank *(explicit)*

The baselines are carefully chosen to make sure that each algorithm is typical in each discussed class. ListRank [Cao *et al.*, 2007; Shi *et al.*, 2010] and CofiRank [Weimer *et al.*, 2007] are mainly for optimizing observed (user, item) correlations, while iMF [Hu *et al.*, 2008; Lin *et al.*, 2014], BPR [Rendle *et al.*, 2009] and GBPR [Pan and Chen, 2013] consider both observed and unobserved correlations. Some details of those methods are discussed in the previous sections.

For all the approaches, the learning rate is set to 0.05, and the latent dimension is set to 10 (i.e., $d = 10$). The regularization coefficient is selected from $\{1, 0.1, 0.01, 0.001\}$; $t_1 = t_2 = 2$; Because the adopted weighted sampling method for SGD, we set e, w_{f_p,f_q} and w_f to 1. For GBPR-MF, the group size is 3.

5.3 Analysis of Preference Embedding

We reconstruct the L_2-norm of the feedback vectors (i.e., $\{\mathbf{v}_{f_i} | i = 0, 1, 2, 3, 4, 5\}$) after learning CPE-s or CPE-ps, and illustrate them by column charts in Figure 3. Due to the limited space, we only provide the results learned by CPE-s.

According to the figure, $||\mathbf{v}_{f_0}||_2$ is much greater than $||\mathbf{v}_{f_{i\in\{1,2,3,4,5\}}}||_2$ for all datasets; it implies that whether a user gives a rating score or not to an item is more significant than which score she chooses. The results are due to the fact that the number of (u, f_0, i) triplets is much larger than other triplets', and the assumption that the members in $\{f_i|1, 2, 3, 4, 5\}$ should be more positive than f_0. Hence, the difference between $||\mathbf{v}_{f_0}||_2$ and $||\mathbf{v}_{f_{i\in\{1,2,3,4,5\}}}||_2$ has an important influence on model optimization. With the same reason, we can also explain why the difference between $||\mathbf{v}_{f_{i\in\{4,5\}}}||_2$ and $||\mathbf{v}_{f_{j\in\{1,2\}}}||_2$ is greater comparing with $||\mathbf{v}_{f_4}||_2 - ||\mathbf{v}_{f_5}||_2$ or $||\mathbf{v}_{f_1}||_2 - ||\mathbf{v}_{f_2}||_2$. Hence, the length of feedback vectors learned by our models can intuitively reflect the behavior distribution on the real-world datasets.

5.4 Analysis of Item Recommendation

The item recommendation results on 8 real-world datasets are listed in Table 2. It is interesting that the selected point-wise and pair-wise algorithms are better than the list-wise ranking methods, which is inconsistent with our intuition. The reason for the outcomes is that the compared list-wise ranking methods mainly consider explicit rating scores and ignore unobserved (user, item) pairs. For our sparse datasets, CofiRank

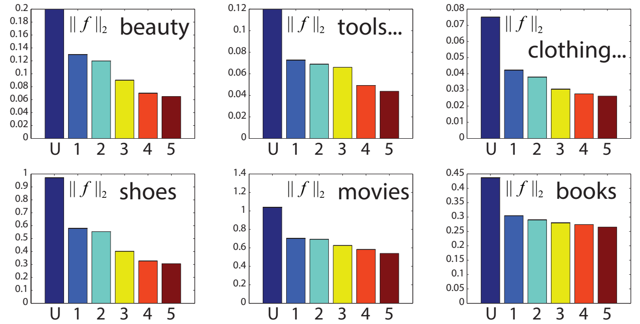

Figure 3: L_2-norm of the learned feedback vectors. Here "U" represents the unobserved (user, item) pairs, and the numerical numbers denote the feedback about related rating scores, with the bar indicating their vector length.

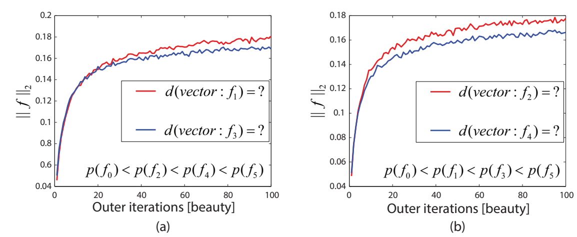

Figure 4: L_2-norm comparison of two feedback vectors learned by CPE with some assistant feedback.

and ListRank cannot take full advantage of the available data due to the shortage of the former information.

It is obvious that our proposed models are better than BPR and GBPR on most datasets. On average, CPE-s (or CPE-ps) can help improve 14.75% precision-5, 15.41% recall-5, and 16.17% NDCG-5 due to the preference embedding assumptions. First, CPE-s and CPE-ps consider 6 different levels of users' feedback, but BPR and GBPR primarily focus on binary information. Therefore, our models can exploit more information than BPR and GBPR. Second, our methods transfer each type of behavior to its related vector rather than restrict it in user and item space. Therefore, they are more effective comparing with other baselines.

The latent dimension $d \in \{3, 5, 10, 15\}$ is changed to test model stabilities of the compared approaches. The outcomes of NDCG-5 on *beauty* and *clothing&accessories* are shown in Figure 6. We find that when d is large, the advantage of CPE is obvious; it is because the feedback vectors may help capture more preference related information as well as loose the low-rank assumption when d is bigger. The benefit is reducing with the decreasing of d, but due to the effectiveness of preference embedding, the performance of CPE is still better than the baselines. Finally, we study NDCG-K, precision-K and recall-K scores with different recommendation size $K \in \{1, 3, 5, 7\}$, and show some selected results in Figure 6. It is clear that CPE-s and CPE-ps are stable when K varies.

Table 2: Prediction, recall and NDCG scores on 8 real-world datasets. The size of recommendation list is 5.

metric	method	beauty[a]	tools& games[a]	clothing& accessories[a]	shoes[a]	industrial &scientific[a]	movies[d]	books[d]	music[d]
Precision-5	iMF	0.0306	0.0075	0.1054	0.1212	0.0434	0.1125	0.0231	0.0378
	BPR	0.0380	0.0093	0.1456	0.1845	0.0597	0.1403	0.0405	0.0570
	GBPR	0.0374	0.0094	0.1502	0.1896	0.0575	**0.1444**	0.0429	0.0598
	ListRank	0.0149	0.0029	0.0482	0.0802	0.0232	0.0903	0.0170	0.0138
	CofiRank	0.0164	0.0039	0.0607	0.0819	0.0285	0.1043	0.0256	0.0242
	CPE-s	0.0431	0.0119	**0.1731**	0.1957	0.0780	0.1434	0.0440	0.0619
	CPE-ps	**0.0443**	**0.0121**	0.1669	**0.2092**	**0.0802**	0.1410	**0.0460**	**0.0637**
Recall-5	iMF	0.1232	0.0288	0.2919	0.3484	0.0889	0.0139	0.0085	0.0128
	BPR	0.1499	0.0358	0.3788	0.4880	0.1181	0.0202	0.0169	0.0230
	GBPR	0.1477	0.0366	0.3887	0.5005	0.1152	**0.0213**	0.0179	0.0247
	ListRank	0.0663	0.0131	0.1656	0.2612	0.0537	0.0106	0.0054	0.0023
	CofiRank	0.0754	0.0174	0.1993	0.2715	0.0641	0.0089	0.0082	0.0045
	CPE-s	0.1678	0.0458	**0.4414**	0.4895	0.1504	0.0207	0.0187	0.0272
	CPE-ps	**0.1729**	**0.0468**	0.4108	**0.5416**	**0.1652**	0.0208	**0.0195**	**0.0278**
NDCG-5	iMF	0.0996	0.0239	0.2327	0.2709	0.0819	0.1134	0.0229	0.0391
	BPR	0.1305	0.0305	0.3281	0.4344	0.1184	0.1427	0.0417	0.0601
	GBPR	0.1273	0.0313	0.3398	0.4509	0.1137	0.1437	0.0441	0.0632
	ListRank	0.0480	0.0083	0.1215	0.1912	0.0408	0.1003	0.0180	0.0142
	CofiRank	0.0526	0.0128	0.1443	0.1978	0.0565	0.1097	0.0271	0.0254
	CPE-s	0.1510	0.0398	**0.3982**	0.4355	0.1518	0.1462	0.0456	0.0673
	CPE-ps	**0.1569**	**0.0411**	0.3685	**0.4870**	**0.1620**	**0.1484**	**0.0483**	**0.0683**

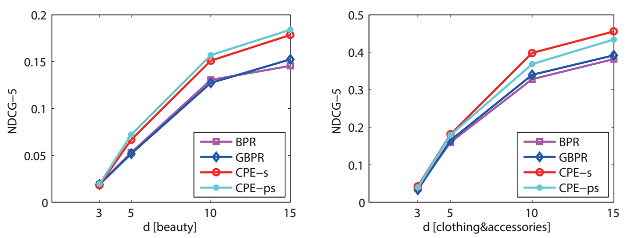

Figure 5: Performance comparison with different latent dimension d.

5.5 Analysis of Semi-CPE (SCPE)

We use rating scores and unobserved correlations to study SCPE's ability on identifying behavior's positiveness. Specifically, two types of feedback f_p and f_q are randomly selected from $\{f_k|k = 0,1,2,3,4,5\}$ to simulate unknown behavior, then every feedback vectors are embedded by SCPE-s, with f_p and f_q not constrained in the optimizing procedure. We will test whether SCPE-s can correctly rank $p(f_p)$ and $p(f_q)$. In the experiments, we choose the following two cases: $\{p = 3, q = 1\}$ and $\{p = 4, q = 2\}$. $||\mathbf{v}_{f_1}||_2$ and $||\mathbf{v}_{f_3}||_2$ learned by SCPE-s on *beauty* dataset are plotted in Figure 4(a), and $||\mathbf{v}_{f_2}||_2$ and $||\mathbf{v}_{f_4}||_2$ are shown in Figure 4(b). It is clear that when the algorithm converges, $||f_1||_2$ is persistently greater than $||f_3||_2$, and $||f_2||_2$ is greater than $||f_4||_2$, which means that, with other assistant feedback, our SCPE-s can automatically infer that "giving 3 scores" is more positive than "giving 1 score", and "giving 4 scores" is more positive comparing with "giving 2 scores". The results are in line with what we expected, and can be explained by collaborative filtering features of SCPE discussed before. Therefore, SCPE may provide a possible way for analyzing and comparing users' heterogeneous feedback for e-commerce websites.

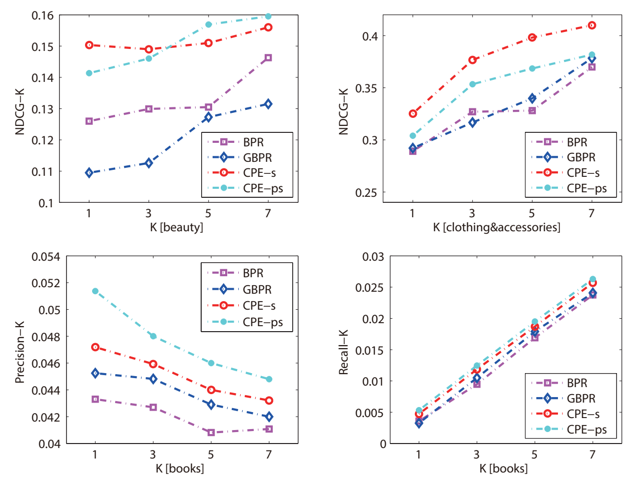

Figure 6: Performance comparison with different recommendation list size.

6 Conclusion

In this paper, we introduced the constrained preference embedding (CPE) assumption and two models (i.e., CPE-s and CPE-ps) for preference learning. We discussed semi-constrained preference embedding (SCPE) and showed its effectiveness on modeling users' feedback of uncertain preference degree. Finally, we demonstrated the relationship between CPE and BPR-MF. In the experiments, CPE is proved effective from different perspectives.

Acknowledgements

This work is supported by National Natural Science Foundation of China (Grant No: 61272303 and Grant No: 61472347). We thank Dr. Zhongyuan Wang in MSRA for helpful discussions on embedding techniques.

References

[Bordes *et al.*, 2013] Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, NIPS'13, pages 2787–2795, 2013.

[Cao *et al.*, 2007] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 129–136. ACM, 2007.

[Chen *et al.*, 2013] Danqi Chen, Richard Socher, Christopher D Manning, and Andrew Y Ng. Learning new facts from knowledge bases with neural tensor networks and semantic word vectors. *arXiv preprint arXiv:1301.3618*, 2013.

[Hu *et al.*, 2008] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Proceedings of the 8th International Conference on Data Mining*, ICDM'08, pages 263–272. IEEE, 2008.

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

[Koren, 2008] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, SIGKDD'13, pages 426–434. ACM, 2008.

[Lin *et al.*, 2014] Christopher H. Lin, Ece Kamar, and Eric Horvitz. Signals in the silence: Models of implicit feedback in a recommendation system for crowdsourcing. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, AAAI'14, pages 22–28, 2014.

[McAuley and Leskovec, 2013] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys'13, pages 165–172. ACM, 2013.

[Pan and Chen, 2013] Weike Pan and Li Chen. Gbpr: Group preference based bayesian personalized ranking for one-class collaborative filtering. In *Proceedings of the 23th International Joint Conference on Artificial Intelligence*, IJCAI'13, pages 2691–2697. AAAI Press, 2013.

[Pan *et al.*, 2008] Rong Pan, Yunhong Zhou, Bin Cao, Nathan Nan Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. One-class collaborative filtering. In *Proceedings of the 8th International Conference on Data Mining*, ICDM'08, pages 502–511. IEEE, 2008.

[Rendle *et al.*, 2009] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence*, pages 452–461. AUAI Press, 2009.

[Salakhutdinov and Mnih, 2008] Ruslan Salakhutdinov and Andriy Mnih. Probabilistic matrix factorization. In *Advances in Neural Information Processing Systems*, volume 20 of *NIPS'08*, 2008.

[Sarwar *et al.*, 2001] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th International Conference on World Wide Web*, WWW'01, pages 285–295. ACM, 2001.

[Shi *et al.*, 2010] Yue Shi, Martha Larson, and Alan Hanjalic. List-wise learning to rank with matrix factorization for collaborative filtering. In *Proceedings of the 4th ACM Conference on Recommender Systems*, RecSys'10, pages 269–272. ACM, 2010.

[Valizadegan *et al.*, 2009] Hamed Valizadegan, Rong Jin, Ruofei Zhang, and Jianchang Mao. Learning to rank by optimizing ndcg measure. In *Advances in Neural Information Processing Systems*, NIPS'09, pages 1883–1891, 2009.

[Wang *et al.*, 2014a] Xin Wang, Weike Pan, and Congfu Xu. Hgmf: Hierarchical group matrix factorization for collaborative recommendation. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 769–778. ACM, 2014.

[Wang *et al.*, 2014b] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, AAAI'14, pages 1112–1119. Citeseer, 2014.

[Weimer *et al.*, 2007] Markus Weimer, Alexandros Karatzoglou, Quoc Viet Le, and Alex Smola. Maximum margin matrix factorization for collaborative ranking. In *Advances in Neural Information Processing Systems*, NIPS'07, 2007.