User Graph Regularized Pairwise Matrix Factorization for Item Recommendation

Liang Du[1,2], Xuan Li[1,2], and Yi-Dong Shen[1]

[1] State Key Laboratory of Computer Science,
Institute of Software, Chinese Academy of Sciences, Beijing 100190, China
[2] Graduate University, Chinese Academy of Sciences, Beijing 100049, China
{duliang,lixuan,ydshen}@ios.ac.cn

Abstract. Item recommendation from implicit, positive only feedback is an emerging setup in collaborative filtering in which only one class examples are observed. In this paper, we propose a novel method, called User Graph regularized Pairwise Matrix Factorization (UGPMF), to seamlessly integrate user information into pairwise matrix factorization procedure. Due to the use of the available information on user side, we are able to find more compact, low dimensional representations for users and items. Experiments on real-world recommendation data sets demonstrate that the proposed method significantly outperforms various competing alternative methods on top-k ranking performance of one-class item recommendation task.

Keywords: Pairwise Matrix Factorization, User Graph, Item Recommendation.

1 Introduction

Recommender systems have become a core component for today's personalized online businesses. Most recent work is on scenarios where users provide explicit feedback, e.g. in terms of ratings. One of the well known example is the Netflix Prize problem. However, Such explicit feedback is hard to collect in many applications because of the intensive user involvement. In fact, most of the feedback in real-world applications is not explicit but implicit. Typical examples include web page bookmarking and Amazon's production recommendation. Therefore, in this paper we focus on item recommendation from implicit, positive only feedback [1]. It is also known as one class collaborative filtering (OCCF) [2]. Its task is to provide personalized ranking on a set of items according to the score of user-item elements based on the feedback information.

In recent years, some matrix factorization models have been proposed to tackle this problem. Pan et.al. [2] proposed to weight and sample unobserved user-item elements and learn a matrix factorization model by minimizing the weighted element-wise squared loss. The essential idea is to treat all missing user-item examples as negative and assign proper weights on these entries. The weighting schemes include uniform, user-specific, and item specific weightings. [3] proposed

J. Tang et al. (Eds.): ADMA 2011, Part II, LNAI 7121, pp. 372–385, 2011.

to approximate the weight matrix with low-rank matrices for large-scale OCCF. [4] proposed to improve OCCF accuracy by exploiting rich context information to weight the binary matrix. One drawback of these methods is that they focus on the optimization of the squared loss on user-item elements, without considering the partial order induced from pairwise element comparison. Such partial order information is valuable in the ranking of items. Rendle et.al. [1] proposed a Bayesian personalized ranking (BPR) optimization framework, which is used for optimizing different kinds of models based on implicit feedback, such as k-nearest-neighbor (KNN) [1], matrix factorization [1] and tensor factorization [5] for the task of item recommendation. BPR's key idea is to make use of partial order of items, instead of using single user-item elements, to train a recommendation model.

On the other hand, the idea of incorporating contextual information to a recommendation algorithm to improve the performance of rating based prediction tasks is used in [6, 7]. Such contextual information includes neighborhood information in the user-item rating matrix [6], content information such as user's occupation and item's genre [6, 7], and social network connections such as social trust/distrust network and social friend relationships [8, 9, 10, 11]. These models can be seen as an integration of memory based and model based approaches. Apparently, these extensions are based on element-wise matrix factorization models for rating prediction tasks.

These observations suggest that it is desirable to combine the pairwise matrix factorization based on BPR and the contextual information in a unified model. In this paper we propose such a model, called *user graph regularized pairwise matrix factorization* (UGPMF), to seamlessly integrate user side contextual information into the pairwise matrix factorization procedure. Specifically, we construct similarity graphs on user side by exploiting the available user information, and use the graphs to regularize the pairwise matrix factorization procedure based on BPR. Our method inherits the advantage of pairwise matrix factorization method [1]. Due to the use of user information, it also has the merits of memory based methods. It can alleviate the overfitting problem suffered by the matrix factorization model by the additional graph regularization. Experiments on real-world data sets show that the proposed method can effectively improve the top-k ranking performance.

The paper is organized as follows. Section 2 introduces some notation and reviews the matrix factorization based on Bayesian personalized ranking [1]. Section 3 describes the details of our model. Section 4 gives the experimental results, and Sction 5 concludes the paper and presents some future work.

2 Matrix Factorization Based on Bayesian Personalized Ranking

Let $U = \{u_1, u_2, \ldots, u_n\}$ be a group of users, $I = \{i_1, i_2, \ldots, i_m\}$ be a set of items, and $\boldsymbol{X} = (X_{ui})_{n\times m} \in \{0, 1\}^{n\times m}$ be the user-item binary preference matrix with n users and m items, such as a typical who-bought-what customer-product matrix. $\boldsymbol{X}_{ui} = 1$ denotes customer-product purchases, while $\boldsymbol{X}_{ui} = 0$ means that no purchase was made.

In a matrix factorization based model, we usually seek two low-rank matrices $\boldsymbol{W} \in \mathcal{R}^{n \times D}$ and $\boldsymbol{H} \in \mathcal{R}^{m \times D}$. The row vectors $\boldsymbol{W}_u, 1 \leq u \leq n$ and $\boldsymbol{H}_i, 1 \leq i \leq m$ represent the low dimension representations of users and items respectively.

2.1 Bayesian Personalized Ranking

BPR's key idea is to use partial order of items, instead of single user-item examples, to train a recommendation model. It allows the interpretation of positive-only data as partial ordering of items. When we observed a positive user-item example of user u on an item i, e.g. user u viewed or purchased item i, we assume that the user prefers this item than all other non-observed items. Formally we can extract a pairwise preference dataset $\mathcal{P} : U \times I \times I$ by

$$\mathcal{P} := \{(u, i, j) | i \in I_u^+ \wedge j \in I \setminus I_u^+\} \tag{1}$$

where I_u^+ and $I \setminus I_u^+$ are the positive item set and missing set associated with user u, respectively. Each triple $(u, i, j) \in \mathcal{P}$ says that user u prefers item i than j. BPR optimization criterion [1] aims to find an arbitrary model class to maximize the following posterior probability over these pairs.

$$\text{BPR-OPT} = -\sum_{(u,i,j) \in \mathcal{P}} \ln \sigma(\hat{x}_{uij}) + \lambda_\Theta(\Theta) \tag{2}$$

where Θ represents the parameter of an arbitrary model class, and λ_Θ are model specific regularization parameters.

Note that extracting pairwise preferences has been widely used in learning to rank tasks [12]. The BPR optimization criterion is actually the cross entropy cost function (logistic loss) over pairs [13].

2.2 Matrix Factorization Based on BPR (BPR-MF)

BPR-MF learns two low-rank matrices $\boldsymbol{W}$ and $\boldsymbol{H}$. For a specific user-item example the score is computed by

$$\hat{x}_{ui} = \sum_{d=1}^{D} W_{ud} H_{id}$$

In order to estimate whether a user prefers one item over another, BPR-MF optimizes the following objective function:

$$\mathcal{O}_1 = -\sum_{u=1}^{n} \sum_{i \in I_u^+} \sum_{j \in I \setminus I_u^+} \ln \sigma(\hat{x}_{uij}) + \alpha(||\boldsymbol{W}||^2 + ||\boldsymbol{H}||^2) \tag{3}$$

where $\hat{x}_{uij} = \hat{x}_{ui} - \hat{x}_{uj}$, σ denotes the logistic function $\sigma(x) = 1/(1 + \exp(-x))$, and α is a regularization parameter for complexity control.

3 User Graph Regularized Pairwise Matrix Factorization

In this section we first describe how to construct user graphs based on different user information, then we propose our UGPMF model based on the computed user graphs.

3.1 User Graph Regularization

Graph regularization has been wildly used in dimensionality reduction [14], clustering [15] and semi-supervised learning [16]. The key assumption of graph regularization is that if two users u and v are similar, the latent features W_u and W_v discovered by pairwise matrix factorization (PMF) procedure are also close to each other. This can be achieved by minimizing the following objective function:

$$\begin{aligned}
\mathcal{O}_2 &= \frac{1}{2}\sum_{u}^{n}\sum_{v}^{n} S_{u,v}||\boldsymbol{W}_{u.} - \boldsymbol{W}_{v.}||^2 \\
&= \frac{1}{2}\sum_{u=1}^{n}\sum_{v=1}^{n}\left[S_{u,v}\sum_{d=1}^{D}(W_{ud} - W_{vd})^2\right] \\
&= \frac{1}{2}\sum_{d=1}^{D}\left[\sum_{u=1}^{n}\sum_{v=1}^{n} S_{u,v}(W_{ud} - W_{vd})^2\right] \\
&= \sum_{d=1}^{D} W_{*d}^T \boldsymbol{L} W_{*d} = \mathrm{tr}(\boldsymbol{W}^T\boldsymbol{L}\boldsymbol{W})
\end{aligned} \tag{4}$$

where $S_{u,v}$ is the similarity between user u and v computed from the available information, $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{S}$ is called the graph Laplacian [17] with $\boldsymbol{D}$ being a diagonal matrix whose diagonal entries are row sums of $\boldsymbol{S}$, $D_{uu} = \sum_v S_{u,v}$, and $\mathrm{tr}(\cdot)$ denotes the trace of a matrix.

The crucial part of graph regularization is the definition of the user graph $\boldsymbol{S}$, which encodes desired information. In the following, we will describe how to construct proper user graph to encode various useful information, such as neighborhood information, demographic information, and social networks.

Neighborhood Information. The neighborhood information of users in the user-item matrix have been widely used in user-oriented memory-based methods [18,19]. The basic assumption of user-oriented memory-based methods is: if two users have similar interests on common items, then they probably have similar interests on the other items. This can be embodied by a user similarity graph defined as follows:

$$S_{u,v} = \begin{cases} sim(X_u, X_v) \text{ if } u \in \mathcal{N}(v) \text{or } v \in \mathcal{N}(u) \\ 0 \qquad\qquad\qquad\qquad \text{otherwise} \end{cases} \tag{5}$$

where $\mathcal{N}(u)$ denotes the K-nearest neighbor of user u, and $sim(X_u, X_v)$ is the similarity between user u and user v computed based on user-item binary matrix.

User's Demographic Information. Demographic information of users is popular in content-based recommendation systems. User demographic information includes age, gender and occupation. We denote by F_u the feature vector which characterizes the demographic information of user u. The above assumption on users similar interests can be captured by a user similarity graph constructed as follows:

$$S_{uv} = sim(F_u, F_v)$$

S_{uv} is the similarity between the feature vectors of user u and user v.

Social Network. Traditional recommender systems ignore social relationships among users. However, the recommendation is sometimes a social activity. For example, we ask our friends for recommendation of movies or cellphones. [11] shows that social friendship can be employed to improve traditional recommender systems. Given a social network, the user graph can be represented by the similarity between two users. It can be defined in terms of the number of shared friends or whether they are friends.

It is important to note that other sources which contain different user information can also be used to construct the user graph, such as user tagging history in a social tagging system, user clickthrough log in web search, etc.

3.2 Integration of PMF and User Graph Regularization

We propose a user graph regularized pariwise matrix factorization (UGPMF) to integrate user information into the pairwise matrix factorization. It combines the above two criteria (3) and (4) and minimizes the following objective function:

$$\begin{aligned} f = &-\sum_{u=1}^{n} \sum_{i \in I_u^+} \sum_{j \in I \setminus I_u^+} \ln \sigma(\hat{x}_{uij}) \\ &+ \frac{\alpha}{2}(||\boldsymbol{W}||^2 + ||\boldsymbol{H}||^2) \\ &+ \frac{\beta}{2} tr(\boldsymbol{W}^T \boldsymbol{L} \boldsymbol{W}) \end{aligned} \tag{6}$$

where β is an additional regularization parameter used to balance the information from the user-item matrix and the user side information.

The above formulation inherits the advantage of pairwise matrix factorization method [1] and has the merits of memory based methods. It can also alleviate the overfitting problem suffered by the matrix factorization procedure with the additional graph regularization.

Due to the huge number of preference pairs (see (1)), it is expensive to update the latent features over all pairs. Hence, like in BPF-MF [1], we adopt a stochastic gradient descent procedure to compute $\boldsymbol{W}$ and $\boldsymbol{H}$. We choose the preference triples (u, i, j) randomly (uniformly distributed) and update the corresponding latent features by the following gradients

$$\frac{\partial f}{\partial W_u} = -\frac{\exp(-\hat{x}_{uij})}{1+\exp(\hat{x}_{uij})}(H_i - H_j) + \alpha W_u \\ + \beta \sum_{v \in \mathcal{N}^+(u)} S_{uv}(W_u - W_v) \\ + \beta \sum_{v \in \mathcal{N}^-(u)} S_{uv}(W_u - W_v) \tag{7}$$

$$\frac{\partial f}{\partial H_i} = -\frac{\exp(-\hat{x}_{uij})}{1+\exp(\hat{x}_{uij})} W_u + \alpha H_i \tag{8}$$

$$\frac{\partial f}{\partial H_j} = \frac{\exp(-\hat{x}_{uij})}{1+\exp(\hat{x}_{uij})} W_u + \alpha H_j \tag{9}$$

where $\mathcal{N}^+(u)$ and $\mathcal{N}^-(u)$ denote the outlink neighbors and inlink neighbors of user u respectively. We use a constant learning rate to update the latent features. The process of estimating the low-rank matrices $\boldsymbol{W}$ and $\boldsymbol{H}$ is described in algorithm 1.

Algorithm 1. Learning procedure of UGPMF

Input: training data $\boldsymbol{X}$, user graph $\boldsymbol{S}$, learning rate η, regularization parameters α and β
Output: $\boldsymbol{W}$ and $\boldsymbol{H}$
1: Compute Laplacian matrix $\boldsymbol{L}$ based on $\boldsymbol{S}$
2: Initialize $\boldsymbol{W}$ and $\boldsymbol{H}$
3: **repeat**
4: draw (u, i, j) uniformly from $U \times I \times I$
5: $\hat{x}_{uij} \leftarrow \hat{x}_{ui} - \hat{x}_{uj}$
6: update W_u, the u-th row of $\boldsymbol{W}$ according to Eq. (7)
7: update H_i, the i-th row of $\boldsymbol{H}$ according to Eq. (8)
8: update H_j, the j-th row of $\boldsymbol{H}$ according to Eq. (9)
9: **until** convergence
10: **return** $\boldsymbol{W}$ and $\boldsymbol{H}$

It is important to note that due to the pairwise comparison in pairwise matrix factorization procedure, the optimization and regularization of PMF are different from other graph regularized element-wise matrix factorization [7, 11]. In [11, 7] they both take a batch algorithm to update the latent features where the gradient of these latent features are accumulated over all pairs. In PMF, the number of preference pairs derived from the original binary dataset is $n\bar{m}(m-\bar{m})$, where $\bar{m}$ is the average positive items for each user. This number is extremely huge. For example the MovieLens dataset (see section 3.1) used in our experiments contains only 100,000 positive examples, but the number of induced pairs is more than 140,000,000. This size is usually prohibitive for batch updating and storage. Hence we resort to stochastic optimization algorithms. In element-wise matrix factorization (MF) approaches, both user and item graphs can be constructed to regularize the MF procedure. However, adding graph regularization on item side in PMF makes the optimization procedure costly; e.g. to update the latent features of each triple (u, i, j), both the in-degree neighbors and out-degree neighbors of item i and j are involved.

4 Experiments

We conduct several experiments to compare the recommendation quality of the proposed method with other state-of-the-art item recommendation methods.

4.1 Datasets

We use two datasets to evaluate our algorithm. The first one, Movielens[1], is a widely used movie recommendation dataset. It contains 100,000 ratings with scale 1-5, which are obtained from 943 users over 1682 movies. To simulate binary responses for item recommendation task, we removed all ratings below 4, and relabeled ratings 4 and 5 as 1. This treatment has also been used in [20]. To evaluate the impact of user demographic information, we extract 30 features (2 features to characterize the user's gender, and 7 features to category the user's age and 21 features to describe the user's occupation) to represent each user.

The second dataset, User-Tag[2] is crawled from a social bookmarking site[3], which has been used in [2]. It contains the tag history of 3000 users on 2000 tags. In total 246,346 posts are recorded.

In our experiments, we choose cosine similarity to construct the user graph based on user-item binary matrix or user demographic information (other similarity measures can also be used).

4.2 Evaluation Criteria

We assess the recommendation performance of each model by comparing the top suggestions of the model to the true positive actions by a user. We consider three measures commonly used to evaluate top-k ranking performance in the IR community. The reason for concentrating on top-k results is that in a recommender system users are usually interested in the top-k results than a sorted order of the entire items. We choose $k = 5$ since many recommender systems recommend a similar number of items for each user.

– Prec@k: The precision at position k of a ranked item list for an given user is defined as

$$\text{Prec@}k = \frac{1}{k} \sum_{i=1}^{k} \mathbf{1}(x_{r_i} = 1)$$

[1] http://www.grouplens.org/

[2] http://www.rongpan.net/

[3] http://del.icio.us/

where $\mathbf{1}(x_{r_i} = 1)$ is 1 if x_{r_i}, the label of item ranked at position i, is positive, and 0 otherwise. Hence, Prec@k considers the positive items and computes the fraction of such items in the top-k elements of the ranked list.

- NDCG@k: The normalized discounted cumulated gain (NDCG) at position k of a ranked item list for an given user is defined by a position discounting function and a grades weighting function. It is normalized by the NDCG value of perfect ranking of these items.

$$\text{NDCG@}k = Z \sum_{i=1}^{k} \frac{2^{x_{r_i}} - 1}{\log(1+i)}$$

where Z is the DCG value of ideal ranking. In our binary item recommendation task, NDCG@k can be viewed as a position discounted and normalized version of Prec@K.

- MAP: MAP denotes the mean of the Average Precision over all test users. Average Precision (AP) is defined as the mean over the precision scores for all positive items. It is given by

$$\text{AP} = \frac{\sum_{i=1}^{m} \text{Prec@i} \cdot \mathbf{1}(x_{r_i} = 1)}{\sum_{i=1}^{m} \mathbf{1}(x_i = 1)}$$

where $\mathbf{1}(x_{r_i} = 1)$ is defined above.

4.3 Comparison Settings

We create random training-test splits of positive user-item entries in the ratio 60%-to-40% respectively. All results reported here are averaged over 10 random rounds. The hyperparameters for all methods are optimized via grid search in the first round and kept constant in the remaining 9 rounds.

In order to show the effectiveness of our recommendation approach, we compare the recommendation results with the following 6 baseline models. The first two models are considered as weak baselines, while the other four are considered as strong baselines.

- PopRank: The first method sorts all the items based on their popularity, so that the top recommended items are the most popular one in terms of the number of times bought by users. This simple measure is supposed to have reasonable performance, as people tend to focus on few popular items.
- AMAN: AMAN stands for All Missing as Negative examples. In AMAN, all non-positive elements are assigned to 0, and the Alternative Least Squares optimization is adopted to optimize the factorization. An ordered list of items can be obtained by sorting the scores calculated by multiplying two low-rank matrices.
- wAMAN (uniform) [2]: It is a weighted version of AMAN where negative elements are treated as zeros but a uniform weight with value less than 1 is additionally imposed. The intuition behind this is that the confidence of

unknown user-item examples being negative is lower than the confidence of positive examples. In particular, we adopt the same weight scheme in [20] and report the best performance over the following weights

$$\delta_k = \frac{n_+}{2^k n_0} \tag{10}$$

where n_+ is the number of positive examples and n_0 is the number of zero-entries in X, with $0 \leq k \leq 5$.
- wAMAN (user) [2]: The weights of zero-entries are proportional to the number of positive items associated with each user. We report the best performance over the following weights $\forall u : w_{ui} = \delta_k \sum_i X_{ui}, k = 0, 1, , 2, 3$, where δ_k is as defined above. The intuition behind this is that if a user has more positive examples, then it is more likely that the other items are less preferred, that is, the missing data for this user is negative with higher probability.
- wAMAN (item) [2]: The weights of zero-entries are linear to the number of users associated with each item. We report the best performance over the weights $\forall i : w_{ui} = \delta_k(m - \sum_u X_{ui}), k = 0, 1, 2, 3$, where δ_k is the same as above. The intuition is that if an item has less positive examples, then the missing data for this item is negative with higher probability.
- BPR-MF: This is the pairwise matrix factorization approach with BPR optimization criterion proposed in [1]. We use the MyMediaLite[4] package in their implementation.

4.4 Results

To make a fair comparison, we report the results of all matrix factorization approaches by setting the number of the latent features $D = 10$. Similar improvement is also observed with other settings of D. The results reported in Table 1 are the top-k rank performance of UGPMF and other baselines on MovieLens dataset. The better results are shown in bold. We observe that our method with user graph constructed by demographic information and neighbor information

Table 1. Performance on the MovieLens dataset

Methods	Prec@5	NDCG@5	MAP
PopRank	0.2375	0.2566	0.1503
AMAN	0.2442	0.2624	0.1440
wAMAN (Uniform)	0.3192	0.3429	0.2279
wAMAN (Item)	0.3417	0.3672	0.2259
wAMAN (User)	0.3556	0.3709	0.2469
BPRMF	0.3590	0.3720	0.2505
UGPMF (User Demographic)	0.3769	0.3906	0.2578
UGPMF (User Neighbor)	**0.3826**	**0.3986**	**0.2633**

[4] `http://www.ismll.uni-hildesheim.de/mymedialite/index.html`

in binary matrix significantly improves the top-k ranking performance of item recommendation, and outperforms the compared baselines consistently. We report the results of UGPMF and compared baselines on User-Tag dataset in Table 2. It is clear that UGPMG achieves better performance in all the measures than the compared baselines. We also conduct a pairwise t-test with a standard 0.05 significance level, which further indicates that all the improvements obtained by UGPMF are significant.

Table 2. Performance on the User-Tag dataset

Methods	Prec@5	NDCG@5	MAP
PopRank	0.2408	0.2497	0.1178
AMAN	0.2409	0.2487	0.1155
wAMAN (Uniform)	0.2552	0.2563	0.1376
wAMAN (Item)	0.2546	0.2640	0.1271
wAMAN (User)	0.2560	0.2645	0.1274
BPRMF	0.2607	0.2682	0.1421
UGPMF (User Neighbor)	**0.2731**	**0.2767**	**0.1489**

We further compare UGPMF with BPR-MF in their performance on users with different number of observed positive examples. The results are shown in Figure 1, from which we see that UGPMF outperforms BPR-MF for all users and the improvement is more significant for uses with only few observed positive examples. This is a very promising property of UGPMF because most users have only a small number of positive examples in real-world situations.

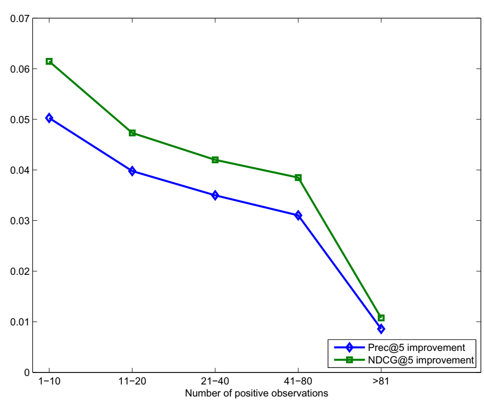

Fig. 1. Top-k ranking performance improvement of UGPMF over that of BPR-MF on different user scales

4.5 Impact of Parameters

UGPMF and BPR-MF have the common parameter D (the number of latent features). UGPMF has two additional parameters, the regularization parameter β and the number of nearest neighbors K in user graphs. We conduct several experiments on the MovieLens dataset to show the impacts of these parameters.

Impact of Parameter β. The main advantage of UGPMF is that it incorporates user information, which helps predict user's preferences. In our model β is used to balance the information coming from the user-item matrix and other user related sources. If $\beta = 0$, we do not use additional user-side information at all and hence our model degenerates to BPR-MF. Table 3 shows the impact of β on Prec@5 and NDCG@5, where we set $D = 10$ and $K = 50$.

As we can see from Table 3, the value of β impacts the performance significantly, which demonstrates that integrating the user-item matrix and user side information greatly improves the top-k recommendation results. It can be also observed that our method achieves the best performance when $\beta \in [0.005, 0.05]$. This relatively wide range shows that the parameter β of our model is easy to tune.

Table 3. The performance of UGPMF vs. parameter β

	0.001	0.005	0.01	0.05	0.1
Prec@5	0.35	0.377	0.38	0.361	0.33
NDCG@5	0.356	0.391	0.398	0.384	0.35
MAP	0.242	0.26	0.263	0.25	0.23

Impact of Parameter K. Similar to memory based collaborative filtering algorithms, the size of neighborhood will affect the performance of our algorithm since it needs to construct K nearest neighbor user graph. We run our algorithm with varying neighborhood size. Figure 2 shows the change of performance (Prec@5 and NDCG@5) when the neighborhood size K increases from 10 to 100 with $\beta = 0.01$ and $D = 10$.

We can see that Prec@5 and NDCG@5 gradually increase as the neighborhood size increases from 10 to 40 since the similarities can be estimated more accurately given more neighbors. However, we also observe that the performance starts to decrease as the neighborhood size exceeds 30, which is due to that many non-similar users introduce incorrect information in regularization as the neighborhood size is too large.

Impact of Parameter D. We report the results of our method and BPF-MF with varying D with $K = 50$ and $\beta = 0.01$. Figure 3 shows that the top-k ranking performance increases as D increases. This agrees with our intuition, that is, the more latent features we have, the more information can be represented by the latent feature vectors. The figure also shows that the improvement in Prec@5 and NDCG@5 becomes smaller as D continues to increase. When D becomes

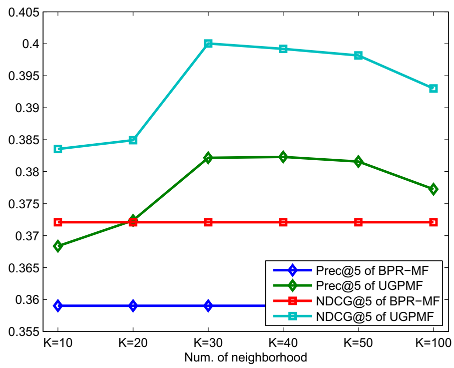

Fig. 2. Top-k ranking performance as a function of size of neighborhood

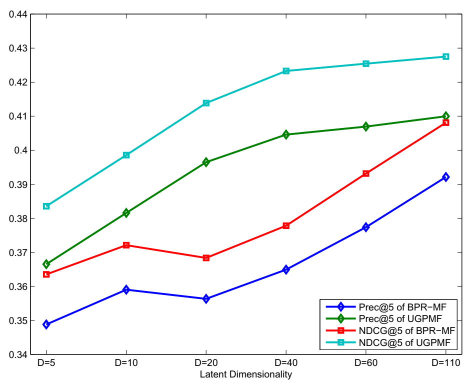

Fig. 3. Top-k ranking performance as a function of latent dimensionality

large enough, there is essentially no significant improvement because the useful information has already been represented well by the existing latent features. It is clear that UGPMF performs better than BPR-MF at all levels.

5 Conclusion

In this paper, we integrate the internal and external user information into the process of pairwise matrix factorization procedure for one class item recommendation task, and propose a unified model UGPMF. One interesting property of the proposed method is that it can alleviate the overfitting problem suffered by most matrix factorization based models [1, 2], since it uses additional user information to regularize the factorization process. Experiments on benchmark datasets demonstrate that the proposed method outperforms many state-of-the-art methods in top-k ranking performance, such as wAMAN [2], BPR-MF [1], etc.

As ongoing work, we are testing our method over more benchmark datasets. We are also investigating more sophisticated models that accommodate valuable user information.

Acknowledgments. We would like to thank all anonymous reviewers for their helpful comments. This work is supported in part by the National Natural Science Foundation of China (NSFC) grants 60970045 and 60833001.

References

1. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence, pp. 452–461. AUAI Press (2009)
2. Pan, R., Zhou, Y., Cao, B., Liu, N., Lukose, R., Scholz, M., Yang, Q.: One-class collaborative filtering. In: IEEE International Conference on Data Mining (ICDM), pp. 502–511 (2008)
3. Pan, R., Scholz, M.: Mind the gaps: weighting the unknown in large-scale one-class collaborative filtering. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 667–676. ACM (2009)
4. Li, Y., Hu, J., Zhai, C., Chen, Y.: Improving one-class collaborative filtering by incorporating rich user information. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management, pp. 959–968. ACM (2010)
5. Rendle, S., Schmidt-Thieme, L.: Pairwise interaction tensor factorization for personalized tag recommendation. In: Proceedings of the Third ACM International Conference on Web Search and Data Mining, pp. 81–90. ACM (2010)
6. Gu, Q., Zhou, J., Ding, C.: Collaborative Filtering: Weighted Nonnegative Matrix Factorization Incorporating User and Item Graphs. In: Proceedings of the SIAM International Conference on Data Mining, pp. 199–210. SIAM (2010)
7. Zhen, Y., Li, W., Yeung, D.: TagiCoFi: tag informed collaborative filtering. In: Proceedings of the Third ACM Conference on Recommender Systems, pp. 69–76. ACM (2009)
8. Ma, H., Yang, H., Lyu, M., King, I.: SoRec: social recommendation using probabilistic matrix factorization. In: Proceeding of the 17th ACM Conference on Information and Knowledge Management, pp. 931–940. ACM (2008)
9. Ma, H., King, I., Lyu, M.: Learning to recommend with social trust ensemble. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 203–210. ACM (2009)
10. Ma, H., Lyu, M., King, I.: Learning to recommend with trust and distrust relationships. In: Proceedings of the Third ACM Conference on Recommender Systems, pp. 189–196. ACM (2009)
11. Ma, H., Zhou, D., Liu, C., Lyu, M., King, I.: Recommender systems with social regularization. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pp. 287–296. ACM (2011)
12. Liu, T.: Learning to rank for information retrieval. Foundations and Trends in Information Retrieval 3(3), 225–331 (2009)
13. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to rank using gradient descent. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 89–96. ACM (2005)

14. Belkin, M., Niyogi, P.: Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation, 1373–1396 (2003)
15. Cai, D., He, X., Han, J., Huang, T.: Graph regularized non-negative matrix factorization for data representation. IEEE Transactions on Pattern Analysis and Machine Intelligence (2010)
16. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. The Journal of Machine Learning Research 7, 2399–2434 (2006)
17. Chung, F.: Spectral graph theory. Regional Conference Series in Mathematics American Mathematical Society, vol. 92, pp. 1–212. American Mathematical Society (1997)
18. Breese, J., Heckerman, D., Kadie, C., et al.: Empirical analysis of predictive algorithms for collaborative filtering. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence, vol. 52, Morgan Kaufmann Publishers, Madison (1998)
19. Sarwar, B., Karypis, G., Konstan, J., Reidl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web, pp. 285–295. ACM (2001)
20. Sindhwani, V., Bucak, S., Hu, J., Mojsilovic, A.: One-class matrix completion with low-density factorizations. In: 2010 IEEE International Conference on Data Mining, pp. 1055–1060. IEEE (2010)