ORTHOGONAL GRAPH-REGULARIZED MATRIX FACTORIZATION AND ITS APPLICATION FOR RECOMMENDATION

Zhenfeng Zhu, Peilu Xin, Shikui Wei, Yao Zhao

Institute of Information Science, Beijing Jiaotong University, Beijing, 100044, China
Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing, 100044, China
zhfzhu@bjtu.edu.cn

ABSTRACT

As one of the most successful approaches for recommendation, matrix factorization based Collaborative Filtering (CF) technique has received considerable attentions over the past years. In this paper, we propose an orthogonal matrix factorization model with graph regularization to preserve the consistency of the local structure both in user and item spaces, respectively. Instead of traditional alternating optimization method, a greedy sequential one is introduced to optimize a pair of coupled factor vector and its corresponding loading vector simultaneously each time, thus the original optimization problem is converted into the well-studied Multivariate Eigen Problem (MEP). Furthermore, multiple pairs of coupled eigen-vectors can be obtained in sequence. To guarantee nonrecurring of repetition of solutions, a novel dual-deflation technique is developed to incorporate into the sequential optimization. Experimental results on MovieLens and Each Movie data sets demonstrate that the proposed method is much more competitive compared with the state of the art matrix factorization based collaborative filtering methods.

***Index Terms*—** Recommendation, Collaborative filtering, Matrix factorization, Multivariate eigenvalue problem, Dual-deflation, Graph model

1. INTRODUCTION

The explosive growth of the information on the web and the emergence of e-commerce have led to urgent demand for personalized recommender systems, providing user-oriented suggestions of fitting users' taste to help them in selecting items from an overwhelming set of choices. Knowing preferences of users for some items, the key issue of building a recommender system is to accurately recommend other items which they will like. To achieve this goal, content-based approach [1] and collaborative filtering method [2] are of two highly influential technologies.

Content-based approaches originate from the field of information retrieval (IR), which rely on profiling the content of the items (such as product information/descriptions, category, title, and author) [1]. The profiles can be used by algorithms to connect user's interests and item's descriptions when generating recommendations. However, it is usually laborious to collect the necessary information about items, and similarly it is often difficult to motivate users to share their personal data to help create the database for the basis of profiling.

On the other hand, the alternative approach, termed collaborative filtering (CF) [2], is usually more feasible by making use of only past user activities (for example, transaction history or user satisfaction expressed in ratings). Collaborative filtering (CF) allows the known preferences of previous users to be propagated to the unknown preferences for other users, thus personalized recommendations or predications for products or services to potential customers can be made. An example of successful putting a collaborative filtering system into use is Amazon.com, where new books are recommended to users based on what they have previously bought as well as their similarity to other users. The underlying assumption of CF is that the active user will prefer those items which the similar user prefers. Since 2006, the well-known Netflix Prize competition has greatly promoted much recent progress in the field of collaborative filtering. Currently, many approaches have been proposed for CF problems, such as memory based methods [3] [4] and model based algorithms [5] [6].

Methods based on matrix factorization (MF) [7], or called latent factor models, are the most representative model-based collaborative filtering methods. As the Netflix Prize competition has demonstrated, Matrix Factorization (MF) based approaches have proven to be efficient for rating-based recommendation systems. In its basic form, matrix factorization characterizes both items and users by vectors of factors inferred from item rating patterns. One of the earliest popular MF models is latent semantic indexing (LSI) [8], which uses singular value decomposition (SVD) to map the content of documents into a lower-dimensional latent semantic space. Some other typical MF based collaborative filtering methods include Regularized Matrix Factorization (RMF), Probabilistic Matrix Factorization (PMF) [9], and Maximum Margin Matrix Factorization (MMMF) [10], et al..

Although based on good mathematical foundation, the above mentioned matrix factorization models fail to take the structure property of data themselves into consideration. To overcome this limitation, an orthogonal graph- regularized matrix factorization (OGRMF) model for recommendation was proposed in this paper to preserve the consistency of local data structure. The difference of our OGRMF model from some other previous graph-regularized matrix factorization models, like [11] [12] [13], lies in an orthogonality constraint is imposed on the loading and factor matrixes, which will be in favor of eliminating the relevance among variables in the latent subspace. Instead of traditional alternative optimization method, a felicitous greedy sequential optimization was introduced, which finally boils down to the well-studied Multivariate Eigen Problem (**MEP**). In addition, a novel dual-deflation was presented to combine with MEP, thus multiple coupled distinct eigen-vectors can be obtained in sequence.

2. GRAPH REGULARIZED MATRIX FACTORIZATION MODEL

Let's begin with introducing some useful notations. Throughout the paper, we use capital letter to denote matrix and lowercase to denote vector. Given a matrix $A = [a_{i,j}] \in \mathbb{R}^{l \times p}$, $a_{i\cdot}$ and $a_{\cdot j}$ represent the i^{th} row vector and the j^{th} column vector of matrix A, respectively. We denote by $\mathrm{Tr}(\cdot)$ the trace of a square matrix, and $\|\cdot\|_F$ represents the Fresenius norm defined as $\|A\|_F = \sqrt{\sum_{i,j} a_{i,j}^2}$.

The essence of matrix factorization models is to map both users and items to a joint latent factor space of lower dimensionality , thus the user-item interactions are modeled as inner products in that space. Given the user-item rating matrix $X = [x_{i,j}] \in \mathbb{R}^{m \times n}$ with m users and n items, $x_{i\cdot} = [x_{i,1}, x_{i,2}, \cdots, x_{i,n}]$ denotes the i^{th} user's ratings on n items, and $x_{\cdot j} = [x_{1,j}, x_{2,j}, \cdots, x_{m,j}]$ the ratings given by m users on the j^{th} item. For the rating matrix X, its factorization refers to seek two mutually coupled matrices $U = [u_{\cdot 1}, u_{\cdot 2}, \cdots, u_{\cdot k}] \in \mathbb{R}^{m \times k}$ and $V = [v_{\cdot 1}, v_{\cdot 2}, \cdots, v_{\cdot k}] \in \mathbb{R}^{n \times k}$ to approximate itself, i.e., $\hat{X} = U \cdot V^T$ and $\hat{X}$ is the approximation of rating matrix X. Typically, k is much smaller than $\min(m, n)$, which means $\hat{X}$ is a low-rank approximation of matrix X.

As one kind of representative matrix factorization models, Regularized Matrix Factorization (RMF) model is given by:

$$\min F_{(U,V)} = \underbrace{\mathcal{L}(X, \hat{X})}_{Factorization\ term} + \alpha \underbrace{\left[\|U\|_F^2 + \|V\|_F^2\right]}_{Regularization\ term} \quad (1)$$

where $\mathcal{L}(X, \hat{X}) = \|X - U \cdot V^T\|_F^2$ is defined as the factorization term, showing the approximation degree of $\hat{X}$ to X, and α is a balancing constant to trade-off the factorization term and regularization term. Note that the regularization term is to restrict the domains of U and V from being over-fitting, so that the resulting model has a good generalization performance.

In this work, instead of forcing the Fresenius norm based regularization as in Eq.1, an orthogonal graph-regularized matrix factorization (OGRMF) model is proposed to preserve the consistency of the local structure in both user and item spaces, respectively. Specifically, the OGRMF is modeled as follows:

$$\begin{aligned} \min F_{(U,V)} = \mathcal{L}(X, \hat{X}) + \alpha \cdot \frac{1}{2} \Bigg[&\sum_{i=1}^{m} \sum_{j=1}^{m} w_{i,j}^u \|u_{i\cdot} - u_{j\cdot}\|_F^2 \\ &+ \sum_{p=1}^{n} \sum_{q=1}^{n} w_{p,q}^v \|v_{\cdot p} - v_{\cdot q}\|_F^2 \Bigg] \quad (2) \\ s.t. \quad U^T \cdot U = I, V^T \cdot V = I & \end{aligned}$$

where $W^u = \left[w_{i,j}^u\right] \in \mathbb{R}^{m \times m}$ and $W^v = \left[w_{p,q}^v\right] \in \mathbb{R}^{n \times n}$ are the well defined edge weight matrices of user graph and item graph, respectively. The aim of imposing orthogonality constraint on loading and factor matrixes is to reduce the relevance among variables in the latent user and item subspaces.

Particularly, the edge weight $w_{i,j}^u$ in the user space is defined by:

$$w_{i,j}^u = \begin{cases} sim(x_{i\cdot}, x_{j\cdot}), & if\ x_{i\cdot} \in \mathcal{N}_{knn}(x_{j\cdot}) \\ & or\ x_{j\cdot} \in \mathcal{N}_{knn}(x_{i\cdot}) \\ 0, & else \end{cases} \quad (3)$$

where $\mathcal{N}_{knn}(x_{i\cdot})$ denotes the set of knn nearest neighbors of user $x_{i\cdot}$, and $sim(x_{i\cdot}, x_{j\cdot})$ reflects the similarity between $x_{i\cdot}$ and $x_{j\cdot}$ in user space (cosine distance is used unless special specification). The similar definition is for $w_{p,q}^v$.

Furthermore, simplifying Eq.2 yields:

$$\begin{aligned} &\min F_{(U,V)} = \mathcal{L}(X, \hat{X}) + \alpha \left[\mathrm{Tr}(U^T L^u \cdot U + V^T L^v V)\right] \\ &s.t. \quad U^T \cdot U = I, V^T \cdot V = I \end{aligned} \quad (4)$$

where $L^u = D^u - W^u$ and $L^v = D^v - W^v$ are two graph Laplacian matrices in User and Item space, respectively; D^u and D^v are two diagonal matrices with $d_i^u = \sum_j w_{i,j}^u$ and $d_p^v = \sum_q w_{p,q}^v$.

3. SOLVING THE OPTIMIZATION FOR OGRMF MODEL

3.1. Greedy sequential optimization

From Eq.4, it is straightforward to notice that the objective function $F_{(U,V)}$ in the OGRMF model is not jointly convex with respect to U and V. But if keeping one of them free while fixing the other, then the objective function is convex for the remained free one. Hence, U and V can be updated alternatively, which is also known as alternative optimization method, and has been widely applied to optimize MF model. But for OGRMF model, the alternative optimization method will not work due to the orthogonality constrains on U and V.

In this work, we propose a more efficient sequential scheme to optimize a pair of coupled factor vector and its corresponding loading vector simultaneously each time, thus the original optimization problem is converted into the well-studied Multivariate Eigen Problem (MEP). Furthermore, multiple pairs of coupled eigen-vectors can be obtained in a greedy sequential way.

Now, for $d = 1, 2, \cdots, k$, suppose that $U_{d-1} = \left[u_{\cdot 1}, u_{\cdot 2}, \cdots, u_{\cdot (d-1)}\right] \in \mathbb{R}^{m\times(d-1)}$ with $U_{d-1}^T \cdot U_{d-1} = I^{[d-1]}$ and $V_{d-1} = \left[v_{\cdot 1}, v_{\cdot 2}, \cdots, v_{\cdot (d-1)}\right] \in \mathbb{R}^{n\times(d-1)}$ with $V_{d-1}^T \cdot V_{d-1} = I^{[d-1]}$ have been obtained at present, where $I^{[d-1]}$ denotes a $d-1 \times d-1$ identity matrix. Substituting $U_d = [U_{d-1} \quad u_{\cdot d}]$ for U and $V_d = [V_{d-1} \quad v_{\cdot d}]$ for V of objective function $F_{(U,V)}$ in Eq. 4 will give rise to:

$$
\begin{aligned}
F_{(U,V)} =& \|X - U_d \cdot V_d^T\|_F^2 + \alpha \cdot \mathrm{Tr}\left[U_d^T \cdot L^u \cdot U_d + V_d^T \cdot L^v \cdot V_d\right] \\
=& \mathrm{Tr}\left[-2v_{\cdot d} \cdot u_{\cdot d}^T \cdot X + \alpha \cdot \left(u_{\cdot d}^T \cdot L^u \cdot u_{\cdot d} + v_{\cdot d}^T \cdot L^v \cdot v_{\cdot d}\right)\right] \\
&+ \sum_{i=1}^{d-1} \mathrm{Tr}\left[-2v_{\cdot i} \cdot u_{\cdot i}^T \cdot X + \alpha \cdot \left(u_{\cdot i}^T \cdot L^u \cdot u_{\cdot i} + v_{\cdot i}^T \cdot L^v \cdot v_{\cdot i}\right)\right] + C_1
\end{aligned} \tag{5}
$$

where $C_1 = \mathrm{Tr}\left[X^T \cdot X + V_d \cdot U_d^T \cdot U_d \cdot V_d^T\right]$ is a constant. To make the illustration more clear, let $f_{(u_{\cdot i}, v_{\cdot i})} = \mathrm{Tr}\left[2v_{\cdot i} \cdot u_{\cdot i}^T \cdot X - \alpha \cdot \left(u_{\cdot i}^T \cdot L^u \cdot u_{\cdot i} + v_{\cdot i}^T \cdot L^v \cdot v_{\cdot i}\right)\right]$, Eq.5 becomes:

$$
F_{(U,V)} = -f_{(u_{\cdot d}, v_{\cdot d})} - \sum_{i=1}^{d-1} f_{(u_{\cdot i}, v_{\cdot i})} + C_1 \tag{6}
$$

Given the obtained $U_{d-1} = \left[u_{\cdot 1}, u_{\cdot 2}, \cdots, u_{\cdot (d-1)}\right] \in \mathbb{R}^{m\times(d-1)}$ and $V_{d-1} = \left[v_{\cdot 1}, v_{\cdot 2}, \cdots, v_{\cdot (d-1)}\right] \in \mathbb{R}^{n\times(d-1)}$, it is explicit that the second term $C_2 = \sum_{i=1}^{d-1} f_{(u_{\cdot i}, v_{\cdot i})}$ of Eq. 6 is also a constant. Thus, minimizing $F_{(U,V)}$ equals to maximizing $f_{(u_{\cdot d}, v_{\cdot d})}$, which leads to the following maximization problem with respect to the coupled factor and loading vectors $u_{\cdot d}$ and $v_{\cdot d}$:

$$
\begin{aligned}
\max f_{(u_{\cdot d}, v_{\cdot d})} =& \mathrm{Tr}\left[2v_{\cdot d} \cdot u_{\cdot d}^T \cdot X\right] \\
&- \alpha \left[u_{\cdot d}^T \cdot L^u \cdot u_{\cdot d} + v_{\cdot d}^T \cdot L^v \cdot v_{\cdot d}\right] \\
s.t. \quad & u_{\cdot d}^T \cdot u_{\cdot d} = 1, v_{\cdot d}^T \cdot v_{\cdot d} = 1
\end{aligned} \tag{7}
$$

For convenience of making difference between $f_{(u_{\cdot d}, v_{\cdot d})}$ and $F_{(U,V)}$, we name hereafter $f_{(u_{\cdot d}, v_{\cdot d})}$ by the individual objective function compared with the ensemble objective function $F_{(U,V)}$. Introducing Lagrange multiplier λ_u^d and λ_v^d, and further differentiating Eq. 7 w.r.t $u_{\cdot d}$ and $v_{\cdot d}$, we have:

$$
\begin{cases}
\frac{\partial f_{(u_{\cdot d}, v_{\cdot d})}}{\partial u_{\cdot d}} = X \cdot v_{\cdot d} - \alpha \cdot L^u \cdot u_{\cdot d} - \lambda_u^d \cdot u_{\cdot d} \\
\frac{\partial f_{(u_{\cdot d}, v_{\cdot d})}}{\partial v_{\cdot d}} = X^T \cdot u_{\cdot d} - \alpha \cdot L^v \cdot v_{\cdot d} - \lambda_v^d \cdot v_{\cdot d}
\end{cases} \tag{8}
$$

Setting Eq.8 to zero and by some mathematical manipulation, the following coupled Eq.9 about $u_{\cdot d}$ and $v_{\cdot d}$ will be obtained:

$$
\begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} u_{\cdot d} \\ v_{\cdot d} \end{bmatrix} = \begin{bmatrix} \lambda_u^d \cdot I^{[m]} & 0 \\ 0 & \lambda_v^d \cdot I^{[n]} \end{bmatrix} \begin{bmatrix} u_{\cdot d} \\ v_{\cdot d} \end{bmatrix} \tag{9}
$$

where $M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$ is a block matrix with $M_{11} = -\alpha \cdot L^u$, $M_{12} = X$, $M_{21} = X^T$, and $M_{22} = -\alpha \cdot L^v$. Noticeably, $u_{\cdot d}$ and $v_{\cdot d}$ can be taken as the d^{th} pair of coupled leading eigen-vectors with their corresponding eigenvalues λ_u^d and λ_v^d.

3.2. Multivariate eigenvalue problem

So far, we can find that obtaining the solutions of $u_{\cdot d}$ and $v_{\cdot d}$ to Eq.9 just equals to the well-studied Multivariate Eigen Problem (**MEP**) which derives its origin from a particular maximum correlation problem. Some distinctive traits of MEP were discussed in [14]. Currently, some algorithms have been developed for solving the MEP mentioned above, including Horst method, Power method and Gauss-Seidel method, et al. Since Gauss-Seidel method takes good convergence performance, it was adopted for solving the MEP with Eq.9. The detail of Gauss-Seidel method is elaborated in Algorithm 1.

Algorithm 1: Gauss-Seidel method for the MEP (**GSMEP**)

```
Input: the block matrix M, the obtained orthogonal matrixes U_{d-1} and V_{d-1}, and the maximum iteration number Iter
Randomly initialize p_1^{(0)} ∈ R^{m×1} and p_2^{(0)} ∈ R^{n×1}
for k = 1, ..., Iter      // Repeat until convergence
   for i = 1, 2      // q_1 and q_2 are calculated in parallel
      q_i^{(k)} := Σ_{j=1}^{i-1} M_{i,j} · p_j^{(k+1)} + Σ_{j=i}^{2} M_{i,j} · p_j^{(k)}
      if i==1
         q_i^{(k)} = q_i^{(k)} − U_{d-1} · U_{d-1}^T · q_i^{(k)}
      else
         q_i^{(k)} = q_i^{(k)} − V_{d-1} · V_{d-1}^T · q_i^{(k)}
      end if
      λ_i^{(k)} := ||q_i^{(k)}||_2
      p_i^{(k+1)} := q_i^{(k)} / λ_i^{(k)}
   end for
end for
λ_u = λ_1^{(Iter)}, λ_v = λ_2^{(Iter)}
u = p_1^{(Iter+1)}, v = p_2^{(Iter+1)}
Output u and v
```

3.3. Dual-deflation

Recall that, for a standard eigenvalue problem of a basis matrix, the multiple distinct eigen-vectors and their corresponding eigen-values are generally obtained using some kinds of iteration methods, e.g. power method. To keep each of eigen-vectors unique against others, the deflation on the above basis matrix should be implemented prior to seeking subsequent eigen-vector and its corresponding eigen-value [15].

As we can see from Algorithm 1, GSMEP can merely obtain a pair of coupled eigen-vectors and their corresponding eigen-values. If we apply directly GSMEP to seek multiple eigen-vectors, say the multiple eigen-vectors u's, the repetitions of them can't be avoided; in other word, some kind of deflation should also be implemented as solving standard eigen-value problem. Considering the fact that Eq.9 is indeed a coupled linear system, it is clear that the traditional deflation methods will not work on it. As a result, a novel dual-deflation scheme is proposed in this paper. Specifically, for the proposed dual-deflation, we have the following Proposition 1.

Proposition 1. *Let* $\{u_{\cdot i}, v_{\cdot i}\}$ *and* $\{\lambda_u^i, \lambda_v^i\}$, $i = 1, ..., d$, *be the multiple pairs of concurrent leading eigen-vectors and corresponding eigen-values for the coupled linear system Eq.9 . If we deflate the block basis matrix M according to Eq.10 in Algorithm 2, the eigen-values* $\{\lambda_u^i, \lambda_v^i\}_{i=1,\cdots,d-1}$ *will be deflated to zeros while keeping* $\{\lambda_u^d, \lambda_v^d\}$ *unchanged, that is, we have :*

$$\begin{bmatrix} M_{11}^d & M_{12}^d \\ M_{21}^d & M_{22}^d \end{bmatrix} \begin{bmatrix} u_{\cdot i} \\ v_{\cdot i} \end{bmatrix} = \begin{cases} \begin{bmatrix} 0 & \\ & 0 \end{bmatrix} \begin{bmatrix} u_{\cdot i} \\ v_{\cdot i} \end{bmatrix} & , i < d \\ \begin{bmatrix} \lambda_u^i \cdot I^{[m]} & 0 \\ 0 & \lambda_v^i \cdot I^{[n]} \end{bmatrix} \begin{bmatrix} u_{\cdot i} \\ v_{\cdot i} \end{bmatrix} & , i = d \end{cases}$$

For the detail proof of **Proposition 1** please refers to Appendix.

Algorithm 2: The dual-deflation method (**DDM**)

1. Input: the block matrix M, the obtained orthogonal matrices $U_{d-1} = [u_{\cdot 1}, ..., u_{\cdot (d-1)}]$ and $V_{d-1} = [v_{\cdot 1}, ..., v_{\cdot (d-1)}]$
2. Update M by the following way:
3. $$\begin{cases} M_{11}^d = M_{11} - U_{d-1} \cdot U_{d-1}^T \cdot M_{11} \cdot U_{d-1} \cdot U_{d-1}^T \\ M_{12}^d = M_{12} - U_{d-1} \cdot U_{d-1}^T \cdot M_{12} \cdot V_{d-1} \cdot V_{d-1}^T \\ M_{21}^d = M_{21} - V_{d-1} \cdot V_{d-1}^T \cdot M_{21} \cdot U_{d-1} \cdot U_{d-1}^T \\ M_{22}^d = M_{22} - V_{d-1} \cdot V_{d-1}^T \cdot M_{22} \cdot V_{d-1} \cdot V_{d-1}^T \end{cases} \quad (10)$$
4. Output $M^d = \begin{bmatrix} M_{11}^d & M_{21}^d \\ M_{21}^d & M_{22}^d \end{bmatrix}$

3.4. Summarization of solving OGRMF model

Following the above discussions, the overall procedure for solving OGRMF model is summarized in **Algorithm 3**.

Algorithm 3: Solving the optimization for OGRMF model)

1. Input: the block matrix M , the available U_{d-1} and V_{d-1}, and the maximum iteration number $Iter$
2. for $d = 1, ...k$
3.   if $d == 1$
4. $[u_{\cdot d}, v_{\cdot d}] =$ ***GSMEP***$(M, U_{d-1}, V_{d-1}, Iter)$
5. $U_d = [u_{\cdot d}]$, $V_d = [v_{\cdot d}]$
6.   else
7. $M^d =$ ***DDM***(M, U_{d-1}, V_{d-1})
8. $[u_{\cdot d}, v_{\cdot d}] =$ ***GSMEP***$(M^d, U_{d-1}, V_{d-1}, Iter)$
9.   end if
10. $U_d = [U_{d-1} \;\; u_{\cdot d}]$
11. $V_d = [V_{d-1} \;\; v_{\cdot d}]$
12. end for
13. $U = U_k, V = V_k$
14. Output U and V.

4. EXPERIMENTAL RESULT AND ANALYSIS

4.1. Data sets and experimental setting

In our experiments, we evaluate the performance of the proposed OGRMF model on two popularly referred movie datasets: MovieLens and EachMovie. MovieLens consists of 100,000 movie ratings (1-5) from 943 users on 1682 movies. EachMovie dataset provides 2.6 million ratings (1-6) with 74,424 users and 1,648 movies. Considering the cold start problem, we discarded items which have less than 5 ratings by users for both data sets and chose a subset of the users for EachMovie.

Table 1 lists the details of the data sets used in the experiment. Particularly, binaryzation into -1 and +1 on all the ratings are preprocessed to show more common choices between agree/disagree or good/bad from users. We randomly select 80% users as the training set, and the rest as the test set. The available records of each test user are split into an observed set and a held-out set. The former is used to predict the held out one. For each user, 10 interested items are held out. To evaluate the top-N recommendation, we use the standard F1 metric with equal weight to both of recall and precision measures.

Table 1: Descriptions of the data sets

Data Sets	*# users*	*# items*	*# ratings*
MovieLens	943	1349	99287
EachMovie	1000	1037	74111

Firstly, we compared the performance of the proposed O-GRMF model with neighbor-based, SVD and RMF methods for recommendation. Table 2 lists the F1 performances obtained using different CF algorithms on MovieLens and EachMovie. Note that the reported results in Table 2 are based on the optimal parameters for each of methods. The results show that the OGRMF model achieves the best performance by F1, which reaches 0.50% improvement over SVD method on MovieLens and 0.34% higher on EachMovie.

Table 2: Performance Comparisons

Dataset	*Neighbor-based*	*SVD*	*RMF*	*OGRMF*
MoviwLens	0.2182	0.2360	0.1928	**0.2410**
EachMovei	0.3698	0.3714	0.3700	**0.3748**

4.2. Impact of key coefficients

For the proposed OGRMF model, there are 3 key coefficients, the trade-off parameter α , the number $userKnn$ of nearest neighbors in user space, and the number $itemKnn$ of nearest neighbors in item space (See Eq.3). Fig. 1 and Fig.2 show the impacts of them on the recommendation performance. Particularly, good results can be obtained when $\alpha = 4$, $userKnn = 15$, and $itemKnn = 10$.

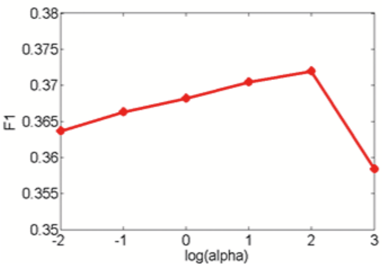

Fig. 1: Trade-off parameter α vs. Recommendation performance

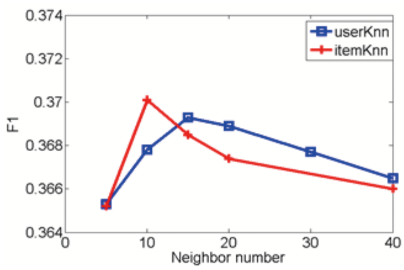

Fig. 2: $userKnn$ and $itemKnn$ vs. recommendation performance

4.3. Convergence analysis

We use the MovieLens dataset to make the convergence analysis of the proposed sequential optimization method. Fig.3 gives the relationship of the optimal individual objective function value $f_{(u_{\cdot d}, v_{\cdot d})}$ (see Eq.7) with the iteration times, from which we can find that the value reaches a stable state by around 7-8 iteration times in average. Furthermore, we also show the variation tendency of the optimal ensemble objective function value $F_{(U,V)}$ in Fig.4, suggesting the dimensionality k of the latent factor space is around 20.

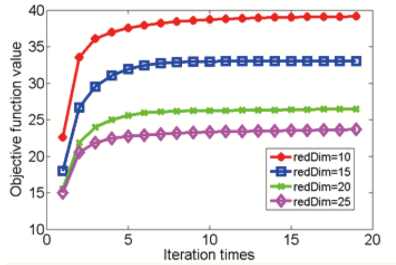

Fig. 3: Convergence Analysis

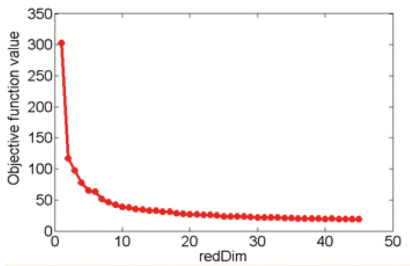

Fig. 4: Variation of $F_{(U,V)}$ with different latent dimensionality $redDim$

5. CONCLUSIONS

To preserve the consistency of local structure in both user and item spaces, an orthogonal graph- regularized matrix factorization (**OGRMF**) model and its application for recommendation was proposed. To solve the optimization problem with OGRMF model, a greedy sequential optimization method was introduced , which finally equals to a coupled linear system . Hence, the well-studied Multivariate Eigen Problem (MEP) can be applied. In addition, a novel dual-deflation was proposed on the coupled linear system, thus multiple distinct eigen-vectors can be obtained in sequence.

6. ACKNOWLEDGEMENTS

This work was supported in part by 973 Program (No.2012CB316400), National Natural Science Foundation of China (No.61025013, No.61172129), PCSIRT(No.IRT201206), Beijing Natural Science Foundation (No. 4112043) and Fundamental Research Funds for the Central Universities (No.2012JBZ012). The authors would also like to thank anonymous reviewers for their constructive and valuable comments.

7. REFERENCES

[1] F. Ricci et al., "Content-based recommender systems: State of the art and trends," Recommender Systems Handbook,Part 1, pp. 73–105. Springer, 2011.

[2] F. Ricci et al., "Advances in collaborative filtering," Recommender Systems Handbook,Part 1, pp. 145–186. Springer, 2011.

[3] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. of WWW Conference*, 2001.

[4] R. Jin, J. Y. Chai,and L. Si, "An automatic weighting scheme for collaborative filtering," in *Proc.of ACM SIGIR*, 2004.

[5] B. Marlin and R. S. Zemel, "The multiple multiplicative factor model for collaborative filtering," in *Proc. of International Conference on Machine Learning*, 2004.

[6] G. Shani, D. Heckerman and R. I. Brafman, "An MDP-based recommender system," *Journal of Machine Learning Research*, vol. 6, pp. 1265–1295, 2005.

[7] Y. Koren, R. M. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Computer*, vol. 42, no. 8, pp. 30–37, 2009.

[8] Michael Pryor, "The effects of singular value decomposition on collaborative filtering," Dartmouth College CS Technical Report, PCS-TR98-338, 1998.

[9] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. of Advances in Neural Information Processing Systems (NIPS)*, 2008.

[10] J. Rennie and N. Srebro, "Fast maximum margin matrix factorization for collaborative prediction," in *Proc. of International Conference on Machine Learning*, 2005.

[11] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Transaction On Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, 2011.

[12] Quanquan Gu, Jie Zhou, and Chris Ding, "Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs," in *Proc. of SIAM DM*, 2010.

[13] Liang Du, Xuan Li, and Yi-Dong Shen, "User graph regularized pairwise matrix factorization for item recommendation," in *Proc. of ADMA*, 2011.

[14] Leihong Zhang and Moody T. Chu, "Computing absolute maximum correlation," *IMA Journal of Numerical Analysis*, vol. 32, pp. 163–184, 2012.

[15] Lester Mackey, "Deflation methods for sparse pca," in *Proc. of Advances in Neural Information Processing Systems (NIPS)*, 2009.

Appendix

Proposition 1. *Let* $\{u_{\cdot i}, v_{\cdot i}\}$ *and* $\{\lambda_u^i, \lambda_v^i\}$, $i = 1, ..., d$ *, be the multiple pairs of concurrent leading eigen-vectors and corresponding eigen-values for the coupled linear system Eq.9 . If we deflate the block basis matrix* M *according to Eq.10 in Algorithm 2, the eigen-values* $\{\lambda_u^i, \lambda_v^i\}_{i=1,\cdots,d-1}$ *will be deflated to zeros while keeping* $\{\lambda_u^d, \lambda_v^d\}$ *unchanged, that is, we have :*

$$\begin{bmatrix} M_{11}^d & M_{12}^d \\ M_{21}^d & M_{22}^d \end{bmatrix}\begin{bmatrix} u_{\cdot i} \\ v_{\cdot i} \end{bmatrix} = \begin{cases} \begin{bmatrix} 0 & \\ & 0 \end{bmatrix}\begin{bmatrix} u_{\cdot i} \\ v_{\cdot i} \end{bmatrix} & , i < d \\ \begin{bmatrix} \lambda_u^i \cdot I^{[m]} & 0 \\ 0 & \lambda_v^i \cdot I^{[n]} \end{bmatrix}\begin{bmatrix} u_{\cdot i} \\ v_{\cdot i} \end{bmatrix} & , i = d \end{cases}$$

Proof. Assume $\{\lambda_u^i, \lambda_v^i\}$ and $\{u_{\cdot i}, v_{\cdot i}\}$, $i = 1, \cdots, d$, are the solutions to the coupled linear system given by Eq.9. Using the deflated basis matrix $M^d = \begin{bmatrix} M_{11}^d & M_{21}^d \\ M_{21}^d & M_{22}^d \end{bmatrix}$ as in Eq.10 to substitute for $M = \begin{bmatrix} M_{11} & M_{21} \\ M_{21} & M_{22} \end{bmatrix}$ in Eq.9, the left hand side of Eq.9 becomes:

$$\begin{aligned} & M_{11}^d \cdot u_{\cdot i} + M_{12}^d \cdot v_{\cdot i} \\ = \; & \left[M_{11} \cdot u_{\cdot i} - U_{d-1} \cdot U_{d-1}^T \cdot M_{11} \cdot U_{d-1} \cdot U_{d-1}^T \cdot u_{\cdot i}\right] + \\ & \left[M_{12} \cdot v_{\cdot i} - U_{d-1} \cdot U_{d-1}^T \cdot M_{12} \cdot V_{d-1} \cdot V_{d-1}^T \cdot v_{\cdot i}\right] \end{aligned} \tag{11a}$$

$$\begin{aligned} & M_{21}^d \cdot u_{\cdot i} + M_{22}^d \cdot v_{\cdot i} \\ = \; & \left[M_{21} \cdot u_{\cdot i} - V_{d-1} \cdot V_{d-1}^T \cdot M_{21} \cdot U_{d-1} \cdot U_{d-1}^T \cdot u_{\cdot i}\right] + \\ & \left[M_{22} \cdot v_{\cdot i} - V_{d-1} \cdot V_{d-1}^T \cdot M_{22} \cdot V_{d-1} \cdot V_{d-1}^T \cdot v_{\cdot i}\right] \end{aligned} \tag{11b}$$

When $i = 1, \cdots, d-1$ and considering $U_{d-1}^T \cdot U_{d-1} = I^{[d-1]}$ and $V_{d-1}^T \cdot V_{d-1} = I^{[d-1]}$, Eq.11a can be rewritten as:

$$\begin{aligned} & M_{11}^d \cdot u_{\cdot i} + M_{12}^d \cdot v_{\cdot i} \\ = \; & \left[M_{11} \cdot u_{\cdot i} - U_{d-1} \cdot U_{d-1}^T \cdot M_{11} \cdot u_{\cdot i}\right] + \\ & \left[M_{12} \cdot v_{\cdot i} - U_{d-1} \cdot U_{d-1}^T \cdot M_{12}^0 \cdot v_{\cdot i}\right] \\ = \; & \left[M_{11} \cdot u_{\cdot i} + M_{12} \cdot v_{\cdot i}\right] - \\ & U_{d-1} \cdot U_{d-1}^T \cdot \left[M_{11} \cdot u_{\cdot i} + M_{12} \cdot v_{\cdot i}\right] \\ = \; & \left[I^{[m]} - U_{d-1} \cdot U_{d-1}^T\right] \cdot \left[M_{11} \cdot u_{\cdot i} + M_{12} \cdot v_{\cdot i}\right] \\ = \; & \left[I^{[m]} - U^{d-1} \cdot U_{d-1}^T\right] \cdot \lambda_u^i \cdot u_{\cdot i} \\ = \; & \lambda_u^i \cdot u_{\cdot i} - \lambda_u^i \cdot U_{d-1} \cdot U_{d-1}^T \cdot u_{\cdot i} \\ = \; & \lambda_u^i \cdot u_{\cdot i} - \lambda_u^i \cdot u_{\cdot i} \\ = \; & 0 \cdot u_{\cdot i} \end{aligned} \tag{12}$$

By similar mathematical manipulation, Eq.11b becomes:

$$M_{21}^d \cdot u_{\cdot i} + M_{22}^d \cdot v_{\cdot i} = \lambda_v^i \cdot v_{\cdot i} - \lambda_v^i \cdot v_{\cdot i} = 0 \cdot v_{\cdot i} \tag{13}$$

But when $i = d$, λ_u^d and λ_v^d will remain to be unchanged since Eq.11 will be:

$$\begin{aligned} M_{11}^d \cdot u_{\cdot i} + M_{12}^d \cdot v_{\cdot i} &= M_{11} \cdot u_{\cdot d} + M_{12} \cdot v_{\cdot d} = \lambda_u^d \cdot u_{\cdot d} \\ M_{21}^d \cdot u_{\cdot i} + M_{22}^d \cdot v_{\cdot i} &= M_{21} \cdot u_{\cdot d} + M_{22} \cdot v_{\cdot d} = \lambda_v^d \cdot v_{\cdot d} \end{aligned} \tag{14}$$

This completes the proof. □