Improving Collaborative Filtering via Hidden Structured Constraint

Qing Zhang, Houfeng Wang*
Key Laboratory of Computational Linguistics (Peking University) Ministry of Education, China
zqicl@pku.edu.cn, wanghf@pku.edu.cn

ABSTRACT

Matrix factorization models, as one of the most powerful Collaborative Filtering approaches, have greatly advanced the recommendation tasks. However, few of them are able to explicitly consider structured constraint for modeling user interests. To solve this problem, we propose a novel matrix factorization model with adaptive graph regularization framework, which can automatically discover latent user communities jointly with learning latent user representations, to enhance the discriminative power for recommendation. Experiments on real-world datasets demonstrate the effectiveness of the proposed method.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Information filtering

General Terms

Algorithms, Design, Experimentation

Keywords

Collaborative Filtering, Structured Constraint, Recommendation

1. INTRODUCTION

Recommender systems play an important role in our daily life, especially in this era of Big Data, which help users to find the information that they are rarely aware but really interested in. With the ability to make recommendations without clear content descriptions of items, Collaborative Filtering (CF) algorithms have been widely applied in various recommender systems. As one of the most powerful CF approaches, Matrix Factorization (MF) models have become popular and achieves the state-of-the-art performance [11]. However, MF approaches have also encountered some problems. Consider music recommendation as example. Different users may have different preferences on music genres, and they can be divided into different communities according to their interests. Therefore, an ideal recommendation system should take this structured property into consideration. To achieve this goal, most existing methods are mainly based on side information, such as social network and item content. In fact, a more fundamental solution is also needed. Recently, directly exploiting structured property for learning matrix factorization has gained much attention [10, 9]. Along this novel direction, we focus on how to incorporate structured prior into matrix factorization models for improving CF performance.

There are two main approaches for explicitly exploiting structured property for recommendation tasks. The first one is from permutation view, based on graph partitioning theories. Typically, [10] first transforms a sparse rating matrix into a bordered block-diagonal structure, and then uses the extracted denser submatrices for rating prediction separately. However, it is a pipeline framework which is independent of specific matrix factorization models. Though it can be used as a black box efficiently, this transforming process may not incorporate task specific information. The second one is from feature selection view, based on the regularization techniques [4]. In this category, [9] considers multiple user interests, and factorizes the rating matrices into latent factor spaces for each with group sparsity regularization. This approach assumes users' interests are determined by different sets of factors, and then could use a subset of the latent factors. The limitation is that the subgroups need to be pre-partitioned. In fact, how to find a reasonable partition is also a problem, which might propagate errors by the inappropriate grouping, into subsequent learning process.

Different from the above approaches, in this paper, we seek to enhance the performance of matrix factorization models through proposing an adaptive graph regularization on latent users, which is learnt automatically from data. The motivation behind the idea is to improve the discriminative power of latent user representations. It is achieved by incorporating structured prior, such that the points within the same communities become more similar, and those within different communities become less similar, in the latent factor space. Therefore, the proposed method can take structured property into consideration jointly with the recommendation task oriented objective, and can benefit from community induced discriminative power with the ability of automatically grouping without pre-partition.

2. PROBLEM FORMULATION

As shown in Definition 1, our goal is to predict the missing values in R, by computing the predicted values $r_{ij} = u_i^T v_j$. Although matrix factorization models also have a nice probabilistic interpretation with Gaussian noise [6], in this paper, we mainly focus on how to exploit structured constraint from algebra view.

DEFINITION 1 (MATRIX FACTORIZATION MODELS). *Given a sparse rating matrix $R = [r_{ij}]$, where the observed r_{ij} denotes*

*Corresponding author

CIKM'15, October 19–23, 2015, Melbourne, Australia.

DOI: http://dx.doi.org/10.1145/2806416.2806623.

the rating of user i on item j, matrix factorization models aim to factorize $R = UV^T$, where U and V are low rank matrices with rows as latent users u_i^T and latent items v_j^T respectively.

To consider structured influence on modeling user preference, the most popular way is to incorporate side information, such as social network [5] and topic structure of item content [1]. This structured information has proven to be useful for building more accurate recommendation models. For the above approaches, the underlying assumption is that additional auxiliary resources with high quality are available in advance besides user rating information. In contrast, we consider a more fundamental case, only using rating matrix without side information. It could be easily extended to the scenarios of the above modeling approaches.

2.1 Laplacian Constraint for Block-diagonality

In this section, we introduce the recently proposed structured constraint [2], i.e., Laplacian constraint, to recommendation tasks, for explicitly capturing the latent user community structures while learning matrix factorization. The motivation could also be roughly explained from overlapping denoising perspective. As a toy example, a middle-school student buys a T-shirt and a pencil. A worker buys a T-shirt and a hammer. The hammer may be recommended to the student due to the overlapping T-shirt. Such noisy overlapping pattern often accounts for the fundamental reason why CF performance is highly sensitive to data in practice, which is seldom addressed explicitly in previous work. To consider this issue, we propose to constrain the overlapping patterns within different latent communities, as hidden structured constraint on global overlapping patterns for CF. In the following, we first show the definition of Laplacian matrix, and then present its connection with the structure of the affinity matrix.

DEFINITION 2 (LAPLACIAN MATIRX). *Consider an affinity matrix $W \in \mathbb{R}^{n\times n}$ of n samples with weights $W(i,i')$. The Laplacian matrix $L_W \in \mathbb{R}^{n\times n}$ is defined as: $L_W = D - W$, where $D = diag(d_1, ..., d_n)$ and $d_n = \Sigma_{i'} W(i,i')$. The normalized version is defined as $L_{W_{sys}} = D^{-\frac{1}{2}} L_W D^{-\frac{1}{2}}$.*

The following well known theorem relates the rank of the Laplacian matrix to the number of blocks in W.

THEOREM 1 ([7]). *Let W be an affinity matrix. The multiplicity k of the eigenvalue 0 of the corresponding Laplacian $L_{W_{sys}}$, equals the number of connected components (blocks) in W.*

Based on the above theorem, we can enforce a general square matrix to be k-block-diagonal to represent different latent communities. For the hidden graph constructed by latent users u_j, we construct an affinity matrix $W(j,j')$ using Gaussian kernel. Then we can define a set of k-block-diagonal matrix (k-BDMS) as the constraint term for optimization in Eq. (2),

$$\begin{aligned}\mathcal{K} =& \{W | rank(L_{W_{sys}}) = n - k, \\ & W(i,i') = w_{ii'} = exp(-\frac{\|u_i - u_{i'}\|_2^2}{\sigma^2})\},\end{aligned} \tag{1}$$

where $\|.\|_2^2$ denotes the $\ell 2$ norm, and σ^2 denotes the deviation. Ideally, the degree of overlapping constraint is expected to be flexibly controlled, which is considered in the proposed framework.

2.2 Adaptive Graph Regularization Framework

To incorporate the structured constraint, the straightforward applying Laplacian constraint on reconstructed rating matrix is not suitable, because this may violate the Nearly Isometric Property (NIP) as mentioned in [11] to undermine the performance. In addition, we expect to flexibly control the overlapping constraint. Therefore, we propose an adaptive hidden graph regularization framework, which is similar to [5], but our graph used for regularization is learnt automatically with block-diagonal prior, rather than a pre-defined one based on side information. We can automatically discover latent user communities jointly with learning latent user representations, to enhance the discriminative power. To achieve the goal, we have the following optimization objective:

$$\begin{aligned}\min_{U,V,W,S} & \frac{1}{2}\sum_{i,j} c_{ij}(r_{ij} - u_i v_j^T)^2 + \frac{\lambda_W}{2}\sum_{i,i'} c_{w_{ii'}}(w_{ii'} - u_i s_{i'}^T)^2 \\ & + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2 + \frac{\lambda_S}{2}\|S\|_F^2 \\ & s.t.\ W \in \mathcal{K}\end{aligned} \tag{2}$$

where $\|\cdot\|_F$ is the Frobenius norm. In Eq.(2), the first term ensures that latent users U and items V can well approximate the observed ratings, where c_{ij} is a confidence parameter [8] for rating r_{ij}, $a > b$. If c_{ij} is large, we trust r_{ij} more, $c_{ij} = \begin{cases} a, r_{ij} = 1 \\ b, r_{ij} = 0. \end{cases}$ The second term in Eq. (2) is the graph regularization term, where $c_{w_{ii'}}$ is the confidence parameter for modeling $w_{ii'} = exp(-\frac{\|u_i - u_{i'}\|_2^2}{\sigma^2}) \in W$ that measures the similarity between latent users. The similarity weight $w_{ii'}$ in hidden graph W is automatically learnt from data with Laplacian constraint, i.e., $rank(L_{W_{sys}}) = n - k$ for pursuing k-block-diagonal structure as discussed in Section 2.1. This constraint $W \in \mathcal{K}$ in Eq. (1) is imposed on the Laplacian matrix of latent users through Gaussian kernel. Thus it can be directly optimized jointly with latent user representations, and then can improve their discriminative power. The remaining terms are used to avoid over-fitting, the parameters λ_U, λ_v and λ_S control the strength of each regularization term respectively. It is noted that the degree of overlapping constraint can not only be controlled by the parameter λ_W but also the number of latent groups flexibly.

2.3 Optimization with Structured Constraint

We employ an alternating optimization method to solve the problem, by updating U, V, S, W iteratively and alternatingly.

2.3.1 Learning U, V, S

The following update rules are obtained by setting the derivative of L with respect to u_i, v_j , and s_k to zero.

$$u_i = (\lambda_U I + V^{\mathrm{T}} D_{c_i} V + S^{\mathrm{T}} D_{w_i} S)^{-1} \cdot (V^{\mathrm{T}} D_{c_i} R_i + S^{\mathrm{T}} D_{w_i} W_i) \tag{3}$$

$$v_j = (\lambda_V I + U^{\mathrm{T}} D_{c_j} U)^{-1} \cdot (U^{\mathrm{T}} D_{c_j} R_j) \tag{4}$$

$$s_{i'} = (\lambda_S I + U^{\mathrm{T}} D_{w_{i'}} U)^{-1} \cdot (U^{\mathrm{T}} D_{w_{i'}} W_{i'}) \tag{5}$$

where I is an identity matrix of the same dimension as that of latent space. S is a matrix with rows as social factor-specific latent feature vectors for the learnt hidden graph W. R_i is a column vector with rating values $[r_{i1}, \ldots, r_{iJ}]^{\mathrm{T}}$. Similarly, $R_j = [r_{1j}, \ldots, r_{I_N j}]^{\mathrm{T}}$. For the hidden graph W, $W_i = [w_{i1}, \ldots, w_{iI_N}]^{\mathrm{T}}$ and $W_{i'} = [w_{1i'}, \ldots, w_{I_N i'}]^{\mathrm{T}}$, I_N and J are the total number of users and items respectively. D_{c_i} is a diagonal matrix with values $diag(c_{i1}, \ldots, c_{iJ})$ and $D_{c_j} = diag(c_{1j}, \ldots, c_{I_N j})$. D_{w_i}, and $D_{w_{i'}}$ are similarly defined with diagonal elements $c_{w_{i,\cdot}}$ and $c_{w_{\cdot,i'}}$ respectively. In addition, $c_{w_{ii'}}$ is the confidence parameters for $w_{ii'}$. The

high confidence value a is set to the learnt user relations within the same communities, and the low confidence value b is set to those within different communities, where $a > b > 0$.

2.3.2 Learning Hidden Graph Matrix, W

After learning U, V, S, we can obtain the hidden graph W_0 using Gaussian kernel as shown in Eq. (1). However, this variable matrix W_0 may move out of the hidden structured constraint set, and no longer guarantees a k-block-diagonal structure. Thus, we need to project it back to the k-BDMS constraint set. The projection essentially finds a matrix W in $\mathcal{K}$ as defined in Eq. (1) which is closest to W_0 in terms of the Euclidean distance. The involved optimization problem can be explicitly written as follows via Augmented Lagrangian Multiplier (ALM) method:

$$\min_{W,\tilde{Z}} \frac{1}{2}\|W - W_0\|_F^2 - \langle J, L_{W_{sys}} \rangle + \frac{\beta}{2}\|\tilde{Z} - L_{W_{sys}}\|_F^2 \quad s.t.\ rank(\tilde{Z}) = n - k \tag{6}$$

where J is the Lagrangian multiplier and β is an increasing weight parameter for the term of enforcing the auxiliary variable $\tilde{Z} = L_{W_{sys}}$. Thus the two constraints are decoupled and we can alternatively optimize W and $\tilde{Z}$. The solution to Eq. (6) is similar to that in [2], and we omit it here due to limited space. The difference is that our W is constructed from latent user representations using Gaussian kernel, with normalized Laplacian constraint.

Algorithm 1: Optimization Algorithm for HGMF

Input: Number of latent factors k, a sparse rating matrix R, standard deviation σ of Gaussian kernel, parameters λ_U, λ_V, λ_S and λ_W, confidence parameters a, b.
Output: U,V,S,W.

```
while t < T do
    Update each u_i^t with Eq. (3);
    Update each v_j^t with Eq. (4);
    Update each s_{i'}^t with Eq. (5);
    Update W^t by solving the problem in (6);
    t = t + 1;
end while
Return U, V, S, W;
```

2.3.3 Implementation Details

The naive calculation of $V^{\mathrm{T}} D_{c_i} V$, $S^{\mathrm{T}} D_{w_i} S$, $U^{\mathrm{T}} D_{c_j} U$ and $U^{\mathrm{T}} D_{w_{i'}} U$, requires time $O(K^2 N_{user})$ or $O(K^2 N_{item})$, where K is dimension of the latent space. Inspired by [3], e.g., we can rewrite $V^{\mathrm{T}} D_{c_i} V = V^{\mathrm{T}}(D_{c_i} - bI)V + bV^{\mathrm{T}}V$, where I is the corresponding identity matrix. Then we can pre-compute $bV^{\mathrm{T}}V$. Since $D_{c_i} - bI$ has only $\overline{N}_{user}$ non-zeros where $\overline{N}_{user} \ll N_{user}$, this sparsity can significantly speed up computation [5].

3. EXPERIMENTS

The experiments were conducted on two public real-world datasets[1]: LastFM and Delicious,with binary ratings, as shown in Table 1.

3.1 Experimental Setups

3.1.1 Baselines and Settings

- **PMF**: This method, described in [6], is a well-known matrix factorization method for CF, only using interactive rating information.

[1]Data available at http://grouplens.org/datasets/hetrec-2011/

Dataset	#User	#Item	Sparsity	#Ratings	#Avg. R
LastFM	1892	18745	0.28%	92834	49
Delicious	1867	69223	0.08%	104799	56

Table 1: Datasets Statistics. R denotes #rated per user.

Dataset	PMF	PMF-LMF	GSMF	GSMF-K	HGMF
LastFM	0.5761	0.5646	0.5923	0.5701	**0.5512**
Delicious	0.9084	0.8623	1.1321	0.8812	**0.8313**

Table 2: Model Comparison for RMSE.

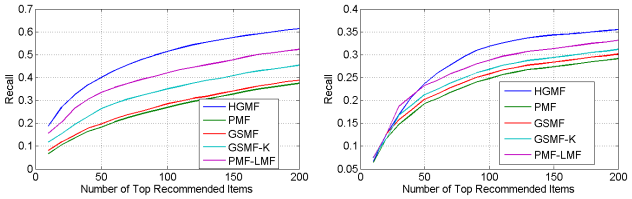
- **PMF-LMF**: This method first uses LMF described in [10] to acquire submatrices as groups and then for each one performs PMF.
- **GSMF**: This method, described in [9], factorizes the rating matrices for multiple interests with group sparsity regularization. (without pre-partition)
- **GSMF-K**: The same GSMF method, described in [9]. GSMF-K denotes using k-means to partition the item set into k groups for GSMF.
- **HGMF**: Our model is described in this paper with Augmented Lagrangian Multiplier for learning hidden graph.

For easier comparison with previous proposed methods, we use Root Mean Square Error (RMSE) to measure the prediction accuracy in this work. For N rating-prediction pairs, i.e., the predicted r_i and true $\widehat{r}_i$, $RMSE = \sqrt{\sum_{i=1}^{N} \frac{(r_i - \widehat{r}_i)^2}{N}}$. We use Recall to measure top-k performance, $Recall@k = \frac{\#relevant\ in\ top{-}k\ list}{\#total\ relevant}$.

The experimental setting is as follows. Following [5, 9], we randomly select 90% of the data for training and the rest for testing. The random selection was carried out 5 times independently, and we report the average results. For all datasets, we set kernel $\sigma = 50$; $a = 1$, $b = 0.01$ in PMF and HGMF; $\alpha = 1$, $\beta = 0.1$, $\lambda = 0.01$ and $K_{pre} = 200$ in GSMF-K. The max iteration number is 200, and the number of latent factors $k = 200$. For two different datasets, **LastFM**: $\lambda_V = 0.1$ in PMF. $\widehat{\rho} = 0.005$ in LMF. $\lambda_U = 0.1$, $\lambda_V = 100$ in GSMF and HGMF. **Delicious**: $\lambda_U = 0.01$, $\lambda_V = 0.1$ in PMF, GSMF and HGMF. $\widehat{\rho} = 0.0016$ in LMF. The remaining parameters vary for our evaluation.

3.2 Results and Analysis

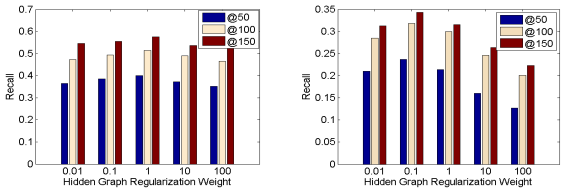
3.2.1 Performance Comparison



(a) LastFM (b) Delicious

Figure 1: Model Comparison for Recall.

Table 2 shows that, on the extremely sparse dataset, Delicious, our method can achieve considerable improvement compared PMF and other structured aware modeling approaches. This could be explained that in extremely sparse case, the common ratings for users are not sufficiently overlap such that the collaborative modeling fails due to the lack of co-occurrence pattern information. Therefore, considering structured property is important for modeling users, which can make the points within the same communi-



(a) LastFM (b) Delicious

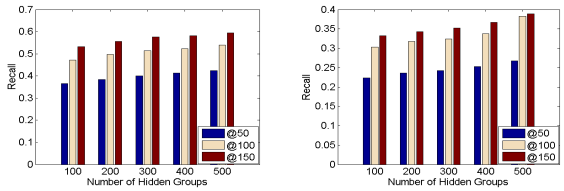
Figure 2: Impact of Hidden Graph Weight.

λ_W	**0.01**	**0.1**	**1**	**10**	**100**
LastFM	0.5709	0.5612	**0.5565**	0.5712	0.7213
Delicious	0.8574	**0.8474**	0.8974	0.9415	1.1306

Table 3: Impact of Hidden Graph Weight (k=200).

ties more similar to distinguish dissimilar patterns. Although PMF-LMF, GSMF and GSMF-K take the structured property into consideration from different perspectives, the results of these methods still perform worse than our method. This could be explained that PMF-LMF is a pipeline approach which may not well extract subdenser matrices without task objective, especially when original data is not relatively dense enough. In contrast, our HGMF explores overlapping constraint in the more denser latent subspace. GSMF is limited to the pre-partitioned groups without adaptive grouping for task objective. Moreover, Figure 1 further validates the effectiveness of the proposed method in terms of Top-N performance.

3.2.2 *Impact of Hidden Graph Weight* λ_W



(a) LastFM (b) Delicious

Figure 3: Impact of the Number of Hidden Groups.

The parameter λ_W is introduced to control the contribution from hidden graph regularization for CF. Therefore, we investigate how our algorithm HGMF is influenced by the graph weight parameter. We vary the value of λ_W as $\{0.01, 0.1\ , 1, 10, 100\}$. Table 3 and Figure 2 show that in general, with the increase of λ_W, the performance in different datasets shows similar patterns: first increasing, reaching its highest value and then decreasing, obviously for Delicious. In particular, for the extremely sparse dataset, Delicious, lower value suggests good performance. For other more denser datasets, relatively higher value will improve the evaluation results. It could be explained that in the extremely sparse case, each latent user may rely on more latent neighbours. Thus, if we restrict its group membership tightly, the performance will decrease. For denser case, the rating matrix may introduces more noisy patterns. Thus, the larger regularization will separate the latent user from the noisy relevant users in the different latent communities.

3.2.3 *Impact of the Number of Hidden Groups*

Different from traditional regularization approaches, our method can naturally bridge different latent spaces with different dimensions, within the proposed adaptive hidden graph regularization framework. Therefore, we further investigate how our algorithm

#hidden group$k$	**100**	**200**	**300**	**400**	**500**
LastFM	0.5665	0.5565	0.5413	0.5395	0.5335
Delicious	0.8522	0.8474	0.8319	0.8205	0.8234

Table 4: Impact of the Number of Hidden Groups.

HGMF is influenced by the number of hidden groups (k in Eq. (6)), i.e., the approximate dimension of the learnt hidden graph space. It is noted that the dimension of latent factor space is 200. We vary the value of the number of hidden groups as $\{100, 200, 300, 400, 500\}$. Table 4 and Figure 3 show that fixing the optimal parameter λ_W, setting larger number of hidden groups will achieve better results in terms of RMSE and Recall. Empirically, we find this parameter is dependent on the parameter λ_W. Thus we can first fix this parameter and then change λ_W, or vise versa. Both cases can achieve the similar results. Intuitively, fixing λ_W, slightly enlarging the number of hidden groups will narrow the range of visible neighbours. It will make the learnt latent user representation more discriminative.

4. CONCLUSION

In this paper, we proposed a novel solution to explicitly exploiting structured property for collaborative filtering, without relying on side information. It could be seen as a principled way of constraining the potentially noisy overlapping patterns locally, while learning the global matrix factorization. This method benefits from the community induced discriminative power for recommendation, jointly with the ability of automatically grouping without prepartition. Experiments on two real-world datasets exhibit the promising performance, compared with some state-of-the-art methods.

5. ACKNOWLEDGMENTS

Our work is supported by National High Technology Research and Development Program of China (863 Program) (No.2015AA015402), National Natural Science Foundation (No.61433015 and No.61370117) and Major National Social Science Fund of China (No.12&ZD227).

6. REFERENCES

[1] D. Agarwal and B. Chen. flda: matrix factorization through latent dirichlet allocation. In *Proceedings of the WSDM*, pages 91–100, 2010.

[2] J. Feng, Z. Lin, H. Xu, and S. Yan. Robust subspace segmentation with block-diagonal prior. In *CVPR*, pages 3818–3825, 2014.

[3] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM*, pages 263–272, 2008.

[4] F. Nie, H. Huang, X. Cai, and C. H. Q. Ding. Efficient and robust feature selection via joint l21-norms minimization. In *Proceedings of NIPS*, pages 1813–1821, 2010.

[5] S. Purushotham and Y. Liu. Collaborative topic regression with social matrix factorization for recommendation systems. In *Proceedings of ICML*, 2012.

[6] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Proceedings of NIPS*, pages 1257–1264, 2007.

[7] U. von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007.

[8] C. Wang and D. M. Blei. Collaborative topic modeling for recommending scientific articles. In *Proceedings of KDD*, pages 448–456, 2011.

[9] T. Yuan, J. Cheng, X. Zhang, S. Qiu, and H. Lu. Recommendation by mining multiple user behaviors with group sparsity. In *Proceedings of AAAI*, pages 222–228, 2014.

[10] Y. Zhang, M. Zhang, Y. Liu, S. Ma, and S. Feng. Localized matrix factorization for recommendation based on matrix block diagonal forms. In *Proceedings of WWW*, pages 1511–1520, 2013.

[11] Y. Zhang, M. Zhang, Y. Zhang, Y. Liu, and S. Ma. Understanding the sparsity: Augmented matrix factorization with sampled constraints on unobservables. In *Proceedings of CIKM*, pages 1189–1198, 2014.