

Exploring Social Activeness and Dynamic Interest in Community-based Recommender System

Bin Yin
Information Science and
Technology Division
Graduate School at
Shenzhen, Tsinghua University
Shenzhen, China
yinb11@mails.tsing-
hua.edu.cn

Yujiu Yang
Information Science and
Technology Division
Graduate School at
Shenzhen, Tsinghua University
Shenzhen, China
yang.yujiu@sz.tsing-
hua.edu.cn

Wenhuang Liu
Information Science and
Technology Division
Graduate School at
Shenzhen, Tsinghua University
Shenzhen, China
liuwh@sz.tsing-
hua.edu.cn

ABSTRACT

Community-based recommender systems have attracted much research attention. Forming communities allows us to reduce data sparsity and focus on discovering the latent characteristics of communities instead of individuals. Previous work focused on how to detect the community using various algorithms. However, they failed to consider users' social attributes, such as social activeness and dynamic interest, which have strong correlations to users' preference and choice. Intuitively, people have different social activeness in a social network. Ratings from users with high activeness are more likely to be trustworthy. Temporal dynamic of interest is also significant to user's preference. In this paper, we propose a novel community-based framework. We first employ PLSA-based model incorporating social activeness and dynamic interest to discover communities. Then the state-of-the-art matrix factorization method is applied on each of the communities. The experiment results on two real world datasets validate the effectiveness of our method for improving recommendation performance.

Categories and Subject Descriptors

H.3.3 [Information Search and Retrieval]: Information filtering; J.4 [Computer Applications]: Social and Behavior Sciences

General Terms

Algorithms, Experimentation

Keywords

Recommender System, Community Detection, Social Activeness, Temporal Dynamic

1. INTRODUCTION

With the rapid growth of information on the Internet, it becomes difficult for users to find relevant information. Recommender systems have become a promising tool to handle information overload in many online application scenarios. Most of recommender systems are based on collaborative filtering (CF), which predict a user's interest in an item by mining patterns from the past rating information of other similar users or items. One remarkable advantage of CF is that CF only relies on users' historical behaviors without collecting content information from all users and items.

Although collaborative filtering has become the standard method for the recommendation problem, it suffers from a few weaknesses that limit its performance, such as data sparsity and cold start problem. Significant advancements have been made in recent years to improve the accuracy of such personalized recommender systems and handle these issues.

One of the most extensively used techniques in these recommender systems is community detection [10]. Community detection in social networks helps us to understand users' collective behavioral patterns better [2]. Community detection techniques in recommender system aim to find subgroups among users that the preference of users within the group is more similar than the outside one. However, previous work focused on either link analysis or content analysis. It failed to consider individuals' attributes such as social activeness and temporal dynamic of interest which are quite important to community detection and recommendation task.

We introduce social activeness to describe the contribution of a user to a community. Intuitively, people's activeness differ from each other in a social network. People who have more friends and higher reputation seem to give more contribution to the community than the less ones. In terms of interest, people's interest is not static, but dynamic. However, to the best of our knowledge, no existing work combined these two critical properties with community-based recommendation algorithms. In this paper, we propose a novel framework, CBMF, which aggregates social activeness and dynamic interest features into community-based recommendation. The main contributions of our framework are summarized as follows:

(1) We propose a two-phase approach to get Top-N recommendation for each user. We first employ a modified

PLSA model to discover communities before applying matrix factorization on each community. Based on the results obtained after matrix factorization, we can get a Top-N list for each user.

(2) We incorporate social activeness and dynamic interest features into community detection phase. These two features have strong correlation to users' preference and can improve recommendation accuracy effectively.

(3) Forming communities for recommendation on a social network reduces the sparsity in the matrix which is one of the most serious limitations of traditional matrix factorization approaches.

(4) Extensive experiments on two real world datasets Epinions and Ciao demonstrate that the proposed approach outperforms other state-of-the-art recommendation algorithms.

The rest of this paper is organized as follows: In Section 2, we discuss related work. In Section 3 and 4, we present our framework in detail. Section 5 gives the results of our experimental study and we conclude in Section 6.

2. RELATED WORK

As we know, community detection itself is a very important research task with great challenges in social network analysis [13]. The work in [10] used communities, extracted from different dimensions of social networks, to capture the similarities of these different dimensions and accordingly help recommender systems to improve the accuracy and mitigate the cold start problem. The work in [12] explored how user-item subgroups can improve the performance of recommender systems. They design a multi-class co-clustering approach that utilizes the explicit ratings to group the users and items. The work in [15] proposed a LDA-based method to uncover community before using matrix factorization in each community. The work in [14] proposed a PLSA-based topic model to detect latent topic and then generated communities.

[7] employed global perspective to reveal the importance of users' activeness in the whole social network. The work in [11] proposed a novel framework LOCABAL taking advantage of both local and global social context for recommendation. There are many algorithms to calculate the social activeness of users in social networks according to their connections [9] and we adopt one of the most popular algorithms PageRank [9] to compute the user activeness scores in our work.

Some work is carried out to investigate the time context in recommendations. User purchase time and item launch time were considered in [5] in order to improve recommendation accuracy. The work which is most directly related to ours is [1], the authors used an exponential time decay function to compute time weights for different items according to each user and each cluster of items. Yehuda [4] presented a methodology and specific technique for modeling time drifting of user preferences based on matrix factorization.

3. COMMUNITY DETECTION AGGREGATING SOCIAL ACTIVENESS AND INTEREST DYNAMIC

Our proposed framework comprises two main phases. The first phase utilizes a modified PLSA method to determine the topic distribution of the items and users. Note that

the concept of community in this paper is slightly different from the traditional one. Communities are groups of users and items which have similar interest and properties. Communities are formed by grouping users and items whose probability of a given topic is bigger than others. The second phase applies matrix factorization on each community to generate a list of items. We then combine these candidate lists to obtain the top-N items for a target user.

We first introduce notations used in this paper. Let $\mathcal{U} = \{u_1, u_2, \dots, u_M\}$ and $\mathcal{V} = \{v_1, v_2, \dots, v_N\}$ be the sets of users and items respectively, where M is the number of users and N is the number of items. Assume that $R \in \mathbb{R}^{M \times N}$ is the user-item rating matrix. If u_i gives a rating to v_j , R_{ij} is the rating score, otherwise we employ 0 to represent the unknown rating from u_i to v_j , i.e. $R_{ij} = 0$. Let $T \in \mathbb{R}^{M \times M}$ denote user-user social relations where $T_{ij} = 1$ if u_i trust u_j and 0 otherwise. Let $C = \{c_1, c_2, \dots, c_K\}$ be the set of communities and K is the number of communities.

3.1 A PLSA-based Method to Detect Community

In our framework, we employ probabilistic Latent Semantic Analysis (PLSA) model [3] to discover community. The advantages of PLSA in community detection are (1) It is more practicable and effective than traditional community detection methods. (2) PLSA model can mine the latent interest feature of users and item, so that it can get more innate character information from limited dataset. (3) The latent topic of PLSA can be well-aligned with the point of interest in the real word. So it is more interpretable and representative. In our work, the original PLSA model is adjusted to the user-item matrix by viewing user as word, item as document:

$$L(U) = \log p(U|\Theta) = \sum_{i=1}^M \sum_{j=1}^N R_{ij} \cdot \log \left[\sum_{k=1}^K p(u_i|z_k) p(z_k|v_j) \right] \quad (1)$$

In our work, EM algorithm is employed to estimate the parameter Θ . Afterwards, the probabilistic distribution of $p(u_i|z_k)$, $p(z_k|v_j)$ are obtained, see Eq.(2). Note that $p(z_k|v_j)$ is very helpful. It denotes the probabilistic distribution of an item v_j on a community z_k , which we can cluster items based on it. A distribution matrix $P \in \mathbb{R}^{C \times N}$ is obtained, where C is the number of communities and N is the number of items. For each column of P , only the top-k biggest elements are retained and each column sums to one by normalizing. Therefore if $p(v_j|z_k)$ is in top k biggest list, we allocate v_j into the k -th community. In this way, all the items are clustered into proper communities according to their latent property, i.e. community detection for items.

$$P(z_k|u_i, v_j) = \frac{p(u_i|z_k)p(z_k|v_j)}{\sum_{l=1}^K p(u_i|z_l)p(z_l|v_j)} \quad (2)$$

Although $p(u_i|z_k)$ can describe the probabilistic distribution of a user u_i on a community z_k to some extent, we employ a new measurement POI-score which aggregates users' activeness and interest temporal dynamic to describe the extent of how a user interested in a community.

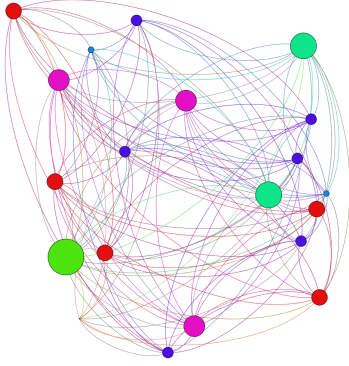


Figure 1: A Toy Example of Different Activeness in A Social Network

3.2 Modeling Social Activeness in Community Detection

In this paper, social activeness is defined to measure a person's degrees (including in-degree and out-degree) in the social network. As we know, people's activeness differs from each other and people having a lot of friends and reputation deserve to have high social activeness score. Fig.1 gives a toy example of different social activeness in a social network. The bigger circle indicates higher activeness or reputation in the social networks. Moreover, some researchers found that ratings from users with high activeness are more likely to be trusted on the Internet [7]. For example, we seem to like to review or repost the information coming from big V users (verified weibo users who have more than 500,000 followers) in Weibo. Therefore our proposed method exploits users' social activeness to weight the importance of user ratings according to their social activeness scores as:

$$W_a(i) = \frac{1}{1 + \log(r_i)} \quad (3)$$

Where r_i is the rank of users obtained by PageRank algorithm:

$$Pagerank(p_i) = \frac{1-d}{N} + d \sum_{p_j \in M(p_i)} \frac{PageRank(p_j)}{L(p_j)} \quad (4)$$

Where M is the number of all users, $M(p_i)$ is the set of users that user u_i trust. $L(p_j)$ is the set of users who trust user u_i . People who have high reputation will have a larger value of W_a .

3.3 Modeling Users' Interest Dynamic in Community Detection

Intuitively, people's interest or preference is drifting over time. On one hand, the popularity of product is constantly changing as new selection emerging. On the other hand, customers' preferences are evolving, leading them to redefine their taste. For example, a man might like rock music when he was young. A few years later, he may turn to like classical music as he grows older. Hence, temporal dynamic should be a crucial feature when designing recommender systems or general customer preference models. In our work, we employ a time decay function to model people's interest temporal dynamic as follows:

$$W_t = \frac{1}{1 + \beta|t_{ij} - t_i^{last}|} \quad (5)$$

Where t_{ij} is the timestamp when user i rated item j . t_i^{last} is the last time when user i rated. A larger value W_t indicates the user's rating is closed to his nearest interest or preference, while a smaller value may indicate ratings long before.

3.4 A Synthesized Model Combining Social Activeness and Interest Dynamic

In previous sections, we have modeled users' social activeness and interest temporal dynamic respectively. In this section, we will aggregate the two crucial factors into a unified community detection model and allocate all users to the proper communities according to their interest. To this end, we first employ a linear model combines the weight of users' social activeness and interest dynamic as follows:

$$W = \alpha W_t + (1 - \alpha) W_a \quad (6)$$

Where W_t is the weight related to users' interest temporal dynamic and W_a is the weight value related to users' social activeness. α is the parameter that controls the impact of temporal dynamic and social activeness.

Afterwards, a novel indicator, POI score, is adopted to describe the extent of how a user interested in a community. The higher POI score implies strong interest in the community. Notice that rating scores by users are valuable feedbacks which reflect users' preference explicitly. If a user gives a rating of 4 or 5, it most likely means he/she likes the film. But a rating of 1 or 2 can be just to the contrary. Hence rating scores should be taken the into consideration. With all the concerns aforementioned, one's POI score can be computed as follows:

$$POI(u_i, z_k) = \sum_{j=1}^N W_{ij} R_{ij} p(v_j | z_k) \quad (7)$$

In this way, we can obtain users' POI scores on each interest circle, $POI \in \mathbb{R}^{C \times M}$. Similarly, we adopt the same mechanism of processing items to assign all users to proper communities.

4. TOP-N RECOMMENDATION FOR EACH USER BASED ON COMMUNITY

4.1 Using matrix factorization in each community

Recently, matrix factorization has drawn much attention due to the advantages of scalability and accuracy, especially for large-scale data, as exemplified by the Netflix contest.

After community detection, the observed user-item pairs are allocated into different communities. Let $R_k \in \mathbb{R}^{M_k \times N_k}$ denote the rating matrix for the k -th community, where $k = 1, \dots, K$. M_k and N_k are the number of users and items in each community respectively. Let $U_k \in \mathbb{R}^{M_k \times d}$ and $V_k \in \mathbb{R}^{N_k \times d}$ denote the latent feature matrices in k -th community, where d denotes the dimension of latent feature. Adopting PMF model in different communities, the model is trained on rating data by minimizing the square error:

$$L_1 = \sum_{i=1}^M \sum_{j=1}^N I_{ij}^k (R_{ij}^{k*} - U_i^k (V_j^k)^T)^2 + \lambda (\|U_i^k\|_F^2 + \|V_j^k\|_F^2) \quad (8)$$

Where I is an indicator function of which $I_{ij} = 1$ if user i rated item j , and 0 otherwise. The second term is regularization term and $\|\cdot\|_F$ is the Frobenius norm, which is introduced to avoid overfitting.

4.2 Merge the result from all the communities

Since we have obtain latent factor vector U_i^k, V_j^k of each community, we should solve the problem of how to get the result from all communities and merge them into a unified user-item matrix. Note that only if the user and item in the same community can we compute its value by :

$$\hat{R}_{ij}^k = \begin{cases} r_m + U_i^k (V_j^k)^T & \text{if } u_i \in C^k \cap v_j \in C^k \\ 0 & \text{otherwise} \end{cases} \quad (9)$$

Here we should handle another problem: since users and items may belong to multiple communities, so the predict ratings should be merged if one user and one item appear in many communities simultaneously. The final results are merged by linear averaging as $R_{ij} = \frac{\sum_k \hat{R}_{ij}^k}{S}$, where S is the number of community where user i and item j appear at the same time. Then for each user, K items with highest predict scores are recommended.

5. EXPERIMENTS

In this section, we conduct extensive experiments to evaluate the performance of our proposed CBMF by comparing with the state-of-the-art recommender systems on two real-world datasets. Also, we report the results for different settings of model parameters.

5.1 Datasets

Our experiments are conducted on two real world datasets: Epinions and Ciao, which are publicly available from Tang's homepage¹. Users in Epinions and Ciao can rate products with scores from 1 to 5. A timestamp is associated with a rating which records the time when the rating is given. Users can also establish social relations with others. Some statistics of these two datasets are presented in Table 1. To evaluate the effect of social activeness, we first remove users without trust relations. We also prune users with fewer than ten reviews and items rated by less than 10 users to ensure sufficient test data for each user. From the table we can see the rating data is very sparse for both Epinions and Ciao. In experiments, we randomly pick 80% of the rating data to generate the training set and the rest to be the test set, and run ten times in each configuration.

5.2 Evaluation Metrics

As discussed before, our framework is applied to give a Top-N list rather than a rating score. Hence, in our experiment several accuracy metric such as MAP (Mean Average Precision), Recall and NDCG (normalized Discounted Cumulative Gain) are adopted to evaluate the performance of the ranking list. MAP (Mean Average Precision) provides a

¹<http://www.public.asu.edu/~jtang20/datasetcode/truststudy.htm>

Table 1: Statistics of Datasets

	Ciao	Epinions
#user	5636	8477
#item	4147	10335
#rating	98529	306112
#relation	111781	296989
#of rating per user	17.48	36.11
#of rating per item	23.76	29.62
#of relation per user	19.83	35.03
Rating sparsity	99.58%	99.66%
Relation sparsity	99.65%	99.59%

single-figure measure of ranking quality across recall levels. The definition of MAP is expressed as below:

$$MAP = \frac{1}{|U|} \sum_{u=1}^{|U|} \frac{\sum_{j=1}^N (rel_u \times P_u @ j)}{\sum_{j=1}^N rel_u(j)} \quad (10)$$

where $rel_u(j)$ is an indicator function equaling 1 if the item at rank j is a relevant item, zero otherwise. In addition, we utilize recall as another evaluation metric, since precision metric is difficult to evaluate (rating 0 for item can imply either the user does not like the item or does not know about the item). Recall only considers the rated items within the top N list. For each user, $Recall@N$ is defined as:

$$Recall@N = \frac{\#hits}{|T|} \quad (11)$$

Where $\#hits$ is the number of relevant items in Top-N list. $|T|$ is the number of all relevant items in the test set. NDCG is a widely used metric for a ranking list. NDCG@K is defined as:

$$NDCG@K = \frac{1}{IDCG} \times \sum_{i=1}^K \frac{2^{r_i-1}}{\log_2(i+1)} \quad (12)$$

Where r_i is 1 if the item at position i is a hit item and 0 otherwise. $IDCG$ is chosen so that the perfect ranking has a NDCG value 1.

5.3 Comparisons

We compare CBMF with the state-of-the-art social recommender system, as well as a basic matrix factorization model and item/user-based collaborative filtering algorithms:

SVD: The user-item matrix is decomposed into three component matrices with k dimension features: $T = U_k S_k V_k^T$, the predicting score of user i to item j equals $\hat{R}_{ij} = \bar{r}_i + U_k \sqrt{S_k}^T(i) \cdot \sqrt{S_k} V_k^T(j)$, where \bar{r}_i is the i -th row average of T , and T is the normalization of R .

PMF: This method is the probabilistic matrix factorization approach proposed by [8], which does not take the social network into account. The estimated value of R is $\hat{R}_{ij} = U_i V_j^T$.

User-based CF: For user-based algorithm, a typically similarity metric: Pearson correlation, is adopted to measure the user-user similarities and predict by taking a weighted averaging of neighbors' rating.

Item-based CF: For item-based algorithm, it first finds a set of the most similar items that the target user has rated and then predicts the rating on the target item based on the ratings on these nearest items.

RSTE: This approach is proposed by [6]. It is a social matrix factorization method that models one's ratings as the combination between this user's own favors and the tastes of his friends. The rating of user i rated on item j can be predict with: $\hat{R}_{u,i} = g(\alpha U_u^T + (1 - \alpha) \sum_{v \in N_u} T_{u,v} U_v^T V_i)$, where $g(*)$ is logistic function.

PLSA-MF: For this method, we first detect community using original PLSA model and then in each community we employed basic PMF model to get the predict values.

CBMF-random: For this method, we first randomly partition users and items to several communities. The rest phases are the same as our proposed model.

CBMF-single: For this method, we assume every user and item can belong to only one community. i.e. It's non-overlapping. The rest is the same as our proposed model.

5.4 Result analysis

Note that we use cross-validation to determine parameters for all baseline methods. For CBMF, α is set to 0.7 and 0.5 for Ciao and Epinions respectively and the number of latent factors d is set to 10 for both datasets. The number of communities C is set to 15. The comparison results are demonstrated in Table 2 and we have the following observations:

- Exploiting community detection can significantly improve recommendation performance in terms of MAP@10, Recall@10, NDCG@10. For example, CBMF, CBMF-single, CBMF obtain 40.78%, 31.03%, 60.79% relative improvement in terms of MAP in Ciao, 94.58%, 93.45%, 109.57% relative improvement in terms of MAP in Epinions, compared to PMF.
- The proposed framework CBMF outperforms most approaches with respect to Recall@10, MAP and NDCG@10. The major reason is that CBMF exploits social activeness and temporal dynamic simultaneously. Although the NDCG@10 performance is worse than the performance of SVD in Epinions, we emphasize again that the precision related metrics are more critical indicators for this recommendation task.
- Our proposed framework CBMF, compared to PLSA-MF, RSTE, obtain 14.22%, 26.77% relative improvement in terms of MAP in Ciao, which indicates our proposed framework outperforms traditional community-based and social MF recommender algorithm.
- The results of CBMF in both Epinions and Ciao are better than CBMF-single, in which the communities are non-overlapping. We can infer that the overlapping communities are consistent with the physical truth and can result in more accurate recommendation. In addition, our proposed CBMF behaves better than CBMF-random, which infers that the communities generated by our framework can approximately represent different items' categories and people's tastes.

5.5 The Effect of Social Activeness and Temporal Dynamic

In this section, we will discuss the contribution of social activeness and temporal dynamic to our proposed model CBMF. To this end, we eliminate the effect of social activeness and temporal dynamic systematically from CBMF by defining the following variants of CBMF:

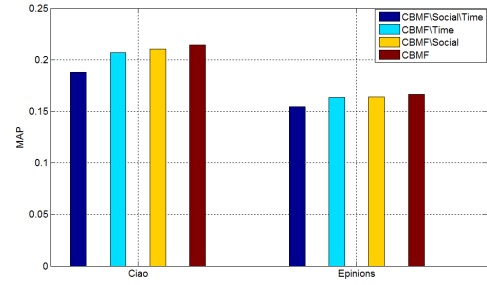


Figure 2: The Performance of Variants of Our Framework in Epinions and Ciao

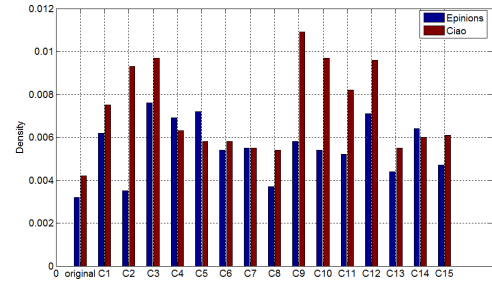


Figure 3: The Comparison of Density From Original Matrix to Submatrices in Epinions and Ciao

CBMF\Social\Time: Eliminating the effect of social activeness and temporal dynamic, i.e. original PLSA-MF model.

CBMF\Social: Eliminating the effect of social activeness by setting $W_a = 1$.

CBMF\Time: Eliminating the effect of temporal dynamic by setting $W_t = 1$.

The comparison results in Epinions and Ciao are shown in Fig.2. Note that we only show the results in Ciao since we have similar observations in Epinions. We can find when eliminating the effect of social activeness and temporal dynamic, the performance compared to **CBMF**, **CBMF\Time** and **CBMF\Social** have 3.41% and 2.05% relative performance reduction in terms of MAP, respectively. It means temporal dynamic is more helpful than social activeness. It proves again the significance of temporal features in recommender system. **CBMF\Time\social** obtains worse performance than both **CBMF\Time** and **CBMF\Social**, suggesting that both social activeness and temporal dynamic are beneficial to Top-N recommendation.

5.6 Data Sparsity Problem

Data sparsity is one of challenges in recommender system, which results in insufficient data to discover the essential attribute. In our framework, we employ community detection to allocate all users and items into proper communities. The density of all submatrices in communities compared to the original one is recorded in Fig 3. We observe that all the submatrices are denser than the original one. This is an exciting result and thus demonstrates our framework can mitigate the sparsity problem effectively.

Table 2: Performance Comparison on Epinions and Ciao Dataset

Model	Epinions			Ciao		
	MAP	Recall@10	NDCG@10	MAP	Recall@10	NDCG@10
user-based CF	0.0572	0.1806	0.1328	0.0738	0.1437	0.0937
item-based CF	0.0120	0.0758	0.0491	0.0409	0.1490	0.0762
SVD	0.0724	0.0300	0.2287	0.0852	0.0606	0.2500
PMF	0.0794	0.0778	0.1303	0.1334	0.2164	0.1736
RSTE	0.1246	0.1268	0.1738	0.1692	0.2395	0.2297
PLSA-MF	0.1545	0.1551	0.2195	0.1878	0.2478	0.1726
CB-Random	0.1094	0.0754	0.1719	0.1312	0.1200	0.1979
CBMF-single	0.1554	0.1233	0.2026	0.1739	0.1593	0.2062
CBMF	0.1683	0.1736	0.2265	0.2151	0.2539	0.2572

6. CONCLUSION

In this paper, we investigate the effect of social activeness and temporal dynamic on recommendation. To exploit social activeness information, we use social activeness scores to weight the importance of users' ratings. To exploit temporal dynamic information, we utilize a time decay function to weight the importance of the latest ratings that can reflect users' latest interest. Then we combine these two weights into PLSA model to discover communities. With these solutions, we propose CBMF, a two-phase framework systematically combines social activeness and temporal dynamic information to improve the quality of recommendations. We first utilize the modified PLSA model to discover communities. Then matrix factorization approach is employed on each community. We carried out extensive experiments to evaluate the performance of our approach on two real world data sets, Epinions and Ciao. The results indicate that the proposed CBMF framework significantly outperforms the state-of-the-art recommender algorithms. Future research direction includes making our algorithm parallelization and deploying our approach on MapReduce framework.

7. ACKNOWLEDGMENTS

The work was Supported by the Shenzhen Key Tech. R&D Program (No.ZD201010210080A) and the Research Fund for the development of strategic emerging industries by Shenzhen (No.JCYJ20120619153002947 and No. JCYJ201304021-45002449). In addition, we thank the anonymous reviewers for their very valuable comments and suggestions.

8. REFERENCES

- [1] Y. Ding and X. Li. Time weight collaborative filtering. In *Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 485–492. ACM, 2005.
- [2] S. Fortunato. Community detection in graphs. *Physics Reports*, 486(3):75–174, 2010.
- [3] T. Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 259–266. ACM, 2003.
- [4] Y. Koren. Collaborative filtering with temporal dynamics. *Communications of the ACM*, 53(4):89–97, 2010.
- [5] T. Q. Lee, Y. Park, and Y.-T. Park. A time-based approach to effective recommender systems using implicit feedback. *Expert systems with applications*, 34(4):3055–3062, 2008.
- [6] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 203–210. ACM, 2009.
- [7] P. Massa and P. Avesani. Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems*, pages 17–24. ACM, 2007.
- [8] A. Mnih and R. Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2007.
- [9] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: bringing order to the web. 1999.
- [10] S. Sahebi and W. W. Cohen. Community-based recommendations: a solution to the cold start problem. In *Workshop on Recommender Systems and the Social Web, RSWEB*, 2011.
- [11] J. Tang, X. Hu, H. Gao, and H. Liu. Exploiting local and global social context for recommendation.
- [12] B. Xu, J. Bu, C. Chen, and D. Cai. An exploration of improving collaborative recommender systems via user-item subgroups. In *Proceedings of the 21st international conference on World Wide Web*, pages 21–30. ACM, 2012.
- [13] T. Yang, R. Jin, Y. Chi, and S. Zhu. Combining link and content for community detection: a discriminative approach. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 927–936. ACM, 2009.
- [14] X. Zhang, J. Cheng, T. Yuan, B. Niu, and H. Lu. Toprec: domain-specific recommendation through community topic mining in social network. In *Proceedings of the 22nd international conference on World Wide Web*, pages 1501–1510. International World Wide Web Conferences Steering Committee, 2013.
- [15] G. Zhao, M. L. Lee, W. Hsu, W. Chen, and H. Hu. Community-based user recommendation in uni-directional social networks. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, pages 189–198. ACM, 2013.