

Recommending Interest Groups to Social Media Users by Incorporating Heterogeneous Resources

Wei Zeng^{1,2} and Li Chen¹

¹ Department of Computer Science,

Hong Kong Baptist University, Hong Kong

² School of Computer Science and Engineering

University of Electronic Science and Technology, China

zwei504@gmail.com, lichen@comp.hkbu.edu.hk

Abstract. Due to the advance of social media technologies, it becomes easier for users to gather together to form groups online. Take the Last.fm for example (which is a popular music sharing website), users with common interests can join groups where they can share and discuss their loved songs. However, since the number of groups grows over time, users often need effective group recommendation (also called affiliation or community recommendation) in order to meet like-minded users. In this paper, based on the matrix factorization mechanism, we have investigated how to improve the accuracy of group recommendation by fusing other potentially useful information resources. Particular, we adopt the collective factorization model to incorporate the user-item preference data, and the similarity-integrated regularization model to fuse the friendship data. The experiment on two real-world datasets (namely Last.fm and Douban) shows the outperforming impact of the chosen models relative to others on addressing the data sparsity problem and enhancing the algorithm's accuracy. Moreover, the experimental results identify that the user-item preference data can be more effective than the friendship in terms of benefiting the group recommendation.

Keywords: Recommending groups, matrix factorization, regularization, user-item preferences, friendship.

1 Introduction

In recent years, social media sites become popular among online users. Take Last.fm as a typical example, in this website, users can not only listen to music, but also be associated with different types of social relations: s/he may create a contact list including her/his friends; s/he could also join in interest groups to build membership with others whom are with some common interests in musics (though he may not know in the offline life). Therefore, in such environment, users should be willing to receive various types of recommendation from the website so as to more effectively establish their social network. However, so far, most research focuses have been put on recommending items (such as music), but less on recommending the relationship, especially the interest groups that

users might be affiliated with in the social media environment. Indeed, as the available user-group data are rather sparse, purely applying the classic recommender technology (like the collaborative filtering) cannot effectively generate the group recommendation. Therefore, in this paper, we have mainly been engaged in studying how to fuse other information resources, such as the user-item preferences (i.e., users' interaction with items) and user-user friendship data, to enhance the group recommendation.

Specifically, considering that the interest group is usually created based on multiple users' comment interests in items, their ratings (or implicit interaction like clicking) on items should reveal their similarity in terms of joining the groups. On the other hand, the friendship data can be supplementary to provide the network relation between users, though they might not be stronger than user-item preferences to reflect the common interests. In our work, the respective effects of the two information resources on improving group recommendation accuracy are empirically studied. Particularly, we take into account the property of data resource (i.e., bipartite or one mode data) when choosing the proper fusion model. More notably, the work has been grounded on the Matrix Factorization (MF) mechanism owing to its well-recognized high algorithm efficiency and accuracy [7]. In the following, we will first describe related works and then in detail present our proposed fusion methods. The experiment setup and results analysis will follow. At the end, we will conclude the major findings.

2 Related Work

Some of related works have emphasized studying how users form group. For instance, [1] investigated the phenomenon of group formation in two large social networks and found that the tendency of an individual to join a community is influenced not only by the number of friends s/he has within the community, but also by how those friends are connected to each another. They proposed decision tree based methods to measure the movement of individuals across communities, and showed how the movements were closely aligned with change in the topic of interest of the community. As another typical work, [6] showed that a group can attract new members through revealing the friendship ties of its current members to outsiders.

As for how to recommend groups to users, in [10], two models were explored, namely the Graph Proximity Model (GPM) and the Latent Factors Model (LFM), to generate community recommendation to users by taking into account their friendship and affiliation networks. Their empirical results indicated that GPM turns out to be more effective and efficient. [3] proposed a collaborative filtering method, called Combinational Collaborative Filtering (CCF), to perform personalized community recommendation. It concretely applied a hybrid training strategy that combines Gibbs sampling and Expectation-Maximization algorithm for fusing semantic info, such as the description of communities and users. The experiment on a large Orkut dataset demonstrated that the approach can more accurately cluster relevant communities given their similar semantics.

In [2], the authors investigated two approaches to generate community recommendation: the first adopted the Association Rule Mining technique (ARM) to discover associations between sets of communities; the second was based on Latent Dirichlet Allocation (LDA) to model user-community co-occurrences with latent aspects. The experiment on Orkut data indicated that LDA consistently outperforms ARM when recommending four or more communities, while ARM is slightly better when recommending up to three communities.

However, these related works did not consider the potential value of incorporating both user-item preferences and friendship into the group recommendation process. Moreover, the model they utilized, such as the Graph Proximity Model (GPM) in [10], is inevitably with high time complexity, that is why we have chosen matrix factorization as the basis mechanism to perform the fusion.

3 Proposed Methodology

Given a system like Last.fm, there are two types of data available, which are: 1) *bipartite data* such as user-item interaction data (in Last.fm they are implicit binary data where 1 means users clicked the item, and 0 otherwise), since there are two types of entities involved in each relationship, and 2) *one mode data* like the user-user friendship since only one type of entity (i.e., the “user”) exists. The user-group membership belongs to the first type. In order to effectively fuse the heterogeneous types of auxiliary resources into the group recommendation process, we have chosen the matrix factorization technology as the basis mechanism given that it could be extended to incorporate both bipartite and one-mode data with low computation complexity.

More specifically, for the one mode data, since it describes the relation between entities which are with the same type, it can be considered as an indicator of closeness. That is, if there is a link between two entities, we can regard that the two entities are closer than the ones without the link. Because of this, most state-of-the-art works leverage regularization model to fuse the one mode data in order to minimize the gap between two entities [9,8,5].

On the other hand, for the bipartite data such as user-item preferences, we argue that it is different from the one mode data in nature since a user indicates her/his interests in the item by interacting/rating it, which is however absent in the one mode data. Therefore, such data would be more suitably addressed by the factorization model, because it can effectively factorize user-item relations into two components and obtain a user’s latent factor model and an item’s latent factor model simultaneously. Previously, we discussed the limitation if bipartite data were handled in the manner of regularization [11]. We also proved that the regularization is better than factorization when dealing with one mode data like friendship. In this paper, we are motivated to consolidate the findings when the recommended object is changed from item to group. That is, would one-mode data still be better handled by regularization when they are fused into the process of recommending groups, and the factorization better suits user-item preferences?

Table 1. Summary of notations

<i>Notation</i>	<i>Description</i>
m, n, l	the numbers of users, items and groups respectively
k	the dimension of the factor vector
X, Y, Z	the user-factor, item-factor and group-factor matrix respectively
x_u, y_i, z_g	the user u , item i and group g factor vector respectively
$p_{ui}, p_{ug}^*, p'_{uf}$	user u 's preference on item i , group g and user f respectively
$p(u), p^*(u), p'(u)$	the vector that contains u 's the preference on all items, all groups and all friends respectively
$c_{ui}, c_{ug}^*, c'_{uf}$	the confidence level indicating how much a user likes an item, a group and a friend respectively
C^u, C^{*u}, C'^u	C^u denotes the $n \times n$ diagonal matrix and $C_{ii}^u = c_{ui}$; C^{*u} denotes the $l \times l$ diagonal matrix and $C_{gg}^{*u} = c_{ug}^*$; C'^u denotes the $m \times m$ diagonal matrix and $C'_{ff} = c'_{uf}$
$F(u)$	the friend set of user u
λ_f	the coefficient of the regularization
α	the coefficient for the collective matrix factorization

Table 1 first summarizes the notations used in the equations of the paper.

3.1 Baseline

To recommend interest groups to a user, we take the user-group matrix as the bipartite data type and use the following factorization equation as the baseline (which is without any fusions of other resources except the available user-group membership data themselves).

$$\min_{u^*, g^*} \sum_{u, g} c_{ug}^* (p_{ug}^* - x_u^T z_g)^2 + \lambda (\sum_u \|x_u\|^2 + \sum_g \|z_g\|^2) \quad (1)$$

where p_{ug}^* equals 1 if the user u joined group g , otherwise it is 0; c_{ug}^* is the confidence level indicating how much a user prefers a group which is set as 1 if no relevant data like “visiting frequency” are available.

The above cost function contains $m * l$ terms, where m is the number of users and l is the number of groups. To optimize it, we apply the Alternating Least Squares (ALS) [7,4], because it can help achieve massive parallelization of the algorithm by computing each z_i independent of the other group factors and computing each x_u independent of the other user factors. It was also demonstrated to be capable of efficiently processing the sparse binary data (such as the user-group relations in our case). Based on ALS, the analytic expressions for x_u and z_g that are used to minimize the above cost function are respectively:

$$x_u = (Z^T C^{*u} Z + \lambda I)^{-1} Z^T C^{*u} p^*(u) \quad (2)$$

$$z_g = (X^T C^{*g} X + \lambda I)^{-1} X^T C^{*g} p^*(g) \quad (3)$$

To generate a top-N recommendation list for each user u , we assume her/his candidate group set (i.e., groups unjoined by the user) is ϕ_u . For each group i in ϕ_u , we calculate a prediction score as follows:

$$p'_{ui} = x_u^T * z_i \tag{4}$$

where x_u^T and z_i are the user’s latent factor model and the group’s latent factor model respectively. Top-N groups with higher scores will then be included the recommendation list and returned to the target user.

3.2 Incorporating Friendship

To inject friendship in the above framework, we tried the factorization approach which was to factorize user-user friendship into two factor vectors. However, as mentioned before, because friendship belongs to one-mode data with only one type of entity existing, the regularization model would be more suitable [5,9]. Grounded on this model, we develop the following equation in order to minimize the gap between the taste of a user and the average taste of her/his friends:

$$\begin{aligned} \min_{u^*,g^*} \sum_{u,g} c_{ug}^* (p_{ug}^* - x_u^T z_g)^2 + \lambda (\sum_u \|x_u\|^2 + \sum_g \|z_g\|^2) \\ + \lambda_f (\|x_u - \frac{1}{|F(u)|} \sum_{f \in F(u)} \widehat{sim}(u, f) x_f\|^2) \end{aligned} \tag{5}$$

In this formula, λ_f is the coefficient for the friendship regularization. $\widehat{sim}(u, f) = sim(u, f) / \sum_{v \in F(u)} sim(u, v)$ denotes the normalized similarity degree between the user u and her/his friend f , which is used to adjust individual friends’ contributions when predicting the target user’s interests. It is worth mentioning that this similarity measure is a special element that we integrate into the regularization process in order to enhance its prediction power. In the experiment, we particularly compared the similarity-integrated regularization method to the one without its integration. We also tested different approaches to calculate the similarity degree, including ones based on common groups (shared by the user and her/his friend), common item preferences, and common friends. The vector space similarity (VSS) is concretely performed: $sim(u, f) = \frac{r_u r_f}{\|r_u\| \|r_f\|}$, where r_u can denote the group vector, friend vector or item vector of user u . The experimental results show that the common-group based similarity measure performs more accurate than others (see Section 4.2).

We then adopt ALS to perform the optimization process. Due to the addition of the friendship’s regularization, the analytic expression for x_u is changed to:

$$\begin{aligned} x_u = (Z^T C^{*u} Z + (\lambda + \lambda_f) I)^{-1} (Z^T C^{*u} p^*(u) \\ + \frac{\lambda_f}{|F(u)|} \sum_{f \in F(u)} \widehat{sim}(u, f) x_f) \end{aligned} \tag{6}$$

The expression for the group factor z_g is the same as in Equation (3).

3.3 Incorporating User-Item Preferences

Another auxiliary resource we considered is the user’s preferences on items (that can be either explicitly stated by users via rating, or inferred from their interaction with items such as “clicking” behavior). Still, we tried both factorization and regularization approaches. We especially investigated the collective matrix factorization (CMF) technique for fusing the data, so that the user-item interaction matrix can be directly factorized into two components: the “user” latent factor and the “item” latent factor.

$$\begin{aligned} & \alpha \min_{u^*,g^*} \sum_{u,g} c_{ug}^* (p_{ug}^* - x_u^T z_g)^2 + \lambda (\sum_u \|x_u\|^2 + \sum_g \|z_g\|^2) + \\ & (1 - \alpha) \min_{u^*,i^*} \sum_{u,i} c_{ui} (p_{ui} - x_u^T y_i)^2 + \lambda (\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2) \end{aligned} \quad (7)$$

where the parameter α is used to adjust the relative weights of user-group matrix and user-item matrix in the factorization. Similar to the definition of confidence level c_{ug}^* when factorizing user-group, we introduce the c_{ui} for user-item, that indicates the confidence level regarding users’ preference over item. Based on ALS, the analytic expressions for x_u and y_i are respectively defined as:

$$\begin{aligned} x_u = & (\alpha Z^T C^{*u} Z + (1 - \alpha) Y^T C^i Y + \lambda I)^{-1} * \\ & (\alpha Z^T C^{*u} p^*(u) + (1 - \alpha) Y^T C^u p(u)) \end{aligned} \quad (8)$$

$$y_i = (X^T C^i X + \lambda I)^{-1} X^T C^i p(i) \quad (9)$$

The expression for z_g is the same as in Equation (3).

For the purpose of comparison, we also developed the regularization-based fusion method, that converts user-item matrix into user-user relationship by means of a weighted scheme:

$$\begin{aligned} & \min_{u^*,g^*} \sum_{u,g} c_{ug}^* (p_{ug}^* - x_u^T z_g)^2 + \lambda (\sum_u \|x_u\|^2 + \sum_g \|z_g\|^2) \\ & + \lambda_f (\|x_u - \frac{1}{N(u)} \sum_{n \in N(u)} \omega_{un}^* * x_n\|^2) \end{aligned} \quad (10)$$

where the weight $w_{un}^* = \frac{|O_{un}|}{\sum_{i \in N(u)} |O_{ui}|}$ (O_{un} is the set of common items interacted by both users u and n , and $N(u)$ is user u 's neighbors who have common items with u).

The analytic expression for x_u in respect of the above model is

$$\begin{aligned} x_u = & (Z^T C^{*u} Z + (\lambda + \lambda_f) I)^{-1} (Z^T C^{*u} p^*(u) \\ & + \lambda_n \frac{1}{|N(n)|} \sum_{n \in N(n)} \omega_{un}^* x_n) \end{aligned} \quad (11)$$

3.4 Incorporating Friendship and User-Item Preferences Together

After fusing friendship and user-item preferences separately, we derive a formula to fuse them together:

$$\begin{aligned}
 & \alpha \min_{u^*, g^*} \sum_{u, g} c_{ug}^* (p_{ug}^* - x_u^T z_g)^2 + \lambda (\sum_u \|x_u\|^2 + \sum_g \|z_g\|^2) + \\
 & \quad \lambda_f (\|x_u - \frac{1}{|F(u)|} \sum_{f \in F(u)} \widehat{sim}(u, f) x_f\|^2) + \quad (12) \\
 & (1 - \alpha) \min_{u^*, i^*} \sum_{u, i} c_{ui} (p_{ui} - x_u^T y_i)^2 + \lambda (\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2)
 \end{aligned}$$

where the friendship is handled by the similarity-integrated regularization and user-item preferences are handled via the factorization. This combination was actually resulted from comparing regularization and factorization models for fusing friendship and user-item preferences respectively in the experiment (see Section 4.2). The analytic expression for x_u is

$$\begin{aligned}
 x_u = & (\alpha Z^T C^{*u} Z + (1 - \alpha) Y^T C^u Y + (\lambda + \alpha \lambda_f) I)^{-1} (\alpha (Z^T C^{*u} \\
 & p^*(u) + \frac{\lambda_f}{|F(u)|} \sum_{f \in F(u)} \widehat{sim}(u, f) x_f) + (1 - \alpha) Y^T C^u p(u)) \quad (13)
 \end{aligned}$$

The expression for z_g is the same as in Equation (3), and for y_i it is the same as in Equation (9).

4 Experiment

4.1 Dataset and Evaluation Metrics

Two real-world datasets, namely Last.fm (www.last.fm) and Douban (www.douban.com), were used to test the performance of the algorithms. The Last.fm is a worldwide popular social music site. The *membership* in the dataset refers to the user's participation in interest groups, the *friendship* was extracted from the user's friend list, and the *item* is referred to the artist (because users' preference over artists can be more stable than their preference over songs). Douban is a popular social media site in China that supports users to freely share movies, books and music. Being different from Last.fm, users of Douban can assign 5-scale ratings to items. Therefore, when assessing algorithms in both datasets, we can identify whether the performance is valid no matter whether the user-item preferences are explicit or implicit. Besides, for the sake of simplicity, we only collected users' data related to one product domain, the movie in Douban. We treat the user-item interaction matrix as 0/1, that is, the element equals to 1 if the user viewed (or rated) the item and 0 otherwise. Moreover, as Douban supports Twitter-like following mechanism, two users were treated

Table 2. Description of two datasets

	Element	Size	Element	Size
Last.fm	#user	100,000	#user-item pair	29,908,020
	#item	22,443	#friendship pair	583,621
	#group	25,397	#user-group pair	1,132,281
Douban	#user	71,034	#user-item pair	12,292,429
	#item	25,258	#friendship pair	273,832
	#group	2,973	#user-group pair	373,239

Table 3. Abbreviations' description

<i>Method</i>	<i>Description</i>
Group.MF	The basic matrix factorization;
Group.MF.F.R	Fusing the friendship by regularization;
Group.MF.F.F	Fusing the friendship by factorization;
Group.MF.I.R	Fusing the user-item preferences by regularization;
Group.MF.I.F	Fusing the user-item preferences by factorization;
Group.MF.FI	Fusing the friendship by regularization and fusing the user-item preferences by factorization;
Group.MF.F.FCos	Fusing the friendship by similarity-integrated regularization based on common friends;
Group.MF.F.GCos	Fusing the friendship by similarity-integrated regularization based on common groups;
Group.MF.F.ICos	Fusing the friendship by similarity-integrated regularization based on common items;
Group.MF.FI.GCos	Fusing the friendship by similarity-integrated regularization based on common groups and fusing the user-item preference by factorization.

as friends only if they follow each other. The details of the two datasets are described in Table 2.

To measure the accuracy of group recommendation, we applied the *leave-one-out* evaluation scheme because user-group pairs are rather sparse so they cannot be divided into subsets to perform the cross-fold validation. Concretely, during each testing round, we randomly selected one of the user's participated groups as the target choice. The measurement goal was hence to identify whether the top- N recommendation list as generated by the tested algorithm contains this target choice or not. Correspondingly, we use the hit ratio metric $Hits@N$ to evaluate the recommendation accuracy. That is, given the total number of users m , $Hits@N$ is defined as $Hits@N = \sum_{u=1}^m hit(u)@N/m$, where $hit(u)@N$ denotes whether user u 's target choice was located in the recommendation list.

4.2 Results

We first compared regularization and factorization models for fusing friendship, and for fusing user-item preferences, respectively. We also tested the model that fuses both data resources together. In total, we assessed 10 different methods via the experiment (see Table 3).

Table 4 shows the results. It can be seen that the regularization model (Group.MF.F.R) outperforms the factorization model (Group.MF.F.F) when fusing the friendship. It further shows that the regularization model integrated with the group-based similarity measure (Group.MF.F.GCos) not only outperforms the originally non-similarity based model, but also ones integrated with other similarity measures (such as item-based Group.MF.F.ICos and friend-based Group.MF.F.FCos).

As for fusing user-item preferences, it shows that the accuracy of factorization model (Group.MF.I.F) is improved with the increase of the density level of the user-item matrix (where $@train.X$ in Table 4 represents that $X\%$ of total user-item pairs are used). In comparison, the accuracy of regularization model (Group.MF.I.R) is lower and does not obviously change when the data density level is varied. This might be because once the user-item matrix is projected into the user-user matrix, a lot of information is lost, so the performance of Group.MF.I.R that fuses the projected matrix can not be improved even in denser user-item matrix.

The above results thus indicate that the regularization model is more suitable than the factorization for fusing one-mode data (friendship), while factorization is more suitable than regularization for fusing bipartite data (user-item preferences). In addition, the comparison between Group.MF.F.GCos (the best method regarding friendship's fusion) and Group.MF.I.F suggests that the user-item preferences act more positive than the friendship in terms of enhancing group recommendation.

Driven by the above results, we finally combined Group.MF.F.GCos and Group.MF.I.F@train.80 for fusing the two resources (friendship and user-item preferences) together, which is shorted as Group.MF.FI.GCos. From Fig. 1, it can be seen that such combination *Group.MF.FI.GCos* achieves accuracy improvement against fusing the two resources separately. Moreover, Group.MF.FI.GCos is better than an alternative combination Group.MF.FI (which is without the similarity integration).

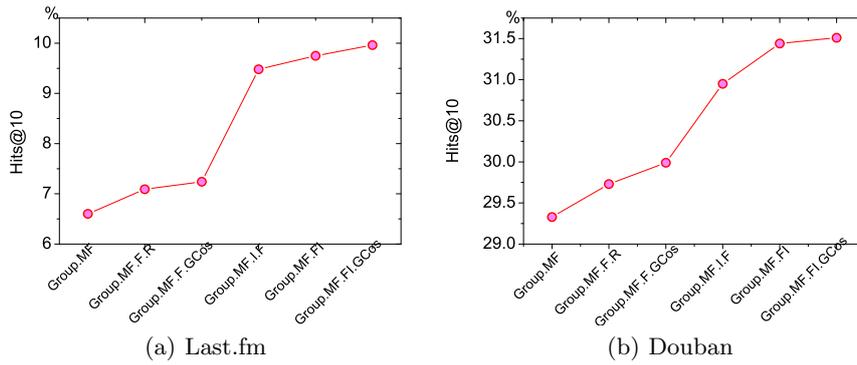
5 Conclusions

In conclusion, in order to solve the user-group sparsity phenomenon that commonly occurs in social media sites, we have proposed to fuse both friendship and user-item preference data to improve the accuracy of recommending interest groups to the target user. In more detail, we explored the matrix factorization technique to incorporate both one mode and bipartite data in a collective, unified framework. We have also proved the outperforming suitability of regularization model for handling the one mode friendship data, and the factorization

Table 4. Algorithms' comparison results

	<i>Last.fm</i>		<i>Douban</i>	
Method	Hits@5	Hits@10	Hits@5	Hits@10
Group.MF (baseline)	0.0530	0.0875	0.1995	0.2933
<i>Fusing user-item preferences (via Factorization)</i>				
Group.MF.I.F@train.20	0.0573	0.0899	0.2030	0.2950
Group.MF.I.F@train.40	0.0678	0.1026	0.2102	0.3013
Group.MF.I.F@train.60	0.0714	0.1068	0.2113	0.3079
Group.MF.I.F@train.80	0.0722	0.1070	0.2120	0.3095
<i>Fusing user-item preferences (via Regularization)</i>				
Group.MF.I.R@train.20	0.0559	0.0885	0.2025	0.2932
Group.MF.I.F@train.40	0.0559	0.0885	0.2026	0.2936
Group.MF.I.R@train.60	0.0560	0.0886	0.2026	0.2936
Group.MF.I.R@train.80	0.0561	0.0887	0.2027	0.2937
<i>Fusing friendship</i>				
Group.MF.F.R	0.0566	0.0910	0.2072	0.2973
Group.MF.F.F	0.0553	0.0876	0.2038	0.2928
Group.MF.F.FCos	0.0549	0.0861	0.2075	0.2974
Group.MF.F.GCos	0.0593	0.0923	0.2093	0.2999
Group.MF.F.ICos	0.0569	0.0897	0.2062	0.2921

Note: the size of user/group latent factors (k) is 10. The other parameters were tuned with optimal values, e.g., for Group.MF.I.F@train.20 $\alpha = 0.8$ in Last.fm dataset and $\alpha = 0.9$ in Douban dataset.

**Fig. 1.** Comparison of different methods

model for processing the user-item bipartite data. The friendship's regularization can be further augmented by integrating the similarity measure, especially the common-group based, to distinguish different friends' contributions.

By comparing the effectiveness of the two types of auxiliary resources, we found that the user-item preferences in general perform more accurate than the friendship to benefit the group recommendation, which might be attributed to its advantage of revealing users' comment interests in items. Furthermore, combining the two auxiliary resources can further increase the recommendation accuracy. Thus, our work points out a promising trend of incorporating heterogeneous data into the group recommendation. In the future, we will continue the work by investigating other potentially useful resources.

Acknowledgements. This research was supported by HKBU/FRG2/11-12/049 and partially supported by ECS/HKBU211912.

References

1. Backstrom, L., Huttenlocher, D., Kleinberg, J., Lan, X.: Group formation in large social networks: membership, growth, and evolution. In: Proc. KDD 2006, pp. 44–54. ACM, New York (2006)
2. Chen, W.-Y., Chu, J.-C., Luan, J., Bai, H., Wang, Y., Chang, E.Y.: Collaborative filtering for orkut communities: discovery of user latent behavior. In: Proc. WWW 2009, pp. 681–690. ACM (2009)
3. Chen, W.-Y., Zhang, D., Chang, E.Y.: Combinational collaborative filtering for personalized community recommendation. In: Proc. KDD 2008, pp. 115–123. ACM (2008)
4. Hu, Y., Koren, Y., Volinsky, C.: Collaborative filtering for implicit feedback datasets. In: Proc. ICDM 2008, pp. 263–272. IEEE Computer Society (2008)
5. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: Proc. RecSys 2010, pp. 135–142. ACM (2010)
6. Kairam, S.R., Wang, D.J., Leskovec, J.: The life and death of online groups: predicting group growth and longevity. In: Proc. WSDM 2012, pp. 673–682. ACM, New York (2012)
7. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. *Computer* 42, 30–37 (2009)
8. Ma, H., Yang, H., Lyu, M.R., King, I.: Sorec: social recommendation using probabilistic matrix factorization. In: Proc. CIKM 2008, pp. 931–940. ACM (2008)
9. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: Proc. WSDM 2011, pp. 287–296. ACM (2011)
10. Vasuki, V., Natarajan, N., Lu, Z., Dhillon, I.S.: Affiliation recommendation using auxiliary networks. In: Proc. RecSys 2010, pp. 103–110. ACM (2010)
11. Yuan, Q., Chen, L., Zhao, S.: Factorization vs. regularization: fusing heterogeneous social relationships in top-n recommendation. In: Proc. RecSys 2011, pp. 245–252. ACM (2011)