



A unified framework for recommending items, groups and friends in social media environment via mutual resource fusion

Li Chen^{a,*}, Wei Zeng^{a,b}, Quan Yuan^c

^a Department of Computer Science, Hong Kong Baptist University, Hong Kong, China

^b School of Computer Science and Engineering, University of Electronic Science and Technology, China

^c Taobao Software Co., Ltd, China

ARTICLE INFO

Keywords:

Fusion of heterogeneous resource
Item recommendation
Group recommendation
Friend recommendation
Matrix factorization
Regularization

ABSTRACT

Up to now, more and more online sites have started to allow their users to build the social relationships. Take the Last.fm for example (which is a popular music-sharing site), users can not only add each other as friends, but also join online interest groups where they shall meet people with common tastes. Therefore, in this environment, users might be interested in not only receiving item recommendations (such as music), but also getting friend suggestions so they might put them in the contact list, and group recommendations that they could consider joining. To support such demanding needs, in this paper, we propose a unified framework that provides three different types of recommendation in a single system: *recommending items*, *recommending groups* and *recommending friends*. For each type of recommendation, we in depth investigate the contribution of fusing other two auxiliary information resources (e.g., fusing *friendship* and *membership* for *recommending items*, and fusing *user-item preferences* and *friendship* for *recommending groups*) for boosting the algorithm performance. More notably, the algorithms were developed based on the matrix factorization framework in order to achieve the ideal efficiency as well as accuracy. We performed experiments with two large-scale real-world data sets that contain users' implicit interaction with items. The results revealed the effective fusion mechanism for each type of recommendation in such implicit data condition. Moreover, it demonstrates the respective merits of regularization model and factorization model: the factorization is more suitable for fusing bipartite data (such as membership and user-item preferences), while the regularization model better suits one mode data (like friendship). We further enhanced the friendship's regularization by integrating the similarity measure, which was experimentally proven with positive effect.

© 2013 Elsevier Ltd. All rights reserved.

1. Introduction

With the rapid growth of the internet and overwhelming amount of information and choices that people are confronted with every day, recommender systems have been widely developed with the purpose of supporting users to make effective decision in the online environment. In particular, given users are now commonly retained in a multi-resource environment, they do not seek for only one kind of recommendation. For example, in Last.fm which is a popular music sharing website, a user can be associated with different types of social relations: s/he may create a friend list (e.g., establishing the friendship) which is in nature a bidirectional relationship as two parties should approve this connection; s/he could join in an interest group, to build membership with others whom s/he may not know in the offline life but with common

interests. Accordingly, users in such environment are likely to be not only interested in receiving music recommendations, but also getting friend & group recommendations.¹

Though in the past years researchers have proposed different methods to address how to recommend items, groups or friends, these works were mostly done separately (that is, each mainly focuses on one type of recommendation only; see related works in Section 2). Few have actually combined them in a unified framework, and more meaningfully, discovered the mutual contribution of the three information resources: *user-item preferences*, *user-group membership* and *user-user friendship*. Indeed, due to the data sparsity phenomenon that is commonly occurring in current online systems, purely considering one type of resource to generate recommendation for itself (such as only involving user-item records

* Corresponding author. Address: 224 Waterloo Road, Kowloon Tong, Kowloon, Hong Kong. Tel.: +852 34117090; fax: +852 34117892.

E-mail address: lichen@comp.hkbu.edu.hk (L. Chen).

¹ The group recommendation described in this paper is different from the concept described in Chen, Cheng, and Chuang (2008b) and Baltrunas, Makcinskas, and Ricci (2010). Their focus was on recommending items to a group of users, whereas ours is on recommending groups to one target user.

to produce item recommendation) cannot effectively help inactive users (who have only expressed few ratings or interacted with few items). To address this problem, fusing other types of information resources should be essentially helpful (Liu & Lee, 2010). Particularly, among others, user-group membership data can be a direct indicator of the user's specific interest in a group's topic. However, it has been rarely investigated as a potential auxiliary resource in existing works. Few attentions have also been paid to augment the group recommendation's accuracy.

Thus, in this manuscript, we aim at addressing existing limitations and emphasizing the triangle relationship among the three types of recommendation: items, groups and friends (see Fig. 1). The ultimate goal was to unify them into a generalized framework. In particular, we chose the matrix factorization (MF) as the basis mechanism due to its well-recognized high algorithm efficiency and accuracy (Koren, Bell, & Volinsky, 2009) (the detailed rationale will be given in Section 3.1). In the experiment, we have compared different MF-based fusion approaches and in-depth explored how different data resources could be mutually contributive to each other for producing recommendation. More notably, we have investigated the fusion performance in a more realistic, typical data condition: it is only with users' implicit interaction records with items. As a matter of fact, many recommender methods that have been proposed so far stand on the assumption that users' explicit ratings are available (Koren et al., 2009; Ma, Zhou, Liu, Lyu, & King, 2011). However, in reality, explicit feedbacks from users are not so easily obtained (Yang, Lee, Park, & Lee, 2012). Thus, a more practical solution might be to derive user preferences from their implicit behavior, such as users' clicking and interaction records (Hu, Koren, & Volinsky, 2008). Unfortunately, in such implicit data condition, by far few studies have been done to incorporate social relations to boost the recommendation. We were hence driven to develop effective fusion methods for dealing with the implicit data.

Below we summarize our major contributions in terms of each type of recommendation:

1.1. Recommending items

This is the main objective of traditional recommender systems, i.e., to recommend a set of top-N items (e.g., music, book, movie) that users might be interested in. Up to now, different recommendation algorithms have been proposed to predict items for users, such as collaborative filtering (Adomavicius & Tuzhilin, 2005), matrix factorization (Hu et al., 2008; Koren et al., 2009; Salakhutdinov & Mnih, 2008b), content-based analysis (Pazzani & Billsus, 2007), latent semantic models (Hofmann, 2004), latent Dirichlet allocation (Blei, Ng, & Jordan, 2003), and so on. However, as mentioned before, some challenging research questions still remain unsolved: (1) *how to effectively incorporate membership, in addition to friendship, into the process of recommending items, so as to fulfill their respective values?* (2) *Could the fusion take significant effect when there are only users' implicit interaction records?* The answer to this question could well address the concern of how to choose appro-

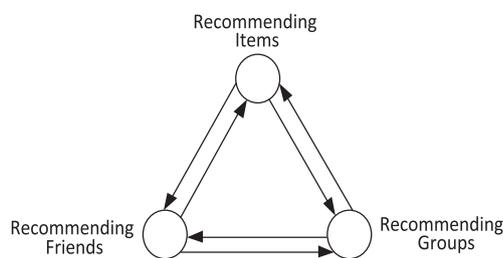


Fig. 1. Triangle relationship among item, friend and group recommendations.

appropriate fusion algorithm in the implicit data condition. (3) *How would the fusion effect be, when there are few of users' interaction data (i.e., the data sparsity phenomenon)?* The answer to this question could help identify whether the fusion of social relations, via the proper modeling, could in practice solve the cold-start problem.

To find solutions to these questions, we have first analyzed two types of relation data's property: one mode relation (e.g., friendship) and bipartite relation (e.g., membership) (see Section 3.2). In terms of algorithm development, we have chosen the matrix factorization technique as the basis mechanism (see Section 3.1). Specifically, we have proposed utilizing collective matrix factorization to fuse membership, and adopting regularization model for fusing friendship. Moreover, we have integrated the similarity measure into the friendship regularization in order to further enhance its prediction power. The experimental results not only demonstrated the proposed algorithms' performance, but also identified the superior effectiveness of membership against friendship. The similarity-integrated regularization model also results in positive outcome.

1.2. Recommending groups

Besides recommending items, it is interesting to recommend to the target user a list of communities (e.g., "interest groups") that s/he might be interested in joining. Membership in nature involves two types of entities: users and groups, which reflect users' participation in groups based on their common tastes. In recent years, some efforts have been devoted to make community recommendation (Chen et al., 2009b; Vasuki, Natarajan, Lu, & Dhillon, 2010), but the proposed methods leveraged only one kind of data: the user-group relations. To the best of our knowledge, few works have exploited the impact from fusing other information resources, such as user-item preferences and friendship, for enhancing the group recommendation. Actually, the three research questions that were raised for item recommendation are also valid for the group recommendation. Thus, to fill in the vacancy, we have explored different fusion methods, and experimentally identified the superior performance of fusing friends via the similarity-enhanced regularization mechanism, and the effect of the collective factorization model for fusing user-item preferences. The item preferences are additionally found more effective than friendship, and the two resources' combination gives the best result.

1.3. Recommending friends

The third issue that we have attempted to resolve was to enhance friend recommendation by incorporating user-item preferences and membership data. Although lately there are increasing interests in generating friend recommendation due to the popularity of social network (Jamali, Huang, & Ester, 2011; Yang et al., 2011), few works have measured the performance when other data resources are leveraged into this process. Moreover, the related works have mainly employed graph-based techniques which are nevertheless with high time complexity. Because matrix factorization (MF) has the inherent advantage of reducing the algorithm's cost (Hu et al., 2008; Koren et al., 2009), we have particularly investigated and compared various MF-based fusion approaches.

In the following, we will first introduce related works and indicate their limitations (Section 2). We then explain the rationale behind our choice of fusion mechanism, and analyze the typical property of involved data (Section 3). The algorithms' details for the three types of recommendation (items, groups and friends) will be respectively presented in Sections 4–6. We then introduce the experiment setup, that includes the description of dataset and the definition of evaluation metrics (Section 7). The experimental

results are then analyzed with respect to each type of recommendation, which show the comparison of various fusion methods (Section 8). At the end, we summarize the major findings and draw the conclusion (Section 9).

2. Related work

According to the type of recommendation that each related work emphasizes, we classify them into three branches: *item recommendation*, *group recommendation* and *friend recommendation*. In the following, we introduce the state-of-the-art of each branch.

2.1. Item recommendation

In order to solve the “cold-start” problem and the sparsity of user-item ratings, Jamali and Ester (2009) proposed a random walk model that combined the trust network so that indirect neighbors were also considered. The experiment on the Epinions dataset showed that the proposed approach outperforms standard collaborative filtering algorithm. Jamali and Ester (2010) utilized the trust relationship under the matrix factorization model, which was called Trust-MF. In Ma, Yang, Lyu, and King (2008), authors proposed a factor analysis approach based on the probabilistic matrix factorization, that took into account both users’ trust network information and rating records. The experiment showed that their method performs more accurate especially when users made few or no ratings.

However, given the difficulty of obtaining actual trust relations in the real online environment, some researchers have attempted to utilize friendship data as they can be more easily obtained from social networking sites (Groh & Ehmig, 2007). A typical work is (Konstas, Stathopoulos, & Jose, 2009) which adopted the generic framework of Random Walk with Restart to model the friendship and social annotation (tagging) in a music track recommendation system. Their experiment showed that the graph model benefits from the additional information embedded to increase recommendation accuracy. In Yang et al. (2011), the authors showed that the information contained in interest networks (i.e., user-service interactions) and friendship networks are highly correlated and mutually helpful. They concretely proposed a friendship based interest propagation (FIP) mechanism which devised a factor-based random walk model to recommend both online services and friends to users. The experiment demonstrates that FIP achieves higher performance regarding both interest targeting and friendship prediction, than systems that use only one source of information. In Jamali et al. (2011), authors proposed a generalized stochastic block model (GSBM), by which both the ratings of items and the friendship can be predicted. Their experiment indicates that although GSBM did not outperform all compared partners in respect of rating prediction, its performance is comparable to the state-of-the-art methods, and it is able to handle multiple tasks. In Ma et al. (2011), the authors proposed two social regularization terms for defining the matrix factorization objective function, with the goal of effectively fusing the friendship information into item recommendation.

However, few attentions have been paid to study the impact of membership data, as another type of auxiliary information, on augmenting the item recommendation’s accuracy. As users’ affiliation with interest groups can more likely reflect their common preferences over items, we believe the membership could be stronger indicator than friendship in terms of enhancing the item recommendation.

2.2. Group recommendation

Regarding group recommendation (or called affiliation or community recommendation in Vasuki et al. (2010)), there are

relatively fewer works. In Vasuki et al. (2010), 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. Chen, Zhang, and Chang (2008a) 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 data set demonstrated that the approach can more accurately cluster relevant communities given their similar semantics. In Chen et al. (2009b), 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. Spertus, Sahami, and Buyukkokten (2005) presented an empirical comparison of six similarity measures for recommending online communities to members in Orkut social network. However, these related works did not explore the potential of fusing other auxiliary resources, especially user-item preferences, for increasing the accuracy of group recommendation.

2.3. Friend recommendation

In Chen, Geyer, Dugan, Muller, and Guy (2009a), authors evaluated several friend recommendation algorithms in an enterprise social networking environment through a user survey (with 500 users) and a field study (with 3000 users). They found that algorithms based on social network information can reveal more known contacts for users, while algorithms that considered the similarity between user-created content were more useful in discovering new friends. Guy, Ronen, and Wilcox (2009) proposed the “Do You Know?” (DYK) widget, by which people recommendations were generated on an aggregated social network that contains various resources across the organization. Their evaluation showed that people recommendation can be effective in increasing the number of social connections. In Symeonidis, Tiakas, and Manolopoulos (2010), a so called FriendTNS algorithm was proposed, that recommends new friends to registered users based on both local and global graph features. Backstrom and Leskovec (2011) developed an algorithm based on Supervised Random Walks, that was targeted to leverage the information from the network structure to predict the occurrence of links between users and was successfully tested on Facebook dataset. In Gou, You, Guo, Wu, and Zhang (2011), authors proposed a novel visual system, called SFViz (Social Friends Visualization), to support users to explore and find friends interactively, by leveraging both semantic structure of activity data and topological structure of social networks. The experiment on Last.fm data indicated that the system can enhance users’ awareness of their social networks under different interest contexts, and help users to seek potential friends who share similar interests in an informative way. Zheng, Zhang, Ma, Xie, and Ma (2011) reports a personalized friend and location recommender for the geographical information system (GIS) on the Web. A framework, referred as a hierarchical-graph-based similarity measurement (HGSM), was proposed to effectively measure the similarity among users based on their location histories. Based on HGSM, the content-based method and the user-based collaborative filtering algorithm were applied to provide not only the item rec-

ommendation but also the friend recommendation. A similar work that considered users' mobile proximity network was proposed by Quercia and Capra (2009). The proximity records were processed based on the geographical proximity and link prediction, which resulted in a personalized and automatically generated list of people whom the target user may know.

2.4. Limitations of related works

Table 1 summarizes several typical works from two aspects: the auxiliary information resource(s) that they fused; and the type of recommendation(s) they emphasized. It can be seen that: (1) as for works that focus on recommending items or friends, few of them have considered utilizing membership information; (2) few have combined all three types of recommendation in a unified framework; (3) few have been targeted to deal with implicit user-item preference data. The work most similar to ours is (De Meo, Nocera, Terracina, & Ursino, 2011), which recommends to the active user a set of similar users, resources and social communities in a cross-network setting. This method considered both explicit relationships among users and implicit ones that connected users given their similar interests and behavior. However, the main limitation of this work is that it did not identify the mutual contributing effects among different types of data resources. For instance, to produce community recommendation, it did not address whether injecting users' existing preferences over items could be helpful to infer users' common interests in groups.

Therefore, given the limitations of related works, we have been motivated to in-depth explore the triangle relationship that is particularly among items, groups (i.e., communities) and friends. Moreover, we have investigated how to optimally fuse these data resources for augmenting the three types of recommendation simultaneously, under the matrix factorization framework, which is novel to the best of our knowledge.

3. Factorization background and data property analysis

Before presenting our fusion methods, in this section, we first introduce the rationale behind our algorithm development, and then show the representative properties of involved data resources.

3.1. Rationale behind our choice of fusion mechanism: matrix factorization

To take into account social relations, there are three typical fusion schemes to choose: the weighted-similarity scheme, the graph-based scheme, and the matrix factorization (MF) scheme. The latter two are more advanced and popular than the first one, among which we finally chose the MF scheme due to the following concerns.

If choosing the graph-based scheme, the common way is applying the random walk technique to produce recommendations based on the graph structure (Fouss, Pirotte, Renders, & Saerens, 2007). We previously tried this method (Yuan, Chen, & Zhao, 2012), but found it is inevitably complex when handling the large-scale dataset. Specifically, random walk is a mathematical formalization of trajectory that consists of taking successive random steps. At each step, the next node in the walk is selected randomly from the neighbors of the last node in the walk. The sequence of visited nodes is a Markov chain, with the transition probability. Since the graph is totally connected, the Markov chain is irreducible. That is, every state can be reached from any of other states. For recommender systems, the random walk technique can be applied to calculate the similarity between two users, including directly connected user nodes and indirectly connected ones, based on the whole graph's knowledge. For instance, a random walker gets off from the node user u_a and arrives at the node user u_b with the probability p_{ab} , or arrives at the node user u_c with the probability p_{ac} . If $p_{ab} > p_{ac}$, it is reasonable to say that u_a is connected to u_b more closely than to u_c , and hence $Similarity(u_a, u_b)$ should be bigger than $Similarity(u_a, u_c)$.

Therefore, suppose there is a set of users for whom we want to provide item recommendations, the computation complexity of the random walk will be $O(m \cdot (N) \cdot |U|)$, where m is the iteration times, N is the number of user-item edges (pairs), and $|U|$ is the total number of users. We can see that for each user, we need to run m times of iteration. The number of edges (N) will increase rapidly as the $|U|$ grows, which will make the computation time unavoidably considerable when the size of $|U|$ is very large. It thus implies that the graph-based method is not an optimal choice for handling the large-scale dataset.

In recent years, matrix factorization (MF) technique has brought more attentions in the area of recommender system, mainly owing to its well-recognized merit in saving the computation cost (Koren et al., 2009). Its original form is the low-rank MF which was proposed to train user-item matrix, under the assumption that a user's preferences are influenced by a set of factors and hence the preference vector is determined by how each factor affects that user (Rennie & Srebro, 2005; Salakhutdinov & Mnih, 2008a). In addition to factorizing only user-item (e.g., user-movie) matrix, Singh and Gordon (2008) introduced collective matrix factorization (CMF) which was to combine the factorization of item-feature (e.g., movie-genre) matrix with the one of user-item (e.g., user-movie) matrix. As to the algorithm's complexity, in the case with implicit interaction data (that we focus in this paper), the Alternating Least Square (ALS) can be used to conduct the optimization process (Hu et al., 2008). The complexity is $O(f^2N + f^3|U| + f^3|I|)$, where N is the number of user-item pairs, $|U|$ is the total number of users, and $|I|$ is the total number of items. Therefore, it can be seen that the running time is linearly increasing with the size of the input. Because the value of f (the number of factors) normally lies between 20 and 200 which is much smaller than the sizes of users and items, the MF technique is obviously less complex and less time

Table 1
Summary of typical related works.

Citation	Type(s) of recommendation	Information resource(s)	Algorithm framework
Konstas et al. (2009)	Item	Item interaction, friendship, social annotation	Random walk
Jamali and Ester (2010)	Item	Item ratings, trust relation, friendship	Matrix factorization
Ma et al. (2011)	Item	Item ratings, friendship	Matrix factorization
Chen et al. (2009b)	Community (group)	User-community data	Latent Dirichlet allocation, association rule mining
Vasuki et al. (2010)	Group	Affiliation network (user-group), friendship	Graph proximity, link prediction
Yang et al. (2011)	Item & friend	Item interaction, friendship	Factor-based random walk
Symeonidis et al. (2010)	Friend	Friendship	Graph proximity and transitivity
Backstrom and Leskovec (2011)	Link (friend)	Co-authorship network (friendship)	Random walk

consuming than the random walk process, especially when there are more than thousands of users.

Based on the above comparison, we decided to use MF as the basic recommendation mechanism. The question to us is then: *how to fuse heterogenous resources under this framework to augment the three types of recommendation simultaneously: items, groups and friends?* As a matter of fact, related factorization-based recommender methods primarily emphasized item recommendation. Few have studied the actual role of MF in handling social recommendation, when membership and friendship are concerned. Therefore, in this paper, we have not only proposed and evaluated the methods of incorporating membership (along with friendship) to improve item recommendation, but also investigated how to fuse different resources to benefit group and friend recommendations based on the matrix factorization mechanism.

3.2. One mode vs. bipartite data

Given a system like Last.fm, there are three types of data available, which are: (1) user-item interaction data (e.g., implicit binary data in Last.fm where 1 means users clicked the item, and 0 otherwise), (2) the user-user friendship, and (3) the user-group membership. We classify these data into two classes: *one mode data* and *bipartite data* (Fig. 2). One mode data refer that the data set only contains one type of entity. For example, in user-user friendship (see Fig. 2(a)), there exists only one type of entity which is the “user”. Bipartite data mean that the data set contains two types of entities (see Fig. 2(b)). For example, in user-item interaction records, one is the “user” and the other is the “item”. In user-group membership, the two entities are the “user” and the “group”.

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 the taste of a user and the taste of her/his friends (Ma et al., 2008, 2011; Jamali & Ester, 2010).

On the other hand, for the bipartite data, we argue that it is different from the one mode data in nature since a user indicates her/his interests in the item (or group) by interacting with it (e.g., rating/joining), 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 (or user-group relations) into two components and obtain a user’s latent factor model and an item’s latent factor model simultaneously. However, if we handle bipartite data in the manner of regularization, we need to first do the one-mode projection,

i.e., transforming the user-group relationship into the user-user relationship, but the one-mode projection is less informative than the bipartite representation. For example, user u_1 and user u_2 joined group a , and user u_1 and user u_3 joined another group b . Group a and group b are two different groups with different discussion topics. If we project such data into a user-user relation graph, u_1 will be linked to both u_2 and u_3 . From the transitivity perspective, u_2 and u_3 should share some common interests, but the fact is not.

Thus, for bipartite relations, we choose the collective matrix factorization (CMF) technique to factorize them. Our experiments (see Section 8) prove that the performance of factorization model for fusing bipartite data is better than the one of regularization model, while the regularization is better than the factorization when dealing with one mode data.

4. Recommending items: fusing friendship and membership

In this section, we mainly present the methods developed for augmenting item recommendation. We first describe the baseline method and then the fusion methods that we propose for injecting the social relations. Table 2 lists all major notations that are used in this paper.

4.1. Baseline: matrix factorization with implicit data

As mentioned in the introduction, the focus of this paper is on proposing effective methods for enhancing the recommendation when there is only implicit user behavior. This data condition is actually quite common in current social media sites because basically they all have implicit records, but not all of them can get explicit ratings from users.

With the implicit binary data as input, a matrix can be built with rows denoting users and columns representing items, which is similar to the explicit user-item matrix. However, instead of putting the exact rating (e.g., from 1 to 5, or 0 if no rating provided) in each cell, it should be either 1 (if the user clicked the item) or 0 (otherwise). The other two differences from processing explicit ratings are that: (1) the factor model should be tailored with varied confidence levels (that indicate how much a user prefers an item). The confidence level can be computed according to the time a user spends on an item, or the frequency a user interacts with an item; (2) the optimization process should account for all $\langle user, item \rangle$ pairs. That is, no matter whether the cell is ‘1’ or ‘0’, the value could be taken into consideration.

With this matrix, we can map both users and items to a joint latent factor space with the dimensionality k . The objective was then to compute a user-factor vector x_u for each user u , and compute an

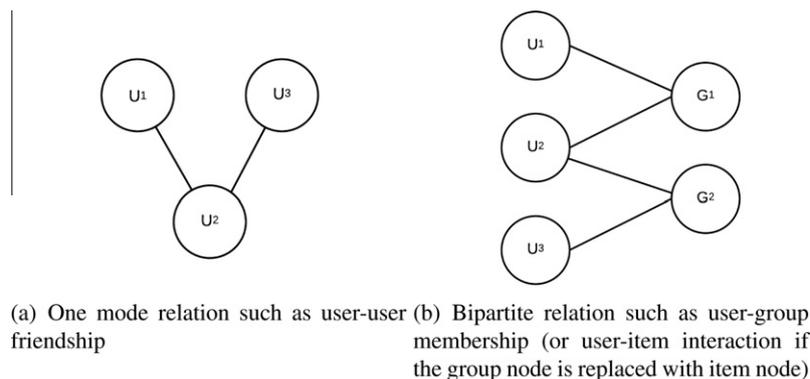


Fig. 2. One mode relation vs. bipartite relation.

Table 2
Notations used in the equations.

Notation	Description
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 $C_{ff}^u = c'_{uf}$
C^i, C^{*g}, C^f	C^i denotes the $m \times m$ diagonal matrix and $C_{uu}^i = c_{ui}$; C^{*g} denotes the $m \times m$ diagonal matrix and $C_{uu}^{*g} = c'_{ug}$; C^f denotes the $m \times m$ diagonal matrix and $C_{uu}^f = c'_{uf}$
$F(u)$	The friend set of user u
λ_f	The coefficient of the regularization
α, β	The coefficients for the collective matrix factorization

item-factor vector y_i for each item i . The rating prediction for user u to item i is based on the inner product of corresponding user-factor and item-factor, i.e., $r_{ui} = x_u^T y_i$. More specifically, factors are computed by minimizing the following cost function:

$$\min_{u,i} \sum_{u,i} c_{ui} (p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right) \quad (1)$$

where p_{ui} measures user u 's preference on item i : it is 1 if u "clicked" the item; otherwise, $p_{ui} = 0$. c_{ui} is the confidence level (it is default set as 1 if no confidence-related data such as "time" and "frequency" are available).

The above cost function contains $m \times n$ terms, where m is the number of users and n is the number of items. To optimize it, we apply the Alternating Least Squares (ALS) due to two primary considerations (as inspired by Koren et al. (2009) and Hu et al. (2008)): (1) though in general another optimization process, Stochastic Gradient Descent (SGD), is easier and faster than ALS, ALS can help achieve massive parallelization of the algorithm because it computes each y_i independent of the other item factors and computes each x_u independent of the other user factors; (2) for implicit data, because the training set cannot be considered sparse, looping over each single training case, as SGD does, would not be practical, but ALS can efficiently handle such case.

Based on ALS, the analytic expressions for x_u and y_i can be respectively formally defined as:

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

$$y_i = (X^T C^i X + \lambda I)^{-1} X^T C^i p(i) \quad (3)$$

In these equations, C_u denotes a diagonal $n \times n$ matrix, where $C_{ii}^u = c_{ui}$. The vector $p(u)$ contains all the preferences of u (see Table 2). This baseline approach is shorted as Item.MF henceforth.

4.2. Fusing friendship by regularization

It can be seen that the baseline method is without any fusions of other information resources, only considering user-item preference data alone. To inject friendship in this framework, we tried the factorization approach, which was to factorize user-user friendship into two factor vectors (Yuan, Chen, & Zhao, 2011). However, as mentioned before, because friendship belongs to one-mode data with only one type of entity existing, the regularization model should be more suitable (Jamali & Ester, 2010; Ma et al., 2011). 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:

$$\min_{u,i} \sum_{u,i} c_{ui} (p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right) + \lambda_f \left(\left\| x_u - \frac{1}{|F(u)|} \sum_{f \in F(u)} \widehat{sim}(u,f) x_f \right\|^2 \right) \quad (4)$$

In the above 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: $sim(u,f) = \frac{r_u^T r_f}{\|r_u\| \|r_f\|}$, where r_u can denote 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 8.1.2).

We then adopt alternating-least-squares to perform the optimization process. Due to the addition of the friendship regularization part, the analytic expression for x_u is changed to:

$$x_u = (Y^T C^u Y + (\lambda + \lambda_f) I)^{-1} \left(Y^T C^u p(u) + \frac{\lambda_f}{|F(u)|} \sum_{f \in F(u)} \widehat{sim}(u,f) x_f \right) \quad (5)$$

For y_i , it is the same as the one defined in Eq. (3).

With the purpose of comparison in the experiment, the regularization equation without the similarity integration is:

$$\min_{u,i} \sum_{u,i} c_{ui} (p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right) + \lambda_f \left(\left\| x_u - \frac{1}{|F(u)|} \sum_{f \in F(u)} x_f \right\|^2 \right) \quad (6)$$

Accordingly, the analytic expression for x_u in Eq. (6) is:

$$x_u = (Y^T C^u Y + (\lambda + \lambda_f) I)^{-1} \left(Y^T C^u p(u) + \lambda_f \frac{1}{|F(u)|} \sum_{f \in F(u)} x_f \right) \quad (7)$$

4.3. Fusing membership by factorization

Compared to the friendship that involves only one type of entity, the membership involves two types of entities which reflect users' participation in groups. Therefore, the user-group interaction matrix can be directly factorized into two components: the "user" latent factor and the "group" latent factor. Therefore, we base collective matrix factorization (Singh & Gordon, 2008) to incorporate the factorization of membership into the item recommendation:

$$\alpha \min_{u,i} \sum_{u,i} c_{ui} (p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right) + (1 - \alpha) \min_{u,g} \sum_{u,g} c_{ug}^* (p_{ug}^* - x_u^T z_g)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_g \|z_g\|^2 \right) \quad (8)$$

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

$$x_u = (\alpha Y^T C^u Y + (1 - \alpha) Z^T C^{*u} Z + \lambda I)^{-1} * (\alpha Y^T C^u p(u) + (1 - \alpha) Z^T C^{*u} p^*(u)) \quad (9)$$

The expression for group factor z_g is:

$$z_g = (X^T C^{*g} X + \lambda I)^{-1} X^T C^{*g} p^*(g) \quad (10)$$

For y_i , it is the same as in Eq. (3).

We also developed a regularization-based membership fusion approach, in order to compare it to the above factorization-based method in the experiment. The concrete idea was to convert user-group matrix into user-user relationship by means of a weighted scheme. For example, if user u and user v joined two common groups, there is a link between user u and user v , with the weight set as two. The cost function is formally defined as:

$$\min_{u,i} \sum_{u,i} c_{ui} (p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right) + \lambda_n \left(\left\| x_u - \frac{1}{|N(u)|} \sum_{n \in N(u)} w_{un} * x_n \right\|^2 \right) \quad (11)$$

where λ_n is the coefficient for the regularization of membership, $N(u)$ is the user u 's neighboring users who have common groups with u , and x_n is the neighbor's factor. w_{un} is the weight between the current user u and the neighbor n , which is calculated as:

$$w_{un} = \frac{|CG_{un}|}{\sum_{i \in N(u)} |CG_{ui}|} \quad (12)$$

where CG_{un} is the set of common groups between user u and n , and $|CG_{un}|$ is the set's size. The analytic expression for x_u is:

$$x_u = (Y^T C^u Y + (\lambda + \lambda_f) I)^{-1} \left(Y^T C^u p(u) + \lambda_n \frac{1}{|N(u)|} \sum_{n \in N(u)} w_{un} * x_n \right) \quad (13)$$

On the other hand, in order to compare factorization and regularization for fusing friendship, we also implemented a factorization-based friendship fusion method, which is essentially

similar to Eq. (8), but with the user-user binary matrix as input where a cell is assigned '1' if two users have the friendship linkage.

4.4. Fusing membership and friendship together

After fusing friendship and membership separately, we derive a formula to fuse them together (see Eq. (14)). Concretely, the factorization of user-item matrix is combined with the similarity-integrated regularization of friendship & the factorization of user-group matrix:

$$\alpha \min_{u,i} \sum_{u,i} c_{ui} (p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right) + \lambda_f \left(\left\| x_u - \frac{1}{|F(u)|} \sum_{f \in F(u)} \widehat{sim}(u,f) x_f \right\|^2 \right) + (1 - \alpha) \min_{u,g} \sum_{u,g} c_{ug}^* (p_{ug}^* - x_u^T z_g)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_g \|z_g\|^2 \right) \quad (14)$$

The analytic expression for x_u is:

$$x_u = (\alpha Y^T C^u Y + (1 - \alpha) Z^T C^{*u} Z + (\lambda + \alpha \lambda_f) I)^{-1} \left(\alpha \left(Y^T C^u p(u) + \lambda_f \frac{1}{|F(u)|} \sum_{f \in F(u)} \widehat{sim}(u,f) x_f \right) + (1 - \alpha) Z^T C^{*u} p^*(u) \right) \quad (15)$$

For the item factor y_i , it is the same as in Eq. (3), and for the group factor z_g , it is the same as in Eq. (10).

An alternative combination model that is without the similarity measure being integrated into the regularization of friendship is:

$$\alpha \min_{u,i} \sum_{u,i} c_{ui} (p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right) + \lambda_f \left(\left\| x_u - \frac{1}{|F(u)|} \sum_{f \in F(u)} x_f \right\|^2 \right) + (1 - \alpha) \min_{u,g} \sum_{u,g} c_{ug}^* (p_{ug}^* - x_u^T z_g)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_g \|z_g\|^2 \right) \quad (16)$$

The analytic expression for x_u to minimize the above cost function is hence:

$$x_u = (\alpha Y^T C^u Y + (1 - \alpha) Z^T C^{*u} Z + (\lambda + \alpha \lambda_f) I)^{-1} \times \left(\alpha \left(Y^T C^u p(u) + \lambda_f \frac{1}{|F(u)|} \sum_{f \in F(u)} x_f \right) + (1 - \alpha) Z^T C^{*u} p^*(u) \right) \quad (17)$$

4.5. Making top-N recommendation

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

$$p'_{ui} = x_u^T * y_i \quad (18)$$

where x_u^T and y_i are the user's latent factor model and the item's latent factor model respectively, obtained from the above described methods.

Top-N items with higher scores will then be included the recommendation list and returned to the target user.

5. Recommending groups: fusing friendship and user-item preferences

5.1. Baseline

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

$$\min_{u^*, g^*} \sum_{u, g} c_{ug}^* (p_{ug}^* - x_u^T z_g)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_g \|z_g\|^2 \right) \quad (19)$$

where p_{ug}^* equals 1 if the user u joined group g , otherwise it is 0; c_{ug}^* is the confidence level (see Table 2). The analytic expressions for x_u and z_g which are used to minimize the above cost function are respectively:

$$x_u = (Z^T C^{su} Z + \lambda I)^{-1} Z^T C^{su} p^*(u) \quad (20)$$

$$z_g = (X^T C^{sg} X + \lambda I)^{-1} X^T C^{sg} p^*(g) \quad (21)$$

The prediction score of a user's preference over an un-joined group can then be calculated through the inner product: $\hat{p}_{ug}^* = x_u^T z_g$.

5.2. Fusing friendship by regularization

To fuse friendship into the group recommendation, we empirically evaluated both regularization and factorization methods (see Section 8.2). Particularly, with the same consideration raised in Section 4, we propose the *similarity-integrated regularization model* to inject the friendship into group recommendation:

$$\min_{u^*, g^*} \sum_{u, g} c_{ug}^* (p_{ug}^* - x_u^T z_g)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_g \|z_g\|^2 \right) + \lambda_f \left(\left\| x_u - \frac{1}{|F(u)|} \sum_{f \in F(u)} \widehat{sim}(u, f) x_f \right\|^2 \right) \quad (22)$$

where $\widehat{sim}(u, f)$ has the same definition as in Eq. (4).

The analytic expression for x_u is

$$x_u = (Z^T C^{su} Z + (\lambda + \lambda_f) I)^{-1} \left(Z^T C^{su} p^*(u) + \frac{\lambda_f}{|F(u)|} \sum_{f \in F(u)} \widehat{sim}(u, f) x_f \right) \quad (23)$$

The analytic expression for the group factor z_g is the same as in Eq. (21).

To save space, the equation without the similarity integration is not listed here, but its general form can be referred to Eq. (6).

Besides, we implemented the *factorization-based* friendship fusion in group recommendation, which is similar to Eq. (24) (see next section), but with the user-user binary matrix as input (rather than the user-item matrix) where a cell is assigned '1' if two users are friends.

5.3. Fusing user-item preferences by factorization

As mentioned before, the user-item preferences in our system are inferred from the implicit data (e.g., the user's "clicking" behavior). Before, we have attempted to incorporate membership into the item recommendation (see Section 4.3). Viceversa, the user-item preferences could also be fused into the process of recommending groups. In this regard, we have developed both factorization and regularization models, but still placed more focus on the *factorization method* due to the "bipartite" data property of user-item relationship:

$$\alpha \min_{u^*, g^*} \sum_{u, g} c_{ug}^* (p_{ug}^* - x_u^T z_g)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_g \|z_g\|^2 \right) + (1 - \alpha) \min_{u^*, i^*} \sum_{u, i} c_{ui} (p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right) \quad (24)$$

The analytic expressions for x_u and y_i are respectively defined as:

$$x_u = (\alpha Z^T C^{su} Z + (1 - \alpha) Y^T C^i Y + \lambda I)^{-1} * (\alpha Z^T C^{su} p^*(u) + (1 - \alpha) Y^T C^i p(u)) \quad (25)$$

$$y_i = (X^T C^i X + \lambda I)^{-1} X^T C^i p(i) \quad (26)$$

The expression for z_g is the same as in Eq. (21).

Alternatively, the *regularization-based* fusion method converts the user-item relation into the user-user weighted relation:

$$\min_{u^*, g^*} \sum_{u, g} c_{ug}^* (p_{ug}^* - x_u^T z_g)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_g \|z_g\|^2 \right) + \lambda_n \left(\left\| x_u - \frac{1}{N(u)} \sum_{n \in N(u)} \omega_{un}^* * x_n \right\|^2 \right) \quad (27)$$

where the weight $w_{un}^* = \frac{|O_{un}|}{\sum_{i \in N(u)} |O_{ui}|}$ (for which 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 the above model is:

$$x_u = (Z^T C^{su} Z + (\lambda + \lambda_f) I)^{-1} \left(Z^T C^{su} p^*(u) + \lambda_n \frac{1}{|N(u)|} \sum_{n \in N(u)} \omega_{un}^* * x_n \right) \quad (28)$$

5.4. Fusing friendship and user-item preferences together

To assess the effect of combining the two information resources, i.e., friendship and user-item preferences, we have fused them together via the equation below:

$$\alpha \min_{u^*, g^*} \sum_{u, g} c_{ug}^* (p_{ug}^* - x_u^T z_g)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_g \|z_g\|^2 \right) + \lambda_f \left(\left\| x_u - \frac{1}{|F(u)|} \sum_{f \in F(u)} \widehat{sim}(u, f) x_f \right\|^2 \right) + (1 - \alpha) \min_{u^*, i^*} \sum_{u, i} c_{ui} (p_{ui} - x_u^T y_i)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right) \quad (29)$$

where the friendship is handled by the similarity-integrated regularization model and user-item preferences are handled via the factorization. This combination was actually devised, after we compared regularization with factorization in terms of fusing friendship and user-item preferences separately in the experiment (see Section 8.2). The analytic expression for x_u is

$$x_u = \left(\alpha Z^T C^{su} Z + (1 - \alpha) Y^T C^i Y + (\lambda + \alpha \lambda_f) I \right)^{-1} \times \left(\alpha \left(Z^T C^{su} p^*(u) + \frac{\lambda_f}{|F(u)|} \sum_{f \in F(u)} \widehat{sim}(u, f) x_f \right) + (1 - \alpha) Y^T C^i p(u) \right) \quad (30)$$

The analytic expression for z_g is the same as in Eq. (21), and for y_i it is the same as in Eq. (26).

Still, to save space, the equation without the similarity measure being integrated into the combination model is not listed here, but its general form can be referred to Eq. (16).

6. Recommending friends: fusing membership and user-item preferences

6.1. Baseline

As to friend recommendation, it is essentially more challenging and difficult than item and group recommendations, because there might be various reasons for two users to become friends. In our baseline method, we propose to add the regularization process into the basis matrix factorization due to the friendship's one mode property. Formally, the cost function is

$$\min_{u, f} \sum_{u, f} c'_{uf} \left(p'_{uf} - x_u^T x_f \right)^2 + \lambda \|x_u\|^2 + \lambda_f \left\| x_u - \frac{1}{|F(u)|} \sum_{f \in F(u)} x_f \right\|^2 \quad (31)$$

where the definitions of c'_{uf} and p'_{uf} can be seen in Table 2. In this equation, the first part shows the factorization and the second part gives the regularization. The analytic expression for x_u is

$$x_u = (X^T C^u X + (\lambda + \lambda_f) I)^{-1} \left(X^T C^u p'(u) + \frac{\lambda_f}{|F(u)|} \sum_{f \in F(u)} x_f \right) \quad (32)$$

The users with higher prediction scores computed from $\hat{p}'_{uf} = x_u^T x_f$ will then be recommended to the target user as her/his friend candidates.

6.2. Fusing user-item preferences by factorization

To fuse user-item preferences, we mainly exploit the factorization approach because the regularization was already embedded in the baseline Eq. (31). Thus, the cost function that is integrated with the factorization of item preferences is:

$$\min_{u, f, i} \sum_{u, f} c'_{uf} \left(p'_{uf} - x_u^T x_f \right)^2 + \lambda_f \|x_u\|^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right) + \alpha \min_{u, i} \sum_{u, i} c_{ui} \left(p_{ui} - x_u^T y_i \right)^2 \quad (33)$$

The analytic expressions for x_u and y_i are respectively defined as:

$$x_u = (X^T C^u X + \alpha Y^T C^u Y + (\lambda + \lambda_f) I)^{-1} \times \left(X^T C^u p(u)' + \alpha Y^T C^u p(u) + \lambda_f \frac{1}{|F(u)|} \sum_{f \in F(u)} x_f \right) \quad (34)$$

$$y_i = (X^T C^i X + \lambda I)^{-1} X^T C^i p(i) \quad (35)$$

6.3. Fusing membership by factorization

Similarly, we apply the factorization model for fusing membership. The cost function is hence:

$$\min_{u, f, g} \sum_{u, f} c'_{uf} \left(p'_{uf} - x_u^T x_f \right)^2 + \lambda_f \left\| x_u - \frac{1}{|F(u)|} \sum_{f \in F(u)} x_f \right\|^2 + \alpha \min_{u, g} \sum_{u, g} c_{ug}^* \left(p_{ug}^* - x_u^T z_g \right)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_g \|z_g\|^2 \right) \quad (36)$$

The analytic expressions for x_u and z_g are respectively:

$$x_u = (X^T C^u X + \alpha Z^T C^g Z + (\lambda + \lambda_f) I)^{-1} \times \left(X^T C^u p(u)' + \alpha Z^T C^g p^*(g) + \lambda_f \frac{1}{|F(u)|} \sum_{f \in F(u)} x_f \right) \quad (37)$$

Table 3
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

$$z_g = (X^T C^{*g} X + \lambda I)^{-1} X^T C^{*g} p^*(g) \quad (38)$$

6.4. Fusing membership and user-item preferences together

The membership and user-item preferences can be then fused simultaneously into the following framework:

$$\min_{u, f, i, g} \sum_{u, f} c'_{uf} \left(p'_{uf} - x_u^T x_f \right)^2 + \lambda_f \|x_u\|^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2 \right) + \alpha \min_{u, i} \sum_{u, i} c_{ui} \left(p_{ui} - x_u^T y_i \right)^2 + \beta \min_{u, g} \sum_{u, g} c_{ug}^* \left(p_{ug}^* - x_u^T z_g \right)^2 + \lambda \left(\sum_u \|x_u\|^2 + \sum_g \|z_g\|^2 \right) \quad (39)$$

where α and β are respectively used to adjust the relative weights of user-item preferences and the user-group membership. The analytic expression for x_u is concretely defined as:

$$x_u = (X^T C^u X + \alpha Y^T C^u Y + \beta Z^T C^{*g} Z + (\lambda + \lambda_f) I)^{-1} \times \left(X^T C^u p(u)' + \alpha Y^T C^u p(u) + \beta Z^T C^{*g} p^*(g) + \lambda_f \frac{1}{|F(u)|} \sum_{f \in F(u)} x_f \right) \quad (40)$$

The analytic expressions for y_i and z_g are the same as the ones in Eqs. (35) and (38) respectively.

7. Experiment setup

7.1. Dataset

Two real-world datasets, namely Last.fm² and Douban,³ were used to test the performance of the above-described algorithms. The Last.fm is a worldwide popular social music site. The *item* in this dataset is referred to the "artist" (because users' preference over artists can be more stable than their preference over songs). The *membership* indicates the user's participation in interest groups, and the *friendship* was extracted from the user's friend list. We first wrapped the user-item interaction data and social relations from Last.fm, and then randomly sampled 100 k users from the dataset to do the evaluation. Douban is a popular social media site in China that supports users to freely share movies, books and music. For the sake of simplicity, we collected users' data only related to movies in Douban. We treat the user-item interaction matrix as 0/1, that is, the cell 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 as friends only if they follow each other. The details of the two datasets are given in Table 3.

To test item recommendation methods, the user-item pairs were first divided into 10 subsets with equal sizes. Two subsets

² www.last.fm

³ www.douban.com

were then randomly chosen: one was the validation set for tuning the equations' parameters, and another was the testing set for testing the algorithm accuracy. Different combinations of remaining 8 subsets represent various levels of data density. For example, train.10 contains 10% user-item pairs of the total data, and train.20 contains 20% user-item pairs.

As for group and friend recommendation, we applied the *leave-one-out* evaluation scheme because user-group pairs and user-user friendship pairs are rather sparse so they cannot be divided into multiple subsets. For the same reason, there is no validation set in this case, because if some groups (or friends) were chosen as the validation set, many users will become isolated nodes. Therefore, according to the *leave-one-out* strategy, during each testing round, one of the user's participated groups (or connected friends) was randomly chosen as her/his target choice, and the evaluation was to assess whether this target choice is included in the top-N recommendation list or not.

7.2. Evaluation metrics

Regarding item recommendation, because we emphasize predicting users' interests in un-clicked items based on their *implicit clicking behavior*, the evaluation was done on the whole set of top-N recommendations (e.g., $N = 5, 10$), instead of for every recommended item. Accordingly, we chose recall and precision as evaluation metrics, rather than NDCG and RMSE (Shani & Gunawardana, 2011), because the latter two have been commonly used to assess explicit ratings, less for implicit, binary data. In addition, as pointed out by Bellogin, Castells, and Cantador (2011), the RMSE-related metrics showed a heavy bias towards known relevant items, that considerably overestimated the performance.

1. **Recall.** It measures the fraction of relevant items as from the user's testing set are recommended to the user. All users' recall values are then averaged to indicate the tested algorithm's overall performance, which is expected to be as high as possible to suggest good accuracy.

$$recall@N = \frac{1}{m} \sum_{u=1}^m \frac{hits(x_u)@N}{|T|} \quad (41)$$

where N is the size of the recommendation list, $|T|$ is the size of each user's testing set T , m is the total number of users, and $hits(x_u)@N$ gives the intersection between the recommendation list and the user x_u 's testing set.

2. **Precision.** This metric measures the fraction of recommended items that appear in the user's testing set. The average of all users' precision values is computed via the following equation:

$$precision@N = \frac{1}{m} \sum_{u=1}^m \frac{hits(x_u)@N}{N} \quad (42)$$

3. **F-measure.** In addition, we calculated F-measure which combines precision and recall through the harmonic mean. It can be interpreted as a weighted average of precision and recall.

$$F1@N = \frac{1}{m} \sum_{u=1}^m 2 * \left(\frac{hits(x_u)@N}{|T|} * \frac{hits(x_u)@N}{N} \right) / \left(\frac{hits(x_u)@N}{|T|} + \frac{hits(x_u)@N}{N} \right) \quad (43)$$

For evaluating group and friend recommendations, we mainly used the metric hit-ratio $Hits@N$ which gives the percentage of users whose target choice (i.e., a group or a friend as randomly withdrawn from the dataset) is located in the top-N recommendation list:

$$Hits@N = \frac{1}{m} \sum_{u=1}^m suc(x_u)@N \quad (44)$$

where $suc(x_u)@N$ is equal to 1 if the user u 's target choice was successfully recommended (otherwise it is 0).

8. Results analysis

In Table 4, we first list the abbreviations of all algorithms that have been described in Sections 4–6.

8.1. Item recommendation

In this section, we first summarize the results from a preliminary testing that was published in Yuan et al. (2011). We then report new results that we get from the current experiment that emphasizes revealing the specific effect of similarity-integrated methods and the combination models.

8.1.1. Summary of previous results

In a prior work with tentative testings (Yuan et al., 2011), we found that fusing friendship and/or membership acts quite active in sparse dataset (e.g., at train.10), but as the training data becomes denser (at train.40 and train.50), the effect tends to vanish. Our explanation for this phenomenon is that when the matrix is very sparse, auxiliary, heterogenous data resources like social relations can take complementary role to infer users' tastes. However, if the user-item matrix is dense enough, introducing auxiliary data resources may likely bring noises, instead of improving the recommendation performance. Thus, we set the data density level at train.10 (i.e., with 10% of user-item pairs) when measuring other fusion-based item recommendation algorithms. Previously, we also identified that fusing friendship/membership can help increase the recommendation accuracy relative to the baseline approach. For instance, in the Last.fm dataset, the accuracy can be improved up to 18.14% at recall@10.

We then compared the two models: *factorization* and *regularization*, in respect of their respective roles in fusing friendship (and membership) for item recommendation. Since two users may join the same group by accident, we only considered users who have at least 2 common groups with the target user, who form the set $N(u)$ in Eq. (12). Fig. 3 shows the performance comparison results. Each line represents the precision of the tested algorithm at a given recall. It can be seen from Fig. 3(a) that fusing friendship by regularization (Item.MF.F.R) is clearly better than by factorization (Item.MF.F.F), which proves our hypothesis that regularization model is more good at minimizing the gap between the taste of a user and the taste of her/his friends. However, for fusing the membership data, factorization outperforms regularization (as shown in Fig. 3(a)).

8.1.2. Impact of integrating similarity into friendship regularization

As the extension of previous work, in the current experiment, we have tested whether integrating the similarity measure in the friendship's regularization (as described in Section 4.2) could obviously increase its prediction power. Table 5 shows the results of comparing different similarity computations: Item.MF.F.ICos (based on common items), Item.MF.F.GCos (based on common groups) and Item.MF.F.FCos (based on common friends), respectively in Last.fm and Douban datasets. It can be seen that all of these similarity-integrated regularization methods perform better than the one without the similarity integration (i.e., Item.MF.F.R) in both datasets. It further shows that the best performance goes to Item.MF.F.GCos which calculates the similarity based on common groups as shared by the user and her/his friend (except that

Table 4
List of compared algorithms.

Abbreviation	Algorithm description	Details
<i>Recommending items</i>		
Item.MF	Basic matrix factorization	Section 4.1
Item.MF.F.R	Fusing the friendship by regularization	Section 4.2
Item.MF.F.F	Fusing the friendship by factorization	Section 4.3
Item.MF.M.R	Fusing the membership by regularization	Section 4.3
Item.MF.M.F	Fusing the membership by factorization	Section 4.3
Item.MF.FM	Fusing the Friendship by regularization and fusing the Membership by factorization	Section 4.4
Item.MF.F.CCos	Fusing the friendship by similarity-integrated regularization based on common friends	Section 4.2
Item.MF.F.GCos	Fusing the friendship by similarity-integrated regularization based on common groups	Section 4.2
Item.MF.F.ICos	Fusing the friendship by similarity-integrated regularization based on common items	Section 4.2
Item.MF.FM.GCos	Fusing the friendship by Item.MF.F.GCos and fusing the membership by factorization	Section 4.4
<i>Recommending groups</i>		
Group.MF	Basic matrix factorization	Section 5.1
Group.MF.F.R	Fusing the friendship by regularization	Section 5.2
Group.MF.F.F	Fusing the friendship by factorization	Section 5.2
Group.MF.I.R	Fusing the user-item preferences by regularization	Section 5.3
Group.MF.I.F	Fusing the user-item preferences by factorization	Section 5.3
Group.MF.FI	Fusing the friendship by regularization and fusing the user-item preferences by factorization	Section 5.4
Group.MF.F.CCos	Fusing the friendship by similarity-integrated regularization based on common friends	Section 5.2
Group.MF.F.GCos	Fusing the friendship by similarity-integrated regularization based on common groups	Section 5.2
Group.MF.F.ICos	Fusing the friendship by similarity-integrated regularization based on common items	Section 5.2
Group.MF.FI.GCos	Fusing the friendship by Group.MF.F.GCos and fusing the user-item preference by factorization	Section 5.4
<i>Recommending friends</i>		
Friend.MF	Basic matrix factorization	Section 6.1
Friend.MF.M.F	Fusing the membership by factorization	Section 6.3
Friend.MF.I.F	Fusing the user-item preferences by factorization	Section 6.2
Friend.MF.MI	Fusing the membership and user-item preferences together	Section 6.4

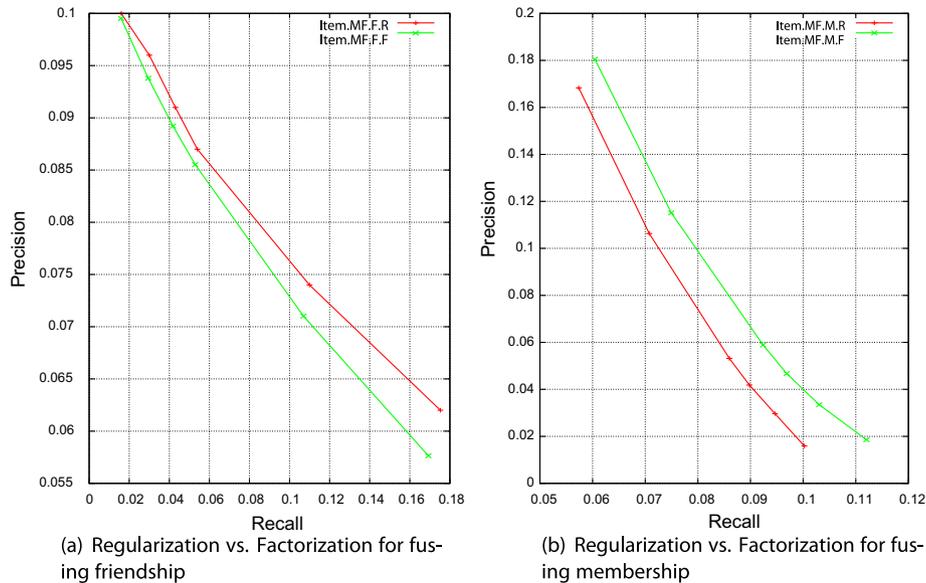


Fig. 3. Regularization vs. factorization (regarding item recommendation) in Last.fm dataset.

the precisions of Item.MF.F.ICos are better than ones of Item.-MF.F.GCos in Douban). It thus infers that the group information might be a better indicator (given that users usually join in a group based on their common interests) and be less noisy when being used to compute the similarity.

On the other hand, though the similarity measure can boost the friendship’s fusion power, the fusion of membership outperforms its best result (i.e., Item.MF.M.F against Item.MF.F.GCos; see Table 5), indicating that membership is more effective than friendship for augmenting the item recommendation’s accuracy.

8.1.3. Fusing friendship and membership together

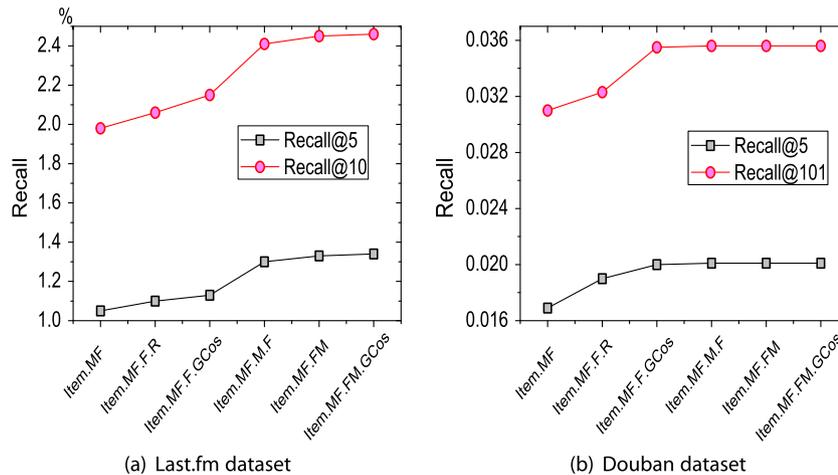
The reported results by far suggest that no matter which type of social relation it is, the performance of fusing it is better than the

baseline Item.MF which is without any data fusions. Driven by these results, we further combined both types of social relationships and incorporated them together into the process of recommending items (as described in Section 4.4). The results (see Table 5 and Fig. 4) show that the combination method did outperform the one that fuses friendship or membership alone. For instance, in Last.fm dataset, the best result from the combination (Item.MF.FM) increased the recall@5 by 26.67% (relative to the baseline), while fusing friendship alone can only achieve maximally 6.67% improvement, and fusing membership alone raises the recall@5 up to 23.80% (which is still lower). According to this observation, we believe that membership and friendship can be complementary to each other for resolving the data sparsity problem (given the reported accuracy values all at train.10). Moreover,

Table 5

Results w.r.t. recommending ITEMS through the validation (Prec: Precision; Rec: Recall; F1: F-measure).

Method	Prec@5	Prec@10	Rec@5	Rec@10	F1@5	F1@10
<i>Last.fm dataset</i>						
Item.MF (baseline)	0.0547	0.0522	0.0105	0.0198	0.0161	0.0259
Item.MF.F.R ($\lambda_f = 1$)	0.0570	0.0540	0.0110	0.0206	0.0168	0.0268
Item.MF.F.FCos ($\lambda_f = 10$)	0.0580	0.0557	0.0112	0.0212	0.0171	0.0277
Item.MF.F.GCos ($\lambda_f = 10$)	0.0581	0.0561	0.0112	0.0214	0.0171	0.0279
Item.MF.F.ICos ($\lambda_f = 10$)	0.0581	0.0560	0.0112	0.0213	0.0171	0.0278
Item.MF.M.F ($\alpha = 0.1$)	0.0654	0.0615	0.0130	0.0240	0.0196	0.0309
Item.MF.FM ($\alpha = 0.2, \lambda_f = 10$)	0.0659	0.0616	0.0133	0.0245	0.0199	0.0312
Item.MF.FM.GCos ($\alpha = 0.1, \lambda_f = 10$)	0.0672	0.0624	0.0134	0.0246	0.0202	0.0314
<i>Douban dataset</i>						
Item.MF (baseline)	0.0368	0.0345	0.0169	0.0310	0.0171	0.0237
Item.MF.F.R ($\lambda_f = 100$)	0.0419	0.0380	0.0190	0.0323	0.0193	0.0254
Item.MF.F.FCos ($\lambda_f = 100$)	0.0420	0.0373	0.0202	0.0355	0.0199	0.0261
Item.MF.F.GCos ($\lambda_f = 100$)	0.0422	0.0383	0.0203	0.0355	0.0200	0.0262
Item.MF.F.ICos ($\lambda_f = 100$)	0.0424	0.0388	0.0191	0.0334	0.0195	0.0261
Item.MF.M.F ($\alpha = 0.1$)	0.0421	0.0381	0.0201	0.0356	0.0198	0.0262
Item.MF.FM ($\alpha = 0.2, \lambda_f = 100$)	0.0427	0.0383	0.0201	0.0356	0.0200	0.0265
Item.MF.FM.GCos ($\alpha = 0.1, \lambda_f = 100$)	0.0429	0.0388	0.0201	0.0356	0.0201	0.0267

Note: λ is set as 10, and the size of user/item latent factors (k) is 10. λ_f and α were optimally tuned via the validation set.**Fig. 4.** Algorithm comparison w.r.t. recommending ITEMS.

the combination model that is integrated with similarity-enhanced regularization (i.e., Item.MF.FM.GCos) can further increase the accuracy (e.g., recall@5 with 27.62% improvement than baseline, versus 26.67% from Item.MF.FM which is without the similarity integration). In Douban dataset, because the number of groups is much less than in Last.fm, the differences between the compared algorithms are not so obvious than in Last.fm dataset, but still, the combination of two relational resources (Item.MF.FM.GCos) can achieve the highest accuracy, in comparison to the separate fusions (i.e., Item.MF.M.F and Item.MF.F.GCos). It thus implies that when the friendship's regularization is further improved by adding the common-group based similarity measure, it can be more effectively combined with the factorization of membership to enable the item recommendation to reach at a higher level of accuracy.

8.2. Group recommendation

In terms of group recommendation, we compared in total ten methods in the experiment (see Table 4). Regarding the comparison of the two fusion models: *regularization* and *factorization*, we first tested their respective roles in injecting user-item preferences into the process of recommending groups (see results in Table 6).

Specifically, the results of using *factorization* model (Group.MF.I.F) at different data density levels indicate that the recommendation accuracy can be improved with the increase of density level. That is, the denser that user-item preferences are when being fused into the factorization model, the more accurate the group recommendation is. For example, in Douban dataset, the Hits@10 was increased from 2950 (at train.20) to 3095 (at train.80), which are both higher than the one of baseline (Group.MF, that is without any auxiliary resources' fusion). Relatively, the accuracy of *regularization* model for fusing user-item preferences (Group.MF.I.R) is lower, and does not obviously change when the data density level is increased. This might be because once the user-item matrix is projected into the user-user matrix, a lot of information is lost, so even with denser user-item matrix, the algorithm's performance cannot be clearly improved. It hence verifies our hypothesis again that the *factorization* model better suits bipartite data, since this was not only proven in dealing with membership (for item recommendation), but also in injecting user-item preferences (for group recommendation).

As for fusing friendship, the comparison of regularization and factorization (i.e., Group.MF.F.R vs. Group.MF.F.F) shows that the former approach outperforms the latter (.0910 against .0876 w.r.t.

Table 6
Results w.r.t. recommending GROUPS.

Method	Last.fm		Douban	
	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.

Hits@10 in Last.fm, and .2973 vs. .2928 in Douban), which reveals the merit of the *regularization* model in fusing the one mode data. We also tested the performance when the similarity measure between users was integrated into the regularization of friendship. Being consistent to its effect on augmenting item recommendation, the similarity-integrated regularization methods are more accurate than the non-similarity based one (Group.MF.F.R), and the similarity computed with users' common groups (Group.MF.F.GCos) is better than the other similarity measures.

From the resource's perspective, we compared user-item preferences and friendship in respect of their fusion effects. It shows that the user-item preferences act much more effective: its maximal improvement is 36.2% (relative to the baseline, w.r.t. Hits@5 in Last.fm) while the friendship's best improvement is 11.9%. Actually, all variations of fusing item-preferences via factorization (from train.40 to train.80) obtain better outcomes than the ones of fusing friendship (the detailed comparisons can be seen in Table 6).

Motivated by the above comparison results, we finally tested the combination of Group.MF.F.GCos and Group.MF.I.F@train.80 (which is shorted as Group.MF.FI.GCos), in order to fuse the two resources (friendship and user-item preferences) together (as described in Section 5.4). From Fig. 5, it can be seen that this

combination achieves higher accuracy than fusing the two resources separately. Moreover, Group.MF.FI.GCos is better than an alternative combination model Group.MF.FI (which is without the common-group based similarity integration), which suggests that it gives the ideal combination since it accommodates all merits as derived from both the regularization of friendship and the factorization of user-item preferences.

8.3. Friend recommendation

As described in Section 6, we have applied the matrix factorization technique to generate friend recommendation, and attempted to fuse other information resources (i.e., membership and user-item preferences) in order to identify their impact. Given the superior performance of regularization model for handling friendship when it was fused to generate item (and group) recommendation, we added the regularization process as a basis part of the baseline method (Friend.MF), and then compared it to various fusion methods that take into account either membership, user-item preferences, or both (see Table 4).

Table 7 shows the comparison results. As for the fusion of user-item preferences, we tested the influence of data density level on algorithm's performance. It surprisingly shows that when the

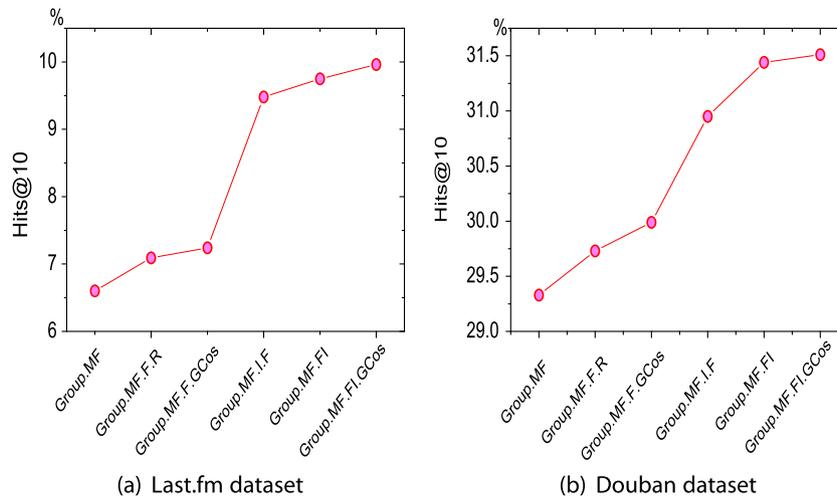


Fig. 5. Algorithm comparison w.r.t. recommending GROUPS.

Table 7
Results w.r.t. recommending FRIENDS.

Method	Last.fm		Douban	
	Hits@5	Hits@10	Hits@5	Hits@10
Friend.MF (baseline)	0.0155	0.0203	0.0112	0.0142
<i>Fusing user-item preferences</i>				
Friend.MF.I.F@train.10	0.0159	0.0212	0.0111	0.0140
Friend.MF.I.F@train.20	0.0157	0.0209	0.0110	0.0140
Friend.MF.I.F@train.40	0.0152	0.0201	0.0111	0.0140
Friend.MF.I.F@train.60	0.0152	0.0201	0.0112	0.0140
Friend.MF.I.F@train.80	0.0151	0.0199	0.0111	0.0140
<i>Fusing membership, and both resources</i>				
Friend.MF.M.F	0.0163	0.0218	0.0117	0.0140
Friend.MF.M.I	0.0159	0.0209	0.0111	0.0140

Note: the size of user/item latent factors (k) is set as 50.

user-item preference data become denser, the accuracy of Friend.MF.I.F (that fuses user-item preferences) is decreased, implying that the fusion of more user-item pairs does not obviously help augment the accuracy. Actually, when the density level reaches at train.40 in Last.fm, the accuracy is lower than the one of baseline (Friend.MF). The phenomena are even worse in Douban dataset, given that the hit ratio of Friend.MF.I.F@train.10 is not better than the Friend.MF's.

With respect to the fusion of membership, Friend.MF.M.F achieves higher accuracy (.0218 w.r.t. Hits@10 in Last.fm) compared to both the baseline Friend.MF and Friend.MF.I.F@train.10, indicating that the membership could be potentially more useful than user-item preferences in terms of benefiting the friend recommendation. The similar improvement occurs in Douban, but only to Hits@5, which may be caused by the implicit friendship that we crawled from the two-way follower/followee relationship (so it is not stable as the friend list that the user created).

Another finding is that the combination approach (Friend.MF.M.I, that fuses user-item preferences @train.10 and membership together) is found not outperforming Friend.MF.M.F (that only fuses membership) in both datasets. This finding implies that fusing membership alone might be enough to accomplish the goal of enhancing friend recommendation, given that the user-item preference might bring some noises and negative influences when it is fused together with the membership. In this experiment, we also tested the effect of integrating similarity measure into the regularization of friendship, but since the performance was not obviously improved, the results are not shown in the table. The implication is then that, for the friend recommendation, attaching the similarity degree as the weight to the target user's friends might not be an effective way to find more relevant friends for the user. However, as shown in previous sections, the similarity measure can help adjust the friends' respective contributions (especially based on common groups) when it is utilized to predict the user's preference over items or groups.

9. Conclusions

In conclusion, this paper presents a unified framework that recommends items, groups and friends in a single system by examining their mutual contributions. Below we summarize the major findings of the work.

- **Recommending items:** (1) with the implicit users' interaction data, we proved that the social relations can be helpful to boost top-N recommendation accuracy, especially in very sparse dataset. It hence suggests that fusing friendship and membership data can well address the cold-start and sparsity problem in the implicit data condition. (2) To the best of our knowledge, this work is the first one that in-depth explored the specific role

of membership and revealed that it performs more effective than friendship in augmenting the item recommendation. (3) We also proved that the system performance can be further enhanced when membership and friendship are combined via the collective factorization mechanism.

- **Recommending groups:** (1) both user-item preferences and friendship exhibit positive effects on increasing the accuracy. (2) The user-item preferences were shown more effective than friendship, especially when higher density level of user-item preference data was contributed. (3) The two resources' combination achieves even higher accuracy than fusing either of them alone.
- **Recommending friends:** membership acts more positive for improving the friend recommendation accuracy, which is even more accurate than the combination model.

From the perspective of algorithm design, two constructive implications can be concluded:

- **Regularization vs. factorization:** through comparing the two models in the cases of recommending items and groups respectively, we found that the factorization model shows better performance when fusing bipartite data (such as user-group membership and user-item preferences), while the regularization model better suits one mode data (such as user-user friendship). It thus demonstrates the two models' respective merits and suggests that it should be contingent on the involved data's property when choosing the appropriate fusion model.
- **Integrating similarity measure:** another finding that can also be suggestive to related researchers is that the similarity measure that we added into the friendship's regularization was proven taking active role. Particularly, the similarity computation based on user-friend common groups can obviously enable both item and group recommendations' accuracy to reach at an upper level of accuracy.

Thus, by means of exploiting the *mutual contributions* among different heterogenous information resources, we identify how to benefit the three types of recommendation (items, groups, and friends) *simultaneously*. The work has practical meaning to the current online social media, in terms of meeting users' increasing needs of receiving different kinds of recommendation in a single platform. In the future, on one hand, we will continue to find real-scenario datasets to test the algorithms. On the other hand, we will endeavor to further improve the friend recommendation's accuracy. As a matter of fact, the limited improvement implies that there is still room to improve the matrix factorization based methods by adopting the advantages of other techniques, such as link prediction (Backstrom & Leskovec, 2011).

Acknowledgment

This research was supported by Hong Kong Baptist University Faculty Research Grant (HKBU/FRG2/11-12/049).

References

- Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transaction on Knowledge and Data Engineering*, 17, 734–749.
- Backstrom, L., & Leskovec, J. (2011). Supervised random walks: Predicting and recommending links in social networks. In *Proceedings of the fourth ACM international conference on web search and data mining (WSDM'11)* (pp. 635–644). ACM.
- Baltrunas, L., Makcinskas, T., & Ricci, F. (2010). Group recommendations with rank aggregation and collaborative filtering. In *Proceedings of the fourth ACM conference on recommender systems (RecSys'10)* (pp. 119–126). ACM.

- Bellogin, A., Castells, P., & Cantador, I. (2011). Precision-oriented evaluation of recommender systems: An algorithmic comparison. In *Proceedings of the fifth ACM conference on recommender systems (RecSys'11)* (pp. 333–336). ACM.
- Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, 3, 993–1022.
- Chen, J., Geyer, W., Dugan, C., Muller, M., & Guy, I. (2009a). Make new friends, but keep the old: Recommending people on social networking sites. In *Proceedings of the 27th international conference on human factors in computing systems (CHI'09)* (pp. 201–210). ACM.
- Chen, W.-Y., Chu, J.-C., Luan, J., Bai, H., Wang, Y., & Chang, E. Y. (2009b). Collaborative filtering for orkut communities: discovery of user latent behavior. In *Proceedings of the 18th international conference on world wide web (WWW'09)* (pp. 681–690). ACM.
- Chen, W.-Y., Zhang, D., & Chang, E. Y. (2008a). Combinational collaborative filtering for personalized community recommendation. In *Proceedings of the 14th ACM SIGKDD international conference on knowledge discovery and data mining (KDD'08)* (pp. 115–123). ACM.
- Chen, Y.-L., Cheng, L.-C., & Chuang, C.-N. (2008b). A group recommendation system with consideration of interactions among group members. *Expert Systems with Applications*, 34, 2082–2090.
- De Meo, P., Nocera, A., Terracina, G., & Ursino, D. (2011). Recommendation of similar users, resources and social networks in a social internet working scenario. *Information Sciences*, 181, 1285–1305.
- Fouss, F., Pirotte, A., Renders, J.-M., & Saerens, M. (2007). Random-walk computation of similarities between nodes of a graph with application to collaborative recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 19, 355–369.
- Gou, L., You, F., Guo, J., Wu, L., & Zhang, X. L. (2011). Sfviz: Interest-based friends exploration and recommendation in social networks. In *Proceedings of the 2011 visual information communication – international symposium (VINCI'11)* (pp. 15:1–15:10). ACM.
- Groh, G., & Ehmgig, C. (2007). Recommendations in taste related domains: Collaborative filtering vs. social filtering. In *Proceedings of the 2007 international ACM conference on supporting group work (Group'07)* (pp. 127–136). ACM.
- Guy, I., Ronen, I., & Wilcox, E. (2009). Do you know? Recommending people to invite into your social network. In *Proceedings of the 14th international conference on intelligent user interfaces (IUI'09)* (pp. 77–86). ACM.
- Hofmann, T. (2004). Latent semantic models for collaborative filtering. *ACM Transaction on Information Systems*, 22, 89–115.
- Hu, Y., Koren, Y., & Volinsky, C. (2008). Collaborative filtering for implicit feedback datasets. In *Proceedings of the 2008 eighth IEEE international conference on data mining (ICDM'08)* (pp. 263–272). IEEE Computer Society.
- Jamali, M., & Ester, M. (2009). Trustwalker: A random walk model for combining trust-based and item-based recommendation. In *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 397–406). ACM.
- Jamali, M., & Ester, M. (2010). A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on recommender systems (RecSys'10)* (pp. 135–142). ACM.
- Jamali, M., Huang, T., & Ester, M. (2011). A generalized stochastic block model for recommendation in social rating networks. In *Proceedings of the fifth ACM conference on recommender systems (RecSys'11)* (pp. 53–60). ACM.
- Konstas, I., Stathopoulos, V., & Jose, J. M. (2009). On social networks and collaborative recommendation. In *Proceedings of the 32nd international ACM SIGIR conference on research and development in information retrieval (SIGIR'09)* (pp. 195–202). ACM.
- Koren, Y., Bell, R., & Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42, 30–37.
- Liu, F., & Lee, H. J. (2010). Use of social network information to enhance collaborative filtering performance. *Expert Systems with Applications*, 37, 4772–4778.
- Ma, H., Yang, H., Lyu, M. R., & King, I. (2008). Sorec: Social recommendation using probabilistic matrix factorization. In *Proceeding of the 17th ACM conference on information and knowledge management (CIKM'08)* (pp. 931–940). ACM.
- Ma, H., Zhou, D., Liu, C., Lyu, M. R., & King, I. (2011). Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on web search and data mining (WSDM'11)* (pp. 287–296). ACM.
- Pazzani, M. J., & Billsus, D. (2007). Content-based recommendation systems. In P. Brusilovsky, A. Kobsa, & W. Nejdl (Eds.), *The adaptive web* (pp. 325–341). Berlin, Heidelberg: Springer-Verlag.
- Quercia, D., & Capra, L. (2009). Friendsensing: Recommending friends using mobile phones. In *Proceedings of the third ACM conference on recommender systems (RecSys'09)* (pp. 273–276). ACM.
- Rennie, J. D. M., & Srebro, N. (2005). Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on machine learning (ICML'05)* (pp. 713–719). ACM.
- Salakhutdinov, R., & Mnih, A. (2008a). Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on machine learning (ICML'08)* (pp. 880–887). ACM.
- Salakhutdinov, R., & Mnih, A. (2008b). Probabilistic matrix factorization. In *Advances in neural information processing systems* (Vol. 20).
- Shani, G., & Gunawardana, A. (2011). Evaluating recommendation systems. In *Recommender systems handbook* (pp. 511–546). Springer.
- Singh, A. P., & Gordon, G. J. (2008). Relational learning via collective matrix factorization. In *Proceeding of the 14th ACM SIGKDD international conference on knowledge discovery and data mining (KDD'08)* (pp. 650–658). ACM.
- Spertus, E., Sahami, M., & Buyukkotken, O. (2005). Evaluating similarity measures: A large-scale study in the orkut social network. In *Proceedings of the 11th ACM SIGKDD international conference on knowledge discovery in data mining (KDD'05)* (pp. 678–684). ACM.
- Symeonidis, P., Tiakas, E., & Manolopoulos, Y. (2010). Transitive node similarity for link prediction in social networks with positive and negative links. In *Proceedings of the fourth ACM conference on recommender systems (RecSys'10)* (pp. 183–190). ACM.
- Vasuki, V., Natarajan, N., Lu, Z., & Dhillon, I. S. (2010). Affiliation recommendation using auxiliary networks. In *Proceedings of the fourth ACM conference on recommender systems (RecSys'10)* (pp. 103–110). ACM.
- Yang, B., Lee, S., Park, S., & Lee, S.-g. (2012). Exploiting various implicit feedback for collaborative filtering. In *Proceedings of the 21st international conference companion on world wide web (WWW'12 companion)* (pp. 639–640). ACM.
- Yang, S.-H., Long, B., Smola, A., Sadagopan, N., Zheng, Z., & Zha, H. (2011). Like like alike: Joint friendship and interest propagation in social networks. In *Proceedings of the 20th international conference on world wide web (WWW'11)* (pp. 537–546). ACM.
- Yuan, Q., Chen, L., & Zhao, S. (2011). Factorization vs. regularization: Fusing heterogeneous social relationships in top-n recommendation. In *Proceedings of the fifth ACM conference on recommender systems (RecSys'11)* (pp. 245–252). New York, NY, USA: ACM.
- Yuan, Q., Chen, L., & Zhao, S. (2012). Augmenting collaborative recommenders by fusing social relationships: Membership and friendship. In *Recommender systems for the social web* (pp. 159–175). Berlin, Heidelberg: Springer-Verlag.
- Zheng, Y., Zhang, L., Ma, Z., Xie, X., & Ma, W.-Y. (2011). Recommending friends and locations based on individual location history. *ACM Transactions on the Web*, 5, 1–44.