Heterogeneous Data Fusion via Matrix Factorization for Augmenting Item, Group and Friend Recommendations

Wei Zeng[a,b] and Li Chen[a]
[a]Department of Computer Science
Hong Kong Baptist University, Hong Kong
[b]School of Computer Science and Engineering
University of Electronic Science and Technology, China
zwei504@gmail.com,lichen@comp.hkbu.edu.hk

ABSTRACT

Up to now, more and more social media 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 interest groups that include people with common tastes. Therefore, in this environment, users might be interested in not only receiving item recommendations, but also friend recommendations whom they might consider putting in the contact list, and group recommendations that they may consider joining in. To support such needs, in this paper, we propose a generalized 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 investigated the algorithm impact of fusing other two information resources (e.g., user-item preferences and friendship to be fused for recommending groups), along with their combined effect. The experiment reveals the ideal fusion mechanism for this multi-output recommender, and validates the benefit of factorization model for fusing bipartite data (such as membership and user-item preferences) and the benefit of regularization model for fusing one mode data (such as friendship). Moreover, the positive effect of integrating similarity measure into the regularization model is identified via the experiment.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Retrieval and Search—*Information Filtering*

General Terms

Algorithms, Experimentation

Keywords

Item recommendation, group recommendation, friend recommendation, matrix factorization, regularization

1. INTRODUCTION

In recent years, recommender systems have been widely developed with the purpose of effectively supporting users' decision-making process 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 any more. For example, in Last.fm which is a popular music sharing site, people should be interested in not only receiving music recommendations, but also friend recommendations (whom they might consider putting in the contact list) and interest-group recommendations (that they might consider joining in). Previously, we have investigated how friendship and membership can benefit the process of generating item recommendations when they were fused [13]. In this paper, we have extended this work and aimed at providing a generalized framework for generating multiple types of recommendation in a single system: recommending items, friends and groups. That is, not only friendship and membership are exploited to enhance *item recommendation*, but also users' existing item preferences plus friendship info are applied to potentially benefit *group recommendation*, and the combination of item preferences and membership could be further utilized to augment *friend recommendation*. Ideally, under the matrix factorization framework, we have compared various fusion approaches and in-depth explored how the three types of data could be mutually contributive to each other. Figure 1 shows the triangle relationship that we have mainly studied for realizing the following three types of recommendation.

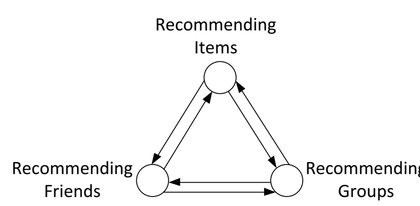

Figure 1: Triangle relationship among item, friend and group recommendations.

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. Considering that there were few works that fused membership and friendship into making item recommendation, we have previously studied their respective and combined roles, and identified their significant fusion effects on addressing the rating sparsity problem [13]. Extending our prior work, in this paper, we have integrated the similarity measure between users based on their common friends, items or groups, with the fusion framework. Our experimental results indicate that taking into account of the similarity can provide better item recommendation. The user-user similarity based on common groups is additionally found more effective than other measures especially in the regularization model.

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 may be interested in joining. Membership essentially involves two types of entities: users and groups, which reflect users' participation in groups based on their common tastes. To the best of our knowledge, few works have explored the algorithm impact of fusing both item preference and friendship on enhancing the accuracy of group recommendation. Thus, to compensate for their limitation, we not only investigated the two resources' respective impacts, but also their combined effect. The experimental results reveal the superior performance from fusing friends via the similarity-enhanced regularization mechanism, and the effect of the collective factorization model for fusing item preferences. More notably, their combination enables even higher accuracy.

Recommending friends. The third issue that we have attempted to resolve is to enhance friend recommendation by incorporating item preferences and membership data together. As a matter of fact, recently, there are increasing efforts devoted to generating the friend recommendation due to the increasing popularity of social networking [12, 7]. However, few attentions have been paid to increase the algorithm's accuracy by considering other information resources (like user-item preferences and membership). Moreover, the related works have been mainly done with graph-based methods (e.g., the link prediction based on random walk) which are nevertheless with high time complexity. Since matrix factorization (MF) has the inherent advantage of reducing the algorithm's cost and advancing the accuracy [5], we have compared variant friend recommendation methods under MF. It is experimentally found that both user-item preference data and user-group membership can help increase its recommendation accuracy, and the latter resource is shown more effective for achieving this goal.

2. RELATED WORK

To the best of our knowledge, few works have combined the three types of recommendations altogether in a generalized framework. Thus, we mainly introduce the state-of-the-art on each topic.

For the item recommendation, because traditional user-based and item-based Collaborative Filtering (CF) algorithms suffer from sparsity and imbalance of rating data, researchers have attempted to incorporate other kinds of data resources, such as user-user friendship network. For instance, Konstas et al. adopted Random Walk with Restart to model the friendship and social annotation (tagging) in a music track recommender system [8]. In [12], a friendship based interest propagation (FIP) framework was proposed which devised a factor-based random walk model to recommend both online services and friends to users. In [7], authors proposed a generalized stochastic block model (GSBM) based on membership stochastic block model which models not only the social relations but also the rating behavior. Both the ratings of items and the friendship can be predicted by the GSBM. In [9], a matrix factorization framework that incorporated the social network information via regularization was developed. Given that little attention has been paid to study the role of membership data, we have previously proposed to utilize membership, as another auxiliary information, to boost the item recommendation accuracy [13].

Regarding group recommendation (or called affiliation or community recommendation), there are relatively fewer efforts spent than for item recommendation. In [11], 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 both their friendship and affiliation networks. The results indicated that GPM turns out to be more effective and efficient. In [3], authors investigated two approaches to personalized community recommendation: the first adopted the Association Rule Mining technique (ARM) to discover associations between sets of communities that are shared across many users; the second was based on Latent Dirichlet Allocation (LDA) to model user-community co-occurrences using latent aspects. The experiment on Orkut data indicated that LDA consistently outperforms ARM when recommending 4 or more communities, while ARM is slightly better when recommending up to 3 communities. However, these works did not in depth explore the value of other resources, especially user-item preferences, for enhancing the group recommendation.

Compared to item and group recommendations, friend recommendation is a more challenging issue because there might be various reasons for two persons to become friends. In [2], authors found that algorithms based on social network information can reveal more known contacts for users, while algorithms that considered the similarity of user-created content were more useful in discovering new friends. Guy et al. 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 [4]. Their evaluation showed that people recommendation can be highly effective in increasing the number of connections. In [10], a so called FriendTNS algorithm was proposed, that recommends new friends to registered users based on both local and global graph features. In [1], friend prediction problem was re-formulated as the link prediction problem, for which an algorithm based on supervised random walk was developed.

From surveying related literatures, we found that few researches actually examined the mutual contributions among item, group and friend recommendations. Motivated by our prior work on fusing social relations into recommending items, we are interested in extending the work to provide different types of recommendation (i.e., groups and friends, in addition to items) in a single system. For example, when producing group recommendation, as users' preferences over items could be a strong indicator of their common tastes, it should be interesting to study their particular impact. The same study could be performed by fusing them to enhance friend recommendation. We have thus be driven to develop more effective and generalized fusion mechanism.

3. DATA PROPERTIES

Given a system like Last.fm, there are three available types of data: 1) the user-item interaction data (e.g., implicit binary data in Last.fm where 1 means that users clicked the item, and 0 otherwise), 2) the user-user friendship, and 3) the user-group membership. Both the user-item preferences and the user-group membership are *bipartite data*, because there are two types of entities involved in each relationship. The user-user friendship is *one mode data* since only one type of entity (i.e., the user) exists.

For the one mode data, some works utilized regularization model in order to minimize the gap between two entities with the same type [9, 6], but they did not prove whether the regularization is better than factorization. For the bipartite data, inspired by our prior work [13], we argue that the factorization model is more suitable, because it could effectively factorize user-item (or user-group) relations into two components and obtain a user's latent factor model and an item's (or a group's) latent factor model. Previously, we discussed the limitation if bipartite data were handled in the manner of regularization. We also experimentally proved that when fusing membership into item recommendation, the collective matrix factorization approach was more effective than the regularization, but regularization was better than factorization when handling friendship data [13]. In this paper, we aim to further validate the prior observations in the other recommending conditions, such as the group recommendation. Table 1 first summarizes major notations used in the following sections.

Table 1: Notations used in this paper

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 $C^u_{ii} = c_{ui}$; C^{*u} denotes the $l \times l$ diagonal matrix and $C^{*u}_{gg} = c^*_{ug}$; C'^u denotes the $m \times m$ diagonal matrix and $C'^u_{ff} = c'_{uf}$
$F(u)$	the friend set of user u
λ_f	the coefficient of the regularization
α, β	the coefficients for the collective matrix factorization

4. RECOMMENDING ITEMS

In this part, we mainly introduce how we fused the friendship and membership into the process of recommending items (note that the baseline without any fusions can be referred to [13]). Concretely, to fuse friendship, we leveraged a regularization model:

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

where p_{ui} measures user u's preference on item i. For example, with Last.fm dataset, if user u "clicked" this item, $p_{ui} = 1$, otherwise $p_{ui} = 0$; c_{ui} is the confidence level indicating how much a user prefers an item (which is set as 1 if no relevant data like "duration" or "frequency" are available). In the above formula, friendship is considered as an indicator of closeness. The regularization model hence aims to minimize the gap between the taste of a user and the average of her/his friends. We adopted alternating-least-squares (ALS) for the optimization process. The analytic expressions for x_u and y_i that minimize the cost function are:

$$x_u = (Y^T C^u Y + (\lambda + \lambda_f)I)^{-1}(Y^T C^u p(u) \quad (2)$$
$$+\lambda_f \frac{1}{| F(u) |} \sum_{f \in F(u)} x_f)$$

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

where C^i denotes the $m \times m$ diagonal matrix and $C^i_{uu} = c_{ui}$, and the vector $p(i)$ contains all user preferences related to i.

Regarding membership, because it is a sort of bipartite data as indicated before, we developed the collective matrix factorization (CMF):

$$\alpha \min_{u*,i*} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda(\sum_u \| x_u \|^2 + \sum_i \| y_i \|^2)+$$
$$(1-\alpha) \min_{u*,g*} \sum_{u,g} c^*_{ug}(p^*_{ug} - x_u^T z_g)^2 + \lambda(\sum_u \| x_u \|^2 + \sum_g \| z_g \|^2) \quad (4)$$

where α is used to adjust the relative weight of user-item matrix and user-group matrix in the factorization model. The analytic expressions for x_u and z_g are:

$$x_u = (\alpha Y^T C^u Y + (1-\alpha) Z^T C^{*u} Z + \lambda I)^{-1} * \quad (5)$$
$$(\alpha Y^T C^u p(u) + (1-\alpha) Z^T C^{*u} p^*(u))$$

$$z_g = (X^T C^{*g} X + \lambda I)^{-1} X^T C^{*g} p^*(g) \quad (6)$$

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

To combine friendship and membership together in an unified fusion framework, we proposed the following equation that concretely deals with friendship via the regularization and membership via CMF:

$$\alpha \min_{u*,i*} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda(\sum_u \| x_u \|^2 + \sum_i \| y_i \|^2) + \lambda_f$$
$$(\| x_u - \frac{1}{|F(u)|} \sum_{f \in F(u)} x_f \|^2) + (1-\alpha) \min_{u*,g*} \sum_{u,g} c^*_{ug}(p^*_{ug} - x_u^T z_g)^2$$
$$+\lambda(\sum_u \| x_u \|^2 + \sum_g \| z_g \|^2) \quad (7)$$

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

$$x_u = (\alpha Y^T C^u Y + (1-\alpha) Z^T C^{*u} Z + (\lambda + \alpha \lambda_f) I)^{-1}$$
$$(\alpha (Y^T C^u p(u) + \lambda_f \frac{1}{|F(u)|} \sum_{f \in F(u)} x_f) + (1-\alpha) Z^T C^{*u} p^*(u)) \quad (8)$$

For the item factor y_i, it is the same as in Equation (3), and for the group factor z_g, it is the same as in Equation (6).

Previously, we performed experiment on a real-world large dataset [13]. The experiment demonstrated the ideal performance achieved by regularization for fusing friendship and CMF for fusing membership. Moreover, it showed that membership is more effective than friendship in boosting the item recommendation's accuracy, and fusing them together can further increase the accuracy. Since the fusion was conducted on implicit matrix factorization with implicit binary data as input (i.e., users' interaction records with items), we proved that social relations can be effective in boosting top-N recommendation in such scenario.

4.1 Extension: Integrating Similarity Measure

Recently, we have been engaged in integrating similarity measure into the regularization model in order to further increase its prediction power. Essentially, the similarity measure between the user and her/his friends was used to adjust these friends' respective contributions:

$$\min_{u*,i*} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda(\sum_u \| x_u \|^2 + \sum_i \| y_i \|^2) \quad (9)$$
$$+\lambda_f(\| x_u - \frac{1}{|F(u)|} \sum_{f \in F(u)} \widehat{sim}(u,f) x_f \|^2)$$

where $\widehat{sim}(u,f) = sim(u,f)/\sum_{v \in F(u)} sim(u,v)$ denotes the normalized similarity degree between the user u and one of her/his friends f. The analytic expression for x_u is (for y_i, it is the same as in Equation (3)):

$$x_u = (Y^T C^u Y + (\lambda+\lambda_f)I)^{-1}(Y^T C^u p(u) + \frac{\lambda_f}{| F(u) |} \sum_{f \in F(u)} \widehat{sim}(u,f) x_f) \quad (10)$$

Specifically, we have tested different approaches to calculate the similarity degree, including ones based on user-friend common groups, 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 performs more accurate than others when being incorporated into the regularization of friendship (see Section 7).

With the similarity-enhanced friendship regularization, the combination model is accordingly revised as follows:

$$\alpha \min_{u*,i*} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda(\sum_u \| x_u \|^2 + \sum_i \| y_i \|^2)$$
$$+\lambda_f(\| x_u - \frac{1}{|F(u)|} \sum_{f \in F(u)} \widehat{sim}(u,f) x_f \|^2) +$$
$$(1-\alpha) \min_{u*,g*} \sum_{u,g} c^*_{ug}(p^*_{ug} - x_u^T z_g)^2 + \lambda(\sum_u \| x_u \|^2 + \sum_g \| z_g \|^2) \quad (11)$$

To generate the 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, the prediction score is calculated as $\hat{p}_{ui} = x_u^T y_i$. The top-N items with higher prediction scores are then recommended to the user.

5. RECOMMENDING GROUPS

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

$$\min_{u*,g*} \sum_{u,g} c^*_{ug}(p^*_{ug} - x_u^T z_g)^2 + \lambda(\sum_u \| x_u \|^2 + \sum_g \| z_g \|^2) \quad (12)$$

where p^*_{ug} equals 1 if the user u joined group g, otherwise it is 0; c^*_{ug} is the confidence level (see its definition in Table 1). The analytic expressions for x_u and z_g which are used to minimize the cost function are:

$$x_u = (Z^T C^{*u} Z + \lambda I)^{-1} Z^T C^{*u} p^*(u) \quad (13)$$

$$z_g = (X^T C^{*g} X + \lambda I)^{-1} X^T C^{*g} p^*(g) \quad (14)$$

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

5.1 Fusing Friendship by Regularization

To fuse friendship into the group recommendation, we empirically evaluated both regularization and factorization methods (see Section 7). Particularly, considering the regularization's performance in item recommendation, we propose the following similarity-enhanced regularization equation for group recommendation:

$$\min_{u*,g*} \sum_{u,g} c^*_{ug}(p^*_{ug} - x_u^T z_g)^2 + \lambda(\sum_u \| x_u \|^2 + \sum_g \| z_g \|^2)$$
$$+\lambda_f(\| x_u - \frac{1}{| F(u) |} \sum_{f \in F(u)} \widehat{sim}(u,f) x_f \|^2) \quad (15)$$

where $\widehat{sim}(u,f)$ has the same definition as in Equation (9). The analytic expression for x_u is

$$x_u = (Z^T C^{*u} Z + (\lambda+\lambda_f)I)^{-1}(Z^T C^{*u} p^*(u) + \frac{\lambda_f}{|F(u)|} \sum_{f \in F(u)} \widehat{sim}(u,f) x_f) \quad (16)$$

The expression for the group factor z_g is the same as in Equation (14). The experiment shows that the common-group based similarity is more effective than others when being integrated into the above Equation to calculate $\widehat{sim}(u,f)$ (see Section 7.2).

5.2 Fusing User-Item Preferences by Factorization

With the user-item preferences data (which can be either explicitly obtained from users' ratings on items, or implicitly inferred from their interaction with items), we were interested in identifying their actual impact on augmenting the group recommendation. Still, we tried both factorization and regularization models, and especially investigated the collective matrix factorization (CMF) technique given the data's bipartite property.

$$\alpha \min_{u*,g*} \sum_{u,g} c^*_{ug}(p^*_{ug} - x_u^T z_g)^2 + \lambda(\sum_u \parallel x_u \parallel^2 + \sum_g \parallel z_g \parallel^2) +$$
$$(1-\alpha) \min_{u*,i*} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda(\sum_u \parallel x_u \parallel^2 + \sum_i \parallel y_i \parallel^2) \quad (17)$$

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

$$x_u = (\alpha Z^T C^{*u} Z + (1-\alpha) Y^T C^i Y + \lambda I)^{-1} * (\alpha Z^T C^{*u} p^*(u) + (1-\alpha) Y^T C^u p(u)) \quad (18)$$

$$y_i = (X^T C^i X + \lambda I)^{-1} X^T C^i p(i) \quad (19)$$

The expression for z_g is the same as in Equation (14).

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

$$\min_{u*,g*} \sum_{u,g} c^*_{ug}(p^*_{ug} - x_u^T z_g)^2 + \lambda(\sum_u \parallel x_u \parallel^2 + \sum_g \parallel z_g \parallel^2) + \lambda_f(\parallel x_u - \frac{1}{N(u)} \sum_{n \in N(u)} \omega^*_{un} * x_n \parallel^2) \quad (20)$$

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 this model is:

$$x_u = (Z^T C^{*u} Z + (\lambda + \lambda_f) I)^{-1} (Z^T C^{*u} p^*(u) + \lambda_n \frac{1}{|N(n)|} \sum_{n \in N(n)} \omega^*_{un} x_n) \quad (21)$$

5.3 Fusing Friendship and User-Item Preferences Together

The two resources, friendship and user-item preferences, can be also fused together for generating group recommendation:

$$\alpha \min_{u*,g*} \sum_{u,g} c^*_{ug}(p^*_{ug} - x_u^T z_g)^2 + \lambda(\sum_u \parallel x_u \parallel^2 + \sum_g \parallel z_g \parallel^2) + \lambda_f(\parallel x_u - \frac{1}{|F(u)|} \sum_{f \in F(u)} \widehat{sim}(u,f) x_f \parallel^2) +$$
$$(1-\alpha) \min_{u*,i*} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda(\sum_u \parallel x_u \parallel^2 + \sum_i \parallel y_i \parallel^2) \quad (22)$$

where the friendship is handled by the similarity-enhanced regularization model and user-item preferences are handled via the factorization. This combination was actually derived from comparing regularization and factorization respectively for fusing friendship and fusing user-item preferences (see Section 7.2). The analytic expression for x_u is

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

The expression for z_g is the same as in Equation (14), and for y_i it is the same as in Equation (19).

6. RECOMMENDING FRIENDS

The challenging issue for recommending friends is that there might be various reasons for two users to become friends, so it is more difficult to predict the friendship than the user-item preference and the membership. In our baseline, we propose to add the regularization process to the matrix factorization for producing the friend recommendation given the friendship's one mode property. Formally, the cost function is

$$\min_{u*,f*} \sum_{u,f} c'_{uf}(p'_{uf} - x_u^T x_f)^2 + \lambda \parallel x_u \parallel^2 + \lambda_f \parallel x_u - \frac{1}{|F(u)|} \sum_{f \in F(u)} x_f \parallel^2 \quad (24)$$

where the definition of c'_{uf} and p'_{uf} can be referred to Table 1. In this equation, the first part represents the factorization and the second part gives the regularization. The analytic expressions for x_u is

$$x_u = (X^T C'^u X + (\lambda + \lambda_f) I)^{-1} (X^T C'^u p'(u) + \frac{\lambda_f}{|F(u)|} \sum_{f \in F(u)} x_f) \quad (25)$$

The users with higher prediction scores as computed from $\hat{p}'_{uf} = x_u^T x_f$ can be recommended to the target user.

6.1 Fusing User-Item Preferences by Factorization

Since regularization was already embedded in the baseline Equation (24), we mainly exploit factorization to fuse the user's item preferences into the friend recommendation. The corresponding cost function is

$$\min_{u*,f*} \sum_{u,f} c'_{uf}(p'_{uf} - x_u^T x_f)^2 + \lambda_f \parallel x_u - \frac{1}{|F(u)|} \sum_{f \in F(u)} x_f \parallel^2 +$$
$$\alpha \min_{u*,i*} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda(\sum_u \parallel x_u \parallel^2 + \sum_i \parallel y_i \parallel^2) \quad (26)$$

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

$$x_u = (X^T C'^u X + \alpha Y^T C^u Y + (\lambda + \lambda_f) I)^{-1} (X^T C'^u p'(u) + \alpha Y^T C^u p(u) + \lambda_f \frac{1}{|F(u)|} \sum_{f \in F(u)} x_f) \quad (27)$$

$$y_i = (X^T C^i X + \lambda I)^{-1} X^T C^i p(i) \quad (28)$$

6.2 Fusing Membership by Factorization

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

$$\min_{u*,f*} \sum_{u,f} c'_{uf}(p'_{uf} - x_u^T x_f)^2 + \lambda_f \parallel x_u - \frac{1}{|F(u)|} \sum_{f \in F(u)} x_f \parallel^2 +$$
$$\alpha \min_{u*,g*} \sum_{u,g} c^*_{ug}(p^*_{ug} - x_u^T z_g)^2 + \lambda(\sum_u \parallel x_u \parallel^2 + \sum_g \parallel z_g \parallel^2) \quad (29)$$

The analytic expressions for x_u and z_g are:

$$x_u = (X^T C'^u X + \alpha Z^T C^{*g} Z + (\lambda + \lambda_f) I)^{-1} (X^T C'^u p'(u) + \alpha Z^T C^{*g} p^*(g) + \lambda_f \frac{1}{|F(u)|} \sum_{f \in F(u)} x_f) \quad (30)$$

$$z_g = (X^T C^{*g} X + \lambda I)^{-1} X^T C^{*g} p^*(g) \tag{31}$$

6.3 Fusing Membership and User-Item Preferences Together

The membership and user-item preferences are then fused simultaneously into the following combination model:

$$\min_{u*,f*} \sum_{u,f} c'_{uf}(p'_{uf} - x_u^T x_f)^2 + \lambda_f \parallel x_u - \frac{1}{|F(u)|} \sum_{f \in F(u)} x_f \parallel^2$$
$$+\lambda \parallel x_u \parallel^2 + \alpha \min_{u*,i*} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda(\sum_u \parallel x_u \parallel^2$$
$$+ \sum_i \parallel y_i \parallel^2) + \beta \min_{u*,g*} \sum_{u,g} c^*_{ug}(p^*_{ug} - x_u^T z_g)^2$$
$$+\lambda(\sum_u \parallel x_u \parallel^2 + \sum_g \parallel z_g \parallel^2) \tag{32}$$

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

$$x_u = (X^T C'^u X + \alpha Y^T C^u Y + \beta Z^T C^{*g} Z + (\lambda + \lambda_f) I)^{-1} (X^T$$
$$C'^u p'(u) + \alpha Y^T C^u p(u) + \beta Z^T C^{*g} p^*(g) + \lambda_f \frac{1}{|F(u)|} \sum_{f \in F(u)} x_f) \tag{33}$$

The expressions for y_i and z_g are the same as in Equations (28) and (31) respectively.

7. EXPERIMENT SETUP AND RESULTS

We crawled a real-world dataset from Last.fm (which is a popular music-sharing site, http://www.last.fm/) for conducting the experiment. Concretely, three types of data resources were available for the testing: user-artist (item) binary matrix (i.e., users' implicit interaction with items), user-group (membership), and user-user (friendship). 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.

The details of the dataset are given in Table 2. To test the accuracy of item recommendation, the user-item pairs were first divided into 10 subsets with equal sizes. Two subsets 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. To test the accuracy of group & 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 ten 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. Concretely, during each testing round, one of the user's participated groups (or connected friends) was randomly chosen as her/his target choice.

In Last.fm, since there is no explicit rating data available, we make top-N recommendation (N = 5, 10), instead of predicting rating for each item. The standard metrics, including precision, recall and F-measure, were adopted to evaluate the item recommendation's accuracy: $Pre@N = hit@N/N$, $Rec@N = hit@N/|T|$ and $F1@N = 2Pre@N * Rec@N/(Pre@N + Rec@N)$, where N is the size of the recommendation list, $|T|$ is the size of the user's testing set T, and $hit@N$ indicates the intersection between the recommendation list and the testing set.

Table 2: Description of Last.fm dataset

Element	Size	Element	Size
#user	100,000	#user-item pair	29,908,020
#item	22,443	#user-user friendship pair	583,621
#group	25,397	#user-group pair	1,132,281

For the group and friend recommendation, we use the hit ratio as the metric: $Hits@N = \sum_{u=1}^{m} hit(u)@N/m$, where $hit(u)@N$ denotes that the user $u's$ targeted group (or friend) was located in the top-N recommendation list, and m is the total number of users.

7.1 Item Recommendation

For the item recommendation, we compared the following 8 approaches, with particular interest in testing the newly integrated similarity measure. Because in the previous work [13], the fusion effect was found performing best at train.10, the current experiment was done at the same data density level.

- **Item.MF:** the basic matrix factorization;
- **Item.MF.F.R:** fusing the friendship by regularization;
- **Item.MF.M.F:** fusing the membership by factorization;
- **Item.MF.FM:** fusing the friendship by regularization and fusing the membership by factorization;
- **Item.MF.F.FCos:** fusing the friendship by similarity-enhanced regularization based on common friends;
- **Item.MF.F.GCos:** fusing the friendship by similarity-enhanced regularization based on common groups;
- **Item.MF.F.ICos:** fusing the friendship by similarity-enhanced regularization based on common items;
- **Item.MF.FM.GCos:** fusing the friendship by similarity-enhanced regularization based on common groups and fusing the membership by factorization.

From Table 3, it can be seen that all of the regularization-based fusion methods (Item.MF.F.R, Item.MF.F.ICos, Item.MF.F.GCos and Item.MF.F.FCos) that take into account of friendship are better than the baseline method Item.MF that is without the fusion of any resources. Among these regularization approaches, it shows that the algorithm's performance can be improved if it further integrates the similarity between the user and her/his friends, and the best performance goes to Item.MF.F.GCos which calculates the similarity based on their common groups. Given that users usually start a group or join in a group based on their common interests, the membership information has been proven as a good indicator and less noisy than friendship and user-item preferences for computing the user-user similarity.

We further compared the friendship-fused winner (i.e., Item.MF.F.GCos) with the membership-fused factorization method (i.e., Item.MF.M.F). It indicates that Item.MF.M.F outperforms Item.MF.F.GCos, inferring that membership is more effective than friendship in boosting the item recommendation's accuracy. In addition, comparing Item.MF.FM.GCos

Table 3: Results w.r.t. Recommending Items (Prec: Precision; Rec: Recall; F1: F-measure)

Method	Prec@5	Prec@10	Rec@5	Rec@10	F1@5	F1@10
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**

Note: λ is set as 10, and the size of user/item latent factors (k) is 10. λ_f and α were tuned with the validation set.

(that combines friendship and membership together) to these methods shows that this combination achieves higher accuracy. It is also superior to the originally proposed combination model (Item.MF.FM that is without the integration of similarity measure). It thus implies that when the friendship's regularization has chance to be improved via the similarity integration, it can be more effectively combined with the factorization of membership to reach at a higher accuracy level.

7.2 Group Recommendation

As for recommending groups, we compared the following 10 methods:

- **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-enhanced regularization based on common friends;
- **Group.MF.F.GCos:** fusing the friendship by similarity-enhanced regularization based on common groups;
- **Group.MF.F.ICos:** fusing the friendship by similarity-enhanced regularization based on common items;
- **Group.MF.FI.GCos:** fusing the friendship by similarity-enhanced regularization based on common groups and fusing the user-item preference by factorization.

Table 4 shows all the results. First of all, for fusing user-item preferences, the results show that the accuracy of factorization model (Group.MF.I.F) can be improved when the density level is increased (where train.X represents that only X% of total user-item pairs were used). In other words, the denser that user-item preferences are when being fused into the factorization model, the more accurate the group recommendation is. Relatively, the accuracy of regularization method (Group.MF.I.R) is lower. It does not obviously change with the increase of data density level. 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 does not obviously improve even in denser user-item matrix. On the other hand, in terms of fusing friendship, we find that the regularization outperforms the factorization model (i.e., Group.MF.F.R against Group.MF.F.F).

Being combined with our previous observations [13], the above results verify our hypothesis that the factorization

Table 4: Results w.r.t. Recommending Groups

Method	Hits@5	Hits@10
Group.MF (baseline)	0.0530	0.0875
Fusing user-item preferences (by Factorization)		
Group.MF.I.F@train.20 ($\alpha = 0.8$)	0.0573	0.0899
Group.MF.I.F@train.40 ($\alpha = 0.9$)	0.0678	0.1026
Group.MF.I.F@train.60 ($\alpha = 0.9$)	0.0714	0.1068
Group.MF.I.F@train.80 ($\alpha = 0.9$)	**0.0722**	**0.1070**
Fusing user-item preferences (by Regularization)		
Group.MF.I.R@train.20 ($\lambda_f = 0.1$)	0.0559	0.0885
Group.MF.I.R@train.40 ($\lambda_f = 0.01$)	0.0559	0.0885
Group.MF.I.R@train.60 ($\lambda_f = 0.01$)	0.0560	0.0886
Group.MF.I.R@train.80 ($\lambda_f = 0.01$)	**0.0561**	**0.0887**
Fusing friendship (by Regularization & Factorization)		
Group.MF.F.R ($\lambda_f = 10$)	0.0566	0.0910
Group.MF.F.F ($\alpha = 0.9$)	0.0553	0.0876
Group.MF.F.FCos ($\lambda_f = 10$)	0.0549	0.0861
Group.MF.F.GCos ($\lambda_f = 10$)	**0.0593**	**0.0923**
Group.MF.F.ICos ($\lambda_f = 0.1$)	0.0569	0.0897

model is indeed more suitable than the regularization model for fusing bipartite data (e.g., user-group membership, user-item preferences), while the regularization model is better for fusing one mode data (e.g., friendship).

We further tested the performance when the similarity measure between users was integrated into the friendship's regularization. Being consistent to its effect on augmenting item recommendation, the integration of similarity based on users' common groups (Group.MF.F.GCos) outperforms other similarity-integrated regularization methods. However, overall speaking, the user-item preferences still act more positive and helpful for enhancing group recommendation, compared to the friendship (see Table 4).

We finally combined Group.MF.F.GCos and Group.MF.I.F@train.80, which is shorted as Group.MF.FI.GCos, for fusing the two resources (friendship and user-item preferences) together. It can be seen that fusing friendship and user-item preferences simultaneously achieves accuracy improvement against fusing them separately.

7.3 Friend Recommendation

Given the superior performance of regularization model for handling friendship when it was fused to produce item (and group) recommendation, we embedded the regularization process into the baseline friend recommendation, which was further compared to three fusion variations:

Table 5: Results w.r.t. Recommending Friends

Method	Hits@5	Hits@10
Friend.MF (baseline)	0.0155	0.0203
Friend.MF.I.F@train.10 (α=0.1, λ_f=1000)	**0.0159**	**0.0212**
Friend.MF.I.F@train.20 (α=0.1, λ_f=1000)	0.0157	0.0209
Friend.MF.I.F@train.40 (α=0.01, λ_f=1000)	0.0152	0.0201
Friend.MF.I.F@train.60 (α=0.01, λ_f=1000)	0.0152	0.0201
Friend.MF.I.F@train.80 (α=0.01, λ_f=1000)	0.0151	0.0199
Friend.MF.M.F (α=0.01, λ_f=1000)	**0.0163**	**0.0218**
Friend.MF.MI (α=0.05, β=0.05, λ_f=1000)	0.0159	0.0209

Note: The size of user/item latent factors (k) is set as 50 for recommending friends.

- **Friend.MF:** the basic matrix factorization;
- **Friend.MF.M.F:** fusing the membership by factorization;
- **Friend.MF.I.F:** fusing the user-item preferences by factorization;
- **Friend.MF.MI:** fusing the membership and user-item preferences together.

Table 5 shows the comparison results. It can be seen that when the data density level is increased, the Friend.MF.I.F (that fuses user-item preferences) decreases, implying that the fusion of more user-item pairs does not help to improve the accuracy. With respect to the fusion of membership, Friend.MF.M.F achieves higher accuracy compared to both the baseline (Friend.MF) and Friend.MF.I.F@train.10 (which is the best among the fusions of user-item preferences), indicating that the membership is more useful than user-item preferences for benefiting the friend recommendation. In the experiment, we also tested the effect of integrating similarity measure into the regularization of friendship, but since the performance was not obviously enhanced, the results are not displayed in the table.

Moreover, it is surprising to find that the combination model (Friend.MF.MI that fuses user-item preferences at train.10 and membership together) is not superior to Friend.MF.M.F (that fuses membership alone). Its accuracy is actually equal to the one of Friend.MF.I.F@train.20. This finding implies that only fusing membership might be sufficient to enhance the friend recommendation. It is indeed the best among all compared methods. For the future work, we are thus motivated to compare this membership-fused approach (based on matrix factorization) to related friend recommendation methods that are under other theoretical framework [1].

8. CONCLUSIONS

In conclusion, this paper presents a generalized framework that recommends items, groups and friends in a single system by examining their mutual contributions. It was found that item recommendation is more accurate when user-group membership and user-user friendship are both considered. The similarity-measure based on users' common groups can further enhance the prediction power of regularization when fusing friendship, and enable the item recommendation's accuracy to achieve an upper level. For the recommendation of groups, the user-item preferences were shown more effective than friendship, especially at higher level of data density. Their combination (with the fusion of both factorization-based user-item preferences and the similarity-integrated regularization of friendship) obtains the best accuracy. Regarding friend recommendation, the fusion of membership acts more positive than the one of user-item preferences.

In addition, the results of comparing regularization and factorization models extend our prior claim (being made for item recommendation [13]) to a larger scope. For group recommendation, the factorization model again shows better suitability for fusing bipartite data (i.e., user-item preferences, in addition to user-group membership), while the regularization model suits better one mode data (i.e., friendship). It thus reveals their respective advantages in different data conditions. Another promising finding is that the similarity measure can take positive effect on augmenting the regularization process, and the similarity based on users' common groups is more effective than alternatives. These phenomena can hence be suggestive to other researchers to improve their recommender applications' performance.

9. ACKNOWLEDGMENTS

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

10. REFERENCES

[1] L. Backstrom and J. Leskovec. Supervised random walks: predicting and recommending links in social networks. In *Proc. WSDM '11*, pages 635–644, New York, NY, USA, 2011. ACM.

[2] J. Chen, W. Geyer, C. Dugan, M. Muller, and I. Guy. Make new friends, but keep the old: recommending people on social networking sites. In *Proc. CHI '09*, pages 201–210. ACM, 2009.

[3] W.-Y. Chen, J.-C. Chu, J. Luan, H. Bai, Y. Wang, and E. Y. Chang. Collaborative filtering for orkut communities: discovery of user latent behavior. In *Proc. WWW '09*, pages 681–690. ACM, 2009.

[4] I. Guy, I. Ronen, and E. Wilcox. Do you know?: recommending people to invite into your social network. In *Proc. IUI '09*, pages 77–86. ACM, 2009.

[5] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *Proc. ICDM '08*, pages 263–272. IEEE Computer Society, 2008.

[6] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proc. RecSys '10*, pages 135–142. ACM, 2010.

[7] M. Jamali, T. Huang, and M. Ester. A generalized stochastic block model for recommendation in social rating networks. In *Proc. RecSys '11*, pages 53–60. ACM, 2011.

[8] I. Konstas, V. Stathopoulos, and J. M. Jose. On social networks and collaborative recommendation. In *Proc. SIGIR '09*, pages 195–202. ACM, 2009.

[9] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *Proc. WSDM '11*, pages 287–296. ACM, 2011.

[10] P. Symeonidis, E. Tiakas, and Y. Manolopoulos. Transitive node similarity for link prediction in social networks with positive and negative links. In *Proc. RecSys '10*, pages 183–190. ACM, 2010.

[11] V. Vasuki, N. Natarajan, Z. Lu, and I. S. Dhillon. Affiliation recommendation using auxiliary networks. In *Proc. RecSys '10*, pages 103–110. ACM, 2010.

[12] S.-H. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha. Like like alike: joint friendship and interest propagation in social networks. In *Proc. WWW '11*, pages 537–546. ACM, 2011.

[13] Q. Yuan, L. Chen, and S. Zhao. Factorization vs. regularization: fusing heterogeneous social relationships in top-n recommendation. In *Proc. RecSys '11*, pages 245–252. ACM, 2011.