Factorization vs. Regularization: Fusing Heterogeneous Social Relationships in Top-N Recommendation

Quan Yuan
IBM Research - China
Zhongguancun Software Park, Haidian District
Beijing, China
quanyuan@cn.ibm.com

Li Chen
Department of Computer Science
Hong Kong Baptist University
Hong Kong China
lichen@comp.hkbu.edu.hk

Shiwan Zhao
IBM Research - China
Zhongguancun Software Park, Haidian District
Beijing, China
zhaosw@cn.ibm.com

ABSTRACT

Collaborative Filtering (CF) based recommender systems often suffer from the sparsity problem, particularly for new and inactive users when they use the system. The emerging trend of social networking sites and their accommodation in other sites like e-commerce can potentially help alleviate the sparsity problem with their provided social relation data. In this paper, we have particularly explored a new kind of social relation, the membership, and its combined effect with friendship. The two type of heterogeneous social relations are fused into the CF recommender via a factorization process. Due to the two relations' respective properties, we adopt different fusion strategies: regularization was leveraged for friendship and collective matrix factorization (CMF) was proposed for incorporating membership. We further developed a unified model to combine the two relations together and tested it with real large-scale datasets at five sparsity levels. The experiment has not only revealed the significant effect of the two relations, especially the membership, in augmenting recommendation accuracy in the sparse data condition, but also identified the ability of our fusing model in achieving the desired fusion performance.

Categories and Subject Descriptors

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

General Terms

Algorithms, Experimentation

Keywords

Social relationships, membership, friendship, factorization, regularization

RecSys'11, October 23–27, 2011, Chicago,Illinois, USA.

1. INTRODUCTION

With the rapid growth of the internet, people are constantly confronted with overwhelming amount of information and choices. Recommender systems have therefore been widely developed to effectively support users' decision-making process in such situations, for example, when they are deciding which music to listen to on *Last.fm*, which product to buy on *Amazon*, which people to connect with on *Facebook*, and so on.

However, most recommender systems perform not so accurate for the inactive or new users as they have only expressed few ratings or interacted with few items. With such sparse data, it is indeed hard to make accurate recommendations if the system purely relies on users' ratings or their interaction records. To address this problem, other types of information sources have been increasingly incorporated into the process of improving recommendations. In particular, social relationship data have been regarded potentially valuable because the relation (e.g., an inactive user's connection with his friends) could be usefully applied to find the user's like-minded neighbors and hence address the rating sparsity limitation. In fact, in current social media sites (e.g., Last.fm), a user might be associated with different types of social relations. For example, he may create a friend list (e.g., the friendship) which is in nature a bidirectional relationship as two parties should approve this connection. The user could also join in an interest group, to establish membership with others whom he may do not know in the offline life.

The focus of this paper is thus on in-depth investigating the respective roles of friendship and membership in augmenting the collaborative recommending process. In this regard, although some researchers have lately attempted to fuse friendship to boost Collaborative Filtering (CF) based recommenders [7, 10, 13], few have fused membership or combine it with friendship to make item recommendation (though the membership has been leveraged to make community recommendation, i.e., recommending to the user interest groups or communities that s/he may be interested in joining [19, 2]). The difference of membership from friendship in nature lies that it involves two types of entities: users and groups, while friendship only involves one: users. Moreover, joining a group can be a direct indicator of the user's specific interest in that group's topic, but friendship might be vague because two people can have various reasons to add each other as friends [1].

Given the limitation of related works (see detailed discussion in next section) and the vacancy of utilizing membership for making *item recommendations* in the general domain of recommender systems, we have been driven to answer the following questions:

1. How to effectively incorporate membership in combination with friendship, into item recommendation process, and take advantage of their respective pros, and would the popular regularization model of the factorization be applicable for achieving the goal?

2. How the fusion effect would be particularly in the sparse data condition (i.e., with few of users' interaction data with items)? The answer to this question could help identify whether the fusion of social relations, via the appropriate modeling, could in practice address the cold-start problem.

3. State-of-the-art works mainly focus on minimizing the rating prediction error, but in most of real applications, explicit ratings are often unavailable. The question is then how effective it would when fusing social relations with implicit data (e.g., users' visiting records) for the purpose of generating top-N recommendation?

To reach our objectives, we have first done analysis on the different properties of one mode relations (e.g. friendship, trust relation) and bipartite relations (e.g. membership). We have then proposed a factorization based fusing framework to process the two types of relation data given their respective properties: collective matrix factorization applied for fusing membership, and regularization model for friendship. It shows that this framework obtains superior results than fusing both relations in a unique regularization model. We have further tested the framework in a real large-scale dataset from Last.fm. The experiment indicates that membership is more effective than friendship in boosting the recommendation accuracy, and fusing them together can further increase the accuracy. On the other hand, since our fusing framework is based on implicit matrix factorization with implicit binary data as input, for the first time in this area, we have proved that social relation data can be effective in boosting top-N recommendation in the implicit rating condition.

In the following content, we first introduce related work and indicate their limitations. We then analyze the pros and cons of two typical fusing frameworks, and identify factorization as our fusing framework for membership. The algorithm steps follow with details of strategies we have proposed to deal with friendship and membership respectively. In the experiment part, we present the results of comparing the fusing algorithm with baseline methods and testing the role of social relation data at varied sparsity levels. At the end, we conclude our work and summarize the major findings.

2. RELATED WORK

Because traditional user-based or item-based Collaborative Filtering (CF) algorithms often suffer from sparse and imbalance of rating data, researchers have started to incorporate other kinds of data sources. For example, [3] proposed to use trust in web-based social networks to create predictive movie recommendations. The trust value was obtained by requiring users to specify how much they trust the people they know. [11] proposed a factor analysis approach based on the probabilistic graphical model, which fuses the user-item matrix with the users' social trust networks by sharing a common latent low-dimensional user feature matrix.

However, given the difficulty of obtaining actual trust relations in the real online environment, some researchers have attempted to utilize friendship because it is easier to obtain in the social networking sites. For instance, Konstas [8] adopted Random Walk with Restart to model the friendship and social annotation (tagging) in a music track recommendation system. In a Munich-based German community, friends are compared to neighbors of collaborative filtering for predicting ratings [4]. Their results showed that the social friendship can benefit the traditional recommender system. [6] also proposed an online social recommender system to use the friends' info for generating recommendations. Regarding membership, it has been mainly adopted to make people/affiliation recommendation, not item recommendation, in the related work. [18] used membership for recommending online communities to members in the Orkut social network. [2] focused on recommending communities through Latent Dirichlet Allocation (LDA) and association rule mining techniques. Indeed, few have utilized both friendship and membership, as auxiliary information, to boost the accuracy of item recommendation.

From the algorithm's aspect, Matrix Factorization (MF) technique can be an alternative and potentially more effective tool for fusing the social relation data, but unfortunately no much work has been done to deeply explore its role. The low-rank matrix factorization methods were actually originally proposed to train user-item matrix, under the assumption that only a small number of factors influences preferences, and that a user's preference vector is determined by how each factor applies to that user [15] [16]. Lately, some researchers have attempted to adopt MF to fuse trust or friend relations. For example, Ma et al. proposed STE, which is a linear combination of basic matrix factorization and a trust network based approach [10]. In their follow-up work [12], two social regularization terms were defined to constrain the matrix factorization objective function, with the goal of effectively fusing the friends' info in MF. Rather than factorizing a single user-item (e.g. user-movie) matrix, [17] introduced Collective Matrix Factorization (CMF) to incorporate the factorization of movie-genre matrix simultaneously when factorizing user-movie matrix. However, according to our knowledge, this approach haven't been used in factorizing social relationships and its ability in handling social data is unknown.

As a matter of fact, from surveying related literatures, we found that little attention has been paid to developing MF-based algorithms for specifically processing the two types of social relation data: friendship and membership. Though some algorithms were proposed to fuse the friends' info, they did not essentially take the advantage of explicit membership network among users. In addition, less focuses have been on in-depth studying the respective impacts of friendship and membership on augmenting item recommendations.

In this paper, we have been therefore engaged in addressing the limitations from both actions of incorporating membership and developing the algorithm. We have not only measured the effect of membership on improving item recommendation, but also investigated how to optimally fuse the social relations via the matrix factorization technique.

3. FUSING SOCIAL RELATIONSHIPS INTO RECOMMENDERS

Before introducing our fusing framework, first we make a classification for the datasets used for recommendation, and then introduce our fusing framework.

3.1 One Mode VS. Bipartite Data

We classify the data into two class: one mode data and bipartite data. One mode data means the dataset only contains one type of entity, e.g. in user-user friendship data, user-user trust relation data, there is only one type of entity, that is user. Bipartite data means the data set contains two type of entities, in most of the cases, one is user and the other is item, for example, user's rating on items, user's participation in social groups, etc.

For the one mode data, it describes the relation between entities, and which can be considered as a indicator of closeness, e.g. if there is a link between two entities, then we think the two entities are more closer. Because of this, most state-of-the-art works [12, 11, 7] leverage regularization model to fusing the one mode data by minimizing the gap between the taste of a user and the taste of her/his friends.

However, for the bipartite data, we argue that it different from the one mode data in that a user explicitly described the interesting topics of oneself by interaction with items, which is absent in one mode data and suitable for a factorization process, which can effectively factorize user-item relations into two components and obtained a user latent factor model and a item latent factor model. By contrary, if we try to handle bipartite data in the manner of regularization, firstly we need to do the one-mode projection, i.e., transform the user-group relationship into user-user relationship. The one-mode projection is always 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 convert this data into a social relation graph, then u_1 will has one link with u_2, and also has a link with u_3. From the transitivity's perspective, u_2 and u_3 should share some common interests, while the fact is not.

In a word, for social relations of bipartite data type, e.g. membership, in order to better leverage the full information in the user-group data, we leverage the CMF approach to factorize them directly into user factors and group-topic factors. Our experiments also prove that the performance of factorization approach for fusing bipartite is better than regularization approach, and regularization is better than factorization when dealing with one mode data.

3.2 Implicit Factorization based Social Fusion

A typical factorization model [9] computes a user-factor vector x_u for each user u, and an item-factor vector y_i for each item i. Then the rating prediction for user u to item i is based on the inner product of corresponding user-factor and item-factor, i.e., $r_{ui} = x_u^T y_i$. However, when handling implicit data, there are two distinctions from factorizing explicit ratings accordign to [5]: firstly, the factor model is tailored for implicit data by associating the data with varying confidence levels; secondly, optimization should account for all u, i pairs, which means that no matter whether the cell is '1' or '0', they should be taken into consideration. Thus, factors are computed by minimizing the following cost function:

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

where p_{ui} measures user u's preference on item i. If user u 'touched' or 'clicked' this item, then $p_{ui} = 1$, otherwise $p_{ui} = 0$; c_{ui} is the confidence level indicating how much a user prefers an item. The confidence level can be computed by the time a user spends on an item, or the frequency a user interacts with an item. In this paper, because the input user-item matrix is a strictly binary matrix, and there is no additional "time" or "frequency" data available, the c_{ui} is set to 1 for all the user-item pairs.

As the cost function contains $m * n$ terms, where m is the number of users and n is the number of items, in order to reduce the computation cost, [5] adopted alternating-least-squares (ALS) for this optimization process, and addressed the computation problem by exploiting the structure of the variables. Their analytic expressions for x_u and y_i that minimize the cost function in equation 1 are described below:

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

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

In these equations, C_u denote a diagonal $n * n$ matrix, where $C^u_{ii} = c_{ui}$, and define the vector $p(u)$ that contains all the preferences of u (the p_{ui} pairs). Throughout this paper, the above approach was called the baseline MF for short.

3.2.1 Fusing Friendship by Regularization

In this section, we describe how we integrate friendship into the before-mentioned matrix factorization framework.

For friendship with only one type of entity, we can not effectively factorize it into two components. Instead, we use it as constraints to the user-item matrix factorization, and aim at minimizing the gap between the taste of a user and the average taste of her/his friends via the regularization model [7, 12].

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

We add the regularization term for the friendship in order to minimize the gap. λ_f is the coefficient of the friendship regularization (hereafter this approach is named MF.F for short). We also adopt alternating-least-squares (ALS) in this optimization process. The new analytic expression for x_u was generated based on 4:

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

For the expression of y_i, it remains the same as in Equation 3.

3.2.2 Fusing Membership by Factorization

Compared to the friendship that only involves one single type of entity (users), the membership involves two types of entities (users and groups) which reflect users' participation in groups. Therefore, the user-item interaction matrix can be directly factorized into two components - the "user" latent factors and "group" factors, indicating users' preferences over groups and groups' distribution on latent features. Therefore, we adopt Collective Matrix Factorization [17] for this purpose, and name this approach MF.M in short. The experimental results (in Section 4) prove that factorization model is more effective than the regularization model in fusing membership.

The formula of fusing membership into factorization is formally as follows:

$$\alpha \min_{u*,i*} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2) + \\ (1-\alpha) \min_{u*,g*} \sum_{u,g} c_{ug}^*(p_{ug}^* - x_u^T z_g)^2 + \lambda(\sum_u \|x_u\|^2 + \sum_i \|z_g\|^2) \tag{6}$$

where the parameter α is used to adjust the weight of user-item matrix and user-group matrix in the factorization. Refer to confidence level c_{ui} in factorizing user-item, we introduce the c_{ug}^* when factorizing user-group, which indicates the confidence level of users' preference towards groups. Similar to c_{ui}'s setting, we set all the c_{ug}^* to 1 for all the user-group pairs because users' participation history in groups is a binary matrix. According to equation 6, the analytic expression for x_u is:

$$x_u = (\alpha Y^T C^u Y + (1-\alpha) Z^T C^{*u} Z + \lambda I)^{-1} * \\ (\alpha Y^T C^u p(u) + (1-\alpha) Z^T C^{*u} p^*(u)) \tag{7}$$

Where C^{*u} and $p^*(u)$ has a similar definition with C_u and $p(u)$ in equation . The expression for group factor z_g is:

$$z_g = (X^T C^{*g} X + \lambda I)^{-1} X^T C^{*g} p^*(g) \tag{8}$$

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

To compare the performance of fusing membership via the factorization model versus the regularization model, we also propose a fusing framework based on 4 (which is called MF.M.REG for short). The idea is to convert user-group matrix into user-user relationship with a weighted schema. For example, if user u and user v have two common groups, there is a link between user u and user v, with weight 2.

$$\min_{u*,i*} \sum_{u,i} c_{ui}(p_{ui} - x_u^T y_i)^2 + \lambda(\sum_u \|x_u\|^2 + \sum_i \|y_i\|^2) + \\ \lambda_n(\|x_u - \frac{1}{|N(u)|} \sum_{n \in N(u)} w_{un} * x_n\|^2) \tag{9}$$

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

$$w_{un} = \frac{|CG_{un}|}{\sum_{i \in N(u)} |CG_{ui}|} \tag{10}$$

where CG_{un} is the common groups between user u and n, $|CG_{un}|$ is the size of common groups.

Similar to the equation 5, the analytic expression of this model is in the following equation 11.

$$x_u = (Y^T C^u Y + (\lambda + \lambda_f) I)^{-1} (Y^T C^u p(u) + \\ \lambda_n \frac{1}{|N(u)|} \sum_{n \in N(u)} w_{un} * x_n) \tag{11}$$

In order to see the factorization model's effectiveness when handling one mode data, we also apply the equation 6 on friendship (a user-user binary matrix with '1' indicates the friendship linkage), and compare it with fusing friendship by regularization in equation 4.

3.2.3 Fusing Membership and Friendship Together

For the next step, it comes to derive the formula for fusing friendship and membership simultaneously. As shown in the following equation 4, we handle friendship by regularization and deal with membership via the collective matrix factorization (this unified fusion framework is called MF.FM hereafter).

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

So we generate the expression for x_u based on 12:

$$x_u = (\alpha Y^T C^u Y + (1-\alpha) Z^T C^{*u} Z + (\lambda + \alpha\lambda_f) I)^{-1} \\ (\alpha (Y^T C^u p(u) + \lambda_f \frac{1}{|F(u)|} \sum_{f \in F(u)} x_f) + (1-\alpha) Z^T C^{*u} p^*(u)) \tag{13}$$

For the item factor y_i, it is the same as in Equation 3; and for the group factor, it is the same as in Equation 8.

3.3 Making Top-N Recommendation

To generate a top-N item list for each user u, we assume her/his candidate item set (items untouched by user) is ϕ_u, and for each item i in ϕ_u, we calculate a predict score $p_{ui}^{'}$ by equation 14. We then rank items according to the predict scores and recommend the top-N to this user:

$$p_{ui}^{'} = x_u^T * y_i \tag{14}$$

where x_u^T and y_i are user latent factor model and item latent factor model respectively, as calculated by above-mentioned fusing framework.

4. EXPERIMENTS

4.1 Datasets

Given that Last.fm (a worldwide popular social music site) provides rich info about users' social relations, we extracted the data by accessing the site's Web Service APIs: the *membership* which describes the user's participation in groups and the *friendship* between users. Furthermore, we think it

(a) Lastfm data overview

Element	Size
#user	100,000
#item	22,443
#group	24,562
#user-item pair	29,205,921
#user-group pair	837,132
#user-user friendship	382,563

(b) Dataset sparsity

Data	Sparsity
train.10	99.91%
train.20	99.82%
train.30	99.74%
train.40	99.65%
train.50	99.57%
test	99.82%

Table 1: Description of LastFm dataset

is more meaningful to recommend artists instead of individual songs since user preference on artists would be more stable, so we use artist as the "item" in our recommendations. In Last.fm, as there is no explicit rating data available, so we use the implicit feedbacks and make top-N recommendation instead of predicting ratings.

Concretely, we first crawled the user-item interaction data and social relations from Last.fm, and then randomly sampled 100K users from the dataset in order to evaluate our recommendation algorithms. For our purpose, we use three types of data sources as the input to our algorithms: user-artist (item) binary matrix, user-group, and user-user (friendship) data. Since we attempted to explore how social information can help build user-item matrix when the data is at different sparsity levels, we made the training and testing set as follows: firstly, we randomly select 20% of user-item pairs as the test set; then we randomly split the rest of the data into 8 slices, each containing 10% user-item pairs. They are incrementally combined to form the training sets with different levels of sparsity. For example, train.10 contains 10% user-item pairs of the total data, and train.20 contains 20% user-item pairs of the whole data, etc. The details are described in Table 1.

4.2 Evaluation Metrics

We adopt standard metrics in the area of information retrieval to evaluate our recommenders. After training the model, we recommend top-N (N = 5, 10, 15, 20 in our experiments) items for each user. We then count the intersection set of the recommended top-N list with users' test set, and define the hit@N as the size of this intersection set. Based on the definition of hit, we define recall and precision in Equation 15 and Equaton 16.

1. **Recall.** The score measures the average (on all users) of the proportion (in percentages) of artists from the test sets that appear among the top n ranked list, for some given n. It should be as high as possible for good performance.

$$Recall@N = \frac{hits@N}{|T|} \tag{15}$$

   where $|T|$ is the size of each user's test set.

2. **Precision.** This metric measures the proportion of recommended items that are ground-truth items.

$$Precsion = \frac{hits@N}{|N|} \tag{16}$$

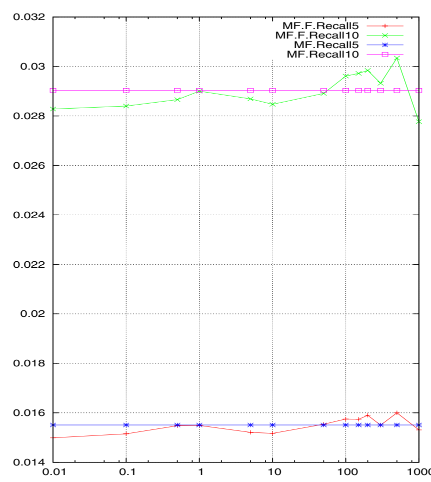
   where $|N|$ is the size of recommendation list.

4.3 Experimental Design and Results

In the experiment, we first select the most sparse dataset train.10, to explore the impact of friendship network with the change of the regularization term λ_f. We then uncover the impact of membership on the performance of recommendation when it was fused with factorization manner (CMF). After that, we compare regularization and factorization's effectiveness when handling one mode and bipartite data, and identify the pros and cons of the two approach. After this, we look into the combinational effect of fusing membership and friendship together, and their overall performance, and compare three types of fusing strategies (MF.F, MF.M, MF.FM) with the baseline MF. Finally, we are interested in exploring the effect of fusing social relation data into user-item matrix when it was at different levels of sparsity.

Since our primary goal was to explore the effect of social relationships on augmenting the performance of the recommendation, we only tune the social relation related parameters, such as λ_f for MF.F, λ_n for MF.M.REG, α for MF.M, etc., while fixing other parameters. In the rest of the paper, λ is default set to 0.015, and the size of user/item factors is set to 10.

4.3.1 *Impact of λ_f on Friendship Fusion*



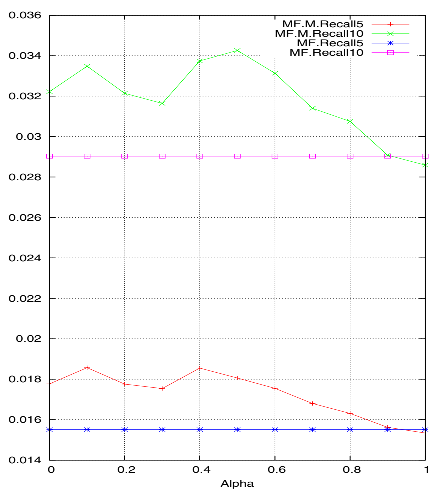
(a) Impact of λ_f on recall

Figure 1: Impact of λ_f on friendship fusion

In equation 4, parameter λ_f controls the influence of friendship network. Larger values of λ_f in the objective function indicate more impact of the friendship on users' behavior. λ_f equals to zero, making the model close to baseline MF as shown in equation 1. However, very large values will make a user's feature vector very close to his/her neighbors' feature vectors in the training stage.

Figures 1 illustrates the recall@5 and recall@10 changes similarly when λ_f changes. On this training.10 data, the best result of recall@5, recall@10, happens for $\lambda_f = 500$. This value of λ_f is much larger than ordinary ones we seen before (e.g. from 0.001 to 5 or so). The reason for such large values of λ_f could be that the train.10 is much sparser (99.91%) than the previous ones, which will cause severe overfitting problem. In order to overcome the overfitting problem, the regularization term should play a much more important role, thus the coefficient is larger than before. Our further experiments also prove this, e.g., the train.50 with denser data (99.57%), achieves the best performance when $\lambda_f = 1$. Fusing friendship improves the recall@10 by 4.62%.

4.3.2 *Impact of α on Membership Fusion*



(a) Impact of α on recall

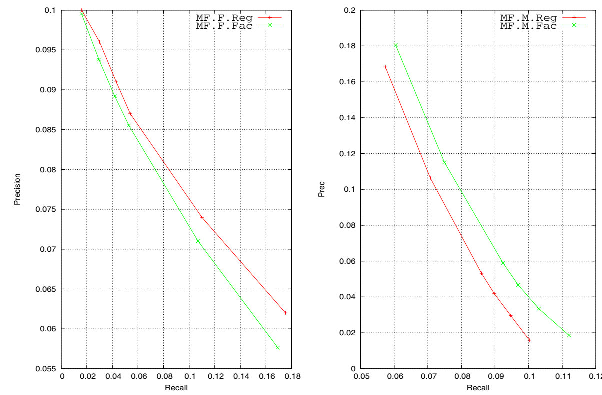
Figure 2: Impact of α on membership fusion

Figure 2 reports the performance of fusing membership via CMF. The parameter α plays an important role in controlling how much the membership should be fused into the factorization process. In extreme case, if we set the α to '1', we only use the user-item binary matrix for making recommendation; while if we set α to '0', the membership information dominates the factorization process. Figure 2 shows how the changes of α affect recommendation accuracy and ranking quality.

From figure 2, we observe that the optimal result for recall@10 was get when $\alpha = 0.5$, while the optimal α for recall@5 is 0.1. For the sake of simplicity, we mainly focus on two metrics: recall@5, recall@10 to identify the best parameters for each algorithm. So in the above situation, $\alpha = 0.5$ is better than $\alpha = 0.1$. Under optimal setting of α, the recall@5 boosted 12.88% and recall@10 boosted 18.14% than baseline MF.

Above experiments for parameters λ_f and α show that purely using the behavioral data (user-item matrix) or purely using the social relational data can not generate better performance than appropriately integrating them together. Compare with the beforementioned friendship fusion, apparently the membership fusion by CMF is much better than friendship fusion. Though the number of user-group pairs is about twice as many as user-user friendship pairs, the impact of membership fusion is four times better than friendship fusion, which proves that the membership closely relates to users' preference. Another finding is that the membership and friendship are both useful in top-N recommendation besides minimizing rating prediction error.

4.3.3 *Regularization VS. Factorizing*



(a) Regularization VS Factorization,Friendship (b) Regularization VS Factorization,Membership

Figure 3: Regularization VS. Factorization

Figure 3 compare the performance of regularization and factorization when handling one mode data (friendship) and bipartite data (membership). In this figure, each line here represents the precision of the algorithm at a given recall.

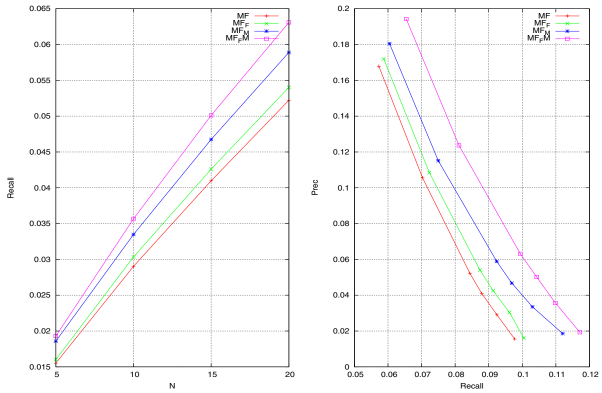
From figure 3(a), we see that fusing friendship by regularization (MF.F.Reg) is clearly better than fusing it with factorization (MF.F.Fac), which proves our former assumptions that regularization model is good at minimizing the gap between the taste of a user and the taste of her/his friends. Vice versa, figure 3(b) proves that when handling bipartite data like membership, factorization (CMF) outperforms regularization-based model not only in accuracy but also in computation efficiency. Since in MF.M.REG, when converting user-group to pairwise user-user relations, there is a pre-processing step and its complexity is $O(|U|^2)$, where $|U|$ is the number of users. By contrast, CMF-based algorithm (MF.M) does not have this step.

In a word, the above results exhibit the advantages of factorization model in fusing bipartite data and regularization model in handling one mode data, thus prove our assumption that membership is different from friendship in nature, thus need a new way to treat this type of information.

4.3.4 *Fusing Friendship and Membership Together*

From the above figures we know that each type of social relationship can augment recommendation quality when fusing them in a proper way, especially the membership information. In this section, we want to further explore the results by fusing two heterogeneous social information together by the approach MF.FM.

Firstly, we look into figure 4(a), which compares MF.FM's performance with other two fusing models and the baseline MF when the size of recommendation changes from 5 to 20. We can see that no matter how long the size of recommendation is, fusing social relation is better than baseline MF. What's more, fusing both types of social relationship together (MF.FM) outperforms fusing the friendship or membership separately. The best result of MF.FM increased the recall@5 by 20.56%, while fusing friendship alone has no impact on recall@5, and fusing membership alone raises the recall@5 by 12.88%. Obviously, the combinational ef-



(a) Recall@5 at N (b) Precision-versus-Recall

Figure 4: Fusing results of membership and friendship

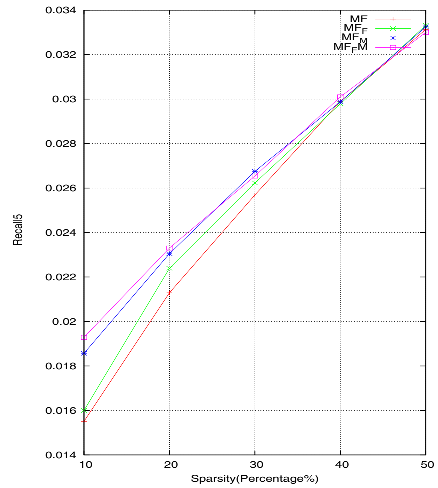
fect (MF.FM is 59.6% improvement over MF.M) of fusing two heterogeneous data together is much more effective than fusing each alone in this sparse condition.

In figure 4(b), again, it confirms that MF.FM outperforms the other fusing algorithms and the baseline MF in terms of precision and recall. Moreover, we notice that combinational effect of membership and friendship is obvious, because the improvement of MF.FM is clearly larger than the sum of improvement of MF.F and MF.M. For instance, in table **??**, let us look at recall@10 of train.10 data, the improvement of MF.MF is 22.84% while the sum of MF.F and MF.M is 20.45%. According to this observation, we believe that membership and friendship is complementary to each other on this very sparse data, and our model MF.FM is a suitable way to fuse them together.

4.3.5 *Impact of Sparsity on Social Fusion*

We were also interested in seeing the effect of social relationship when the sparsity of behavioral matrix changes. So we repeated our experiments on 5 training datasets with different sparsity levels as shown in table 1.

Figure 5(a) illustrates the recall on different datasets. The x-axis is the sparsity level, 10 means that the training data contains 10% user-item pairs of the total dataset, and the y-axis is the value of recall@5 metric. We found that on the most sparse data, the effects of friendship and membership are quite obvious, while as the training data becomes denser, the effect begins to decrease. On train.20 and train.30, social relations still play a positive role in recommendation, while on train.40, the effect of fusing social data is often ignored. Furthermore, on train.50, in some cases, particularly for MF.FM algorithm, the result of social fusion is even worse than baseline MF. Our explanation for this phenomenon is that the user-item behavioral matrix is the best data source to reflect users' preferences on items. Only when user-item is very sparse, other auxiliary information like social relationships will be good in inducing users' taste. If user-item matrix is dense enough, the introducing of auxiliary data sources will bring in noise in modeling users' preferences therefore impaired the performance of recommendation. This is similar to [14]'s work which proves that even 10 ratings of a new movie are more valuable than its metadata



(a) Recall changes with sparsity

Figure 5: Fusing results changes with sparsity

Method	Rec@5	Rec@10	Rec@15	Rec@20
Train.10: MF	.016	.029	.041	.052
$MF\_F(\lambda = 500)$	.016(0)	.030(4.62)	.043(3.98)	.054(3.88)
$MF\_M(\alpha = 0.5)$	.018(12.88)	.034(18.14)	.048(17.8)	.061(17.96))
$MF\_M\_REG(\lambda = 0.01)$	.016(0)	.030(2.31)	.042(2.15)	.053(2.31)
$MF\_FM(\alpha = 0.5)$	.019(20.56)	.036(22.84)	.050(22.31)	.063(21.32)
Train.20: MF	.021	.039	.055	.069
$MF\_F(\lambda = 1000)$	.022(5.17)	.041(4.02)	.057(3.52)	.071(3.26)
$MF\_M(\alpha = .4)$	.023(8.27)	.042(6.87)	.058(6.37)	.073(6.15)
$MF\_FM(\alpha = .4)$	.023(9.44)	.042(7.93)	.059(7.21)	.073(6.83)
Train.30: MF	.026	.046	.064	.080
$MF\_F(\lambda = 1000)$	.026(2.30)	.047(1.60)	.065(1.47)	.080(1.17)
$MF\_M(\alpha = 0.5)$	.027(4.13)	.048(2.94)	.066(2.68)	.082(2.71)
$MF\_FM(\alpha = 0.5)$	.027(3.31)	.048(2.83)	.066(2.63)	.082(2.53)
Train.40: MF	.030	.052	.072	.089
$MF\_F(\lambda = 10)$	.030(-.33)	.052(-.15)	.072(-.13)	.089(-.30)
$MF\_M(\alpha = 0.6)$	.030(-.03)	.053(.86)	.072(.85)	.090(.79)
$MF\_FM(\alpha = 0.6)$	.030(.70)	.053(.76)	.072(.61)	.089(.55)
Train.50: MF	.033	.058	.078	.096
$MF\_F(\lambda = 1)$	.033(.57)	.058(.19)	.078(-.04)	.096(.13)
$MF\_M(\alpha = 0.6)$	.033(.36)	.058(.16)	.078(.08)	.096(.39)
$MF\_FM(\alpha = 0.6)$	.033(-.39)	.057(-.55)	.078(-.29)	.096(-.17)

Table 2: Recall of fusing algorithms on five datasets

for predicting user ratings. In short, the users' behavioral data on items is the most proper data source which directly reflect users' preferences on items.

4.3.6 *Overall Accuracy Comparison*

Finally, as recall is the primary metric in top-N recommendation [5], table 2 listed the best recall in the five datasets at different sparsity levels. The number in the bracket is the improvement of corresponding algorithms compared with baseline MF in each dataset.

From these results, we can derive several insights. First of all, in the implicit feedback dataset, the social relationships are found very helpful in augmenting recommendation accuracy, especially under the sparse data condition. Particularly, the membership, which contains users' participation history in groups, is shown better than friendship in boosting top-N recommendation accuracy. On the other hand, as the user-item matrix becomes denser, the impact of social relationships decrease, and in some cases, it may be harmful especially when fusing two type of relations together by MF.FM.

5. CONCLUSION

In conclusion, in this paper, we have targeted to address the sparsity problem for new and inactive users by introducing social relation data. To be concrete, we have first examined the two types of relations (membership and friendship), and proposed a fusing framework to incorporate the two heterogeneous data. According to our analysis and experiments, we found the regularization model is suitable for one mode data, and factorization model is good at fusing bipartite data. Since there is not too much work on fusing social relation with implicit data, our method was compared to the baseline implicit matrix factorization approach in the experiment. Moreover, the experiment assessed the performance of different fusion methods at varied sparsity data levels. The distinguished effectiveness of social relationships in the sparse data condition was demonstrated. More notably, the significant role of membership in achieving the goal was identified, as well as the combined effect of heterogeneous relations in very sparse data. On the other hand, in the dense data condition, the social relations did not show significant improvement on recommendation accuracy. Furthermore, the combinational effect of fusing membership and friendship together is oblivious on sparse data, as the data become denser, it disappears. In conclusion, these findings illustrate meaningful insights for social recommendation, and verify the significant impact of social relations on addressing the sparsity problem.

6. REFERENCES

[1] N. K. Bayma and A. Ledbetterb. Tunes that bind? predicting friendship strength in a music-based social network. In *Information, Communication and Society*, pages 408–427, 2009.

[2] 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 *Proceedings of the 18th international conference on World wide web*, WWW '09, pages 681–690, New York, NY, USA, 2009. ACM.

[3] J. Golbeck. Generating predictive movie recommendations from trust in social networks. In *Trust Management*, pages 93–104. 2006.

[4] G. Groh and C. Ehmig. Recommendations in taste related domains: collaborative filtering vs. social filtering. In *GROUP '07*, pages 127–136, New York, NY, USA, 2007. ACM.

[5] Y. Hu, Y. Koren, and C. Volinsky. Collaborative filtering for implicit feedback datasets. In *ICDM '08: Proceedings of the 2008 Eighth IEEE International Conference on Data Mining*, pages 263–272, Washington, DC, USA, 2008. IEEE Computer Society.

[6] H. G. K. Hummel, B. V. D. Berg, A. J. Berlanga, H. Drachsler, J. Janssen, R. Nadolski, and R. Koper. Combining social based and information based approaches for personalised recommendation on sequencing learning activities. *Int. J. Learn. Technol.*, 3(2):152–168, 2007.

[7] M. Jamali and M. Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 135–142, New York, NY, USA, 2010. ACM.

[8] I. Konstas, V. Stathopoulos, and J. M. Jose. On social networks and collaborative recommendation. In *SIGIR '09: Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 195–202, New York, NY, USA, 2009. ACM.

[9] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42:30–37, August 2009.

[10] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '09, pages 203–210, New York, NY, USA, 2009. ACM.

[11] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM '08: Proceeding of the 17th ACM conference on Information and knowledge management*, pages 931–940, New York, NY, USA, 2008. ACM.

[12] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*, WSDM '11, pages 287–296, New York, NY, USA, 2011. ACM.

[13] P. Massa and P. Avesani. Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems*, RecSys '07, pages 17–24, New York, NY, USA, 2007. ACM.

[14] I. Pilászy and D. Tikk. Recommending new movies: even a few ratings are more valuable than metadata. In *Proceedings of the third ACM conference on Recommender systems*, RecSys '09, pages 93–100, New York, NY, USA, 2009. ACM.

[15] J. D. M. Rennie and N. Srebro. Fast maximum margin matrix factorization for collaborative prediction. In *Proceedings of the 22nd international conference on Machine learning*, ICML '05, pages 713–719, New York, NY, USA, 2005. ACM.

[16] R. Salakhutdinov and A. Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th international conference on Machine learning*, ICML '08, pages 880–887, New York, NY, USA, 2008. ACM.

[17] A. P. Singh and G. J. Gordon. Relational learning via collective matrix factorization. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '08, pages 650–658, New York, NY, USA, 2008. ACM.

[18] E. Spertus, M. Sahami, and O. Buyukkokten. Evaluating similarity measures: a large-scale study in the orkut social network. In *KDD '05: Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 678–684, New York, NY, USA, 2005. ACM.

[19] V. Vasuki, N. Natarajan, Z. Lu, and I. S. Dhillon. Affiliation recommendation using auxiliary networks. In *Proceedings of the fourth ACM conference on Recommender systems*, RecSys '10, pages 103–110, New York, NY, USA, 2010. ACM.