Published: 16 September 2015

Citation in BibTeX format

RecSys '15: Ninth ACM Conference on Recommender Systems
September 16 - 20, 2015
Vienna, Austria

Conference Sponsors:
SIGCHI

RESEARCH-ARTICLE

Overlapping Community Regularization for Rating Prediction in Social Recommender Systems

HUI LI, The University of Hong Kong, Hong Kong, Hong Kong

Data Mining, Information Retrieval, Data Management

DINGMING WU, The University of Hong Kong, Hong Kong, Hong Kong

WENBIN TANG, The University of Hong Kong, Hong Kong, Hong Kong

NIKOS MAMOULIS, The University of Hong Kong, Hong Kong, Hong Kong

Overlapping Community Regularization for Rating Prediction in Social Recommender Systems*

Hui Li, Dingming Wu, Wenbin Tang, Nikos Mamoulis
Department of Computer Science
The University of Hong Kong, Hong Kong
{hli2,dmwu,wbtang,nikos}@cs.hku.hk

ABSTRACT

Recommender systems have become de facto tools for suggesting items that are of potential interest to users. Predicting a user's rating on an item is the fundamental recommendation task. Traditional methods that generate predictions by analyzing the user-item rating matrix perform poorly when the matrix is sparse. Recent approaches use data from social networks to improve accuracy. However, most of the social-network based recommender systems only consider direct friendships and they are less effective when the targeted user has few social connections. In this paper, we propose two alternative models that incorporate the overlapping community regularization into the matrix factorization framework. Our empirical study on four real datasets shows that our approaches outperform the state-of-the-art algorithms in both traditional and social-network based recommender systems regarding both cold-start users and normal users.

1. INTRODUCTION

Recommender systems have become essential tools for suggesting items of potential interest to users and they have successfully been deployed in the industry, with applications such as movie recommendations (Netflix), product recommendations (Amazon), and music recommendations (Last.fm).

The various definitions of the recommendation problem all boil down to predicting the ratings of a *target* user on items (e.g., movies) that the user has not rated before (e.g., unwatched movies). Specifically, consider a set of m users and a set of n items in a rating-based recommender system: each user u can rate any item by giving it a score. Given a target user u, for each item i that u has not rated, the system predicts the rating, based on the existing ratings of other users. Then, the unrated items with high predicted rating scores are offered as suggestions to u.

Traditional recommender systems [9, 3, 11, 24, 10, 5, 26, 31, 14] are effective for target users who have rated many items, since it is easy to find other users that have rated these items. However, they perform poorly for *cold-start* users who have very few ratings; in this case, it becomes hard to find similar users to the target user, in order to generate recommendations based on their ratings.

The increasing popularity of online social networks offer chances to improve the accuracy of rating predictions; as sociologists postulate, people tend to relate to people with similar preferences and people that influence each other become more similar [27]. [2] also confirm that a social network provides an independent source of information which can be exploited to improve the quality of rating predictions. Based on the rationale that a user's interest is similar to or influenced by her/his friends, several social-based recommender systems [17, 16, 7, 18, 15, 30, 12] have recently been proposed. Experiments demonstrate that making recommendations based on the ratings of the users socially connected to the target user improves traditional techniques, especially when the user-item rating matrix is sparse. However, in the literature, the most effective social-based recommenders systems [7, 18, 30] only consider direct friendships in the network. As shown in our empirical study, they become less effective for target users who have few ratings (*rating-cold-start* users) or few social connections (*social-cold-start* users).

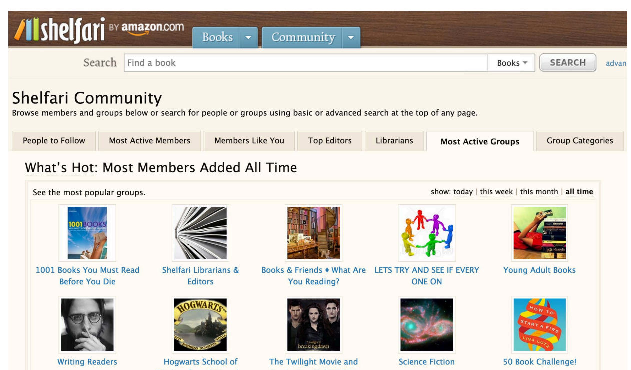


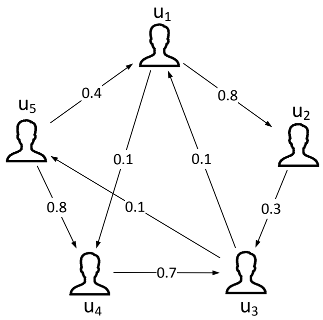
Figure 1: Communities in Shelfari

In this paper, we exploit information about the communities formed by users in social networks, to improve the recommendation accuracy. Social network users tend to establish relationships with people who share similar interests with them. For example, Figure 1 illustrates user communities in a book recommender system[1], based on different topics. The members in the same community usually share characteristics and can be an alternative of direct friends for social recommender systems. Aiming at solving the problems of social-based recommenders discussed in the previous paragraph, we propose two models that incorporate the overlapping community regularization into the matrix factorization framework differently. The communities are detected based on the social network structure; a user may belong to multiple communities with differ-

*Work supported by grant HKU 715413E from Hong Kong RGC.

RecSys'15, September 16–20, 2015, Vienna, Austria.
© 2015 ACM. ISBN 978-1-4503-3692-5/15/09 ...$15.00.
DOI: http://dx.doi.org/10.1145/2792838.2800171.

[1] `http://www.shelfari.com`



(a) Social Graph

	u_1	u_2	u_3	u_4	u_5
u_1	0	0.8	0	0.1	0
u_2	0	0	0.3	0	0
u_3	0.1	0	0	0	0.1
u_4	0	0	0.7	0	0
u_5	0.4	0	0	0.8	0

(b) Adjacency Matrix

	v_1	v_2	v_3	v_4	v_5
u_1	5	?	3	1	?
u_2	2	?	?	4	?
u_3	?	5	?	?	?
u_4	?	?	2	?	?
u_5	4	3	2	?	?

(c) Rating Matrix

	v_1	v_2	v_3	v_4	v_5
u_1	5	**3.3**	3	1	**4.6**
u_2	2	**4**	**3.2**	4	**2.8**
u_3	**4**	5	**2.4**	**4.9**	**1.7**
u_4	**3**	**4.3**	2	**4.7**	**2.3**
u_5	4	3	2	**3.2**	**4.3**

(d) Output

Figure 2: An Example of Social Recommender System

ent interests. One of our models (MFC) ensures that the distance between the latent feature vectors of users u and u' is low if u and u' belong to the same community c. Our other model (MFC$^+$) forces the user latent feature vectors to be close to those of her/his communities. Empirical studies on four real datasets show that our approaches outperform the state-of-the-art traditional and social-based recommenders by 6%-42% for general users. Moreover, we put emphasis on *cold-start* users. While the problem of rating-cold-start users is studied in previous research on social recommender systems, *social-cold-start* users are ignored by previous work. Our methods consider both cases and beat baselines by 7%-32% for *rating-cold-start* users and 4%-37% for *social-cold-start* users.

The rest of the paper is organized as follows. Section 2 formally defines the problem. Section 3 reviews major rating prediction approaches in the literature. Section 4 introduces our MFC and MFC$^+$ models. The results of our empirical study are reported in Section 5. We conclude in Section 6.

2. PROBLEM DEFINITION

In a traditional ratings-based recommender system, there are m users $\{u_1, \cdots u_m\}$ and n items $\{v_1, \cdots v_n\}$. The users' ratings on items form a $m \times n$ rating matrix $R = [r_{ij}]$ where r_{ij} is the rating of user u_i on item v_j. Typically, 5-scale or 10-scale rating systems are used.

In a social recommender system, we have a social network graph where each node represents a user and edges model the social relationships between users (e.g., friendship or influence). The social graph can also be modeled by a $m \times m$ *adjacency* matrix $S = [s_{ij}]$, where s_{ij} represents the similarity between users u_i and u_j or how much user u_i trusts user u_j. Figure 2 shows a toy example of a social recommender system with 5 users and 5 items. Figure 2(a) is the social network graph and Figure 2(b) is the corresponding adjacency matrix $S = [s_{ij}]$ where a positive s_{ij} indicates a social edge between user u_i and user u_j. Figure 2(c) is an exemplary user-item rating matrix where questionmarks are unknown ratings. Figure 2(d) illustrates the possible output of the social recommender system where any unknown ratings are predicted.

The basic task of a social recommender is as follows: given the user-item rating matrix $R = [r_{ij}]$ and the adjacency matrix $S = [s_{ij}]$, predict an unknown rating r_{ij} for user u_i on item v_j.

3. RELATED WORK

This section reviews important rating prediction approaches in traditional and social-based recommender systems.

3.1 Traditional Approaches

One of the most commonly-used and successfully-deployed rating prediction approaches in traditional recommender systems is collaborative filtering (CF). Two classes of CF methods are widely used. *Memory-based* methods predict ratings for the target user based on the ratings of similar users [9] or the computed information of items similar to those chosen by the target user [3, 24]. *Model-based* methods make predictions using a trained compact model from the user-item rating matrix. Various training models have been investigated, such as the clustering model [10], the aspect models [5, 26], the Bayesian hierarchical model [31], and the ranking model [14]. None of these traditional approaches take social network data into account.

3.2 Social-based Approaches

Most of the social-based approaches [17, 16, 7, 18, 15, 30] follow the matrix factorization (MF) framework [11, 23], due to its effectiveness and efficiency in dealing with large user-item rating matrices. Let R be an $m \times n$ matrix with the ratings of m users on n items. The basic MF method shown in Figure 3(a) predicts the rating matrix R by multiplying a d-rank user-specific matrix $U \in \mathbb{R}^{d\times m}$ with a d-rank item-specific matrix $V \in \mathbb{R}^{d\times n}$, i.e., $R \approx U^T V$, where $d \ll \min(m, n)$. Column vectors U_i and V_i represent the d-dimensional latent feature vectors of user u_i and item v_j, respectively. The latent vectors can be learnt by minimizing the following sum-of-squared-errors objective function with two quadratic regularization terms to avoid overfitting:

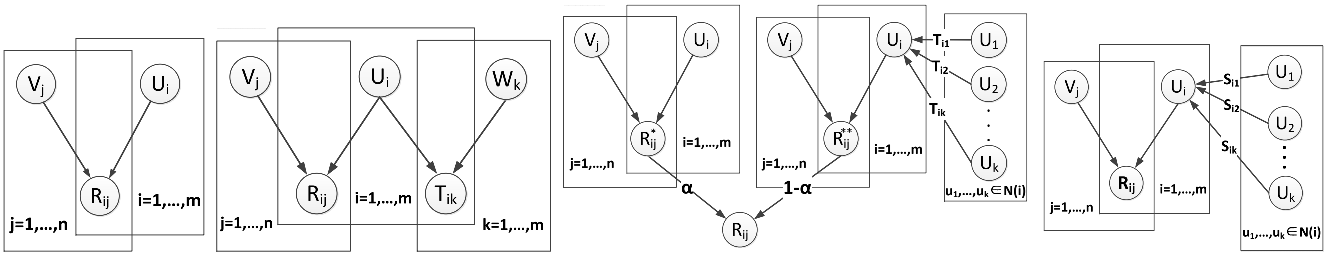
$$\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^R (R_{ij} - U_i^T V_j)^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2,$$

where $\|\cdot\|_F^2$ denotes the Frobenius norm and I_{ij}^R is equal to 1 if user u_i rated item v_j and equal to 0 otherwise. Gradient descent can be applied to find a local minimum. Having latent feature vectors U_i and V_j, the unknown rating on item v_j for user u_i is predicted as $\hat{R}_{ij} = U_i^T V_j$.

SoRec [17] extends the basic MF model by integrating the social network, as shown in Figure 3(b). T is the matrix representation of the social network; I_{ik}^T=1 if users u_i and u_k are friends and I_{ik}^T=0 otherwise. Matrix T is factorized into a user-specific matrix U and a factor-specific matrix W. The latent feature vectors of users are learnt based on both the rating and social network matrices. The objective function to be minimized is:

$$\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^R (R_{ij} - U_i^T V_j)^2 + \frac{\lambda_T}{2}\sum_{i=1}^{m}\sum_{k=1}^{m} I_{ik}^T (T_{ik} - U_i^T W_k)^2$$
$$+ \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2 + \frac{\lambda_W}{2}\|W\|_F^2.$$

Later, as shown in Figure 3(c), STE [16] modified the basic MF model so that each rating R_{ij} in the user-item matrix R reflects (i)



(a) Basic MF (b) SoRec (c) STE (d) SocialMF and SR

Figure 3: Matrix Factorization (MF) Based Models

user u_i's favor on item v_j and (ii) the favors of user u_i's friends ($N(i)$ indicates u_i's friends) on item v_j. The objective function is:

$$\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^{R}\Big(R_{ij} - \big(\alpha U_i^T V_j + (1-\alpha)\sum_{u_k \in N(i)} T_{ik} U_k^T V_j\big)\Big)^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2,$$

where T is the matrix representation of the social network, $N(i)$ is the friend set of user u_i, and α controls the effect of friends on the rating estimation.

A recent model, SocialMF [7], learns the latent feature vectors of users based on the latent feature vectors of their friends, as shown in Figure 3(d). SR model [18] improves SocialMF by treating friends with dissimilar tastes differently, so as to consider the diversity of each user's friends. SR's objective function is:

$$\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^{R}(R_{ij} - U_i^T V_j)^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2 + \frac{\beta}{2}\sum_{i=1}^{m}\sum_{u_k \in N(i)} S_{ik}\|U_i - U_k\|_F^2,$$

where S_{ik} is the similarity between users u_i and his/her friend u_k. Concerning the social-cold-start users who have few friends in the social network, in the SR$^+$ model [15], the latent feature vectors of users depend on the latent feature vectors of both their friends and the users with high similarities. However, SR$^+$ requires an a priori similarity threshold. The CircleCon model [30] refines SocialMF by considering category-specific friends and the intuition is that a user may trust different subsets of users regarding different domains. GSBM [8] extends the mixed membership stochastic blockmodel [1] to capture both the social relations and the rating behavior for groups of users and items. However, its rating prediction accuracy is worse than that of SocialMF.

PSLF [25] is a unified probabilistic model for social recommendation. It extracts the social factor vectors of users from the social network based on the mixture membership stochastic blockmodel [1] and integrates them into the user-item space.

In summary, most of the social-based approaches only consider direct friendships in the social network. They become less effective when a user has few social connections.

4. OVERLAPPING COMMUNITY REGULARIZATION

We propose two models, MFC and MFC$^+$, that incorporate the overlapping community information as regularization terms into the widely used MF framework. We first introduce the concept of overlapping community in Section 4.1. The two models MFC and MFC$^+$ are presented in Sections 4.2 and 4.3. The time complexity is analyzed in Sections 4.4.

4.1 Overlapping Community

Community structures are quite common in social networks. The users in the same community share characteristics (e.g., they may have common locations, interests, occupations, etc.). In some real applications (e.g., Douban which is an online recommender system for book, movie and music and Shelfari which is a recommender system for book), there are some manually formed communities which represents members' interests. It is natural that a user may belong to multiple communities and overlapping communities can represent the users' diverse characteristics. For example, a user may join a group for mystery novels and a group for historical novels at the same time.

We derive the rating vector of a community as the mean vector of the rating vectors of all the users in the community. We adopt the widely used Pearson Correlation Coefficient (PCC) [22] to measure the similarity S_{ij} between two users based on their rating vectors. The interest Z_{ih} of a user u_i on community c_h is the PCC between the rating vectors of user u_i and community c_h. We map PCC into range $[0, 1]$ using function $f(x) = (x+1)/2$. PCC is defined as:

$$sim_{ij} = \frac{\sum_{p\in P}(x_{ip} - \overline{x}_i)(x_{jp} - \overline{x}_j)}{\sqrt{\sum_{p\in P}(x_{ip} - \overline{x}_i)^2}\sqrt{\sum_{p\in P}(x_{jp} - \overline{x}_j)^2}},$$

where P is the intersection of the two vectors.

4.2 MFC Model

The SR model [18, 15] improves ratings prediction by imposing similarity constraints between users and their friends. Given the social network, model SR can be easily extended by considering all the members in the community where the target user belongs to. However, taking the individual influence based on the user-to-user similarity constraints may cause overfitting.

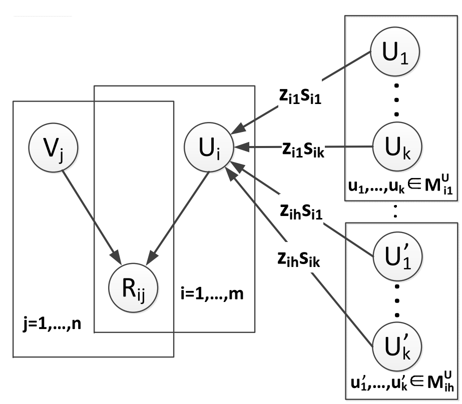


Figure 4: MFC Model

Our proposed MFC model, shown in Figure 4, injects community interest constraints into the SR model. Our motivation is that users who belong to different communities should be treated differently, as opposed to the SR model, which treats them equally. If the target user is more interested in music than sports, users belonging to the music community should be weighed higher. In MFC, the latent feature vector of user u_i depends on the users belonging to the same communities as u_i. The regularization term in MFC considers both the user similarity S_{ik} and the community interest Z_{ih}:

$$\frac{\lambda_Z}{2}\sum_{i=1}^{m}\sum_{h=1}^{l} I_{ih}^{Z} Z_{ih} \sum_{u_w \in M_{ih}^{U}} S_{iw} \| U_i - U_w \|_F^2,$$

where M_{ih}^{U} contains the users in the same community c_h as user u_i, I_{ih}^{Z} equals 1 if user u_i belongs to c_h and equals 0 otherwise. The objective function E to be minimized is

$$\begin{aligned} E =& \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^{R}(R_{ij} - U_i^T V_j)^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2 \\ &+ \frac{\lambda_Z}{2}\sum_{i=1}^{m}\sum_{h=1}^{l} I_{ih}^{Z} Z_{ih} \sum_{u_w \in M_{ih}^{U}} S_{iw} \| U_i - U_w \|_F^2. \end{aligned}$$

A local minimum of the above function E can be found by performing gradient descent in U_i and V_j:

$$\begin{aligned} \frac{\partial E}{\partial U_i} =& \sum_{j=1}^{n} I_{ij}^{R}(U_i^T V_j - R_{ij})V_j + \lambda_U U_i \\ &+ \lambda_Z \sum_{h=1}^{l} I_{ih}^{Z} Z_{ih} \sum_{u_w \in M_{ih}^{U}} S_{iw}(U_i - U_w) \\ &- \lambda_Z \sum_{u_p \in G} \sum_{c_h \in F} Z_{ph} S_{pi}(U_p - U_i), \\ \frac{\partial E}{\partial V_j} =& \sum_{i=1}^{m} I_{ij}^{R}(U_i^T V_j - R_{ij})U_i + \lambda_V V_j, \end{aligned}$$

where $G = \{u_p | \exists h, u_i \in M_{ph}^{U} \ \&\ I_{ph}^{Z} = 1\}$ and $F = \{c_h | u_i \in M_{ph}^{U} \ \&\ I_{ph}^{Z} = 1\}$.

4.3 MFC$^+$ Model

In real life, it is common for users to seek advice from members in different groups and summarize suggestions as an overall view of a group. For example, John, who joined two groups *Comedy* and *Romance* in a social-based movie recommender system, wants to watch a movie on the weekend. After reading comments, he chooses *Four Weddings and a Funeral* with high average ratings in these two groups.

Model MFC$^+$, shown in Figure 5, learns the latent feature vector of the target user u_i based on the latent feature vectors of the communities where u_i belongs. The latent feature vector C_h of community c_h is defined based on the latent feature vectors of the users that belong to c_h and the interests of these users for c_h. A user with high interest in community c_h contributes more to the latent feature vector of c_h. Also, each user's latent feature vector can contribute to multiple community latent feature vectors. The latent feature vector of a community is calculated as:

$$C_h = \frac{\sum_{u_w \in M_h^C} Z_{wh} U_w}{\sum_{u_w \in M_h^C} Z_{wh}},$$

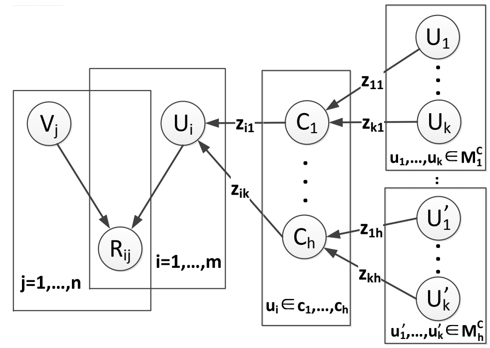


Figure 5: MFC$^+$ Model

where M_h^C contains the users belonging to community c_h. Note that C_h is a latent feature vector, while the community vector we used in Section 4.1 to calculate Z is a ratings vector. The overlapping community-based regularization term in the MFC$^+$ model is $\frac{\lambda_Z}{2}\sum_{i=1}^{m}\sum_{h=1}^{l} I_{ih}^{Z} Z_{ih} \| U_i - C_h \|_F^2$. The objective function E in MFC$^+$ is:

$$\begin{aligned} E =& \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}^{R}(R_{ij} - U_i^T V_j)^2 + \frac{\lambda_U}{2}\|U\|_F^2 + \frac{\lambda_V}{2}\|V\|_F^2 \\ &+ \frac{\lambda_Z}{2}\sum_{i=1}^{m}\sum_{h=1}^{l} I_{ih}^{Z} Z_{ih} \Big\| U_i - \frac{\sum_{u_w \in M_h^C} Z_{wh} U_w}{\sum_{u_w \in M_h^C} Z_{wh}} \Big\|_F^2. \end{aligned}$$

A local minimum of objective function E can be found by performing gradient descent in U_i and V_j:

$$\begin{aligned} \frac{\partial E}{\partial U_i} =& \sum_{j=1}^{n} I_{ij}^{R}(U_i^T V_j - R_{ij})V_j + \lambda_U U_i \\ &+ \lambda_Z \sum_{h=1}^{l} I_{ih}^{Z} Z_{ih} \Big(U_i - \frac{\sum_{u_w \in M_h^C} Z_{wh} U_w}{\sum_{u_w \in M_h^C} Z_{wh}} \Big) \\ &- \lambda_Z \sum_{u_p \in G} \sum_{c_h \in F} \Big(\frac{Z_{ph} Z_{ih}}{\sum_{u_w \in M_h^C} Z_{wh}} \Big(U_p - \frac{\sum_{u_w \in M_h^C} Z_{wh} U_w}{\sum_{u_w \in M_h^C} Z_{wh}} \Big) \Big), \\ \frac{\partial E}{\partial V_j} =& \sum_{i=1}^{m} I_{ij}^{R}(U_i^T V_j - R_{ij})U_i + \lambda_V V_j, \end{aligned}$$

where $G = \{u_p | \exists h, u_i \in M_h^C \ \&\ I_{ph}^{Z} = 1\}$ and $F = \{c_h | u_i \in M_h^C \ \&\ I_{ph}^{Z} = 1\}$.

4.4 Time Complexity

Let d be the dimensionality of the latent space, m be the number of users, n be the number of items, $\bar{r}$ be the average number of ratings per user gave, $\bar{f}$ be the average number of communities where to each user belongs, and $\bar{w}$ be the average number of members per community. The complexity of evaluating the objective function is $O(\bar{r}md + \bar{f}\bar{w}md)$, while the cost of computing the gradients is $O(\bar{r}nd + \bar{f}\bar{w}md)$. These costs are linear with respect to $\bar{r}$ and $\bar{f}\bar{w}$. Since the rating matrix is very sparse, $\bar{r}$ is relatively small. According to the analysis of real networks including information network (Wikipedia), content-sharing network (Flickr) and social networks (Facebook, Google+ and Twitter) [29], $\bar{f}$ and $\bar{w}$ are also small.

5. EMPIRICAL STUDY

We conducted an empirical study using four public datasets, Yelp, Flixster, Douban and Dianping, to evaluate MFC and MFC$^+$ compared to the state-of-the-art approaches.

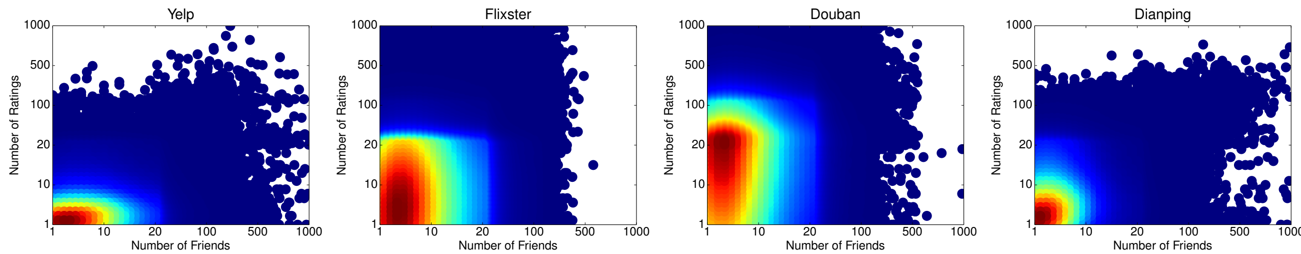


Figure 6: Distributions of Four Datasets

5.1 Data

Datasets Yelp, Flixster, Douban are widely used in previous studies of social recommender systems. We crawled Dianping from a social recommender system.

- **Yelp**[2] is provided by the fourth round of the Yelp Dataset Challenge. It is a local business recommendation platform with a social networking feature.
- **Flixster**[3] [7] is a social networking service where users can rate movies.
- **Douban**[4] [18] is an online community service, providing recommendations for movies, books and music.
- **Dianping** was crawled from the real social network-based recommender system Dianping[5], which is a leading local business search and review platform in China. The dataset contains business items in Shanghai, a social network of users, and the ratings from April 2003 to November 2013.

General statistics of the four datasets are shown in Table 1. Rating sparsity is defined as $1 - |R|/mn$ and edge sparsity is calculated as $1-2|E|/[m(m-1)]$, where $|R|$ is the number of ratings and $|E|$ is the number of social connections. In each dataset, every user has at least one friend and has rated at least one item. Each item has at least one rating. Figure 6 illustrates the relationship between the number of friends and the number of ratings per user in the four datasets and the intensity of color shows the density of users. In general, more than half of the users have few ratings and few social connections. The number of users in red regions are shown in Table 1. The whole social network and randomly selected 80% ratings are used for training. The remaining 20% ratings are held out for testing. The random selection was carried out 5 times independently and we report the average results.

Table 1: General Statistics of Four Datasets

Statistics	Yelp	Flixster	Douban	Dianping
Users	123,369	137,925	111,210	147,918
Users in Red Region	69,393	61,382	39,957	75,366
Items	41,958	48,758	57,934	11,123
Ratings	804,791	8,071,979	15,221,584	2,149,675
Social Connections	956,020	1,269,373	855,901	629,618
Rating Sparsity	99.98%	99.88%	99.76%	99.87%
Edge Sparsity	99.99%	99.99%	99.99%	99.99%

[2]http://www.yelp.com/dataset_challenge
[3]http://www.flixster.com
[4]http://www.douban.com
[5]http://www.dianping.com

5.2 Performance Metric

We adopt the *Root Mean Square Error* (RMSE) to measure the accuracy of the predicted ratings because it is widely used in the evaluation of rating-based recommendations [4]. RMSE is defined as follows:

$$RMSE = \sqrt{\frac{1}{|N|}\sum_{u_i,v_j}(R_{ij}-\hat{R}_{ij})^2},$$

where $|N|$ denotes the number of tested ratings, R_{ij} is a real rating, and $\hat{R}_{ij}$ is a predicted rating.

5.3 Competitors

To evaluate the effectiveness of our proposals, we compare with the following state-of-the-art approaches:

- **SCF** [13] is a social collaborative filtering method. Unlike the traditional user-based collaborative filtering (CF) that considers the top-k similar users, SCF makes predictions based on users' direct friends. SCF has been reported superior to CF.
- **BaseMF** [23] is the basic matrix factorization based approach that does not take the social network into account.
- **SR** [18] is a social regularization model, which adopts the individual-based regularization.
- **SR$^+$** [15]: an improved version of SR by using similar users (i.e., users' similarity is larger than θ) regarding to the target user rather than just direct friends in SR.
- **CircleCon** [30] is an extension of SocialMF and uses category-specific friends.

5.4 Community Detection

In some applications, it may be difficult to obtain the structure information of communities. To the best of our knowledge, there are no public datasets for the research of social recommender systems including such information. As a replacement, we use three prominent overlapping community detection methods to identify overlapping communities.

- **CPM** is the *clique percolation method* [20] based on *k-clique*. A *k-clique* is the complete subgraph of k nodes (e.g., a 3-clique is equivalent to a triangle), such that every two nodes in the subset are connected by an edge. Two k-cliques are adjacent if they share $k-1$ nodes. A community is a maximal set of cliques, such that every clique can be reached from every other clique through a series of adjacent cliques. CPM only considers the structure of social network.
- **BIGCLAM** is the *cluster affiliation model for big networks* [28] based on a novel observation that overlaps between communities are densly connected. It is efficient and scale to large networks but only considers the structure of social network.

- **CESNA** is one of the few overlapping community detection methods that consider both structure and node attributes [29]. Since CESNA also takes user attributes into account, we use a binary category vector from users to rated items to represent each user's attributes. For example, if user u_i rated item v_j of which the categories are *Fast Food* and *Ice Cream* then we set the corresponding entities of his/her binary category vector to 1. It has a linear runtime in the network size.

We use the fast clique percolation algorithm[6] introduced in [21] for CPM and the implementations from Stanford Network Analysis Project[7] for BIGCLAM and CESNA. For each user who does not belong to any community, we form a simple community by taking the user and all his/her direct friends.

5.5 Parameter Settings

We performed 5-fold cross-validation on the training set to empirically tune parameters, so that each method achieves its own best result.

λ_U and λ_V are set to 0.01 for all methods. In SR and SR$^+$, β is set to 0.35, 1.1, 0.6 and 0.55 on Yelp, Flixster, Douban, and Dianping, respectively. SR$^+$ requires an additional parameter θ which is hard to know a priori and we adopt 0.75, used in the original paper, in our evaluation. For CircleCon, β are set to 20 for Yelp and Dianping. For MFC, λ_Z is set to 0.0001 on Douban dataset and 0.001 on other datasets. For MFC$^+$, λ_Z is set to 0.1 for Douban, 0.5 for other datasets. We list the results for $d = 10$ where d is the dimensionality of the latent space. As d varies, our methods outperform the competitors consistently.

For community detection approaches BIGCLAM and CESNA, default parameters are used. For CPM, we report the results when $k = 3$. When k is less than 6, our methods beat SR method and achieve their best result when $k = 3$. When k exceeds 6, the number of communities that can be detected begins to decline sharply and our methods degrade to BaseMF.

5.6 Performance on All Users

We report the results of all tested users in Table 2. The standard deviations of the results are around 0.0015. The notation p, b, c denote that community structures are generated from CPM, BIGCLAM and CESNA, respectively. The percentages are the highest improvements of MFC or MFC$^+$ over the competitors. For example, the best results in Flixster (1.0134) is achieved by MFC$_b^+$ and the percentages in that row mean the percentages of the improvements of MFC$_b^+$ over the baselines. Note that datasets Douban and Flixster do not contain additional information about item category. Therefore, CESNA cannot be used in Douban and Flixster and the corresponding results are denoted by "None".

Our proposed models beat all competitors in all the four datasets, since the community information plays a positive role in rating predictions. The SR model only considers direct friends, hence the predictions for the users having few friends may not be accurate. Also, some of the friends may have inconsistent interest with the target user, causing the SR model to make inaccurate predictions. Although the SR$^+$ model takes highly similar users into account, our approaches further refine the similarities using community interest. The effectiveness has been proved by the result. Our proposals beat the state-of-the-art SR model by 7%-16% and SR$^+$ model by 5%-11% on four datasets. As an indication about the significance of this improvement, the well-known Netflix Prize[8] of one million US dollar was awarded to a team for reducing the RMSE by 10% compared to the state-of-the-art. As another evidence, the improvements reported in previous work are around 5% for SocialMF compared with STE in dataset Flixster and 1.42%-2.33% for SR compared with STE in dataset Douban.

CircleCon divides direct friends into different circles and each circle corresponds to one category. When making prediction, only one circle is considered. To compare our approaches with CircleCon, we evaluate the prediction results of category *Restaurants* in datasets Yelp and Dianping since datasets Douban and Flixster do not have the information about item category. *Restaurants* is the largest category in both two datasets. Table 3 displays the results for *Restaurants* category. According to the results, our approaches outperform CircleCon, since CircleCon has the same limitation as does SR that only direct friends are considered.

From experiment results, we can also find that our methods outperform baselines no matter which overlapping community detection methods is employed. MFC$_b$ and MFC$_b^+$ are slightly better than MFC$_p$ and MFC$_p^+$. When additional information about users is available, MFC$_c$ and MFC$_c^+$ perform much better than MFC$_b$, MFC$_b^+$, MFC$_p$ and MFC$_p^+$. It is reasonable because CESNA considers both node attributes and structure while CPM and BIGCLAM only take structure into account. Our methods are independent of the communitiy detection methods and the reduction of RMSE may go up if better community detection method is used. When explicit community structure is known, our models are expected to perform even better since they outperform baselines though current community structure is generated from community detection methods.

5.7 Performance on Cold-Start Users

In this section, we consider two types of cold-start users in order to evaluate the performance of our approaches:*rating cold-start* users, who have few ratings, and *social cold-start* users, who have few social connections. Recommender systems must be capable of matching the characteristics of an item against relevant features in the user's profile. In order to do this, it must first construct a sufficiently-detailed model of the user's tastes and preferences through preference elicitation. Rating cold-start users is challenging for traditional recommender systems, because there is not enough information about them that can be utilized to generate a recommendation. A motivation behind social recommender systems is to utilize social information to improve prediction accuracy for rating cold-start users. Previous studies [17, 7] illustrate the effectiveness of taking direct friendships into account when making predictions for rating cold-start users. However, social cold-start users are ignored in most evaluations of social recommender systems. On average, 44.70% of users are rating cold-start users and 50.62% of them are social cold-start users in four datasets used in our experiments (i.e., users in red regions in Figure 6). Due to the significant number of both rating cold-start and social cold-start users, the effectiveness of any social recommendation approach is important for both of these two types of users.

We assess the performance of our approaches for rating cold-start users with less than 5 ratings (following the setting in [7, 19, 6]) and the results are reported in Table 4. We also evaluate the performance for soical cold-start users (with less than 5 direct friends) and the result is reported in Table 5. CircleCon cannot model circle influence when user does not have enough friends who have rated items belonging to a specific category. CESNA fails to identify community for cold-start users due to the lack of their ratings. Hence, their results are omitted. As expected, all methods perform worse compared to the results in Table 2. However, our proposed methods still reduce RMSE by 5%-10% for rating cold-start users

[6] `http://github.com/aaronmcdaid/MaximalCliques`
[7] `http://snap.stanford.edu/snap`
[8] `http://www.netflixprize.com`

Table 2: Performance Comparison

Dataset	SCF	BaseMF	SR	SR$^+$	MFC$_p$	MFC$_p^+$	MFC$_b$	MFC$_b^+$	MFC$_c$	MFC$_c^+$
Yelp	1.4730 24.35%	1.2498 10.84%	1.2216 8.78%	1.2032 7.39%	1.1618	1.1617	1.1543	1.1602	1.1324	**1.1143**
Flixster	1.1761 13.83%	1.1853 14.50%	1.1041 8.21%	1.0823 6.37%	1.0427	1.0436	1.0325	**1.0134**	None	None
Douban	1.2788 31.37%	1.0478 16.24%	0.9441 7.04%	0.9322 5.86%	0.8952	0.8961	**0.8776**	0.8823	None	None
Dianping	1.3642 41.86%	1.0598 25.16%	0.9449 16.05%	0.9012 11.98%	0.8678	0.8748	0.8812	0.8721	**0.7932**	0.8211

Table 3: Performance Comparison with CircleCon

Dataset	CircleCon	MFC$_p$	MFC$_p^+$	MFC$_b$	MFC$_b^+$	MFC$_c$	MFC$_c^+$
Yelp	1.1907 7.71%	1.1332	1.1345	1.1287	1.1333	1.1001	**1.0989**
Dianping	0.8234 14.84%	0.7754	0.7801	0.7562	0.7612	0.7122	**0.7012**

Table 4: Performance on Rating Cold-start Users

Dataset	SCF	BaseMF	SR	SR$^+$	MFC$_p$	MFC$_p^+$	MFC$_b$	MFC$_b^+$
Yelp	1.7832 28.9%	1.4270 11.16%	1.3769 7.92%	1.3865 8.56%	1.3082	1.3079	**1.2678**	1.2867
Flixster	1.6086 27.42%	1.3573 13.98%	1.2589 7.25%	1.2321 5.23%	1.1747	1.1769	**1.1676**	1.1832
Douban	1.2194 23.90%	1.1198 17.13%	1.0373 10.54%	0.9823 5.53%	**0.9280**	0.9287	0.9331	0.9423
Dianping	1.6098 31.71%	1.3344 17.62%	1.1868 7.37%	1.2012 8.48%	1.1331	1.1197	1.1023	**1.0993**

Table 5: Performance on Social Cold-start Users

Dataset	SCF	BaseMF	SR	SR$^+$	MFC$_p$	MFC$_p^+$	MFC$_b$	MFC$_b^+$
Yelp	1.6472 25.2%	1.3902 11.44%	1.3521 8.94%	1.3234 6.97%	1.2938	1.2935	1.2421	**1.2312**
Flixster	1.4080 24.34%	1.2160 12.39%	1.1911 10.56%	1.1342 6.07%	1.0953	1.0956	1.0732	**1.0653**
Douban	1.4320 36.29%	1.1432 20.20%	1.0119 9.84%	1.0012 8.88%	0.9432	0.9234	**0.9123**	0.9321
Dianping	1.3724 35.00%	1.1027 19.10%	0.9847 9.40%	0.9321 4.29%	0.9109	0.9246	**0.8921**	0.9012

and 4%-10% for social cold-start users, compared with SR and SR$^+$ indicating that community regularization handles both rating cold-start users and social cold-start users better than baselines.

5.8 Impact of parameter λ_Z

Parameter λ_Z controls the influence of communities in the prediction process of MFC and MFC$^+$. A large λ_Z indicates strong influence of communities. Figure 7(a) show the impact of λ_Z over dataset Dianping when CPM is employed and similar trends can be observed over other datasets. MFC$^+$ achieves its best performance when $\lambda_Z = 0.5$, while the best results of MFC are when λ_Z is small (0.001).

The two models exhibit different behaviors because MFC$^+$ minimizes the distance between the target user and her/his interested communities, while MFC tries to minimize the distances between the target user and the users in the same communities. This means that λ_Z is considered more times for MFC than for MFC$^+$ in the calculation of gradient descent, since the number of communities to which the target user belongs is generally much smaller than the numbers of users in these communities. Thus, a large λ_Z can help MFC$^+$ achieve a good result, while a small λ_Z can avoid the over-influence of communities to MFC.

5.9 Comparison between MFC and MFC$^+$

From the experimental results, we can see that MFC and MFC$^+$ have similar performance. Intuitively, MFC should perform better when the communities include diverse users since it minimizes the distance between community members, while MFC$^+$ should outperform MFC when community members have consistent tastes, because it minimizes the distances between members and community center. To analyze differences in the performance of these two methods for different kinds of communities, we use Root Mean Square Distance (RMSD) for comparison. RMSD is defined as:

$$\bar{S}_h = \frac{2\sum_{u_i,u_j\in c_h} S_{ij}}{|C_h|\,(|C_h|-1)},$$

$$RMSD = \sqrt{\frac{2\sum_{u_i,u_j\in c_h}(S_{ij}-\bar{S}_h)^2}{|C_h|\,(|C_h|-1)}},$$

where $|C_h|$ is the number of users in community c_h and S_{ij} is the PCC defined in Section 4.1. Small RMSD means community members have consistent tastes.

Take for example dataset Dianping, where CPM is employed. From Figure 7(b), we can see that MFC$^+$ performs better than MFC for users in communities with small RMSD. When RMSD exceeds 0.3, the results of MFC are superior.

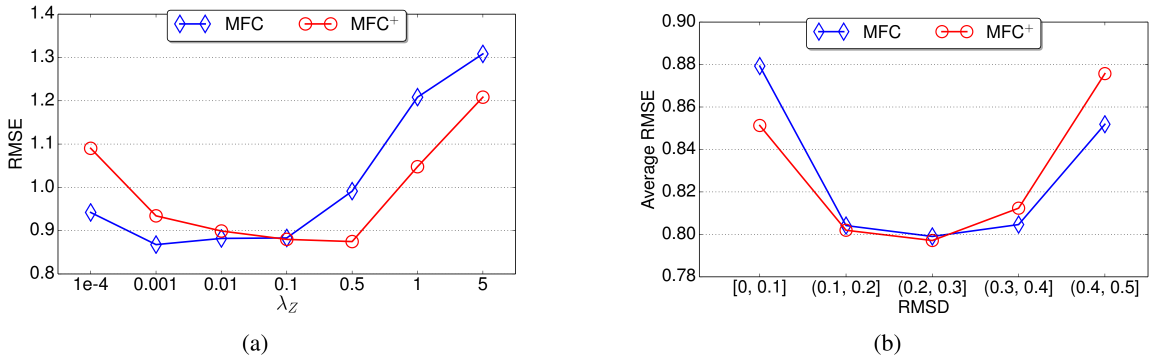


Figure 7: (a) Impact of parameter λ_Z on RMSE (Dianping) when CPM is employed. (b) Performance of MFC and MFC$^+$ over different kinds of communities in Dianping when CPM is employed.

6. DISCUSSION

In this paper, we improved the effectiveness of social-network based recommendation, by proposing two models that incorporate the overlapping community regularization into the matrix factorization framework differently.

The idea of utilizing communities to enhance the accuracy of rating prediction should not be confined to social network based recommender systems. When more information is available, it is also possible to consider *communities of items* and plug item-community regularization into our models. Communities of items can be obtained via item clustering based on the item features or the user-item bipartite network. In addition, explicit relationships (e.g., similar tastes and frequent interactions) can also be taken into consideration instead of only considering implicit social relationships. By considering information from both the implicit relationships and item network, traditional recommender systems without a supporting social network can benefit from social recommendation models and the idea of social collaborative filtering can be applied in a much broader context. Besides the above possible enhancements, in our future work, we also intend to further analyze the relationship between different community members and explore how these relationships affect users' rating behaviors, and, in turn, improve our approaches based on these results.

7. REFERENCES

[1] E. Airoldi, D. M. Blei, S. E. Fienberg, and E. P. Xing. Mixed membership stochastic blockmodels. In *NIPS*, pages 33–40, 2008.

[2] D. J. Crandall, D. Cosley, D. P. Huttenlocher, J. M. Kleinberg, and S. Suri. Feedback effects between similarity and social influence in online communities. In *KDD*, pages 160–168, 2008.

[3] M. Deshpande and G. Karypis. Item-based top-*N* recommendation algorithms. *ACM Trans. Inf. Syst.*, 22(1):143–177, 2004.

[4] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, pages 5–53, 2004.

[5] T. Hofmann. Collaborative filtering via gaussian probabilistic latent semantic analysis. In *SIGIR*, pages 259–266, 2003.

[6] M. Jamali and M. Ester. Trustwalker: a random walk model for combining trust-based and item-based recommendation. In *KDD*, pages 397–406, 2009.

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

[8] M. Jamali, T. Huang, and M. Ester. A generalized stochastic block model for recommendation in social rating networks. In *RecSys*, pages 53–60, 2011.

[9] R. Jin, J. Y. Chai, and L. Si. An automatic weighting scheme for collaborative filtering. In *SIGIR*, pages 337–344, 2004.

[10] A. Kohrs and B. Mérialdo. Clustering for collaborative filtering applications. In *CIMCA*, 1999.

[11] Y. Koren, R. M. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *IEEE Computer*, 42(8):30–37, 2009.

[12] H. Li, D. Wu, and N. Mamoulis. A revisit to social network-based recommender systems. In *SIGIR*, pages 1239–1242, 2014.

[13] F. Liu and H. J. Lee. Use of social network information to enhance collaborative filtering performance. *Expert Syst. Appl.*, 37(7):4772–4778, 2010.

[14] N. N. Liu and Q. Yang. Eigenrank: a ranking-oriented approach to collaborative filtering. In *SIGIR*, pages 83–90, 2008.

[15] H. Ma. An experimental study on implicit social recommendation. In *SIGIR*, pages 73–82, 2013.

[16] H. Ma, I. King, and M. R. Lyu. Learning to recommend with social trust ensemble. In *SIGIR*, pages 203–210, 2009.

[17] H. Ma, H. Yang, M. R. Lyu, and I. King. Sorec: social recommendation using probabilistic matrix factorization. In *CIKM*, pages 931–940, 2008.

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

[19] P. Massa and P. Avesani. Trust-aware recommender systems. In *RecSys*, pages 17–24, 2007.

[20] G. Palla, I. Derényi, I. Farkas, and T. Vicsek. Uncovering the overlapping community structure of complex networks in nature and society. *Nature*, 435(7043):814–818, 2005.

[21] F. Reid, A. F. McDaid, and N. J. Hurley. Percolation computation in complex networks. In *ASONAM*, pages 274–281, 2012.

[22] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl. Grouplens: An open architecture for collaborative filtering of netnews. In *CSCW*, pages 175–186, 1994.

[23] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *NIPS*, pages 1257–1264, 2007.

[24] B. M. Sarwar, G. Karypis, J. A. Konstan, and J. Riedl. Item-based collaborative filtering recommendation algorithms. In *WWW*, pages 285–295, 2001.

[25] Y. Shen and R. Jin. Learning personal + social latent factor model for social recommendation. In *KDD*, pages 1303–1311, 2012.

[26] L. Si and R. Jin. Flexible mixture model for collaborative filtering. In *ICML*, pages 704–711, 2003.

[27] S. Wasserman and K. Faust. *Social Network Analysis*. Cambridge Univ. Press, 1994.

[28] J. Yang and J. Leskovec. Overlapping community detection at scale: a nonnegative matrix factorization approach. In *WSDM*, pages 587–596, 2013.

[29] J. Yang, J. J. McAuley, and J. Leskovec. Community detection in networks with node attributes. In *ICDM*, pages 1151–1156, 2013.

[30] X. Yang, H. Steck, and Y. Liu. Circle-based recommendation in online social networks. In *KDD*, pages 1267–1275, 2012.

[31] Y. Zhang and J. Koren. Efficient bayesian hierarchical user modeling for recommendation system. In *SIGIR*, pages 47–54, 2007.