Additive Co-Clustering with Social Influence for Recommendation

Xixi Du
Beijing Key Lab of Traffic Data Analysis and Mining
Beijing Jiaotong University
Beijing, China 100044
15120391@bjtu.edu.cn

Huafeng Liu
Beijing Key Lab of Traffic Data Analysis and Mining
Beijing Jiaotong University
Beijing, China 100044
15120419@bjtu.edu.cn

Liping Jing
Beijing Key Lab of Traffic Data Analysis and Mining
Beijing Jiaotong University
Beijing, China 100044
lpjing@bjtu.edu.cn

ABSTRACT

Recommender system is a popular tool to accurately and actively provide users with potentially interesting information. For capturing the users’ preferences and approximating the missing data, matrix completion and approximation are widely adopted. Except for the typical low-rank factorization-based methods, the additive co-clustering approach (ACCAMS) is recently proposed to succinctly approximate large-scale rating matrix. Although ACCAMS efficiently produces effective recommendation result, it still suffers from the cold-start problem. To address this issue, we propose a **S**ocial **I**nfluence **A**dditive **C**o-**C**lustering method (SIACC) by making use of user-item rating data and user-user social relations.

The main idea of SIACC is to extract the social influences from the social network, integrate them to additive co-clustering for effectively determining the user clusters and item clusters, minimize the loss error by backfitting the residuals of data approximation in the previous iteration, and finally improve the recommendation performance. In order to take advantage of social influence, we present a **g**raph-regularized **w**eighted-**F**uzzy **C**-**M**eans algorithm (gwFCM) to cluster users. gwFCM has ability to identify user groups from both rating and social information. Specifically, gwFCM makes sure that a pair of users have similar cluster membership if they have direct social relation (denoted as local social influence), and that the user with higher reputation (denoted as global social influence) plays a dominate role in clustering process. The reasonable user clusters obtained by gwFCM can benefit the item clustering, which will leverage the additive co-clustering processing and further improve the recommendation performance. A series of experiments on three real-world datasets have shown that SIACC outperforms the existing popular recommendation methods (PMF and ACCAMS) and social recommendation methods (SoReg, TrustMF, Locabal and SPF), especially on the cold-start users recommendation and running time.

CCS CONCEPTS

•Information systems →Recommender systems; Clustering and classification;

RecSys’17, August 27–31, 2017, Como, Italy

ISBN 978-1-4503-4652-8/17/08…$15.00
DOI: http://dx.doi.org/10.1145/3109859.3109883

KEYWORDS

Recommendation System, Additive Co-Clustering, Social Influence, Cold-Start Users

1 INTRODUCTION

As an indispensable technique to tackle the information overload problem, recommender system is nowadays ubiquitous in various domains and e-commerce platforms. It aims to provide the online users with the potentially interesting information, such as recommendation of products at Amazon, books/movies/musics at Douban, research articles at CiteULike and etc. A high quality recommendation system can not only capture users’ individualized preferences, but also increase both satisfaction for users and revenue for content providers.

Among the existing recommendation methods, matrix approximation techniques, including low-rank matrix factorization and additive co-clustering, are widely adopted in the academia and the industry. The low-rank matrix factorization is a typical model-based collaborative filtering approach [1], which is relatively simple and effective. This kind of methods attempt to automatically predict the interest of users through matrix factorization [2, 21]. Even though these methods have promising results, they are faced with two main challenges: cold-start and computation complexity. When there are a large amount of users or items, they have to incur the cost of latent factor identification from the high-dimension matrix. Furthermore, these methods can not provide satisfactory recommendations for cold-start users or cold-start items.

In order to improve the performance on cold-start users, researchers have demonstrated that recommendation models with the aid of social relations can provide better recommendations. This kind of recommendation is based on the social rationale, i.e., the user’s interest is affected by his/her connected neighbors. To date, a series of matrix factorization based social recommendation methods have been proposed by taking available social relations as constraints to design models. For example, the social relations are employed to adjust the latent user factor [6, 14] and latent user preference [5] respectively, or determine the user latent factor by simultaneously factorizing user-user social matrix and user-item rating matrix [8, 15, 23, 27, 28]. Although these methods take advantage of the social relations to improve the recommendation accuracy, the additional social information makes the model more complicated and slows down the recommendation process, i.e., their computational complexity increases with the increasing size of users and items.

For handling large-scale user-item rating data, Beutel et al. [3] recently proposed an additive co-clustering model to succinctly approximate the rating matrix (ACCAMS). ACCAMS takes linear combinations of co-clusterings to encode the factorial nature of large-scale matrix without incurring the cost of high-dimension matrix factorization. ACCAMS has been theoretically and empirically proved efficient, however, it only considers the preference matrix and still suffers from cold-start problem. Therefore, in this work, we present a Social Influence Additive Co-Clustering method (SIACC) by making use of user-item rating data and social relations among users to improve the prediction quality and speed up the recommendation process.

The main idea of SIACC is to integrate the social influence to the ACCAMS. Firstly, the global and local social influence of users are extracted from social network. Global social influence indicates the reputation of each user in the whole social network. Local social influence reveals the correlations between users and their neighborhoods in the social network. Secondly, a **g**raph-regularized **w**eighted **f**uzzy **c**-**m**eans clustering algorithm (gwFCM) is proposed to identify the user clusters and the corresponding cluster centers. In gwFCM, the local social influence is used to constraint the user cluster membership via a graph regularizer term so that a pair of users obtain similar cluster memberships if they have local social influence. The global social influence of each user is used to weight data fitting term so that the user with higher reputation plays a dominate role in clustering process. Thirdly, the item clusters are identified by applying k-means (KM) clustering algorithm on the user cluster centers. Following ACCAMS, SIACC iteratively applies gwFCM and KM to minimizing the approximation error by backfitting the residuals of the data approximation in the previous iteration.

The rest of the paper is organized as follows. The related work about matrix approximation-based recommendation systems is discussed in Section 2. Section 3 gives the proposed SIACC recommendation framework for rating matrix approximation, including the new user clustering algorithm (gwFCM) which exploits both rating data and social influence. In Section 4, we give the details of experiments on three real-world datasets and discuss the results. Finally, a brief conclusion and future work are given in Section 5.

2 RELATED WORK

Before reviewing the matrix approximation techniques for recommendation systems, we first introduce notations used in this paper. Suppose there are n users and m items. Let $R = \left[R_{ij}\right]_{n\times m}$ denote the user-item rating matrix. If the i-th user gives a rating to the j-th item, R_{ij} is the rating score, otherwise $R_{ij} = 0$. Users can establish social relations to each other. The unknown cell set in R is denoted as Ω, i.e., the values of R_Ω have to be predicted. Let $A = [A_{iz}]_{n\times n}$ denote the user-user social relations where $A_{iz} = 1$ if the i-th and z-th users have social relation, otherwise $A_{iz} = 0$. Next, we will discuss the popular matrix factorization and additive co-clustering recommendation approaches.

2.1 Matrix Factorization

Matrix factorization (MF) techniques have been widely employed for recommendation systems [2, 10, 21, 22]. The basic idea of these approaches is employing low-rank matrix factorization on the rating matrix R to identify the user factor and item factor, which can be implemented by solving the following optimization problem,

$$\min \frac{1}{2}\Big(\sum_{R_{ij}\neq 0} (R_{ij} - {h_i}^{\mathrm{T}} b_j)^2 + \lambda(\|H\|_F^2 + \|B\|_F^2)\Big). \tag{1}$$

where $H = [h_1, h_2, \cdots, h_n] \in \mathbb{R}^{r\times n}$ with h_i indicating the i-th user in the latent space, $B = [b_1, b_2, \cdots, b_m] \in \mathbb{R}^{r\times m}$ with b_j indicating the j-th item, and r is the number of features in the latent space. Then, the incomplete rating matrix R can be approximated by $H^T B$, and the unknown component R_{ij} can be estimated by $h_i^T b_j$.

The MF-based recommendation methods usually assume that users are independent and identically distributed, which ignores the connection among users. Meanwhile, they can not provide satisfactory recommendations to the users having few or no rating information. Recently, more and more researchers investigated the social recommendation based on the social rationale, i.e., the user's social neighbors can affect his/her preference. The main idea is to integrate the social relations into the MF model via two strategies. The first strategy mainly factorizes the rating matrix and adjusts the latent user factor (H) [6, 8, 9, 13, 14, 16] or latent user preference [5] by his/her neighbors. The second strategy simultaneously factorizes rating matrix and social matrix to identify the latent user/item factor (H) [15, 23, 27, 28]. Even though these methods improve the prediction quality, the social relation information makes the models more complicated and aggravates the computational complexity of model training process.

2.2 Additive Co-Clustering

Recently, Beutel et al. [3] presented an additive co-clustering model to approximate the rating matrix succinctly (ACCAMS). To predict the missing values of rating matrix R, ACCAMS aims to iteratively learn s stencils $\left\{\mathcal{S}(T^l, c^l, d^l)\right\}_{l=1}^{s}$ and approximate R via

$$\min_{\{T^l, c^l, d^l\}} \left\| R - \sum_{l=1}^{s} S\left(T^l, c^l, d^l\right) \right\|_F^2 \tag{2}$$

In the l-th iteration, $c^l \in \{1, \ldots k_n\}^n$ is a user-cluster index vector, $d^l \in \{1, \ldots k_m\}^m$ is a item-cluster index vector, k_n is the number of user clusters, and k_m for item clusters. The user and item clusters are determined by applying k-means on the l-th input data R^l, where $R^{l+1} = R^l - S^l$ (with $R^1 = R$) as follows.

$$\min \sum_{i=1}^{n} \sum_{k=1}^{k_n} U_{ik} \left\| R_{i\cdot}^l - X_{k\cdot} \right\|_2^2 \tag{3}$$

where$X = [X_{kj}] \in \mathbb{R}^{k_n \times m}$ with $X_{k\cdot}$ is the center of the k-th cluster, $U = [U_{ik}] \in \mathbb{R}^{n\times k_n}$ (s.t. $U_{ik} \in \{0,1\}$, $\sum_{k=1}^{k_n} U_{ik} = 1\ \forall i$) is the user cluster membership matrix, $U_{ik} = 1$ denotes that the i-th user belongs to the k-th cluster, otherwise $U_{ik} = 0$. In this case, user-cluster index vector c^l can be identified by

$$c_i^l = \{k | U_{ik} = 1\}, i = 1, \ldots n. \tag{4}$$

Similarly, the item clusters and d^l can be identified by applying k-means on the user cluster centers X. The template $T^l \in \mathbb{R}^{k_n \times k_m}$ is generated according to c^l, d^l and the input data R^l, so that T_{pq}^l

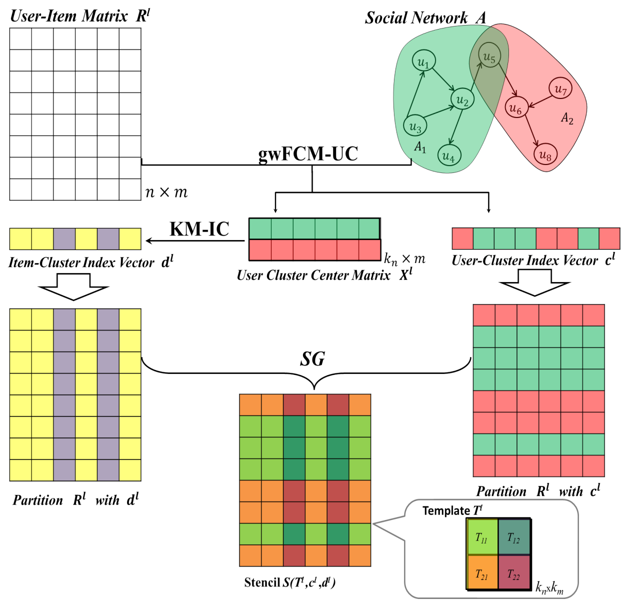


Figure 1: The Framework of SIACC Recommendation.

is the mean of $\{R^l_{ij} | c^l_i = p, d^l_j = q\}$. Finally, the rating that the i-th user gives to the j-th item can be predicted by $\sum_{l=1}^{s} S(T^l, c^l, d^l)_{ij} = \sum_{l=1}^{s} T^l_{pq}$.

Although ACCAMS has been proved efficient and effective for recommendation, it still has some drawbacks. Firstly, each user is required to have a distinct membership to one single cluster because of k-means, however, ambiguity may exist in real world datasets, thus it is necessary to allow each user have memberships in all clusters. Secondly, only the sparse user-item preference information is used in ACCAMS, which results in cold-start problem like the classical MF model. In order to solve these problems, we will propose a new social influence additive co-clustering recommendation framework (SIACC) to make use of both user preference information and user social information, and efficiently approximate the incomplete user-item rating matrix.

3 THE PROPOSED SIACC RECOMMENDATION METHOD

The key step of ACCAMS is to get the user and item clusters, so that users in the same cluster are likely to share similar tastes. Although the user-item rating data can provide some preference clues to find the user clusters, it is limited due to its sparsity. The booming online social networking websites, as complementary resources, offer more opportunities to mine the correlations between users. In social network, social relations are established according to the users' trust of one another. Such social relations include two perspectives: local and global. From the local view, according to social correlation theories [17, 19], the user correlations indicate that users who have direct social relations usually share similar tastes with each other. From the global view [18], the user correlations indicate that the users with higher reputation in social network, to a large extent, influence other users' decision when they seek suggestions. In literatures, more and more researches focus on exploiting such correlations to improve the performance of recommendation [23], but most of them focus on local social correlations [6, 8, 9, 13, 14, 16, 27, 28].

In this section, we will present a social influence additive co-clustering recommendation framework (SIACC) as shown in Figure 1. It aims to integrate the global and local user correlations into additive co-clustering for efficiently approximate the incomplete rating matrix. SIACC is an iterative matrix approximation process including three parts: user clusters generation (gwFCM-UC), item clusters generation (kM-IC), and stencil generation (SG), which will be described next.

3.1 User Cluster Generation (gwFCM-UC)

In real applications, one user may have various tastes. For example, a young person may like soap opera as well as war movie. Similarly, in social network A (as shown in Figure 1), the user u_5 may belong to the left social community as well as the right community. Thus, ambiguity really exists in the user's cluster membership. Given the input data R^l in the l-th iteration, for capturing such ambiguity, we adopt fuzzy c-means [7] instead of k-means as the base clustering method to find the user clusters by minimizing

$$\sum_{i=1}^{n} \sum_{k=1}^{k_n} U_{ik}^{\beta} \left\| R^l_{i\cdot} - X_{k\cdot} \right\|_2^2 \tag{5}$$

Here $\beta \geqslant 1$ is the fuzzy factor, and its value is set as 2 following [7]. $X \in \mathbb{R}^{k_n \times m}$ is the user cluster center set, $U \in \mathbb{R}^{n \times k_n}$ (s.t., $U_{ik} \in [0, 1]$, $\sum_{k=1}^{k_n} U_{ik} = 1, \forall i$) is the user cluster membership matrix with component U_{ik} indicating the probability that the i-th user belongs to the k-th cluster.

According to social influence theory[18], the global perspective of social relation reveals the reputation of a user in the whole social network, the user with high reputation usually has a greater influence. In other words, user reputation plays an important role in recommendation just like celebrity effect [23]. Following [20], the reputation of each user can be calculated via

$$\omega_i = \frac{1}{1 + log(r_i)}, \tag{6}$$

where $r_i \in [1, n]$ donates the the i-th user's reputation rank which can be obtained by the most popular algorithm (e.g., PageRank). For example, $r_i = 1$ indicates that the i-th user has the highest reputation in the whole social network, then its reputation ω_i should have the largest value. In this case, we can weight each user when determining the user clusters by minimizing

$$\sum_{i=1}^{n} \sum_{k=1}^{k_n} \omega_i U_{ik}^2 \left\| R^l_{i\cdot} - X_{k\cdot} \right\|_2^2, \tag{7}$$

having the same constraints with (5). This weighted FCM has ability to let the users with higher reputation dominate the whole clustering process.

Meanwhile, social correlation theories[17] indicate that users with similar tastes are more likely to be socially connected. In other words, users with stronger social correlations are more likely to share similar interests than those who have weak or no correlations. Researchers have shown that exploiting such direct social relations among connected users can improve the recommendation quality [6, 8, 9, 13, 14, 16, 23, 27]. Following [26], such social relation strength (denoted as $Q \in \mathbb{R}^{n \times n}$) can be measured via the user preference cosine similarity as follows.

$$Q_{iz} = \begin{cases} \frac{\sum_j R_{ij} R_{zj}}{\sqrt{\sum_j R_{ij}^2}\sqrt{\sum_j R_{zj}^2}} & u_i \text{ connected with } u_z, \text{ i.e., } A_{iz} = 1, \\ 0 & otherwise \text{ i.e., } A_{iz} = 0. \end{cases} \tag{8}$$

Recall the weighted fuzzy c-means model (7), it outputs a cluster membership vector for each user, then we can enforce the local social correlations on the cluster membership vectors of the corresponding pair of users (the i-th and z-th users) via minimizing

$$\sum_{i,z} Q_{iz} \left\|U_{i\cdot} - U_{z\cdot}\right\|_2^2 \tag{9}$$

This term is also called graph regularizer [4]. Minimizing this term can make the user memberships have the similar local structure with that demonstrated in the social relation strength Q.

By combining (7) and (9), the user clusters can be identified by integrating global and local social influence into fuzzy c-means by minimizing

$$\begin{aligned} J_u = &\sum_{i=1}^{n} \sum_{k=1}^{k_n} \omega_i U_{ik}^2 \left\|R_{i\cdot}^l - X_{k\cdot}\right\|_2^2 \\ &+ \lambda \sum_{i,z} Q_{iz} \left\|U_{i\cdot} - U_{z\cdot}\right\|_2^2 \end{aligned} \tag{10}$$

s.t:

$$U_{ik} \in [0,1], \forall i,k; \qquad \sum_{k=1}^{k_n} U_{ik} = 1, \forall i. \tag{11}$$

where the parameter λ is introduced to control the contributions of rating information and social influence to user clustering process, which can be tuned via cross-validation technique. We call this model as graph-regularized and weighted FCM (gwFCM).

Minimizing (10) is a constrained optimization problem, which can be solved using Lagrange Multiplier Method. Its Lagrange function is

$$\begin{aligned} J_u = &\sum_{i=1}^{n} \sum_{k=1}^{k_n} \omega_i U_{ik}^2 \left\|R_{i\cdot}^l - X_{k\cdot}\right\|^2 \\ &+ \lambda \sum_{i,z} Q_{iz} \left\|U_{i\cdot} - U_{z\cdot}\right\|^2 + \eta_i \left(\sum_{k=1}^{k_n} U_{ik} - 1\right) \end{aligned} \tag{12}$$

where η_i is Lagrange Multiplier.

The local minimum about U and X is the point where the first-order derivative is 0 as follows.

$$\frac{\partial J_u}{\partial X_{kj}} = -2 \sum_{i=1}^{n} \omega_i U_{ik}^2 \left(R_{ij}^l - X_{kj}\right) = 0 \tag{13}$$

$$\begin{aligned} \frac{\partial J_u}{\partial U_{ik}} = &\, 2\omega_i U_{ik} \left\|R_{i\cdot}^l - X_{k\cdot}\right\|_2^2 \\ &+ 2\lambda \sum_{z} Q_{iz} \left(U_{ik} - U_{zk}\right) + \eta_i = 0 \end{aligned} \tag{14}$$

By substituting η_i, we can get the iterative formula for updating X_{kj} and U_{ik}:

$$X_{kj} = \frac{\sum_{i=1}^{n} \omega_i R_{ij}^l U_{ik}^2}{\sum_{i=1}^{n} \omega_i U_{ik}^2} \tag{15}$$

$$U_{ik} = \frac{\sum_{c=1}^{k} \omega_i \left\|R_{i\cdot}^l - X_{c\cdot}\right\|_2^2 + \lambda \sum_z Q_{iz} U_{zk}}{\omega_i \left\|R_{i\cdot}^l - X_{k\cdot}\right\|_2^2 + \lambda \sum_z Q_{iz}} \tag{16}$$

Once having the user cluster membership U, the user cluster index vector c^l can be determined by

$$c_i^l = p, \text{ where } p = \arg\max_k \{U_{ik}\}_{k=1}^{k_n}. \tag{17}$$

3.2 Item Cluster Generation (KM-IC)

The set of user cluster centers ($X \in \mathbb{R}^{k_n \times m}$), to some extent, represents the current rating matrix R^l in a high level. Thus, for the sake of computational simplicity, we apply k-means on the user cluster centers X instead of the original data R^l to find the item clusters by minimizing

$$\min \sum_{j=1}^{m} \sum_{k=1}^{k_m} V_{jk} \left\|X_{\cdot j} - Y_{\cdot k}\right\|_2^2 \tag{18}$$

s.t.

$$V_{jk} \in \{0,1\}, \forall j,k; \qquad \sum_{k=1}^{k_m} V_{jk} = 1, \forall j.$$

Here $Y \in \mathbb{R}^{k_n \times k_m}$ is the item cluster center set, $V \in \mathbb{R}^{m \times k_m}$ is the user cluster indicator matrix with component $V_{jk} = 1$ if the j-th item belongs to the k-th cluster, otherwise $V_{jk} = 0$. In this case, we can determine the item cluster index vector d^l by

$$d_j^l = q, \text{ where } V_{jq} = 1. \tag{19}$$

3.3 Stencil Generation (SG)

Following [3], we can generate the stencil $S(T^l, c^l, d^l)$ according to c^l, d^l and the input data R^l. More specifically, the template T^l is a $k_n \times k_m$ matrix, and its (p,q)-th cell value is calculated by

$$T_{pq}^l = mean\left\{R_{ij}^l | c_i^l = p, d_j^l = q\right\} \tag{20}$$

Then the stencil $S \in \mathbb{R}^{n \times m}$ can be identified via

$$S\left(T^l, c^l, d^l\right)_{ij} = T_{c_i^l d_j^l}^l. \tag{21}$$

For each iteration, after obtaining the stencil $S(T^l, c^l, d^l)$, we can calculate the residual between the input data R^l and the stencil, which is used as the input data for the next co-clustering iteration, i.e.,

$$R^{l+1} = R^l - S\left(T^l, c^l, d^l\right). \tag{22}$$

3.4 SIACC Recommendation

By iteratively co-clustering the residual between input data and the corresponding stencil, we can minimize the loss between the original rating matrix R and the linear-combination of all stencils. In other words, the user-item rating matrix R can be well approximated by the linear-combination of stencils and then do recommendation. The process of stencil generation will be repeated until the stopping condition is satisfied, as shown in Algorithm

1. The linear combination of stencils can be taken as an additive model of co-clustering on both users and items.

Algorithm 1 SIACC Recommendation Method

Require: Rating matrix R with n users and m items, social network A, number of user clusters k_n, number of item clusters k_m, number of stencils s, set of unknown cells Ω.
  Compute the reputation ω_i for each user via (6) based on A;
  Compute social relation strength matrix Q via (8) based on A and R;
  Set $R^1 = R$, $\hat{R}_\Omega = 0$;
  for $l = 1$ to s **do**
    Compute user-cluster index vector c^l and user cluster centers X from R^l and Q via (17) and (15);
    Compute item-cluster index vector d^l from X via (19);
    for $p, q \in \{1, \ldots k_n\} \times \{1, \ldots k_m\}$ **do**
      Calculate the template T^l_{pq} using (20);
    end for
    Fill in the stencil $S\left(T^l, c^l, d^l\right)$ via (21);
    $R^{l+1} \leftarrow R^l - S\left(T^l, c^l, d^l\right)$;
    $\hat{R}_\Omega = \hat{R}_\Omega + S\left(T^l, c^l, d^l\right)_\Omega$.
  end for
  Output the predicated values $\hat{R}_\Omega$ on unknown cells.

The space complexity of the proposed SIACC framework depends on the size of stencil. Its cost includes storing user-cluster index vector c, item-cluster index vector d and template T, and finally has a upper bound $(n \log_2 k_n + m \log_2 k_m + 32 k_n k_m)$ bits [3]. Comparing with the MF-based model with $32(n + m)r$ bits (the value of each user/item on each latent feature is recorded as a float with 32 bits), SIACC needs fewer space because in real application the number of latent features r is usually larger than the clusters $32 \log_2 k_n$ and $32 \log_2 k_m$.

For running time, the main computation complexity of SIACC is to iteratively find c, d and template T. In each iteration, the time of updating c is $O\Big((nnz(R) + nnz(Q))k_n\Big)$, the time of updating d is $O(mk_nk_m)$, the time of updating T is $O(k_nk_m)$, thus the total computation complexity is $O\Big(s(nnz(R) + nnz(Q) + mk_m)k_n\Big)$ with s stencils. Comparing with the MF-based social recommendation model (e.g., Locabal [23]) with $O\Big(t\,(nnz(R) + nnz(Q))\,r\Big)$ (t is the number of iterations), SIACC is much faster because in real application r is usually larger than k_n and k_m, and t is much greater than s. In next section, we will demonstrate the computation complexity using a series of experiments.

4 EXPERIMENTS

In this section, we will demonstrate the effect of parameters on SIACC, and the recommendation performance of the proposed SIACC by comparing it with four popular existing recommendation methods on three real-world datasets.

4.1 Dataset

In order to sufficiently validate the performance of SIACC, we consider the recommendation data including social information among users such as Ciao[1], Epinions[2], and Douban[3] which are usually used in literatures. The preference/rating scores in these three datasets are in five levels (from 1 to 5). The relations in the first two datasets are recorded in a who-trust-whom social network, i.e., a directed trust network, while in Douban, the relations are from friendship between users, i.e., they are recorded in a undirected network. In the experiments, we ignore the effect of direction, i.e., all social networks are considered as undirected networks. The details of three datasets are listed in Table 1. It can be seen that both rating matrix and social network are sparse, which exacerbates the difficulties of recommendation.

Table 1: Summary of experimental datasets

	Ciao	Epinions	Duban
♯ users (n)	7,375	49,290	129,490
♯ items (m)	106,797	139,738	58,541
♯ ratings	284,086	664,824	16,830,839
♯ relations	111,781	487,183	1,692,952
rating density	0.036%	0.010%	0.222%
social density	0.2055%	0.0201%	0.0202%

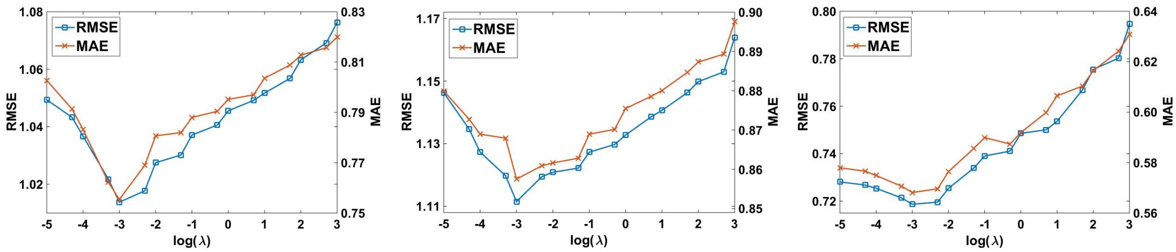
Five-fold cross-validation technique is used for training and testing. Specially, each data set is randomly split into five equal sized subsets, four subsets are used as the training set and the left one as testing set in each fold. Five folds are conducted to ensure all subsets are tested, and the average test performance is recorded as the final results.

4.2 Methodology

To comparatively demonstrate the performance of our method, the classical matrix factorization-based collaborative filtering method (PMF) [21] and the additive co-clustering model (ACCAMS) [3] are taken as baselines which only consider rating information to do recommendation. Since SIACC takes both rating and social information into account, it can be taken as social recommendation. The existing social recommendation methods can be roughly divided into two categories according to the usage of social information, thus four representative methods (two from each category) are used as baselines to compare SIACC. Among them SoReg [16] and SPF[5] employ the social relations to adjust the latent user factors. TrustMF[28] and Locabal [23] simultaneously factorize social matrix and rating matrix to learn the common latent user factor.

The optimal experimental settings for all methods are determined either by experiments or suggested by previous works. In the PMF, SoReg, and Locabal, the number of features in latent space is 10 for three datasets. The regularization parameter on latent factors is 0.001 for Douban, and 0.1 for Ciao and Epinions for these three methods. The parameter to control the contribution of social information is set to 1 in SoReg and Locabal. For ACCAMS, we tuned its parameters including number of user clusters k_n, number of item clusters k_m and number of stencils s.

[1] http://www.jiliang.xyz/trust.html
[2] http://www.trustlet.org/downloaded_epinions.html
[3] https://www.cse.cuhk.edu.hk/irwin.king.new/pub/data/douban



(a) Ciao (b) Epinions (c) Douban

Figure 2: Effect of parameter λ on SIACC for (a) Ciao, (b) Epinions and (c) Douban datasets in terms of RMSE and MAE.

Table 2: Comparisons of different recommendation methods from two views (All Users and Cold-Start (CS) Users).

	Datasets	Metrics	PMF	SoReg	TrustMF	Locabal	SPF	ACCAMS	SIACC
All Users	Ciao	MAE	0.8256	0.7964	0.7650	0.7561	0.7658	0.8084	**0.7554**
		RMSE	1.1310	1.0919	1.0309	1.0214	1.0330	1.0615	**1.0138**
	Epinions	MAE	1.1206	1.0506	1.0369	0.9374	1.0225	0.8971	**0.8515**
		RMSE	1.3654	1.3034	1.2859	1.1548	1.2489	1.1701	**1.1115**
	Douban	MAE	0.6230	0.6163	0.5931	0.5757	0.5976	0.5818	**0.5679**
		RMSE	0.7699	0.7636	0.7340	0.7190	0.7373	0.7309	**0.7183**
CS Users	Ciao	MAE	0.9343	0.9179	0.8493	0.8296	0.8593	0.7965	**0.7394**
		RMSE	1.1876	1.1548	1.0852	1.0774	1.0879	1.0441	**1.0221**
	Epinions	MAE	1.3508	1.2525	1.2459	1.0877	1.1062	0.9445	**0.8856**
		RMSE	1.4477	1.3492	1.3357	1.2651	1.2961	1.2207	**1.1971**
	Douban	MAE	0.8433	0.7952	0.7245	0.7244	0.7373	0.6570	**0.6389**
		RMSE	1.0239	0.9789	0.8881	0.8836	0.9180	0.8344	**0.8323**

In order to validate the prediction quality, two well known evaluation metrics, Root Mean Square Error(RMSE) and Mean Absolute Error(MAE), are adopted in experiments, which are defined by:

$$RMSE = \sqrt{\frac{1}{|\Omega|}\sum_{i,j}(R_{ij} - \hat{R}_{ij})^2} \tag{23}$$

$$MAE = \frac{1}{|\Omega|}\sum_{i,j}\left|R_{ij} - \hat{R}_{ij}\right| \tag{24}$$

where Ω is the unknown cells set and $|\Omega|$ is the number of test ratings. R_{ij} is the true rating value that the i-th user gave to the j-th item in testing data. $\hat{R}_{ij}$ is the predicted value from different methods. The smaller RMSE and MAE values indicate better recommendation result.

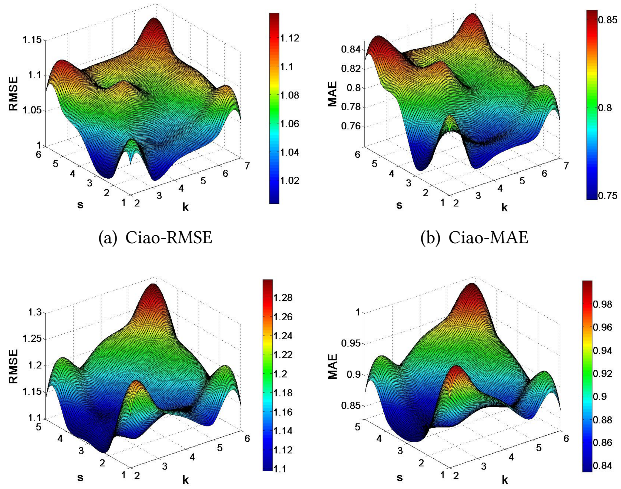
4.3 Results and Discussion

In the proposed SIACC method, there are four parameters to tune, including λ in (10) to trade off the contribution of rating information and social information on user clustering, and three parameters to generate the stencil (number of user clusters k_n, number of item clusters k_m and number of stencils s). Thus, the first experiments are conducted to test the effects of these parameters. Then a series of experiments are conducted to compare SIACC with the existing four methods on recommendation quality and running time.

4.3.1 Effect of parameter λ: Figure 2 shows the effect of regularizer parameter λ on three datasets. The results demonstrate that SIACC performs better as λ increases, reaches the best value at round $\lambda = 0.001$ for three datasets, and then decreases in performance as λ grows larger. We believe this is because a smaller λ can not efficiently make use of the social information to determine the user clusters, while a larger λ may ignore the importance of preference information, thus decreasing the efficacy of the recommendation. These results confirm that social information is helpful to cluster users and further improve the recommendation performance.

4.3.2 Effect of parameters k and s: For the sake of simplicity, we set $k_n = k_m = k$ in experiments following [3]. The effect of both k and s on SIACC is shown in Figure 3 for two sparse datasets (Ciao and Epinions). From Figure 3, it can be seen that the performance is destroyed when s is large, the main reason is that more stencils will overfit the training data. Meanwhile, we find that SIACC is relatively stable under varying k when fixing s. Thus, small k is enough to approximate the rating matrix.

4.3.3 Comparing with existing methods: In order to deep investigate the performance of recommendation, we compare SIACC with the baselines from two views. First, All Users view indicates that all ratings of unknown set Ω as the testing set. Second, Cold-start Users (CS Users) view means that the users who rate less than five items will be involved in the testing set. The best results are marked



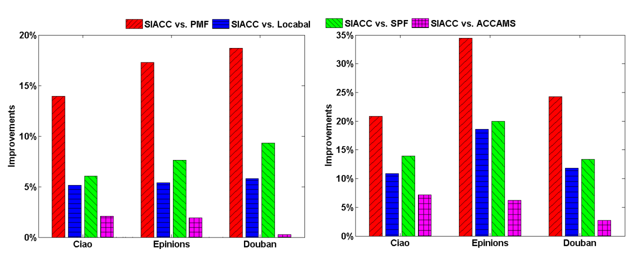
(a) Ciao-RMSE (b) Ciao-MAE

(c) Epinions-RMSE (d) Epinions-MAE

Figure 3: Effect of parameters $k = k_n = k_m$ and s on SIACC for Ciao and Epinions datasets in terms of RMSE and MAE.

in bold as shown in Table 2. It can be seen that SIACC consistently outperforms the baselines. In the first category of social recommendation methods, SPF is better than SoReg. In the second category, Locabal outperforms TrustMF, which means that considering both local and global social influence is more useful than only considering local social influence. Thus, the following comparisons focus on SIACC and PMF, ACCMAS, SPF, and Locabal.

The relative improvements that SIACC achieves relative to four baselines on three datasets are calculated. Since it is challenging to recommend items to cold-start users, we take the cold-start users as an example, as shown in Figure 4. Obviously, SIACC performs better than ACCAMS, which demonstrates that social relations benefit the additive co-clustering process. SIACC is superior to the existing social recommendation methods (Locabal and SPF), which means that integrating global and local social influence into additive co-clustering framework is more useful than into the matrix factorization model to approximate the rating matrix.



(a) RMSE (b) MAE

Figure 4: The relative improvements of SIACC vs. four baselines on could-start users in three datasets in terms of (a) RMSE and (b) MAE.

Even though the relative improvements are small, small improvements can lead to significant differences of recommendations in practice [11]. Thus, we assess the statistical significance using t-test. Table 3 lists the p-value between SIACC and four baselines for MAE on the cold-start users in three datasets. To be excited, the p values are 0.006 or better for all cases. Similarly, p values are 0.01 or better for RMSE, thus we can say SIACC significantly improve the recommendation performance.

Table 3: Statistical significance (p-value) obtained by SIACC vs. four baselines for MAE on Cold-Start Users.

Datasets	vs. PMF	vs. Locabal	vs. SPF	vs. ACCAMS
Ciao	5.54E-07	1.59E-04	2.12E-05	6.42E-03
Epinions	4.06E-09	9.22E-10	4.97E-10	1.66E-04
Douban	2.56E-11	2.99E-08	1.29E-04	4.93E-03

Table 4: Comparisons of SIACC vs. Locabal on three datasets in terms of running time (seconds).

	Ciao	Epinions	Douban
Locabal	302.37	871.32	1592.91
SIACC	113.37	219.52	932.60

Finally, we evaluate the efficiency of SIACC by comparing it with Locabal, because these two methods consider both global and local social influence. Meanwhile, Locabal is the best social recommendation method among the baselines. These two algorithms are implemented in C++ at the hots with Intel(R) Xeon(R) 2.0GHz CPU E7-4820 v2 having 64GB memory. The operating system is Red Hat Enterprise Linux. Table 4 lists the average running time of five times (for 5-fold corss-validation) on three datasets. Again, the proposed SIACC is faster than Locabal, and the performance improvements of SIACC over Locabal are 1.7× or better. This result demonstrates that additive co-clustering framework is much efficient than low-rank factorization on matrix approximation even integrating extra information, which is consistent to the result obtained in [3].

5 CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a new social influence additive co-clustering method for social recommendation. Both global and local social relations among users are effectively integrated into co-clustering process to generate user/item clusters and stencils. This strategy can iteratively co-cluster the residual between input data and the corresponding stencil, and then minimize the loss between rating matrix and the linear combination of stencils. The unknown rating values can be effectively and efficiently predicted by linear combining the stencils. The experiments on benchmark datasets have shown that SIACC consistently and significantly outperforms the popular existing methods.

In this work, we ignore the directional property of social network, however, it plays an important role to identify the social relations [27]. Meanwhile, only the direct social relations are considered here, indirect social relations (for the users who are not directly connected) [12, 24] have been proved useful for social recommendation. Thus, it is interesting to consider them to extend SIACC. Furthermore, we could integrate other available information such as reviews and item content, like [25], to design more effective and explainable recommender systems. Last but not least, we plan to design a model to simultaneously clustering users and items instead of the current two-phase strategy.

ACKNOWLEDGMENTS

The authors would like to thank Alex Beutel for discussion on co-clustering programming and experiments constructing. This work was partially supported by the NSFC Grant (61375062, 61632004) and CCF-Tencent Open Research Fund (RAGR20150116).

REFERENCES

[1] G. Adomavicius and A. Tuzhilin. 2005. Toward the next generation of recommender systems : A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering* 17, 6 (2005), 734–749.

[2] M. Aharon, M. Elad, and A. Bruckstein. 2006. SVD: An Algorithm for Designing Overcomplete Dictionaries for Sparse Representation. *IEEE Transactions on Signal Processing* 54, 11 (2006), 4311–4322.

[3] A. Beutel, A. Ahmed, and A. Smola. 2015. ACCAMS: Additive Co-Clustering to Approximate Matrices Succinctly. In *Proceedings of International Conference on World Wide Web*. 119–129.

[4] D. Cai, X. He, J. Han, and T. Huang. 2011. Graph regularized non-negative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 8 (2011), 1548–1560.

[5] A. Chaney, D. Blei, and T. Eliassi-Rad. 2015. A probabilistic model for using social networks in personalized item recommendation. In *Proceedings of ACM Conference on Recommender Systems*. 43–50.

[6] G. Guo, J. Zhang, and N. Yorkesmith. 2015. TrustSVD: Collaborative Filtering with Both the Explicit and Implicit Influence of User Trust and of Item Ratings. In *Proceedings of AAAI Conference on Artificial Intelligence*. 123–129.

[7] R. Hathaway and J. Bezdek. 2001. Fuzzy c-means clustering of incomplete data. *IEEE Transactions on Systems Man and Cybernetics Part B Cybernetics* 31, 5 (2001), 735–744.

[8] M. Jamali and M. Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of ACM Conference on Recommender Systems*. 135–142.

[9] M. Jiang, P. Cui, R. Liu, Q. Yang, F. Wang, W. Zhu, and S. Yang. 2012. Social contextual recommendation. In *Proceedings of ACM Conference on Information and Knowledge Management*. 45–54.

[10] L. Jing, P. Wang, and L. Yang. 2015. Sparse probabilistic matrix factorization by Llaplace distribution for collaborative filtering. In *Proceedings of the International Conference on Artificial Intelligence*. 1771–1777.

[11] Y. Koren, R. Bell, and C. Kolinsky. 2010. Factor in the neighbors: scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data* 4, 1 (2010), 1–24.

[12] H. Li, D.Wu, W. Tang, and N. Mamoulis. 2015. Overlapping community regularization for rating prediction in social recommender systems. In *Proceedings of ACM Conference on Recommender Systems*. 27–34.

[13] H. Li, D. Wu, and N. Mamoulis. 2014. A revisit to social network-based recommendation systems. In *Proceedings of ACM Special Interest Group on Information Retrieval*. 1239–1242.

[14] H. Ma, I. King, and M. Lyu. 2009. Learning to recommend with social trust ensemble. In *Proceedings of International ACM SIGIR Conference on Research and Development in Information Retrieval*. 203–210.

[15] H. Ma, H. Yang, M. Lyu, and I. King. 2008. SoRec:social recommendation using probabilistic matrix factorization. In *Proceedings of ACM Conference on Information and Knowledge Management*. 931–940.

[16] H. Ma, D. Zhou, C. Liu, M. Lyu, and I. King. 2011. Recommender systems with social regularization. In *Proceedings of ACM Web Search and Web Data Mining*. ACM, 287–296.

[17] P. Marsden and N. Friedkin. 1993. Network studies of social influence. *Sociological Methods and Research* 22, 1 (1993), 127–151.

[18] P. Massa. *Trust in E-services: Technologies, Practices and Challenges*. IGI Publishing.

[19] L. McPherson and J. Cook. 2001. Birds of a feather: homophily in social networks. *Annual Review of Sociology* 27, 1 (2001), 415–444.

[20] L. Page, S. Brin, R. Motwani, and T. Winograd. 1999. *Bringing order to the web.* Technical Report. Technical Report 1999-66, Stanford InfoLab.

[21] R. Salakhutdinov and A. Mnih. 2007. Probabilistic matrix factorization. In *Proceedings of International Conference on Machine Learning*. 880–887.

[22] R. Salakhutdinov and A. Mnih. 2008. Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. In *Proceedings of International Conference on Machine Learning*. 880–887.

[23] J. Tang, X. Hu, H. Gao, and H. Liu. 2013. Exploiting local and global social context for recommendation. In *Proceedings of International Joint Conference on Artificial Intelligence*.

[24] J. Tang, S. Wang, X. Hu, D. Yin, Y. Bi, Y. Chang, and H. Liu. 2016. Recommendation with social dimensions. In *Proceedings of AAAI Conference on Artificial Intelligence*. 251–257.

[25] C. Wu, A. Beutel, A. Ahmed, and A. Smola. 2016. Explaining reviews and ratings with PACO: poisson additive co-clustering. In *Proceedings of International Conference on World Wide Web*. 127–128.

[26] R. Xiang, J. Neville, and M. Rogati. 2010. Modeling relationship strength in online social networks. In *Proceedings of International Conference on World Wide Web*. 981–990.

[27] B. Yang, Y. Lei, J. Liu, and D. Liu. 2016. Social collaborative filtering by trust. *IEEE Transactions on Pattern Analysis and Machine Intelligence, online publication* (2016).

[28] B. Yang, L. Yu, D. Liu, and J. Liu. 2013. Social collaborative filtering by trust. In *Proceedings of International Joint Conference on Artificial Intelligence*. 2747–2753.