The Journal of Systems and Software 99 (2015) 109–119

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Recommender systems based on social networks

Zhoubao Sun[a], Lixin Han[a,*], Wenliang Huang[a], Xueting Wang[a], Xiaoqin Zeng[a], Min Wang[a], Hong Yan[b,c]

[a] College of Computer and Information, Hohai University, Nanjing 210024, China
[b] Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong, China
[c] School of Electrical and Information Engineering, University of Sydney, Sydney, Australia

ARTICLE INFO

Article history:
Received 15 February 2014
Received in revised form 31 July 2014
Accepted 17 September 2014
Available online 5 October 2014

Keywords:
Recommender system
Social network
Social-based recommender system

ABSTRACT

The traditional recommender systems, especially the collaborative filtering recommender systems, have been studied by many researchers in the past decade. However, they ignore the social relationships among users. In fact, these relationships can improve the accuracy of recommendation. In recent years, the study of social-based recommender systems has become an active research topic. In this paper, we propose a social regularization approach that incorporates social network information to benefit recommender systems. Both users' friendships and rating records (tags) are employed to predict the missing values (tags) in the user-item matrix. Especially, we use a biclustering algorithm to identify the most suitable group of friends for generating different final recommendations. Empirical analyses on real datasets show that the proposed approach achieves superior performance to existing approaches.

1. Introduction

Recommender systems have attracted much attention in the past decade. A recommender system is a software tool that supports users in identifying the most interesting items. There has been much work done on developing new approaches to recommender systems (Adomavicius and Tuzhilin, 2005; Brusilovsky and David, 2013). The research topic is still popular because of the abundance of practical applications that help users to deal with information overload and their great commercial value. Examples of the applications include recommending books, movies and some other commercial systems.

With the development of Web 2.0, the study of social-based recommender systems started. The traditional ones (Adomavicius and Tuzhilin, 2005; Brusilovsky and David, 2013) always ignore social interactions among users which can improve recommender systems. The fact is, when we are confused by multiple choices, we may turn to our related friends for the best recommendations, since they are those who we can reach for immediate advice. Hence, in order to provide more accuracy and personalized recommendation results, the social network information should be incorporated. Based on the above viewpoints, a few trust-based recommender systems (Jamali and Ester, 2010; Massa and Avesani, 2004; Ozsoy and Polat, 2013; Massa and Avesani, 2009; Bedi et al., 2007; Nazemian et al., 2012; Ma et al., 2008) which move an important progress forward have been proposed. The methods utilize the unilateral trust information to further improve traditional recommender systems. However, these methods have several inherent limitations and weaknesses that need to be solved. The noticeable weakness is the unilateral "trust relationship" problem. It is different from the concept "social relationship" which refers to the cooperative and mutual relationship between users. In addition, the other weaknesses are the impracticable hypothesis and the weak generalization ability. Obviously, the trust-based recommender systems may no longer be suitable. Therefore, the study of real social-based recommender systems appears on the screen. Additionally, the integration of social networks can theoretically improve the performance of traditional recommender systems. First, in terms of the prediction accuracy, the friendships among users improve the understanding of user ratings. Therefore, we can interpret user preferences more precisely. Second, as a matter of fact, the friendship between two users already indicates that they have things in common. Thus, the cold-start problem can be alleviated (Jamali and Ester, 2010; Massa and Avesani, 2004).

In order to solve the problems mentioned above, in our research, we focus on the social-based recommender system and, similar to Ma et al. (2011), propose an approach named RSboSN (Recommender Systems based on Social Networks) that integrates social network graph and the user-item matrix to improve the prediction accuracy of the traditional recommender systems. In the process of

* Corresponding author. Tel.: +86 83787200.
E-mail address: lixinhan2002@hotmail.com (L. Han).

http://dx.doi.org/10.1016/j.jss.2014.09.019
0164-1212/

recommendation, we mainly use the friendships among users and the tags labeled by the users to recommend. The user-item-tag can be considered as a two-dimensional matrix. We cluster the similar users to calculate the similarity between users and the correlation between a user and an item. The purpose in clustering is to identify the most suitable friends for realistic recommendation tasks. Based on the approach in Ma et al. (2011), the above two detailed aspects of social network information are employed in designing social regularization terms. We also take the situation into consideration that different friends may have dissimilar or even opposite tastes. Even if the friends of the same group focus on the same item, they may have different favorite degree. We have conducted experiments on real dataset to evaluate the performance of our approach on the prediction accuracy. The experiments show significant improvement over traditional and state-of-art social-based recommender systems in those aspects.

The remainder of the paper is organized as follows. Section 2 presents the overview of related work. Section 3 defines the problem and presents the details of the proposed approach. Section 4 presents the experiments results. Finally, we draw the conclusion in Section 5.

2. Related work

In this section, we review the approaches to recommender systems, including traditional recommender systems, trust-based recommender systems and social-based recommender systems.

2.1. Traditional recommender systems

The major traditional approaches are usually classified into three categories: collaborative filtering (Bellogin et al., 2013; Gunes et al., 2013; Chang and Hsiao, 2011; Bergner et al., 2012), content-based filtering (Illig et al., 2011; Barragáns-Martínez et al., 2010), and hybrid filtering (Adomavicius and Tuzhilin, 2005).

Collaborative filtering may be the most commonly used approach. The existing approaches can be grouped into two general classes: memory-based (Bellogin et al., 2013; Gunes et al., 2013) and model-based (Chang and Hsiao, 2011; Bergner et al., 2012). Memory-based approaches identify interesting items from other similar users' opinions by finding the nearest neighbor from a rating matrix. They essentially are heuristics that make rating perditions based on the entire collection of previously rated items by the users. Model-based approaches use the collection of ratings to learn a model, which is then used to make rating predictions. The cold-start and data sparsity problems are the drawbacks of collaborative filtering based approaches. Collaborative systems rely solely on users' preferences to make recommendations. Therefore, until a new item is rated by a substantial number of users, the recommender system would not be able to recommend it.

Content-based filtering approaches (Illig et al., 2011; Barragáns-Martínez et al., 2010) use tags to infer recommendations. Hence, the user may be recommended items similar to the ones the user preferred in the past. The approaches are limited by the tags that are explicitly associated with the items that the systems recommend. Another limitation is that two different items are indistinguishable if they are represented by the same tags. Another problem with limited content analysis is that, if there is a new user with few ratings, it would not be able to produce accurate recommendations.

Hybrid filtering based approach (Adomavicius and Tuzhilin, 2005), on the other hand, combines both content-based filtering and collaborative filtering approaches to produce a recommendation. It can address the problems of content-based and collaborative systems. Different ways to combine collaborative and content-based approaches into a hybrid recommender system can generate different results.

2.2. Trust-based recommender systems

The development of the traditional approaches is very mature, but they are all based on the assumption that users are independent. However, there are actually friendships among users. Based on the above assumption, many researchers have paid attention to trust-based recommender systems which combine the trust social information to further improve traditional approaches.

Many trust-based approaches have been proposed and widely applied to various areas in academia and industry (Massa and Avesani, 2004; Ozsoy and Polat, 2013; Massa and Avesani, 2009; Bedi et al., 2007; Nazemian et al., 2012; Ma et al., 2008). Among these works, the model proposed in Massa and Avesani (2009) is the most popular one which has already been used in a practical application. Bedi et al. (2007) proposed one for the semantic web. The approach uses the web of trust to generate the recommendations. Nazemian et al. (2012) proposed a trust-aware approach which uses distrust metric to improve the accuracy of recommender systems. The relationships between users are calculated by propagating trust. Traditional approaches are used here, such as collaborative filtering. The experiments on the real datasets demonstrate that it can improve accuracy of recommender systems significantly while not reducing the coverage. Ma et al. (2008) proposed a factor analysis approach based on the probabilistic graphical model. It regards the user feature matrix as the link between user-item matrix and the users' trust network. The results prove that it is effective. However, the recommender progress with the real world recommendation process is inconsistent. Hence, they proposed another ensemble approach which calculates users' ratings by considering their own taste and the trusted users' favors at the same time. The experiments show that this approach can be used to develop a better recommendation model.

The trust-based recommender systems have been proved effective and achieved great progress forward. However, as analyzed in Section 1, they have several inherent limitations and weaknesses that need to be solved.

2.3. Social-based recommender systems

In this paper, the concept "social recommender systems" is defined as combining the social network information which can affect personal behaviors on the web, such as the interactive information among users and the information of tags, to improve recommender systems. With the development of social network in recent years, how to utilize social network information has become a hot issue and been studied in many applications.

He and Chu (2010) proposed a probabilistic model to make personalized recommendations from the social network information, especially the influence from social friends. The experiments on real dataset reveal that the friends have a similar tendency to select the same items. In addition, the approach can solve the data sparsity and cold start issues. Domingos and Richardson (2001), Richardson and Domingos (2002) proposed approaches to detect the potential customers from the social network. They calculated the conditional probability to know whether a customer would purchase a product given the adoption values of his friends. The customers have the higher values if they have more social contacts and influence them. The approach regards the probability as the combinational production of the user's internal probability of purchasing and the external effects from his friends. Lu et al. (2010) proposed a generic framework to improve the review quality prediction by incorporating authors' identities and social networks. In addition, they added some constrains to the text-based predictor. Liu and Lee

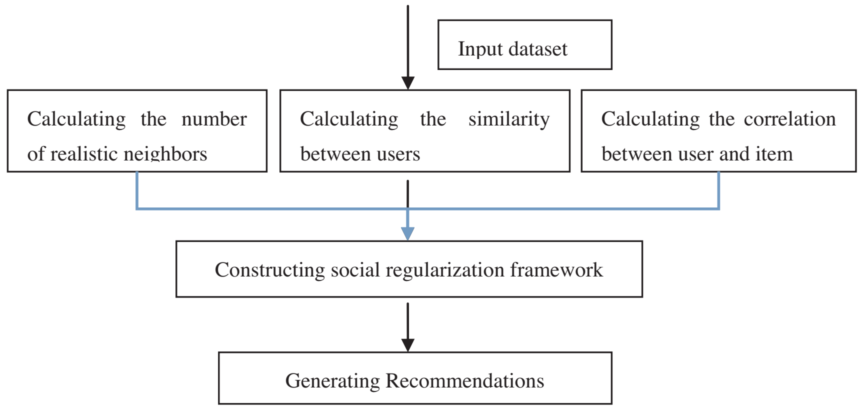


Fig. 1. The flowchart of the RSboSN framework.

(2010) developed a way to increase recommendation effectiveness by incorporating social network information into collaborative filtering. Mei et al. (2008) used social network analysis to help solve the problem of text mining. Chang and Chu (2013) proposed a recommendation mechanism which calculates similarity among users and users' trust-ability and analyzes information collected from the social networks. Ma et al. (2011) proposed methods to improve the prediction accuracy of traditional recommender systems. They developed a matrix factorization framework with social regularization and a factor analysis approach based on probabilistic matrix factorization to solve the data sparsity and poor predication accuracy problems by utilizing users' social network. Based on the approaches in Ma et al. (2011), Wang and Huang (2014) incorporated the friendships as the regularization term to improve the final prediction accuracy.

However, the social recommendation problem is not well studied in the above work. For example, in Ma et al. (2011), the authors did not incorporate the practical situation of users' friendships to generate recommendations. They assumed that a user's preference should be similar to that of him/her social network. This situation does not conform to the actual situation. He and Chu (2010), Domingos and Richardson (2001), Richardson and Domingos (2002) assume the users are correlated when they establish social relations, the users' favors are similar or influenced by their connected friends whom they are socially connected to. However, social relations can be friendships, trust relations or memberships, sometimes the low cost of social relation formation may lead to social relations with heterogeneous results. In Lu et al. (2010), Liu and Lee (2010), Mei et al. (2008), Chang and Chu (2013), they used social network nominally or just simplified the concept of social network. In Wang and Huang (2014), the authors integrated the friendships to generate the recommendations, but they did not distinguish the different friendships between users. They treated all the friendships equally in all situations.

Compared with the above mentioned work, in this paper, we treat friends who have different favors in different ways. We calculate the similarity between users based on the favors of each friend, which can improve the accuracy of recommendation. We cluster the suitable group of friends and calculate the pair-wise user similarities and the similarities among users and items. We use a biclustering algorithm to mine the "true" friends who have the similar favors to generate the final recommendations rather than those who only have the connections but with different or even opposite favors. The "true" friends play decisive roles in generating final recommendations. We integrate friendships and tags information to analyze the social recommendation problem based on matrix factorization framework.

3. Social recommendation framework

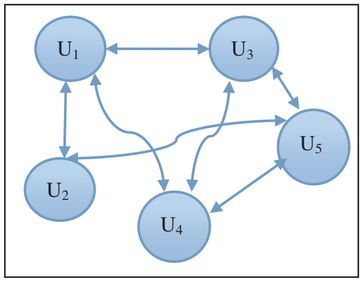
In this section, we first use a synthetic example to illustrate some definitions and abbreviations to social recommendation which is used throughout the paper (see Section 3.1). Then we describe the model which integrates with social network information (see Section 3.2). The brief flowchart of our algorithm is shown in Fig. 1. We cluster the users to obtain friendships and calculate the similarities (see Section 3.3). Lastly, we interpret how to utilize regularization terms to model the framework (see Section 3.4).

3.1. *Definition*

Fig. 2(a) shows the typical friends network graph. There are 5 users (nodes) with 7 relations (edges) between users. Each bidirectional edge represents the connection between two users, which is different from the trust network. The users often rate some items either on a 5-point integer scale (the bigger the better) or on tags to express the level of the favor of each item. The target is to predict the missing values or tags of the user-item matrix which is illustrated in Fig. 2(c). The relationships among users are illustrated in Fig. 2(b); the value 1 refers to the connection between two users.

In Fig. 2(c), T_{ij} denotes the tags (value) user *i* gave to item *j*. We give an example below to explain the similarities/favors between users.

For example, in Fig. 2(a), we can see that U_1 has friendships with U_2, U_3 and U_4. U_2 has friendships with U_1 and U_5. In Fig. 2(c), U_1 pays attention to items I_1, I_3 and I_7 and uses tags (T_{11}, T_{13} and T_{17}) to label them respectively. U_3 pays attention to item I_4 besides items I_1, I_3 and I_7. U_4 pays attention to item I_4 besides item I_2. On the contrary, U_2 pays attention to items I_2, I_5 and I_6 which are quite different from the concerns of U_1. Who would U_1 turn to for help when he/she is confused about item I_4? Obviously, U_1 would not ask U_2 for help because of the different favors. Although U_4 is the friend of U_1 and he/she also pays attention to item I_4, U_1 would turn to U_3 rather than U_4 for help to make the final decisions when consulting something about item I_4. It is obvious that U_1 and U_3 have the most similar favors. The overall trend of their concerns is consistent. So we can regard U_1 and U_3 as most of the same favors. The similarity between them is higher. Of course, due to being friends, when U_1 wants to know items I_5 which he/she does not pay attention to, he/she would ask U_2 rather than randomly select a user for help. Likewise, although U_1 and U_5 who are the friends of U_2 pay attention to items I_3 and I_7, U_2 may turn to U_5 rather than U_1 for help when consulting something about I_3 and I_7. It is because of the higher similarity between U_2 and U_5. U_2 and U_5 focus on several same items including I_5 and I_6. On the contrary, U_2 and U_1 have



(a) Social network

	U_1	U_2	U_3	U_4	U_5
U_1		1	1	1	
U_2	1				1
U_3	1			1	1
U_4	1		1		1
U_5		1	1	1	

(b) The friendship matrix

	I_1	I_2	I_3	I_4	I_5	I_6	I_7
U_1	T_{11}		T_{13}				T_{17}
U_2		T_{22}			T_{25}	T_{26}	
U_3	T_{31}		T_{33}	T_{34}			T_{37}
U_4		T_{42}		T_{44}			
U_5	T_{51}		T_{53}		T_{55}	T_{56}	T_{57}

(c) The user-item matrix

Fig. 2. A synthetic example.

completely different favors. From the above example we can see that each user would choose the most appropriate one from their own circle of friends to make the final decision. Because of the situations mentioned above, U_1 can turn to the friends U_2 and U_3 for different help and acquire the final correct recommendations when he is confused about different items. The different friends' groups of each user play key roles in generating recommendations. Table 1 describes the symbols that we use throughout the rest of the paper. We use the capital and bold letters to represent the matrices, vectors and sets. The value for each variable or variable set is represented by the corresponding lowercase letter.

3.2. Model description

The traditional recommender systems ignore the friendships among users. They just employ the user-item matrix to generate recommendation. In fact, when we travel or go shopping, we prefer to listen to the recommendations of friends intentionally or unintentionally according to the following rules: (1) The recommendations of the users who have the same or similar tastes or favors. Most of the recommender systems utilize the users' friendships to recommend based on the assumption that the user and his/her friends have the similar tastes or favors. However, we live in the circle of people who are widely distributed, such as family, neighbors, classmates, colleagues and so on. Some people have the common favors with us, some others are opposite. When we need advices, we certainly turn to the friends with the similar tastes, because the recommendations are more valuable. (2) The recommendations of the experts in some field. We usually turn to friends who are familiar with the field which we come to contact or prepare to learn and understand.

Based on the above two considerations, we propose a matrix factorization framework with social regularization. The brief flowchart of our algorithm is shown in Fig. 1. As we can see, in order to achieve better recommender results, we cluster the suitable group of friends and calculate the pair-wise user similarities and the similarities among users and items. We combine friendships and tags as regularization terms to constrain the matrix factorization framework. In this paper, we take the realistic situation that the friends with different favors recommend different results into consideration.

3.3. The similarity

3.3.1. Clustering

In this paper, compared with the prior work (Ma et al., 2011; Wang and Huang, 2014), we cluster the users to obtain the "suitable" groups of friends for improving the recommendation accuracy. The user-item dataset can be regarded as a two-dimensional matrix or a three-dimensional matrix which the tags are considered as the third dimension. In order to facilitate clustering and take the data density into consideration, we take the dataset as a two-dimensional matrix. As shown in Fig. 2(b), we utilize a biclustering algorithm which is based on CTWC (Getz et al., 2000) to cluster the dataset. CTWC (Getz et al., 2000) has been proved to be an effective algorithm for mining co-expression subpatterns. The users are firstly classified into stable groups based on the items and vice versa. The subsets of items' set **I** and the subsets of users' set **U** are respectively chosen to generate stable subset matrix to prepare for biclustering. The iterative procedure runs until no new stable biclusters are obtained. The final biclusters can overlap. It means one user may have various favors which accords with the practical situation. The algorithm is illustrated as follows.

Algorithm 1. The biclustering algorithm

Table 1
The symbols.

Symbol	Definition and description	Symbol	Definition and description
$\mathbf{U}$	The user list	$\mathbf{I}$	The item list
$\mathbf{T}$	The user-item rating matrix	$\mathbf{S}_i$	The vector of user i
$\mathbf{R}_i$	The tag vector of user i	$\mathbf{R}$	The item-tag matrix
$F(i)$	The friend list of user i	$\mathbf{V}_j$	The vector of item j

```
Biclustering algorithm
Input: user-item matrix T
Output: the biclusters set P
(1) u0 ← U, i0 ← I
(2) U ← {u0}, I ← {i0}
(3) W ← φ, P ← φ
(4) repeat
(5)    for (u,i) in (U × I) \ W do
(6)       apply the cluster-based K-means on the users of u,
return new stable cluster C(u);
(7)       U ← U + C(u)
(8)       apply the cluster-based K-means on the items of i,
return new stable cluster C(i)
(9)       I ← I + C(i)
(10) W ← W + (u,i)
(11)   for new stable cluster u in U or I do
(12)      P ← P + (u0,ui)
(13)   until no new cluster add to U or I
(14) return P
```

Line 1 initializes clusters u_0 and i_0 with all users and items; line 2 adds the clusters to user set and item set. Line 5 to line 10 utilize the cluster-based *K*-means algorithm to cluster (u,i) in $(\mathbf{U} \times \mathbf{I})$ and add the obtained stable clusters to user set **U** and item set **I** respectively. At the same time, (u,i) is added to set W which is used to prune $(\mathbf{U} \times \mathbf{I})$. Line 11 and line 12 add the new stable cluster in the form of (u_0, u_i) to the final biclusters set P. If u is the stable cluster in user set **U**, u_0 is the set of users, and u_i is the set of corresponding items. When obtaining the final biclusters, the similarity between any two users who have friendships can be calculated preliminarily.

3.3.2. *The similarity*

The collaborative filtering recommender systems utilize the rating matrix to calculate the similarity between uses. However, lots of other information (such as the tags) can be used to calculate the similarity. In Del.icio.us dataset, the average number of tags for each user labeling is about 29.28, and the average number of items that each user pays attention to 7.46. In addition, different friends play different roles when we make choices. We calculate the similarity between users based on not only the "realistic" friendships but also the "personal" favors. This means that, when user u needs recommendation, even if his two friends are also concerned about the same item, they may label the items with different or opposite tags.

(a) The similarity between users

The users may use the same tag for different items. Traditional approaches do not consider the number of tags which one user gave to one item. Hence, it is not fair that recommender systems use the account of user-tag co-occurrences to represent the tag's weight. In this paper, we utilize a method to calculate the weights.

$$\mathrm{w}_{ut} = \sum_i \frac{1}{\left|\mathrm{M}_{ui}\right|} \quad \text{if} \quad t \in \mathrm{M}_{ui} \quad i \in \mathbf{I} \tag{1}$$

where w_{ut} denotes the weight of tag t labeled by user u, M_{ui} is the tag list that user u gave to item i, and $|\mathrm{M}_{ui}|$ is the number of tags. The more tags user u gave to item i; the smaller that the weight of each tag. We combine the biclusters with the corresponding tags to calculate the similarity between users, which is different from the similarity used in Wang and Huang (2014). In Wang and Huang (2014) they calculate the similarities between the user and his/her all friends. In a word, they treat all the friends equally and generate the final recommendations through calculating the cosine similarities ($\cos(\mathbf{T}_i,\mathbf{T}_j)$) among the vectors of tags which the users labeled. However, the friends play different roles in generating different recommendations, in this paper, we calculate the similarity between the friends who are or not in the same cluster separately. If the users are in the same cluster, $\sum_{j=1}^{n} \cos(\mathbf{T}_{uj}, \mathbf{T}_{fj})/N$ is used as the metric to calculate the similarities between users, otherwise, the tags that the users labeled play the key role in calculating the similarities. The formula is defined as follows.

$$\begin{cases} S(u,f) = \lambda * B(u,f^{+}) + (1-\lambda)F(u,f^{-}) \\ B(u,f^{+}) = \dfrac{\sum_{j=1}^{n} \cos(\mathbf{T}_{uj}, \mathbf{T}_{fj})}{N} \quad \text{and} \quad n = \left|\mathbf{S}_u \cap \mathbf{S}_{f+}\right|, \\ \qquad N = \left|\mathbf{S}_u \cup \mathbf{S}_{f+}\right| \qquad 0 < \lambda < 1 \\ F(u,f^{-}) = \dfrac{\sum_{l=1}^{m} \cos(\mathbf{T}_{ul}, \mathbf{T}_{fl})}{N} \quad \text{and} \quad m = \left|\mathbf{S}_{B(u,f^{+})} \cap \mathbf{S}_{f-}\right| \end{cases} \tag{2}$$

where $B(u, f^{+})$ denotes the cosine similarity between u and f who have friendship with each other belong to the same cluster, and $F(u, f^{-})$ denotes the cosine similarity between u and f but are not in the same cluster. In this paper, we only take the immediate friends who are just one hop away from each other in a social network graph into consideration. f^{+} denotes that f and u are in the same cluster, and f^{-} denotes that f and u are not in the same cluster, N denotes the total number of items that the two users labeled, n denotes the total number of the same items that u and f labeled, and m represents the total number of the same items f and the users in cluster **B** labeled. When employing a large value of λ, the users who are in the same cluster dominate the similarity calculating process, otherwise the tags of u and f in the user-item matrix play a key role in calculating the similarity between users. For example, when consulting something about item I_2, U_1 follows the advice of U_4 to make the decision. Although U_1 and U_4 have different favors, U_4 is an expert of his/her friends in this field.

By the way, cold-start problem refers to the situation that a new user has no reviews. It is an extreme case of data sparsity. The traditional recommender systems, even social-based approaches, cannot make a recommendation to the new user since the approaches cannot find the similar users for him/her. However, in this paper, when a new user join the system, he/she must pay attention to the existing users in the system, we treat the interactions as the initial friendships of this new user to generate recommendations. We simply use friends' rating distribution on the target item as the result from friend inference.

(b) User-item correlation

The traditional recommender systems often ignore the correlation between user and item. In this paper, in order to calculate the correlation between user and item, we map the user and item to the tag space based on Ma et al. (2011) and calculate the similarity according to the formula as follows:

$$l(u,j) = vss(\mathbf{R}_u, \mathbf{V}_j) \tag{3}$$

where $\mathbf{R}_u$ denotes the tag vector of user u; and $\mathbf{V}_j$ denotes the vector of item j. Similar to (1), the weights of tags are defined according to the following formula:

$$w_{jt} = \sum_u \frac{1}{\left|M_{uj}\right|} \quad \text{if} \quad t \in M_{uj} \quad u \in U \tag{4}$$

where w_{jt} denotes the weight of tag t of item j, M_{uj} is the tags list and $|\mathrm{M}_{uj}|$ is the number of tags which user u gave to item j. The metric here is also cosine similarity. In addition, the dimensions of factorized user-space and item-space play a certain influence on the results. We discuss the problem in the experiment part.

3.4. *Social regularization*

A matrix factorization framework with social regularization was proposed by Ma et al. (2011). It firstly incorporates all the social connections of each user. However, it does not comply with the practical situation. Based on the intuition that we should identify the most suitable group of friends for different recommendation tasks, we incorporate the users' friendships into the matrix factorization framework. In this paper, we only consider the individual-based regularization approach.

Based on the social recommendation model proposed by Ma et al. (2011), the objective function is defined as follows.

$$\min_{S,V} L(\mathbf{T},\mathbf{S},\mathbf{V}) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{I}_{ij}(\mathbf{T}_{ij}-\mathbf{S}_i^T\mathbf{V}_j)^2 + \frac{\beta}{2}\sum_{i=1}^{m}\sum_{f\in F(i)} S(i,f)\left\|\mathbf{S}_i-\mathbf{S}_f\right\|_F^2 + \frac{\lambda_1}{2}\left\|\mathbf{S}\right\|_F^2 + \frac{\lambda_2}{2}\left\|\mathbf{V}\right\|_F^2 \tag{5}$$

where $\mathbf{I}_{ij}$ is the indicator function that is equal to 1 if user i rated item j, $\mathbf{T}_{ij}$ denotes the tags which user i gave to item j, $\mathbf{S}_i$ denotes the item vector of user i, $\mathbf{V}_j$ is the vector of item j, $\|\bullet\|_F^2$ represents the Frobenius norm, $\beta>0$, and F(i) is the friend list of user i. The last two regularization terms in formula (5) are used to avoid overfitting. In the above objective function, a social regularization term is imposed:

$$\frac{\beta}{2}\sum_{i=1}^{m}\sum_{f\in F(i)} S(i,f)\left\|\mathbf{S}_i-\mathbf{S}_f\right\|_F^2 \tag{6}$$

where S(i, f) denotes the friendship between user i and f, the function allows the regularization term to treat users' friends differently. The more details can be found in Ma et al. (2011). In this paper, we integrate not only the friendships among users but also the correlation between the user and item into the model. We propose the following regularization terms to impose constraints between one user and their friends individually:

$$\frac{\alpha}{2}\sum_{i=1}^{m}\sum_{f\in F(i)} l(j,f)\left\|\mathbf{S}_i-\mathbf{S}_f\right\|_F^2 + \frac{\beta}{2}\sum_{i=1}^{m}\sum_{f\in F(i)} S(i,f)\left\|\mathbf{S}_i-\mathbf{S}_f\right\|_F^2 \tag{7}$$

where $\alpha>0$, $\beta>0$, l(j, f) denotes the correlation between item j and user f. The function can treat the friends differently based on the item. If user i gave tags to item j, and the correlation between item j and user f is high, such as l(j, f) = 0.95, which means user f makes great contributions to the tastes of user i. S(i, f) denotes the friendship between user i and f. A small value of S(i, f) or l(j, f) represents the distance between feature vectors $\mathbf{S}_i$ and $\mathbf{S}_f$ should be larger. The objective function can be defined as follows:

$$\min_{S,V} L(\mathbf{T},\mathbf{S},\mathbf{V}) = \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{I}_{ij}(\mathbf{T}_{ij}-\mathbf{S}_i^T\mathbf{V}_j)^2 + \frac{\alpha}{2}\sum_{i=1}^{m}\sum_{f\in F(i)} l(j,f)\left\|\mathbf{S}_i-\mathbf{S}_f\right\|_F^2 + \frac{\beta}{2}\sum_{i=1}^{m}\sum_{f\in F(i)} S(i,f)\left\|\mathbf{S}_i-\mathbf{S}_f\right\|_F^2 + \frac{\lambda_1}{2}||\mathbf{S}||_F^2 + \frac{\lambda_2}{2}||\mathbf{V}||_F^2 \tag{8}$$

We can apply gradient descent algorithm to feature vector $\mathbf{S}_i$ and $\mathbf{V}_j$ to obtain a local minimum of the objective function. The pseudo code of RSboSN is presented as follows.

Algorithm 2. The RSboSN approach

```
Approach: RSboSN
Input: R, η, α, β, max_iter, L, S_{|S|×|S|}, l_{S×|V|}
Output: S_{|S|×k_u}, V_{|V|×k_v}
(1) Initialize S, V with random small values
(2) for max_iter > 0
(3)    for T_ij in T do
(4)       Calculating ∂L/∂S_i and utilizing S_{|S|×|S|}, l_{|S|×|V|}, S_{|S|×k_u}, V_{|V|×k_v}, T_ij
(5)       for u ← 1, 2···k_u do
(6)       S_{i,u} ← S_{i,u} − η(∂L/∂S_i)
(7)       end for
(8)       Calculating ∂L/∂V_j and utilizing S_{|S|×|S|}, l_{S×|V|}, S_{|S|×k_u}, V_{|V|×k_v}, T_ij
(9)       for v ← 1, 2···k_v do
(10)        V_{i,v} ← V_{i,v} − η(∂L/∂V_j)
(11)        end for
(12) end for
(13) max_iter
(14) if the objective function L converges
(15)    break
(16) end loop
```

In Algorithm 2, η denotes the learning ratio, and max_iter is the maximum iterations, $\mathbf{S}_{|\mathbf{S}|\times|\mathbf{S}|}$ is the similarities between users and $l_{|\mathbf{S}|\times|\mathbf{V}|}$ denotes the correlations between user and item. $\mathbf{S}_{|\mathbf{S}|\times k_u}$ and $\mathbf{V}_{|\mathbf{V}|\times k_v}$ are the feature vectors. $|\mathbf{S}|$ and $|\mathbf{V}|$ are the number of users and items respectively, and k_u and k_v are the feature dimensions of user and item respectively.

Overall, when obtaining the user and item biclusters and calculating the similarities between users and items, we incorporate the friend and item regularization terms into the proposed approach to improve the prediction accuracy. In Eq. (8), the approach adopts the incremental iterative optimization method and the gradient descent algorithm to optimize the objective function. Here, the learning ratio η is set to 0.5. Line 1 initializes the matrices $\mathbf{S}$ and $\mathbf{V}$ with random small values, the remainder lines iteratively calculate the objective function on the dataset. Specifically speaking, lines 3 to line 15 calculate each tag vector $\mathbf{T}_{ij}$ of the user-item matrix iteratively. For each $\mathbf{T}_{ij}$, line 4 to line 7 and line 8 to line 11 respectively use the gradient descent algorithm to optimize the feature vectors $\partial\mathbf{L}/\partial\mathbf{S}_i$ and $\partial\mathbf{L}/\partial\mathbf{V}_j$ for finding the local minimum of the objective function. The formulas are presented as follows.

$$\frac{\partial L}{\partial S_i} = \sum_{j-1}^{n}\mathbf{I}_{ij}(\mathbf{S}_i^T\mathbf{V}_j-\mathbf{T}_{ij})\mathbf{V}_j + \alpha\sum_{f\in F(i)} l(j,f)(\mathbf{S}_i-\mathbf{S}_f) + \beta\sum_{f\in F(i)} S(i,f)(\mathbf{S}_i-\mathbf{S}_f) + \lambda_1\mathbf{S}_i \tag{9}$$

$$\frac{\partial L}{\partial V_j} = \sum_{i=1}^{m}\mathbf{I}_{ij}(\mathbf{S}_i^T\mathbf{V}_j-\mathbf{T}_{ij})\mathbf{S}_i + \lambda_2\mathbf{V}_j \tag{10}$$

Hereafter, line 5 to line 6 and line 9 to line 10, respectively, update the feature vectors of user and item through inputting the parameters $\mathbf{S}_{|\mathbf{S}|\times|\mathbf{S}|}$, $l_{|\mathbf{S}|\times|\mathbf{V}|}$, $\mathbf{S}_{|\mathbf{S}|\times k_u}$, $\mathbf{V}_{|\mathbf{V}|\times k_v}$ and $\mathbf{T}_{ij}$. Finally, when the objective function converges at local minimum value or reaches the maximum number of iterations, the algorithm terminates and we can obtain the feature vectors. There are several parameters in the algorithm; we will discuss the impact of each one in the experiment part.

4. Experimental analysis

In this section, we conduct experiments on real dataset to validate the effectiveness of our approach. The proposed approach is implemented in MATLAB7.1. All the experiments are conducted on a Linux virtual machine with Intel processors (2.5 GHz) and 2 GB memory.

```
{
    "updated": "Thu, 10 Sep 2009 08:27:38 +0000",
    "links": [
        {
            "href": "http://www.usabilityweb.nl/2009/09/ikea-haar-slimme-toepassing-van-mbti-principes/",
            "type": "text/html",
            "rel": "alternate"
        }
    ],
    "title": "Usabilityweb ? Blog Archive ? IKEA haar slimme toepassing van MBTI principes",
    "author": "NicoDruif",
    "comments": "http://delicious.com/url/5b3bc7d41a07f732dae9f3bac8859c50",
    "guidislink": false,
    "title_detail": {
        "base": "http://feeds.delicious.com/v2/rss/recent?min=1&count=100",
        "type": "text/plain",
        "language": null,
        "value": "Usabilityweb ? Blog Archive ? IKEA haar slimme toepassing van MBTI principes"
    },
    "link": "http://www.usabilityweb.nl/2009/09/ikea-haar-slimme-toepassing-van-mbti-principes/",
    "source": {},
    "wfw_commentrss": "http://feeds.delicious.com/v2/rss/url/5b3bc7d41a07f732dae9f3bac8859c50",
    "id": "http://delicious.com/url/5b3bc7d41a07f732dae9f3bac8859c50#NicoDruif",
    "tags": [
        {
            "term": "design",
            "scheme": "http://delicious.com/NicoDruif/",
            "label": null
        },
        {
            "term": "usability",
            "scheme": "http://delicious.com/NicoDruif/",
            "label": null
        },
        {
            "term": "inspiration",
            "scheme": "http://delicious.com/NicoDruif/",
            "label": null
        },
        {
            "term": "interactiondesign",
            "scheme": "http://delicious.com/NicoDruif/",
            "label": null
        }
    ]
}
```


Fig. 3. An example of Del.icio.us data.

4.1. Dataset

With the rapid development of Web 2.0 technology, a lot of data has been produced on the internet every day. People influence each other through social network services. In this paper, we choose Del.icio.us (Delicious, 2013) as the data source to evaluate the proposed approach. Fig. 3 shows an example of Del.icio.us data. The user is the author "NicoDruif", the item is the URL resource, and the labeled tags are "design", "usability", "inspiration" and "interaction design".

Del.icio.us is a well-known tool that is easy and free to save, organize and discover interesting links on the web. The users can share the interesting resource and get in touch with other users who have the same interests. The users in the same community can easily get the new tags of the other users without the need to access them. In this paper, we suppose the users in the same group have the friendship between each other. The dataset from Del.icio.us contains social network information, item and tag. Social network information contains 1867 users and 15,328 edges. The data density is 0.44%. There are 437,593 <USER, URL, Tags> information entries, with 64,305 tags, 69,225 items (URLs) and 1867 users. As a result, we select the "good" data to do experiments according to the following rules: (1) the 1500 users with the most tags; (2) the 5000 items are tagged most frequently by the users in (1); (3) the 5000 tags are used most frequently based on (1) and (2). At last, we collect 99,499 pieces of information. The dataset is divided into training set and test set according to the ratio of 8:2. In addition, the average number of each user's friends is 7.2, and the maximum number is 82.

4.2. Metrics

In the experiment, we use the popular metrics, precision and recall,to measure the prediction quality of the proposed approach. The precision and recall are defined as follows:

$$\text{precision} = \frac{|R(u) \cap T(u)|}{|R(u)|} \tag{11}$$

$$\text{recall} = \frac{|R(u) \cap T(u)|}{|T(u)|} \tag{12}$$

where $R(u)$ denotes the tags that user u may label, and $T(u)$denotes the actual tags that user u labeled. The precision refers to the number of items which u labeled takes the proportion of the entire recommendation items. It reflects the possibility that u is interested in recommender item. The recall refers to the number of items which u labeled takes the proportion of all the items. It reflects the possibility that an item which u labeled may be recommended. As usual, with the increasing number of recommender items, the precision decreases.

4.3. Comparisons

In this section, in order to show the performance improvement of the proposed approach, we compare our approach with several common methods: the Pop approach, the collaborative filtering approach (CF) (Bergner et al., 2012), and the social regularization approach (Ma et al., 2011). The Pop approach is to recommend the most utilized items corresponding to the tags which the users use frequently. Based on the above ideas, the prediction formula is defined as follows:

$$p(u.i) = \sum_t w_{ut} w_{ti},\ t \in T_u \tag{13}$$

where the tag set T_u is labeled by user u, w_{ut} and w_{ti} are the weights. As for the collaborative filtering approach, we choose the

Table 2
The accuracy of the approaches.

Approach	P@1	P@3	P@5	R@5
Pop	10.99	9.21	8.47	5.62
CF	12.16	11.66	9.19	6.81
SoRec	13.29	13.93	11.38	7.67
RSboSN-p@80	16.73	15.22	13.65	10.71

user-based method. Given a user u, we find the neighbors set N_u and the prediction formula is defined as follows:

$$p(u.i) = \sum_{v \in N_u} s(u, v) f(v, i) \tag{14}$$

where $s(u, v)$ denotes the similarity between user u and user v, and $f(v, i) = 1$ denotes that user v gave tag to item i, otherwise $f(v, i)$ equals to 0. The approach in Ma et al. (2011) is a social-based recommendation method that incorporates friendships and user-item rating matrix to generate recommendation. The proposed approach in this paper only focuses on social regularization based on personal favors.

4.4. Experimental results

The experiment is conducted on Del.icio.us dataset. Pop, CF, SoRec and RSboSN-p@k stands for the four kinds of approaches mentioned above. k denotes the dimension of feature space. Here, we conduct the experiment under the situation that the dimensionality equals to 80 to compare the accuracy of each approach. The reason why we choose 80 is that the prediction accuracy is the highest. Later experiments illustrate the evaluation of each parameter. In algorithm 2, the parameter λ which is employed to calculate the similarities between users is set to 0.8, and the regularization parameters α and β are set to 0.01, the learning ratio is 0.5. Specifically, we use the metrics P@1, P@3, P@5 and R@5 to measure the results, where P@i denotes the precision value when the user recommends the number of i items, and R@5 denotes the recall value while the user recommends 5 items. Table 2 illustrates the accuracy of the mentioned approaches.

From Table 2 we can observe that RSboSN outperforms the compared approaches. For example, compared with SoRec (Ma et al., 2011) approach, on average, RSboSN improves the accuracy by 25.88%, 9.26%, 19.95% and 39.63% relative to P@1, P@3, P@5 and R@5. The improvements show that RSboSN is a promising recommender approach.

Fig. 4 shows the experiment results of the compared approaches. It is obvious that RSboSN performs better than the other existing approaches. The reason why the proposed approach can achieve

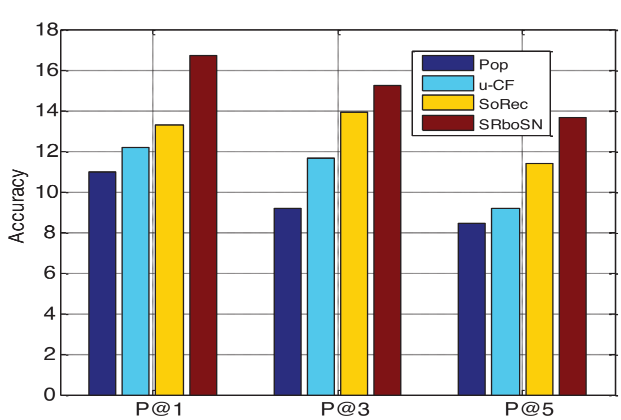


Fig. 4. The accuracy of the compared approaches.

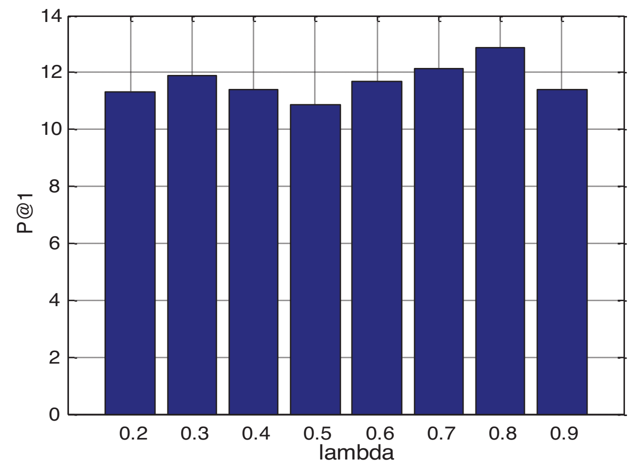


Fig. 5. The impact of parameter λ.

better results is that the friends, especially with the same or similar favors, recommend the valuable and practical recommendations. Comparing with the way of utilizing all the social friendships of each user to recommend, the small and accurate groups of friends could produce better commendations.

4.5. Impacts of parameters

4.5.1. Impact of λ

In this paper, we calculate the similarity between users based on not only the "realistic" friendships but also the "personal" favors. When calculating the similarity between users is based on Eq. (2), we utilize the actual group of friends as well as the corresponding labeled tags. There is a weighting threshold parameter λ. The value of parameter λ may explain which similarity calculation method can better express the friendship between two users. In the extreme case, if we employ a large value of λ, the users who are in the same cluster dominate the similarity calculating process. Otherwise, the tags of u and f in the user-item matrix play a key role in calculating the similarity between users. In this paper, we set the parameter range from 0.2 to 0.9. Fig. 5 illustrates the accuracy with different λ values. The dimensionality equals to 30 and the metric P@1 is used to validate the results.

We can see from Fig. 5 that the prediction accuracy is more precise when the parameter λ equals to 0.8. It also verifies that utilizing the groups of friends can effectively enhance the similarities among uses which would prepare for later recommendations. The result shows that the similar favors of users rather than the users who only have pure friendships with different favors still play a primary role in calculating the similarity between users.

4.5.2. Impact of α and β

Calculating the similarities between users is the initial step of the proposed framework. The main step is to solve the matrix factorization objective function with friendships and tags information regularization term. We incorporate the users' friendships and tags information to generate the recommendations. In the proposed approach, there are two important parameters α and β which play key roles in optimizing the social regularization framework. They control how much the proposed approach should incorporate the friendships and tags information. If the two parameters equal to 0, we only use the user-item rating matrix for matrix factorization to generate the final recommendations. We should find the suitable values to balance the parts of the regularization term. In this experiment, the impacts of α and β have the same trends, we discuss how the changes of α and β can affect the final prediction accuracy. We

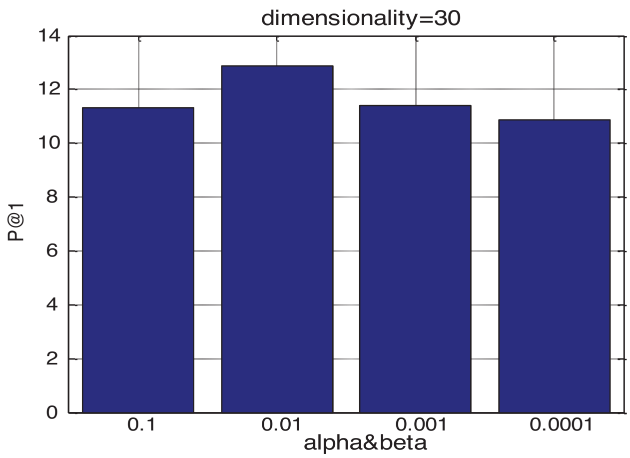


Fig. 6. Parameters α and β.

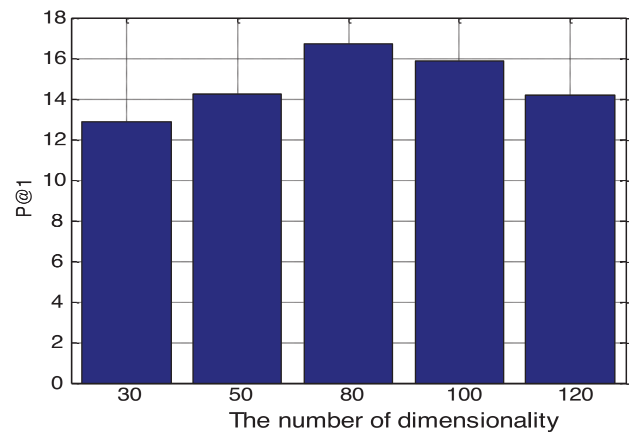


Fig. 7. The effectiveness of dimension.

set the dimensionality of feature space equals to 30; and the metric P@1 is used to validate the results. Fig. 5 illustrates the accuracy with different α and β values.

From the results in Fig. 6, we can observe that as the parameters α and β increase, the precision values increase at first, and as the parameters α and β go up to 0.01, the precision values decrease. So in this paper, we set the parameters α and β equal to 0.01 to predict the final recommendations accuracy.

4.5.3. Impact of feature dimensionality

In addition, the dimensionality of feature space plays an important role in predicting the final recommendations. In the matrix factorization framework with social friendship information, we analyze how the changes of dimension can affect the final recommendation accuracy. In this experiment, the values of parameters α and β are set to 0.01 according to the above experiments. We select the dimension $D = \{30, 50, 80, 100, 120\}$ to validate the effectiveness of the approach. By the way, in order to make the algorithm converge fast, the learning ratio decreases ten percent. The metric P@1 is used to validate the results. Fig. 7 illustrates the prediction accuracy on different number of dimensions.

As we can see from Fig. 7, when the dimension is 80, the prediction accuracy is the highest. Increasing or decreasing the dimension can reduce the accuracy of the approach. That is the reason why we set the dimension equals to 80 to evaluate the effectiveness of the proposed framework.

4.5.4. Performance comparison on different ratio of training set

Moreover, we also investigate the behavior of different approaches as we vary the ratio of training samples. Meanwhile, we only test the results under the situation that the dimensionality equals to 80 and the metric P@1 is used to validate the results. The values of the parameters are set according the above experiments. We select a certain proportion of data as the training set and the remaining data as the testing set. We use 10-flod cross validation to evaluate the performance of each approach. Fig. 8 shows the results of the compared approaches. We can see that, along with the increase of the training samples, the accuracy increases.

4.6. Performance analysis

In terms of time complexity, the main computation of the proposed framework including clustering the user and item groups and using gradient descent algorithm to optimize the objective function **L**. Assume there are m stable user clusters and n stable item clusters, the time complexity of biclustering is $O(m\log n)$. The computational complexity of evaluating the objective function **L** is $O(C\left|\mathbf{S}\right| + C\left|\mathbf{V}\right|)$, C is the dimensionality of feature vector, $\left|\mathbf{S}\right|$ and $\left|\mathbf{V}\right|$ is the number of users and items respectively. $O(C\left|\mathbf{S}\right|)$ and $O(C\left|\mathbf{V}\right|)$

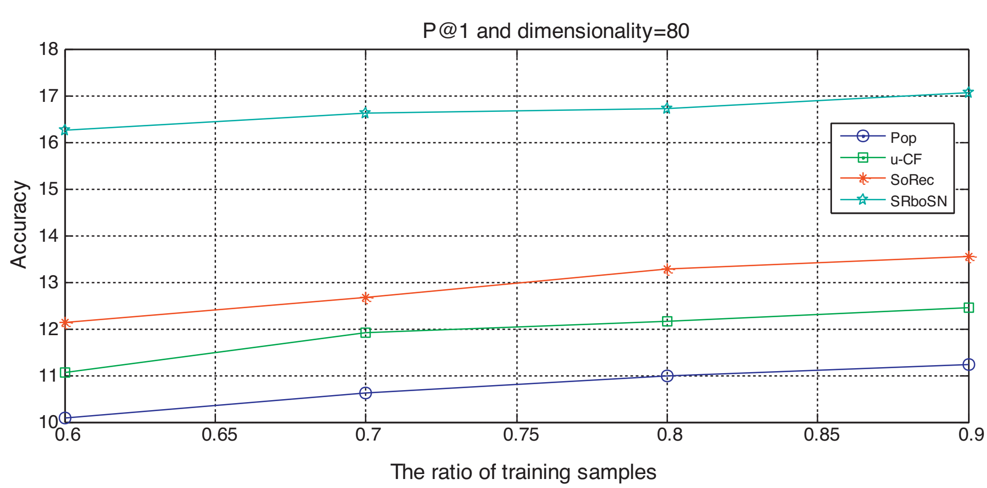


Fig. 8. The performance of different approaches with respect to the ratio of training samples.

are the computational complexities for gradients $\partial L/\partial \mathbf{S}$ and $\partial L/\partial \mathbf{V}$ respectively. Therefore the overall time complexity is $O(m\log n)$, which is linear logarithmic with respect to the number of clusters in the matrix. The complexity shows the proposed approach is efficient and can scale to large datasets. In addition, the proposed approach is quiet general; it can be extended to incorporate other information, such as time-series information, the dynamic variations information and the place information. We can incorporate them into the framework to further improve the performance of recommender systems. In addition, the proposed approach can be applied to solve trust-based recommendation problems, even the trust information from distance friends who are multiple hops away. The experiments show that the proposed approach indicates a promising future.

5. Conclusions and future work

In this paper, based on the observation that the friendships among users can improve the prediction quality, we propose a social regularization approach which incorporates social network information to benefit recommender systems. We employ both friendships among users and rating records (tags) to predict the missing values (tags) in the user-item matrix. The two aspects of social network information are employed in designing social regularization terms. We firstly cluster the dataset to obtain the groups of friends with similar favors to calculate the similarities. The experiments on real large dataset show that our approach outperforms the other popular traditional methods.

As the rapid growth of social network sites continues, the social-based recommender systems become more important. In this paper, we cluster the dataset to obtain the smaller groups with the similar tastes for generating good recommendations. However, we need to investigate the following problems: the cold-start problem, the influence from distance friends who are multiple hops away, the time-series information, the place information and the dynamic variations among users, and how to incorporate the information to improve performance and accord with the practical situation. For example, the recommender systems that take the time information (seasons, weekends) or temporal context into consideration can recommend items with different time effectiveness. These are interesting work to be explored in the future.

Acknowledgments

This work is supported by grants from City University of Hong Kong (project 7004051), the National Natural Science Foundation of China (project 60571048, project 60873264 and project 60971088), and the Qing Lan Project.

References

Adomavicius, G., Tuzhilin, A., 2005. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. IEEE Trans. Knowl. Data Eng. 17 (6), 734–749.

Barragáns-Martínez, A.B., Costa-Montenegro, E., Burguillo, J.C., et al., 2010. A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition. Inf. Sci. 180 (22), 4290–4311.

Bedi, P., Kaur, H., Marwaha, S., 2007. Trust based recommender system for semantic web. Artif. Intell., pp. 2677–2682.

Bellogin, A., Castells, P., Cantador, I., 2013. Improving memory-based collaborative filtering by neighbour selection based on user preference overlap. In: Proceedings of the 10th Conference on Open Research Areas in Information Retrieval, Lisbon, Portugal, May 15–17, pp. 145–148.

Bergner, Y., Droschler, S., Kortemeyer, G., et al., 2012. Model-based collaborative filtering analysis of student response data: machine-learning item response theory. In: Proceedings of the 5th International Conference on Educational Data Mining, Chania, Greece, June 19–21, pp. 95–102.

Peter, B., Chin, D.N., 2013. Preface to the special issue on personalization in social web systems. User Model. User Adapt. Interact. 23 (2–3), 83–87.

Chang, C.-C., Chu, K.-H., 2013. A recommender system combining social networks for tourist attractions. In: Proceedings of the 5th International Conference on Computational Intelligence, Communication Systems and Networks, Madrid, Spain, June 5–7, pp. 42–47.

Chang, T.M., Hsiao, W.F., 2011. Model-based collaborative filtering to handle data reliability and ordinal data scale. In: Proceedings of the 8th Conference on Fuzzy Systems and Knowledge Discovery (FSKD), Shanghai, China, July 26–28, pp. 2065–2069.

Delicious [EB/OL]. 2013. https://delicious.com/

Domingos, P., Richardson, M., 2001. Mining the network value of customers. Knowl. Discov. Data Min., pp. 57–66.

Getz, G., Levine, E., Domany, E., 2000. Coupled two-way clustering analysis of gene micro array data. Proc. Natl. Acad. Sci. U. S. A. 97 (22), 12079–12084.

Gunes, I., Bilge, A., Polat, H., 2013. Shilling attacks against memory-based privacy-preserving recommendation algorithms. Internet Inf. Syst. 7 (5), 1272–1290.

He, J., Chu, W., PhD Dissertation 2010. A Social Network-Based Recommender System (SNRS). University of California at Los Angeles, CA, USA.

Illig, J., Hotho, A., Jäschke, R., et al., 2011. A comparison of content-based tag recommendations in folksonomy systems. In: Wolff, K.E., Palchunov, D.E., Zagoruiko, N.G., Andelfinger, U. (Eds.), Knowledge Processing and Data Analysis. Lecture Notes in Computer Science, vol. 6581, pp. 136–149.

Jamali, M., Ester, M., 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In: Proceedings of the 4th ACM Conference on Recommender Systems, Barcelona, Spain, September 26–30, pp. 135–142.

Liu, F., Lee, H.J., 2010. Use of social network to enhance collaborative filtering performance. Exp. Syst. Appl. 37 (7), 4772–4778.

Lu, Y., Tsaparas, P., Ntoulas, A., Polanyi, L., 2010. Exploiting social context for review quality prediction. In: Proceedings of International Conference on WWW, Raleigh, North Carolina, USA, April 26–30, pp. 691–700.

Ma, H., Yang, H., Lyu, M.R., King, I., 2008. SoRec: social recommendation using probabilistic matrix factorization. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, Napa Valley, California, USA, October 26–30, pp. 931–940.

Ma, H., Zhou, D., Liu, C., et al., 2011. Recommender systems with social regularization. In: Proceedings of the 4th ACM International Conference on Web Search and Data Mining, WSDM 2011, Hong Kong, China, February 9–12, pp. 287–296.

Massa, P., Avesani, P., 2004. Trust-aware collaborative filtering for recommender systems. Lect. Notes Comput. Sci. (3290), 492–508.

Massa, P., Avesani, P., 2009. Trust metrics in recommender systems. Comput. Soc. Trust, 259–285.

Mei, Q., Cai, D., Zhang, D., Zhai, C., 2008. Topic modeling with network regularization. In: Proceedings of the 17th International Conference on World Wide Web (WWW'2008), Beijing, China, April 21–25, pp. 101–110.

Nazemian, A., Gholami, H., Taghiyareh, F., 2012. An improved model of trust-aware recommender systems using distrust metric. In: Proceedings of the 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining, Istanbul, Turkey, August 26–29, pp. 1079–1084.

Ozsoy, M.G., Polat, F., 2013. Trust based recommendation systems. In: Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM'13), Niagara, ON, Canada, August 25–29, pp. 1267–1274.

Richardson, M., Domingos, P.,2002. Mining knowledge-sharing sites for viral marketing. In: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. ACM Press, New York, pp. 61–70.

Wang, X., Huang, W., 2014. Research on social regularization-based recommender algorithm. Comput. Mod. 1, 77–80.

Zhoubao Sun is now a Ph.D. candidate of Computer Science and Technology at the Department of Computer and Information Science, University of Hohai. His research interests include machine learning and pattern recognition. He has worked on developing algorithms and simulation software in the area of data analysis. He is currently working mainly on the technology analysis and the advanced applications of social media.

Lixin Han received the Ph.D. degree in computer science from Nanjing University, Nanjing, China. He has been a Post-Doctoral Fellow with the Department of Mathematics, Nanjing University, and a Research Fellow with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong. He is currently a Professor with the Institute of Intelligence Science and Technology, Hohai University, Nanjing, China. He has published over 30 research papers. Prof. Han is an Invited Reviewer for several renowned journals and has been a Program Committee Member of many international conferences. He is listed in Marquis' Who's Who in the World and Marquis' Who's Who in Science and Engineering.

Wenliang Huang received the B.S. from Beijing University of Posts and Telecommunications, Beijing, China, the M.S. degree from College of Computer and Information Science, Hohai University, Nanjing, China. His research interests include machine learning and pattern recognition. He always works on developing algorithms in the area of recommender systems. He is currently working mainly on the advanced applications of social media in BAIDU, Beijing, China.

Xueting Wang received the B.S. and M.S. degree from College of Computer and Information Science, Hohai University, Nanjing, China. Her research interests include social networks and recommender system.

Xiaoqin Zeng received the B.S. degree from Nanjing University, Nanjing, China, the M.S. degree from Southeast University, Nanjing, and the Ph.D. degree from Hong Kong Polytechnic University, Kowloon, Hong Kong, all in computer science. He is currently a Professor, a Ph.D. student Supervisor, and the Director of the Institute of Intelligence Science and Technology, Hohai University, Nanjing. His current research interests include machine learning, neural networks, pattern recognition, and graph grammar. Prof. Zeng is an Associate Editor of the *IEEE Transactions on Systems, Man, and Cybernetics—Part B*. He is the Principal Investigator of several research projects sponsored by the Natural Science Foundation of China.

Min Wang received the Ph.D. degree in computer science from Nanjing University of Aeronautics and Astronautics, Nanjing, China. She is now an associate Professor with the Institute of Intelligence Science and Technology, Hohai University, Nanjing, China. She has published many research papers. Her research interests include compute vision, pattern recognition and machine learning. She is the Principal Investigator of one research projects sponsored by the Nature Science Foundation of China.

Hong Yan received his Ph.D. degree from Yale University. He has been Professor of Imaging Science at the University of Sydney and is currently Professor of Computer Engineering at City University of Hong Kong. He is elected a Fellow of the Institute of Electrical and Electronic Engineers (IEEE) for contributions to image recognition techniques and applications and a Fellow of the International Association for Pattern Recognition (IAPR) for contributions to document image analysis. He is also a Fellow of the Institution of Engineers, Australia (IEAust) and a member of the International Society for Computational Biology (ISCB). Professor Yan has published numerous technical papers in these areas.