Information Processing and Management 54 (2018) 463–474

Contents lists available at ScienceDirect

Information Processing and Management

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

# A novel recommendation method based on social network using matrix factorization technique

Xu Chonghuan

*School of Business Administration, Zhejiang Gongshang University, Hangzhou, 310018, PR China*

ARTICLE INFO

*keywords:*
Recommendation method
Social network
K-harmonic means
Particle swarm optimization
Matrix factorization

ABSTRACT

The rapid development of information technology and the fast growth of Internet have facilitated an explosion of information which has accentuated the information overload problem. Recommender systems have emerged in response to this problem and helped users to find their interesting contents. With increasingly complicated social context, how to fulfill personalized needs better has become a new trend in personalized recommendation service studies. In order to alleviate the sparsity problem of recommender systems meanwhile increase their accuracy and diversity in complex contexts, we propose a novel recommendation method based on social network using matrix factorization technique. In this method, we cluster users and consider a variety of complex factors. The simulation results on two benchmark data sets and a real data set show that our method achieves superior performance to existing methods.

## 1. Introduction

The rapid development of information technology and the fast growth of the Internet have facilitated an explosion of information which has accentuated the information overload problem. In recent years, recommender systems have proven to be an effective technique to deal with this problem and become extremely common in a variety of applications. They predict users' potential future likes and interests by using users' past preferences data. The quality of the results of a recommender system is determined mainly by the recommendation algorithms it adopts. Designing an excellent algorithm is crucial to the performance of a recommender system. Accordingly, various kinds of recommendation algorithms have been proposed, including collaborative filtering (CF) (Aligon, Gallinucci, Golfarelli, Marcel, & Rizzi, 2015; Kumar, Pujari, Sahu, Kagita, & Padmanabhan, 2017; Tsai, Steinberger, Pająk, & Pulli, 2016), content-based filtering (Khodambashi et al., 2015; Narducci et al., 2016; Puglisi, Javier, Forné, & David, 2015; Soares & Viana, 2015), K-Nearest Neighbor (K-NN) (Park, Park, Jung, & Lee, 2015; Adeniyi et al., 2016; Maillo et al., 2017; Yesilbudak, Sagiroglu, & Colak, 2017), diffusion approach (Gan, 2016; Ju & Xu, 2014; Zhou et al., 2010), and hybrid recommendation approach (Dooms, Pessemier, & Martens, 2015; Kaššák, Kompan, & Bieliková, 2016; Nilashi, Ibrahim, & Ithnin, 2014). Collaborative filtering is the most popular information filtering technique which usually works by searching a large group of users and to find a smaller set with tastes similar to target user. Content-based filtering method tries to recommend items to the active user similar to those rated positively in the past. It is based on the correlation between the content of the objects and the users' preferences. K-NN is a non-parametric method used for classification and regression. In K-NN, $k$ is a user-defined constant, and an unlabeled vector is classified by assigning the label which is most frequent among the $k$ training samples nearest to that query point. Diffusion approach is based on specific transformations of the input data to object-object networks. Personalized recommendations for an individual user are then obtained by using the user's past preferences as "sources" in a given network and propagating them to yet unevaluated objects. Hybrid recommendation approach is usually used to solve the cold-start problem, by combining collaborative and content data in such a way that even a new

*E-mail address:* talentxch@zjgsu.edu.cn.

https://doi.org/10.1016/j.ipm.2018.02.005
Received 15 April 2017; Received in revised form 26 December 2017; Accepted 21 February 2018
Available online 27 February 2018
0306-4573/ 

object that has never been rated before can be recommended. In fact, a number of studies have demonstrated that hybrid methods can provide more accurate recommendation results than independent approaches.

However, with the development of E-commerce and the increasingly complicated user context, how to fulfill personalized needs has become a new trend in personalized recommendation service studies. Psychological and sociological researches show that users' decisions in adopting information are influenced by their preferences and social relationships. Bandura (2001) gave a social cognitive theory of mass communication and argued that users' decisions were influenced by two pathways. In direct pathway, users' preferences determined their decisions. In indirect pathway, their decisions were influenced by friendship networks. Furthermore, Benjamin (1974) showed the similar opinion that factors such as cognition, feeling, taste, interest and interpersonal relationship developed the users' social behaviors. Bond and Smith (1996) found that people's decisions were easily influenced by other people's behaviors to some extent.

Therefore, scholars tend to incorporate complex context factors into the study of recommender systems. The merge of social network and E-commerce has led to enriched user information dimensions, resulting in more accurate personalized recommendation results. This paper gets its recommendation results by designing a personalized recommendation model based on social network context, combining the effect of user preference and user social relationship, and adopting matrix factorization technique. In this approach, we propose an improved clustering algorithm K-harmonic means (KHM) which has the same advantage as K-means but less sensitive to initial conditions. Then, matrix factorization technique is used to compute the similarities between clustered users. Especially in the process of similarity computation, users' preferences, social relationships and associations between users and items are considered. Detailed numerical analysis on benchmark data sets *MovieLens* and *Book-Crossing* and a real data set indicate that our algorithm outperforms other algorithms. Specifically, the recommendation results are not only accurate but also diverse.

The target of our work is to provide a recommendation method which can bring high accuracy and certain diversity under complex context. The main contributions of this paper are summarized as follows:

(1) In order to obtain accurate users classification for subsequent recommending, we propose a good performance hybrid clustering algorithm which composes of K-harmonic means (KHM) and Particle Swarm Optimization (PSO). It overcomes the sensitivity of initial conditions.
(2) In the process of similarity computation, we consider users' preferences, social relationships and associations between users and items. Especially, we use matrix factorization technique to alleviate the data sparsity and cold-start problems.

The remainder of this paper is organized as follows. Section 2 describes the relate works and Section 3 introduces some relevant methods. In Section 4, a socialized recommendation method based on matrix factorization (SRM-MF) is proposed. Section 5 provides experimental results of SRM-MF on two data sets. Finally, we draw implications and conclusions in Section 6.

## 2. Relate works

The problems (such as resource-overload and information-mislead) brought by big data have become increasingly serious. For individual user, how to acquire useful content from massive information quickly and accurately has become one prior issue. While for an enterprise user, how to mine customers' potential needs efficiently, enhance intelligence level of information searching and pushing, improve individualized service quality in this fierce competitive environment, has been put top in modify list in its E-commerce activities. To a certain extent, the creation of personalized recommendation technology has solved the dilemma between information diversity and customer needs specialization. Almost all the E-commerce platforms, such as Amazon, Alibaba., has applied various kinds of recommendation system more or less. However, with the development of E-commerce and the increasingly complicated user context, how to fulfill personalized needs has become a new trend in study of personalized recommendation service.

In recent years, many socialized recommendation methods have emerged. Ma, King, and Lyu (2011) proposed a novel probabilistic factor analysis framework which naturally fused the users' tastes and their trusted friends' favors together. The proposed framework was quite general, and it could also be applied to pure user-item rating matrix even if they did not have explicit social trust information among users. In this framework, they coined the term *social trust ensemble* to represent the formulation of the social trust restrictions on the recommender systems. Jiang, Cui, Wang, Zhu, and Yang (2014) investigated the social recommendation problem on the basis of psychology and sociology studies, which exhibited two important factors: individual preference and interpersonal influence. In this work, they first presented the particular importance of these two factors in online behavior prediction. Then they proposed a novel probabilistic matrix factorization method to fuse them in latent space. Sun et al. (2015) proposed a social regularization approach that incorporated social network information to benefit recommender systems. Both users's friendships and rating records (tags) were employed to predict the missing values in the user-item matrix. Han et al. (2016) proposed an easy-to-compute metric, *Community Similarity Degree* (CSD), to estimate the degree of interest similarity among multiple users in a community. They demonstrated that selecting communities with larger CSD could achieve higher recommendation precision. Li, Ma, and Shi (2016) examined the problem of social collaborative filtering to recommend items of interest to users in a social network setting. Many social networks captured the relationships among the nodes by using trust scores to label the edges. In this paper, they proposed a model-based approach for recommendation employing matrix factorization after removing the bias nodes from each link, which naturally fused the users' tastes and their trusted friends' favors together. Feng, Sharma, Srivastava, Wu, and Tang (2016) proposed a Social network regularized Sparse Linear Model (SocSLIM) with its extensions incorporating local learning (LocSocSLIM). SocSLIM learned sparse coefficient matrix for users by solving a sparse representation problem over user-item rating/purchase matrix and user–user social network′s adjacency matrix at the same time by sharing coefficient matrix. The coefficient matrix was used to

predict the recommendation scores, which were then combined with a proposed item based Distance regularized Sparse LInear Model (DSLIM) to generate recommendations for the users. Jhamb and Fang (2017) proposed a probabilistic latent factor model by incorporating two different types of latent factors to represent the user-oriented and event-oriented characteristics of groups. Pairwise learning was used to exploit unobserved RSVPs by modeling the individual probability of preference via Logistic and Probit sigmoid functions.

Unlike existing approaches, we consider more complex factors, then propose a novel recommendation method based on social network using matrix factorization technique. It can alleviate sparsity problem, increase accuracy and diversity of recommendation results simultaneously.

## 3. Relevant methods

### 3.1. K-harmonic means

K-harmonic means (KHM) takes the sum over all data points of the harmonic average of the squared distance from a data point to all the centers as its performance function, which is different from K-means (Zhang, Hsu, & Dayal, 1999). Let $X = \{x_1,x_2,\ldots, x_n\}$ be a set of $n$ items, $C = \{c_1,c_2,\ldots, c_k\}$ be a set of $k$ cluster centers, $m(c_j|x_i)$ is the membership function defining the proportion of data point $x_i$ that belongs to center $c_j$. $w(x_i)$ denotes the weight function defining how much influence data point $x_i$ has in recomputing the center parameters in the next iteration.

Basic algorithm for KHM clustering is shown as follows:

Step 1. Initialize the cluster centers $C$ randomly.
Step 2. Calculate objective function value according to

$$KHM(X, C) = \sum_{i=1}^{n} \frac{k}{\sum_{j=1}^{k} \frac{1}{\| x_i - c_j \|^p}} \tag{1}$$

Where $p$ ($p \geq 2$) is an input parameter. In general, $p = 2$.

Step 3. For each data point $x_i$, compute its membership $m(c_j|x_i)$ in corresponding center $c_j$ according to

$$m(c_j|x_i) = \frac{\|x_i - c_j\|^{-p-2}}{\sum_{j=1}^{k} \|x_i - c_j\|^{-p-2}} \tag{2}$$

Step 4. For each data point $x_i$, compute its weight $w(x_i)$ according to

$$w(x_i) = \frac{\sum_{j=1}^{k} \|x_i - c_j\|^{-p-2}}{\left(\sum_{j=1}^{k} \|x_i - c_j\|^{-p}\right)^2} \tag{3}$$

Step 5. For each center $c_j$, recompute its location from all data points $x_i$ according to their memberships and weights:

$$c_j = \frac{\sum_{i=1}^{n} m(c_j|x_i) w(x_i) x_i}{\sum_{i=1}^{n} m(c_j|x_i) w(x_i)} \tag{4}$$

Step 6. Do iterative computation according to steps 4 until $KHM(X,C)$ does not change significantly.
Step 7. Assign data point $x_i$ to cluster $j$ with the biggest $m(c_j|x_i)$.

It is demonstrated by Zhang et al. (1999) that KHM is essentially insensitive to the initialization of the centers.

### 3.2. Particle swarm optimization

Particle swarm optimization (PSO) is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality (Kennedy & Eberhart,1995). In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. During the searching process in a hyperspace, the $i$th particle is represented by two vectors, a position vector $X_i = [x_{i1},x_{i2},\ldots,x_{iD}]$ and a velocity vector $V_i = [v_{i1},v_{i2},\ldots,v_{iD}]$, where $D$ represents the dimensionality of the solution space. The vector $X_i$ is regarded as a candidate solution to the problem while the vector $V_i$ is treated as a searching direction and step size of the $i$th particle. Acceleration is weighted by a random term, with separate random

numbers being generated for acceleration toward *gbest* and *pbest* positions. At each iteration, the particle movement is computed as follows:

$$v_{ij}(t+1) = w^*v_{ij}(t) + b_1^*rand_1^*(pbest_{ij}(t) - x_{ij}(t)) + b_2^*rand_2^*(gbest_{ij}(t) - x_{ij}(t)) \tag{5}$$

$$x_{ij}(t+1) = x_{ij}(t) + v_{ij}(t+1) \tag{6}$$

In Eqs. (5) and (6), $x_{ij}(t)$ and $v_{ij}(t)$ represent the $i$th particle's position and velocity in the $j$th dimension at generation $t$, respectively. $pbest_{ij}(t)$ is the best position found by particle $i$ itself so far, $gbest_{ij}(t)$ is the best position found by the whole swarm so far, $w$ is an inertia weight scaling the previous time step velocity, $b_1$ and $b_2$ are two acceleration coefficients that scale the influence of the best personal position of the particle $pbest_{ij}(t)$ and the best global position $gbest_{ij}(t)$. $rand_1$ and $rand_2$ are two random numbers generated in the interval [0,1]. In general, $b_1 = b_2 = 2$ and $w \in [0.7, 1.2]$.

The steps of PSO algorithm are summarized as follows:

(1) Initialize a population of particles with random positions and velocities in the search space.
(2) For each particle $i$, update the velocity of $i$ according to Eq. (5).
(3) For each particle $i$, update the position of $i$ according to Eq. (6).
(4) Map the position of particle $i$ in the solution space and evaluate its fitness value according to the fitness function.
(5) Update $pbest_{ij}(t)$ and $gbest_{ij}(t)$.
(6) If termination conditions are met (reach the maximum or minimum value of fitness function, or the number of iterations), than stop.

### 3.3. Matrix factorization

Matrix factorization is simply a mathematical tool for playing around with matrices, and is therefore applicable in many scenarios where one would like to find out something hidden under the data. It plays an important role in engineering operation, model analysis and so on.

At present, there are many different matrix factorizations. Thereinto, spectral decomposition is the most common method of matrix factorization. It decomposes a normal matrix as the product of eigenvectors and eigenvalues. We assume $A$ as a square and normal matrix and the formula of $A$ can be expressed as

$$A = \sum_{i=1}^{n} \lambda_i A_i \tag{7}$$

where $\lambda_i$ denotes eigenvalue and $\{A_1,\ldots,A_n\}$ meet the equation

$$A_i^* = A_i \neq 0 \quad (i = 1, 2, \ldots, n) \tag{8}$$

$$A_i A_j = 0 \quad (i \neq j) \tag{9}$$

The target of our work is to design a socialized recommendation method which gives users high accuracy and certain diversity recommendation results. In addition, our method will alleviate sparsity and cold-start problems as much as possible. At the end, the experimental results show that our method is better than many other recommendation approaches.

## 4. Socialized recommendation method based on matrix factorization

With the complication of social context, existing recommendation methods do not meet the increasing personalized needs. Actually, people usually accept the recommendation that comes from friends rather than generated by computers. In recommender systems, the similarity between two users is influenced by many factors such as commodity popularity, user preference, social relationship, association between user and commodity, etc. So we need to design a novel recommendation model to fulfill the complex conditions.

The existing studies on recommender systems show rare existence of hybrid approaches that take user attributes, context and social relationships into consideration. In order to increase the accuracy and diversity of recommendation results in complex social context, we propose a novel recommendation method based on social network. Firstly, we design a hybrid clustering algorithm to cluster users which has a better performance than the similar clustering algorithms. Then we introduce multiple factors into similarity computational model and use matrix factorization technique to calculate users' preferences. Our method gets more accurate and diverse recommendation results, meanwhile alleviates the two classical problems. Fig. 1 shows the framework of the proposed recommendation method.

### 4.1. K-harmonic means clustering algorithm using PSO technique

The steps of K-harmonic means clustering algorithm using PSO technique are described as follows:

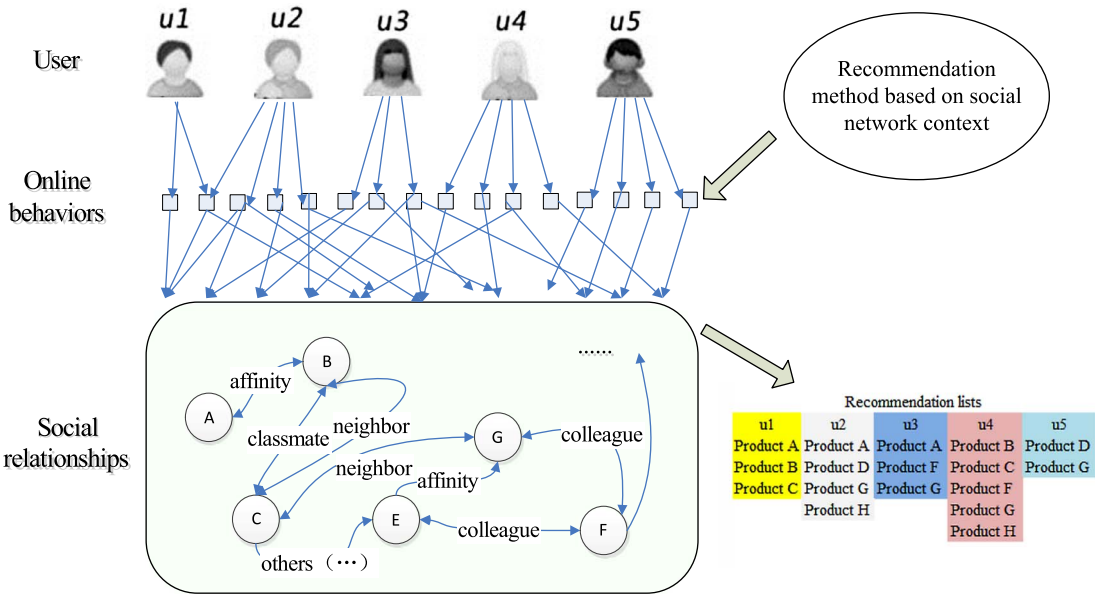


**Fig. 1.** Recommendation framework based on social network context.

Step 1. Initialize the global iterative count $It_1$, PSO's iterative count $It_2$, KHM's iterative count $It_3$ ($It_3 < It_2 < It_1$), a population of particles with random positions (Take the positions of each particle as the initial cluster centers. For example, initialize the position of particle $i$ denoted as $x_i$ with $k$ randomly selected data samples) and velocities (A particle is a vector of numbers of dimension $k*d$, where $k$ is the number of clusters and $d$ is the dimension of data to be clustered. It indicates that the variation of the center of the cluster will always be located within the clustering space.), inertia weight $w$, acceleration coefficients $b_1$ and $b_2$. Let the K-harmonic means' objective function Eq. (1) as the PSO's fitness function, the smaller the fitness value is, the better the cluster result will be.
Step 2. For each particle, apply PSO algorithm to adjust its position (cluster centers), iterate for $It_2$ times. The best current position (cluster centers) is expressed as:

$$P_i(t+1) = \begin{cases} P_i(t) & f(X_i(t+1)) \geq f(P_i(t)) \\ X_i(t+1) & f(X_i(t+1)) < f(P_i(t)) \end{cases} \tag{10}$$

where $P_i(t+1)$ denotes the best current position (cluster centers) of particle $i$.
Update *pbest* and *gbest* (compare the best positions (*pbest*) of each particle and get the global best position (*gbest*)).
Step 3. Use K-harmonic means to update the positions of each particle, iterate for $It_3$ times.
Step 4. Repeat steps 2–3, if the number of iteration equals to $It_1$ or the fitness value is minimized, stop.
Step 5. Assign data points to the corresponding clusters with the minimum fitness value.

The hybrid clustering algorithm searches data cluster centers using the sum over all data points of the harmonic average of the distance from a data point to all the centers as a metric. Using the same metric, the hybrid algorithm improves the convergence speed of PSO.

### 4.2. Recommendation model based on matrix factorization

As we know, collaborative filtering is the most widely used technique to produce user specific recommendations of items based on patterns of ratings or usage without need for exogenous information about either items or users. While user-based or item-based collaborative filtering methods are simple and intuitive, matrix factorization technique is usually more effective because it allows us to discover the latent features underlying the interactions between users and items.

We assume that there are $m$ users and $n$ items in a recommendation model. Each user has selected some items, also, each item has selected by some users. Then, the relationships between users and items can be described by a bipartite network. Let $U = \{u_1,u_2,\ldots,u_m\}$ denote users-set and $O = \{o_1,o_2,\ldots,o_n\}$ denote items-set, the recommendation model can be described by an adjacency matrix $A = \{a_{ij}\}$, where $a_{ij} = 1$ when user $i$ selects item $j$; otherwise, $a_{ij} = 0$. Standard cosine similarity computation based on resources diffusion can be expressed as follow.

$$S(i,f) = \frac{1}{\sqrt{k(u_i)k(u_f)}} \sum_{j=1}^{n} a_{ij} a_{fj} \tag{11}$$

where $S(i,j)$ represents the similarity between user $i$ and user $j$. $k(u_i)$ denotes the degree of the user $i$, namely, how many items are selected by this user. And, we apply social relationships to similarity computational model. The improved formula is as follows.

$$S(i,f) = \frac{1}{\sqrt{k(u_i)k(u_f)}} \sum_{j=1}^{n} a_{ij} a_{fj} \left[ \left( \frac{1}{1 + |r_{ij}/\overline{r_i} - r_{fj}/\overline{r_f}|} \right) \frac{1}{k(o_j)} \right]^{\mu} \tag{12}$$

where $r_{ij}$ and $r_{fj}$ denote the score of item $j$ obtained from user $i$ and user $f$, respectively. In reality, the score can reflect interest degree of a user. $\overline{r_i}$ and $\overline{r_f}$ denote the average score of user $i$ and user $f$, respectively. It means that the differentiation effect of scale of score obtained from different users can be reduced. $k(o_j)$ denotes the degree of item $j$, namely, how many users select this item. Formula $|r_{ij}/\overline{r_i} - r_{fj}/\overline{r_f}|$reflects trust relationship among users (Hwang & Chen, 2007), $\mu$is a freely adjustable parameter. The similarity is proportional to $\left[ \left( \frac{1}{1+|r_{ij}/\overline{r_i} - r_{fj}/\overline{r_f}|} \right) \frac{1}{k(o_j)} \right]^{\mu}$, if $\mu = 0$, the modified similarity computation reduces to the standard cosine similarity computation.

Given the difference of social relationships, we divide users' relationships into friendships and non-friendships. For user $i$ and user $f$, if they are not friends, we set the similarity of them as $\delta S(i, f)$; if they are friends, we set as $(1-\delta)S(i,f)$, where $0 < \delta < 0.5$. That is to say friendships play a more important role. So we get the new similarity formula is as follows.

$$S'(i,f) = \begin{cases} \delta S(i,f) & non - finendships \\ (1-\delta)S(i,f) & friendships \end{cases} \tag{13}$$

Different from traditional prediction method, we use matrix factorization technique to compute the scores of unselected items. We factorize the user-item rating matrix into two intermediated latent matrices including user latent feature matrix and item feature matrix, and utilize them to make further missing data prediction. Let $S$ denote user latent feature matrix and $V$ denote item latent feature matrix. $S_i$ is the feature vector of user $i$ and $V_j$ is the feature vector of item $j$. A low-rank matrix factorization approach seeks to approximate the rating matrix $R$ by a multiplication of $h$-rank factors.

$$R \approx S^{\mathrm{T}} V \tag{14}$$

where $S \in R^{h \times m}$, $V \in R^{h \times n}$, $h < \min(m,n)$. Traditionally, the matrix factorization method is utilized to approximate a rating matrix $R$ by minimizing.

$$\min(R, S, V) = \frac{1}{2} \|R - S^{\mathrm{T}} V\|_F^2 \tag{15}$$

where $\|\cdot\|_F^2$denotes the Frobenius norm. Due to the reason that $R$ contains a large number of missing values, we need to transform the formula. The improved formula is as follows.

$$\min(R, S, V) = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} (R_{ij} - S_i^{\mathrm{T}} V_j)^2 \tag{16}$$

where $a_{ij}$ reflects a selection between user $i$ and item $j$. In order to avoid overfitting, we add two regularization terms into formula (16) that is introduce 2-norm of user latent feature matrix and item feature matrix. Hence, the changed formula can be expressed as follows.

$$\min(R, S, V) = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} (R_{ij} - S_i^{\mathrm{T}} V_j)^2 + \frac{\lambda_1}{2} \|S\|_F^2 + \frac{\lambda_2}{2} \|V\|_F^2 \tag{17}$$

where $\lambda_1, \lambda_2 > 0$. However, the social recommendation is not well considered in this study. We incorporate the relationships between users as well as between users and items based on studies by Ma et al. (2011) and Sun et al. (2015). The objective function is defined as follows.

$$\begin{aligned} \min_{S,V}(R, S, V) \;=\; & \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} (R_{ij} - S_i^{\mathrm{T}} V_j)^2 + \frac{\alpha}{2} \sum_{i=1}^{m} \sum_{j \in O_i, k \notin O_i}^{n} a_{ij} \|V_j - V_k\|_F^2 \\ & + \frac{\beta}{2} \sum_{i=1}^{m} \sum_{f \in F(i)} S'(i,f) \|S_i - S_f\|_F^2 + \frac{\lambda_1}{2} \|S\|_F^2 + \frac{\lambda_2}{2} \|V\|_F^2 \end{aligned} \tag{18}$$

where $\alpha, \beta > 0$, $\|V_j - V_k\|_F^2$ denotes the correlation between two items. In reality, an item is selected by a user means that it must have some functions or characteristics fulfill the user need. When item $j$ is selected by user $i$, a similar item $k$ may be selected by him further. $S'(i,f)\ \|S_i - S_f\|_F^2$ denotes the influence of relationship between user $i$ and user $f$. In some cases, friends can be affected with each other. When value of $S'(i,j)$ is small, the distance between $S_i$ and $S_f$ is large, vice versa. This formula reflects that the corresponding user is not the more similar the more effect. For solving user latent feature matrix $S$ and item latent feature matrix $V$, we use the gradient descent algorithm to optimize them, respectively. The formulas are presented as follows.

$$\frac{\partial L}{\partial S_i} = \sum_{j=1}^{n} a_{ij}(S_i^{\mathrm{T}} V_j - R_{ij})V_j + \beta \sum_{f \in F(i)} S'(i,f)(S_i - S_f) + \lambda_1 S_i \tag{19}$$

$$\frac{\partial L}{\partial V_i} = \sum_{i=1}^{m} a_{ij}(S_i^{\mathrm{T}} V_j - R_{ij})S_i + \alpha \sum_{i=1}^{m} a_{ij}(V_j - V_k) + \lambda_2 V_j \tag{20}$$

## 5. Experimental results and analysis

In this section, we conduct experiments on two benchmark data sets and a real data set to evaluate the performance of our approach. The program is written in Python running on Ubuntu 14.04. The tests were performed on three Servers which consist of Xeon(R) X3323 2.5 GHz with 8 GMB Memory and 1 TB Hard disk. A brief introduction of these data sets is as follows:

(1) The *MovieLens* is an internationally recognized benchmark data set. It consists of 1682 movies, 943 users and 100,000 ratings. Each user has rated at least 20 movies by using a discrete number on the scale of 1–5. In this dataset, there are three kinds of information tables: demographic information about the users, information about the items (movies), scores about the movies.
(2) *Book-Crossing* is another internationally recognized benchmark data set and formed a sparse user-book score matrix. The data set contains user data, book data and score data, and the score table also includes three dimensions: 278,858 users (anonymized but with demographic information), 271,379 books and 1,149,780 ratings (explicit/implicit). There are explicit and implicit scores, while explicit scores are discrete values ranging from 1 to 10, with 0 for implicit scores, the lowest score 1 means the user do not like it, the highest score of 10 indicates a user favorite, while 0 means the user does not show any preference.
(3) Real-world data set comes partly from mobile shopping data provided by our co-operative enterprises (they are delivered by mobile ISPs and e-business platforms) and partly from user-related data acquired by scrapping the Internet with a web-crawler. There are five data tables in our data set, the first one is consisted of basic properties of 689,922 mobile users, the second one is a collection of basic properties of 384,374 products, the third one is the interactive information among users (the data show only which users interact with each other), the fourth one contains 1,096,680 actions (including click, favorite, cart and purchase, each labeled with a different numeric value) to 384,374 products made by 689,922 users in 4 months, and the last one is a score (discrete values ranging from 1 to 5) records table with 4,377,223 ratings to 384,374 products made by 689,922 users in 4 months. Location information (Normalized Longitude and Latitude data) can be acquired by mobile access records of the user.

We believe user preference can be indicated by ratings, the higher the score is, the higher the affection degree is. With real data set, we assume that user will like only those movies and products with score more than 3. Furthermore, we generate a link i.e. $a_{ij} = 1$ between users and the products that they rated 3 or more, otherwise no link is generated i.e. $a_{ij} = 0$, and set the preference value to a product of a user to $R_{ij} = \{3,4,5\}$ (set preference values equivalent to rating scores to facilitate the calculation process). With *MovieLens* data set, the links with ratings no less than 3 are considered and $R_{ij} = \{3,4,5\}$.With *Book-Crossing* data set, we take 5 or more as a criteria in determining whether the book is liked by a user and generate a link between them as $a_{ij} = 1$, otherwise $a_{ij} = 0$. It must be explicitly explained that those 0 s in this data set indicate implicit user preference; we have no idea whether a user likes the book or not, so we cannot simply remove them arbitrarily. For this reason, a link is drawn between the user and the book that has a score of 0 rated by the user (i.e. $a_{ij} = 1$), and preference values are set to $R_{ij} = \{1,5,6,7,8,9,10\}$, while 1 stands for implicit user preference corresponded with a 0 book score. We divide the processed data set into two parts: The training set contains 80% of the data, and the remaining 20% of data constitutes the test. The training set is treated as known information, while no information in the test set is allowed to be used for recommendation.

Generally, scholars employ different metrics to test a recommendation algorithmic performance. In this paper, we adopt six metrics to measure accuracy and diversity of the proposed algorithm, and diversity is as important as accuracy. The six metrics are Precision, Recall, *F*-measure, Root Mean Square Error (RMSE), Intra-similarity and Hamming distance.

(1) Precision is defined as the ratio of the number of recommended objects selected by users appearing in the test set to the total number of recommended objects. This measure is used to evaluate the effectiveness of a given recommendation list. The Precision can be formulated as $a/L$, in which $a$ represents the number of recommended products selected by users appearing in test set, and $L$ is the total number of recommended products. In general, the number of recommended products is no more than 100.
(2) Recall is defined as the ratio of the number of recommended objects selected by users appearing in the test set to the total number of the objects actually selected by these users. The larger Recall corresponds to the better performance. The Recall can be formulated as $a/M$, in which $a$ represents the number of recommended products selected by users appearing in test set, and $M$ is the total number of these users' actual buying.
(3) *F*-measure is a comprehensive metric which is composed of Precision and Recall to evaluate the accuracy of recommendation methods

$$f\left(P, R\right) = \frac{2\cdot(\text{Precision}\cdot\text{Recall})}{\text{Precision} + \text{Recall}} \tag{21}$$

In pattern recognition and information retrieval with binary classification, Precision is the fraction of retrieved instances that are

relevant, while Recall is the fraction of relevant instances that are retrieved. Both Precision and Recall are therefore based on an understanding and measure of relevance. The higher the *F*-measure is, the better the algorithmic performance will be.

(4) Root Mean Square Error (RMSE) is the square root of the ratio of the square of the deviation between observed and real value and the number of observations. The real values have to be substituted by the most reliable values, due to the limitation of number of times to undertake observation in real measurement. In the process of evaluating personalized recommendations, RMSE calculates all the values of the square root of the mean of the quadratic sum of the deviation of real values and predicted values, the smaller the value is, the higher the accuracy of the recommender's prediction is, the formula can be expressed as

$$RMSE = \sqrt{\frac{1}{T}\sum_{i=1}^{T}(R_i - R_i')^2} \tag{22}$$

where $R_i$ stands for real value, while $R_i'$ stands for predicted value and $T$ stands for times of observation.

(5) Intra-similarity evaluates the similarity between objects inside users' recommendation lists. A good recommendation algorithm is expected to give fruitful recommendation results and has the ability to guide or help the users exploit their potential interest fields. Therefore, it calls for a lower Intra-similarity. There are many similarity metrics between objects. Here we adopt the widely used one, that is, cosine similarity to measure objects' similarity. For two objects $w$ and $j$, their similarity is defined as

$$S_{wj} = \frac{\sum_{l=1}^{m} a_{lw}a_{lj}}{\sqrt{k(o_w)k(o_j)}} \tag{23}$$

For an arbitrary user $l$, the number of recommendation objects is $L$. Firstly, we need to calculate $L(L\text{-}1)/2$ couple of objects' similarity, and then average these values to get $I_l = < S_{wj} >$ . Finally, we use the mean value of $I$ of the overall users to measure the diversity in recommendation lists.

(6) Hamming distance can measure the strength of personalization. If the overlapped number of objects in user $i$ and user $l$'s recommendation lists is $Q$, their Hamming distance is

$$H_{il} = 1 - \frac{Q}{L} \tag{24}$$

Generally speaking, a more personalized recommendation list should have long Hamming distances to other lists. Accordingly, we use the mean value of Hamming distance $S = < H_{il} >$ of the overall user-user pairs to measure the strength of personalization

In this experiment, we compare our algorithm (SRM-MF) with four widely used recommendation algorithms: CF, Modified collaborative filtering (MCF) (Liu, Jia, Zhou, Sun, & Wang, 2009), Non-normalized Cosine Neighborhood (NNCosNgbr) (Ju & Xu, 2014), Model-based approach (MB) (Li et al., 2016). MCF considers the influence of a node's degree, and then presents a modified collaborative filtering based on resource diffusion to substitute the standard cosine similarity. NNCosNgbr predicts the rating for a user on an item as the weighted average of the ratings of similar items. MB employs matrix factorization after removing the bias nodes from each link, which naturally fuses the users' tastes and their trusted friends' favors together.

### 5.1. Parameters determination

In this section, we will get the optimal values of corresponding parameters. According to the research proposed by Sun et al. (2015), we set the value of $k$ (dimension of feature space) as 80 and the value of $\alpha$, $\beta$, $\lambda_1$ or $\lambda_2$ as 0.01. Firstly, we want to demonstrate the advantage of our hybrid clustering algorithm which can overcome the local optimum and sensitivity problems. We compare our hybrid clustering algorithm with standard KHM in metric $D/L$. $D$ is the internal distance within clusters and $L$ is the external distance between clusters, the detailed descriptions are as follows.

(1) Internal distance within clusters

Internal distance within clusters is defined as the sum of internal distance within each cluster. The formula can be expressed as.

$$D = \sum_{i=1}^{k}\sum_{p\in C_i}|p - m_i| \tag{25}$$

In which, $k$ denotes the number of clusters, $p$ is an arbitrary object in cluster $C_i$, $m_i$ is the mean value of the distance between each data and its center.

(2) External distance between clusters

External distance between clusters is defined as the sum of distance between each cluster center and globe data center.

**Table 1**
The proportion of hybrid PSO and KHM to KHM in metric $D/L$, where $b_1 = b_2 = 2$, $w = 1.1$.

| Data set | Number of instances | Number of attributes | $k$ | $D/L$ Proportion of hybrid PSO and KHM to KHM |
|---|---|---|---|---|
| Balance | 625 | 4 | 3 | 0.861 |
| Cancer | 569 | 30 | 2 | 0.851 |
| Cancer-Int | 699 | 9 | 2 | 0.853 |
| Credit | 690 | 15 | 2 | 0.864 |
| Dermatology | 366 | 34 | 6 | 0.866 |
| Diabetes | 768 | 8 | 2 | 0.869 |
| Ecoli | 327 | 7 | 5 | 0.870 |
| Glass | 214 | 9 | 6 | 0.868 |
| Heart | 303 | 75 | 2 | 0.852 |
| Horse | 364 | 27 | 3 | 0.849 |
| Iris | 150 | 4 | 3 | 0.853 |
| Thyroid | 215 | 5 | 3 | 0.861 |
| Wine | 178 | 13 | 3 | 0.862 |

$$L = \sum_{i=1}^{k} |m_i - m| \tag{26}$$

In which, $k$ denotes the number of clusters, $m$ is the mean value of the distance between each data and the center of globe data, $m_i$ is the mean value of the distance between each data and its center.

The smaller the value of $D/L$ is, the higher the quality of clustering will be. We set parameters $b_1$ and $b_2$ as 2. After iterative calculation, we get the optimal value of parameter $w$ which is 1.1. Table 1 shows the comparison between our clustering algorithm and KHM.

Comparing the hybrid clustering algorithm with KHM, as is seen in Table 1, the values of $D/L$ can be further reduced by 15% or so in all experimental data sets. That is to say, our algorithm is better than KHM clustering algorithm.

For parameters $\delta$and $\mu$, we predict the optimal value of $\delta$between the range of 0.1 and 0.4 and $\mu$around 1.86 (According to many literatures on diffusion based approaches, for instance the former research of Liu, Liu, Jia, Sun, and Wang (2010)). In this case, algorithm tends to recommend unpopular products to users. To find the optimal values of parameters $\delta$and $\mu$ rapidly, we execute the iterative computation based on the strategy of binary search. In the process of iterative computation, we set the interval between each $\delta$as 0.01, so as $\mu$. Finally, we get $\delta = 0.3$ and $\mu = 1.85$ in obtaining the minimal RMSE.

### 5.2. Results and analysis

For *MovieLens* and *Book-Crossing* data sets, we set the lengths of recommendation list $L$ as 20, 30 and 40, respectively. For real data set, we set the lengths of recommendation list $L$ as 10, 20 and 30, respectively. Under normal circumstances, the number of products (items) recommended by the e-commerce platform to the target user is between 10 and 20. The metrics Precision, Recall, $F$-measure and RMSE are used to test the accuracy and the others are used to test the diversity. The results can be seen in Tables 2–4.

Comparing SRM-MF with MCF and $L = 20$, as is seen in Table 2, on *MovieLens*, the Precision can be further increased by 32.2%. Comparing SRM-MF with NNCosNgbr and MB, the Precision can be increased by 33.7% and 11.2%, respectively. Similarly, our algorithm has higher Recall than the other four kinds of recommendation algorithms. For the rest of the metrics, our algorithm is also the best.

Although *Book-Crossing* is similar to *MovieLens*, it is much sparser. Comparing SRM-MF with MCF and $L = 20$, as is seen in Table 3, the Precision, Recall and $F$-measure can be further increased by 21%, 10.6% and 17.7%, respectively. In addition, the RMSE and Intra-similarity can be further reduced by 16.6% and 31.1%, the Hamming distance can be further increased by 13%. Comparing SRM-MF with other algorithms mentioned in experiment, we can also observe that it is the best in six metrics. Table 3 also shows that our algorithm exceeds the other four algorithms in the recommendation list of 30 and 40. The main reason is that our method finds more relationships and associations among users or items, and increases calculation weight of similarities of users which are in the same cluster.

In addition, we consider whether the accuracy, diversity etc. can be improved by increasing the length $L$ of the recommendation list. We adjust the length of the recommended list to vary between 50, 60 and 70, and find that although a significant degradation of the results of the metrics (such as Precision, $F$-measure) emerges, the performance of our algorithm was still the best. After analysis we find that most of the actual hit items are among the top 30 data and rarely emerged between 50 and 70.

What we can see from the above table is that the proposed algorithm has better performance than other three algorithms in all the six metrics: higher Precision, higher Recall, higher $F$-measure, lower RMSE, smaller Intra-similarity and larger Hamming distance. The real data set exhibits the characters of larger in volume and sparser compared to benchmark data sets, and is accompanied with the existence of a large variety of types of items. Therefore, benefit by the consideration of more factors, although the experimental

**Table 2**
Algorithmic performance on *MovieLens* data set. The Precision, Recall, *F*-measure, RMSE, Intra-similarity and Hamming distance are corresponding to $L = 20, 30, 40$. For MCF, $\alpha = 1.85$and for SRM-MF, $\delta = 0.3$, $\mu = 1.85$ and $\alpha = \beta = \lambda_1 = \lambda_2 = 0.01$. Each number presented in this table is obtained by averaging over five runs with independently random division of 80% training set and 20% test set.

| | Precision | Recall | *F*-measure | RMSE | Intra-similarity | Hamming distance |
|---|---|---|---|---|---|---|
| $L = 20$ | | | | | | |
| CF | 0.079 | 0.307 | 0.126 | 0.842 | 0.334 | 0.548 |
| MCF | 0.090 | 0.328 | 0.141 | 0.729 | 0.318 | 0.609 |
| NN- CosNgbr | 0.089 | 0.327 | 0.140 | 0.731 | 0.317 | 0.607 |
| MB | 0.107 | 0.352 | 0.164 | 0.694 | 0.254 | 0.696 |
| SRM-MF | 0.119 | 0.378 | 0.181 | 0.614 | 0.203 | 0.734 |
| $L = 30$ | | | | | | |
| CF | 0.076 | 0.320 | 0.123 | 0.834 | 0.337 | 0.541 |
| MCF | 0.084 | 0.353 | 0.136 | 0.720 | 0.320 | 0.602 |
| NN- CosNgbr | 0.083 | 0.352 | 0.134 | 0.722 | 0.319 | 0.601 |
| MB | 0.099 | 0.376 | 0.157 | 0.687 | 0.257 | 0.690 |
| SRM-MF | 0.113 | 0.399 | 0.176 | 0.608 | 0.207 | 0.729 |
| $L = 40$ | | | | | | |
| CF | 0.071 | 0.331 | 0.117 | 0.826 | 0.341 | 0.537 |
| MCF | 0.081 | 0.366 | 0.133 | 0.712 | 0.323 | 0.597 |
| NN- CosNgbr | 0.080 | 0.365 | 0.132 | 0.715 | 0.322 | 0.595 |
| MB | 0.091 | 0.394 | 0.148 | 0.679 | 0.261 | 0.684 |
| SRM-MF | 0.104 | 0.416 | 0.166 | 0.603 | 0.210 | 0.723 |

**Table 3**
Algorithmic performance on *Book-Crossing* data set. The Precision, Recall, *F*-measure, RMSE, Intra-similarity and Hamming distance are corresponding to $L = 20, 30, 40$. For MCF, $\alpha = 1.85$and for SRM-MF, $\delta = 0.3$, $\mu = 1.85$ and $\alpha = \beta = \lambda_1 = \lambda_2 = 0.01$. Each number presented in this table is obtained by averaging over five runs with independently random division of 80% training set and 20% test set.

| | Precision | Recall | *F*-measure | RMSE | Intra-similarity | Hamming distance |
|---|---|---|---|---|---|---|
| $L = 20$ | | | | | | |
| CF | 0.067 | 0.178 | 0.097 | 0.917 | 0.318 | 0.521 |
| MCF | 0.081 | 0.188 | 0.113 | 0.770 | 0.305 | 0.563 |
| NN- CosNgbr | 0.080 | 0.187 | 0.112 | 0.772 | 0.303 | 0.561 |
| MB | 0.089 | 0.199 | 0.123 | 0.701 | 0.234 | 0.603 |
| SRM-MF | 0.098 | 0.208 | 0.133 | 0.642 | 0.210 | 0.636 |
| $L = 30$ | | | | | | |
| CF | 0.064 | 0.199 | 0.097 | 0.909 | 0.326 | 0.514 |
| MCF | 0.078 | 0.212 | 0.114 | 0.761 | 0.310 | 0.558 |
| NN- CosNgbr | 0.077 | 0.211 | 0.113 | 0.763 | 0.308 | 0.556 |
| MB | 0.086 | 0.218 | 0.123 | 0.693 | 0.247 | 0.598 |
| SRM-MF | 0.095 | 0.228 | 0.134 | 0.634 | 0.226 | 0.624 |
| $L = 40$ | | | | | | |
| CF | 0.061 | 0.226 | 0.096 | 0.901 | 0.335 | 0.508 |
| MCF | 0.073 | 0.240 | 0.112 | 0.753 | 0.317 | 0.553 |
| NN- CosNgbr | 0.072 | 0.239 | 0.111 | 0.754 | 0.314 | 0.552 |
| MB | 0.082 | 0.246 | 0.123 | 0.685 | 0.257 | 0.594 |
| SRM-MF | 0.090 | 0.257 | 0.133 | 0.627 | 0.239 | 0.618 |

results of the five algorithms are not high, the advantages of our algorithm over others are still obvious.

Finally, for an online recommender system, we need to consider the processing time and memory consumption of its recommendation algorithm. The computational time complexity of SRM-MF should be denoted as $(m^2 + mn)$, in which memory store are $m^2$, $m$ denotes the number of users and $n$ denotes the number of items. Furthermore, $m$ and $n$ meet the condition which is $m >> n$. So the time complexity varies mainly against the variation of $m$, and thus can be denoted with big $O$ notation as $O(m^2)$.

**Table 4**
Algorithmic performance on real data set. The Precision, Recall, *F*-measure, RMSE, Intra-similarity and Hamming distance are corresponding to $L = 10, 20, 30$. For MCF, $\alpha = 1.85$and for SRM-MF, $\delta = 0.3$, $\mu = 1.85$ and $\alpha = \beta = \lambda_1 = \lambda_2 = 0.01$. Each number presented in this table is obtained by averaging over five runs with independently random division of 80% training set and 20% test set.

| | Precision | Recall | *F*-measure | RMSE | Intra-similarity | Hamming distance |
|---|---|---|---|---|---|---|
| $L = 10$ | | | | | | |
| CF | 0.049 | 0.281 | 0.083 | 0.984 | 0.402 | 0.528 |
| MCF | 0.064 | 0.328 | 0.107 | 0.832 | 0.386 | 0.602 |
| NN- CosNgbr | 0.063 | 0.329 | 0.106 | 0.834 | 0.383 | 0.601 |
| MB | 0.076 | 0.348 | 0.139 | 0.702 | 0.335 | 0.650 |
| SRM-MF | 0.092 | 0.411 | 0.150 | 0.614 | 0.297 | 0.682 |
| $L = 20$ | | | | | | |
| CF | 0.047 | 0.286 | 0.081 | 0.981 | 0.433 | 0.521 |
| MCF | 0.062 | 0.331 | 0.104 | 0.828 | 0.410 | 0.588 |
| NN- CosNgbr | 0.062 | 0.330 | 0.104 | 0.828 | 0.408 | 0.586 |
| MB | 0.073 | 0.350 | 0.135 | 0.692 | 0.359 | 0.631 |
| SRM-MF | 0.088 | 0.414 | 0.145 | 0.612 | 0.312 | 0.664 |
| $L = 30$ | | | | | | |
| CF | 0.042 | 0.288 | 0.073 | 0.979 | 0.455 | 0.517 |
| MCF | 0.061 | 0.333 | 0.103 | 0.817 | 0.428 | 0.564 |
| NN- CosNgbr | 0.060 | 0.332 | 0.102 | 0.819 | 0.424 | 0.563 |
| MB | 0.071 | 0.354 | 0.130 | 0.676 | 0.387 | 0.619 |
| SRM-MF | 0.085 | 0.416 | 0.141 | 0.609 | 0.330 | 0.651 |

In addressing the sparsity and cold-start (if a new item is added to the collection or a new user is registered to the recommender system) problems, our method is more superior to the other algorithms. Because the missing data will be assigned predictive values by using matrix factorization technique. In addition, we consider a variety of complex factors which make recommendation results more accuracy and diversity.

## 6. Implications and conclusions

Recommender systems have become extremely common in recent years, and are applied in a variety of applications. They collect information on the preferences of its users for a set of items and predict users' possible future likes or interests. In order to increase accuracy and diversity of recommendation results and alleviate the sparsity and cold-start problems, we proposed a novel recommendation method based on social network. In this method, we consider user preferences and the social relationships among users. Detailed numerical analysis on two benchmark data sets and a real data set, indicate that the presented algorithm is more accurate and diverse than other algorithms.

This research reveals that merging social network context can improve the accuracy of recommendation result and increase its diversity. Meanwhile, Matrix Factorization technique proves to be a good solution to data sparsity and cold start. This personalized recommendation strategy can be applied to various situations and fields, such as friend recommendation with social network and product recommendation in E-commerce platform. It has most significant effect in cooperation with user multi-dimension information. This analysis extent research idea of personalized recommendation strategy, enriches its method system.

Concerning future work, we will research in the following aspects:

(1) How to keep the robustness of recommendation algorithm when it meets hostile attacks. Hostile attacks mean that someone makes hostile and large number of invalid ratings or evaluations to recommender systems. Through hostile attacks, it is possible to affect the availability of the recommender systems.
(2) Research on the recommendation method by the fusion of multi-source and heterogeneous data. Construct a more complex model which considers implicit information of users extracted from different domains. For example, users reputation reflected by number of fans or others. And these fans may come from different SNS.

## Acknowledgments

This research was supported by Philosophy and Social Science Foundation of Zhejiang Province (16NDJC189YB); National Natural Science Foundation of China (71702164); Natural Science Foundation of Zhejiang Province (LY15G010001).

## References


Adeniyi, D. A., Wei, Z., & Yongquan, Y. (2016). Automated web usage data mining and recommendation system using K-Nearest Neighbor (KNN) classification method. *Applied Computing and Informatics, 12*(1), 90–108.

Aligon, J., Gallinucci, E., Golfarelli, M., Marcel, P., & Rizzi, S. (2015). A collaborative filtering approach for recommending OLAP sessions. *Decision Support Systems, 69*, 20–30.

Bandura, A. (2001). Social cognitive theory of mass communication. *Media Psychology, 3*(3), 265–299.

Benjamin, L. S. (1974). Structural analysis of social behavior. *Psychological Review, 81*(5), 392–425.

Bond, R., & Smith, P. B. (1996). Culture and conformity: A meta-analysis of studies using asch's (1952,1956) line judgment task. *Psychological bulletin, 119*(1), 111–137.

Dooms, S., Pessemier, T. D., & Martens, L. (2015). Online optimization for user-specific hybrid recommender systems. *Multimedia Tools and Applications, 74*(24), 11297–11329.

Feng, X., Sharma, A., Srivastava, J., Wu, S., & Tang, Z. (2016). Social network regularized sparse linear model for Top-N recommendation. *Engineering Applications of Artificial Intelligence, 51*, 5–15.

Gan, M. (2016). TAFFY: incorporating tag information into a diffusion process for personalized recommendations. *World Wide Web, 19*(5), 933–955.

Han, X., Wang, L., Farahbakhsh, R., Cuevas, Á., Cuevas, R., Crespi, N., et al. (2016). CSD: A multi-user similarity metric for community recommendation in online social networks. *Expert Systems with Applications, 53*(1), 14–26.

Hwang, C.-S., & Chen, Y.-P. (2007). Using trust in collaborative filtering recommendation. *International conference on industrial, engineering and other applications of applied intelligent systems* (pp. 1052–1060). Springer.

Ju, C., & Xu, C. (2014). Personal recommendation via heterogeneous diffusion on bipartite network. *International Journal on Artificial Intelligence Tools, 23*(3), 1–17.

Jiang, M., Cui, P., Wang, F., Zhu, W., & Yang, S. (2014). Scalable recommendation with social contextual information. *IEEE Transactions on Knowledge and Data Engineering, 26*(11), 2789–2802.

Jhamb, Y., & Fang, Y. (2017). A dual-perspective latent factor model for group-aware social event recommendation. *Information Processing & Management, 53*(3), 559–576.

Kumar, V., Pujari, A. K., Sahu, S. K., Kagita, V. R., & Padmanabhan, V. (2017). Proximal maximum margin matrix factorization for collaborative filtering. *Pattern Recognition Letters, 86*, 62–67.

Khodambashi, S., Perry, A., & Nytrø, Ø. (2015). Comparing user experiences on the search-based and content-based recommendation ranking on stroke clinical guidelines-A case study. *Procedia Computer Science, 63*, 260–267.

Kaššák, O., Kompan, M., & Bieliková, M. (2016). Personalized hybrid recommendation for group of users: Top-N multimedia recommender. *Information Processing and Management, 52*(3), 459–477.

Kennedy, J., & Eberhart, R. C. (1995). Particle swarm optimization. *Proceedings of IEEE international conference on neural networks. Vol. 4. Proceedings of IEEE international conference on neural networks* (pp. 1942–1948).

Li, H., Ma, X.-P., & Shi, J. (2016). Incorporating trust relation with PMF to enhance social network recommendation performance. *International Journal of Pattern Recognition and Artificial Intelligence, 30*(6), 1–13.

Liu, R.-R., Jia, C.-X., Zhou, T., Sun, D., & Wang, B.-H. (2009). Personal recommendation via modified collaborative filtering. *Physica A, 388*, 462–468.

Liu, R.-R., Liu, J.-G., Jia, C.-X., Sun, D., & Wang, B.-H. (2010). Personal recommendation via unequal resource allocation on bipartite networks. *Physica A, 389*(16), 3282–3289.

Maillo, J., Ramírez, S., Triguero, I., & Herrera, F. (2017). kNN-IS: An iterative spark-based design of the k-nearest neighbors classifier for big data. *Knowledge-Based Systems, 117*, 3–15.

Ma, H., King, I., & Lyu, M. R. (2011). Learning to recommend with explicit and implicit social relations. *ACM Transactions on Intelligent Systems and Technology (TIST), 2*(3), 29–46.

Narducci, F., Basile, P., Musto, C., Lops, P., Caputo, A., Gemmis, M.d., et al. (2016). Concept-based item representations for a cross-lingual content-based recommendation process. *Information Sciences, 374*, 15–31.

Nilashi, M., Ibrahim, O. B., & Ithnin, N. (2014). Hybrid recommendation approaches for multi-criteria collaborative filtering. *Expert Systems with Applications, 41*(8), 3879–3900.

Puglisi, S., Javier, P.-A., Forné, J., & David, R.-M. (2015). On content-based recommendation and user privacy in social-tagging systems. *Computer Standards and Interfaces, 41*, 17–27.

Park, Y., Park, S., Jung, W., & Lee, S.-g. (2015). Reversed CF: A fast collaborative filtering algorithm using a k-nearest neighbor graph. *Expert Systems with Applications, 42*(8), 4022–4028.

Soares, M., & Viana, P. (2015). Tuning metadata for better movie content-based recommendation systems. *Multimedia Tools and Applications, 74*(17), 7015–7036.

Sun, Z., Han, L., Huang, W., Wang, X., Zeng, X., Wang, M., et al. (2015). Recommender systems based on social networks. *Journal of Systems and Software, 99*, 109–119.

Tsai, Y.-T., Steinberger, M., Pająk, D., & Pulli, K. (2016). Fast ANN for high-quality collaborative filtering. *Computer Graphics Forum, 35*(1), 138–151.

Yesilbudak, M., Sagiroglu, S., & Colak, I. (2017). A novel implementation of kNN classifier based on multi-tupled meteorological input data for wind power prediction, 135:434–444.

Zhou, T., Kuscsikv, Z., Liu, J. G., Medo, M., Wakeling, J. R., & Zhang, Y. C. (2010). Solving the apparent diversity-accuracy dilemma of recommender systems. *PNAS, 107*(10), 4511–4515.

Zhang, B., Hsu, M., & Dayal, U. (1999). *K-harmonic means-a data clustering algorithm* Technical Report HPL-1999-124.