Applied Soft Computing 66 (2018) 548–556

Contents lists available at ScienceDirect

Applied Soft Computing

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

SCFM: Social and crowdsourcing factorization machines for recommendation

Yue Ding [a], Dong Wang [b], Xin Xin [b], Guoqiang Li [b,*], Daniel Sun [c,b], Xuezhi Zeng [d], Rajiv Ranjan [e]

[a] Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
[b] School of Software, Shanghai Jiao Tong University, Shanghai, China
[c] Data61, CSIRO, ACT, Australia
[d] Australian National University, Australia
[e] Chinese University of Geosciences, China

ARTICLE INFO

Article history:
Received 2 August 2016
Received in revised form 14 June 2017
Accepted 14 August 2017
Available online 14 October 2017

Keywords:
Social recommendation
Crowd computing
Factorization machines

ABSTRACT

With the rapid development of social networks, the exponential growth of social information has attracted much attention. Social information has great value in recommender systems to alleviate the sparsity and cold start problem. On the other hand, the crowd computing empowers recommender systems by utilizing human wisdom. Internal user reviews can be exploited as the wisdom of the crowd to contribute information. In this paper, we propose social and crowdsourcing factorization machines, called SCFM. Our approach fuses social and crowd computing into the factorization machine model. For social computing, we calculate the influence value between users by taking users' social information and user similarity into account. For crowd computing, we apply LDA (Latent Dirichlet Allocation) on people review to obtain sets of underlying topic probabilities. Furthermore, we impose two important constraints called social regularization and domain inner regularization. The experimental results show that our approach outperforms other state-of-the-art methods.

© 2017 Elsevier B.V. All rights reserved.

1. Introduction

Recommender Systems (RS) [1,29] aim to generate a list of items that may interest the users. Collaborative filtering (CF) techniques are widely used for building RS. Traditional CF methods can be classified into memory based and model based methods. Memory based CF methods rely on the assumption that similar users may have similar interests [1], therefore computing user-user or item-item similarity is the critical step. Memory based CF has good result explanation, but it suffers from its inherent weaknesses: sparsity and cold start. The density of user-item rating matrix for real commercial RS is usually less than 1% in practice. Nearly all of the memory based CF algorithms cannot deal with users who never rated any item. Model based CF methods train a predefined model by adopting machine learning techniques to obtain proper parameters to generate unknown ratings. Since the Netflix price, the Matrix factorization (MF) [12] approach has been a general and effective factorization based CF method. MF approximates the observed rating matrix by two low-rank matrices. Andriy proposes Probabilistic Matrix Factorization(PMF) [19] by using a probabilistic perspective on base MF. PMF performs well on very large and sparse Netflix dataset. PMF and other related methods [30] use the same assumption that the user-specific and item-specific vectors are independent and identically distributed (*i. i. d*.). However, in social network, a user's behavior may influence others. For example, if a friend updates a comment "The Star Wars VII is a great movie" on Facebook and many other friends agree with it, we may be influenced by our friends and have a great interest in this movie. Jiang et al. [10] points out that both of individual preference and interpersonal influence have contributions on rating. Apparently, the *i. i. d*. assumption between users is inappropriate for social network analysis. With the rapid development of web2.0, online social websites and mobile apps have produced a huge volume of social information. Incorporating this important social information into recommender systems will definitely help to improve recommendation quality.

* Corresponding author.
E-mail addresses: dingyue@sjtu.edu.cn (Y. Ding), wangdong@sjtu.edu.cn (D. Wang), xinxin@sjtu.edu.cn (X. Xin), li.g@sjtu.edu.cn (G. Li), daniel.sun@data61.csiro.au (D. Sun), xuezhi.zeng@anu.edu.au (X. Zeng), rranjans@gmail.com (R. Ranjan).

https://doi.org/10.1016/j.asoc.2017.08.028
1568-4946/© 2017 Elsevier B.V. All rights reserved.

Typical CF based social recommendations use rating information and social information as inputs, therefore the general social recommendation framework includes a basic CF model and a social information model [31]. Following the classification of traditional CF methods, social recommendation can be categorized into memory based and model based approaches. The MF related techniques are widely used in model based social recommendation due to its flexibility and nice probabilistic interpretation with Gaussian noise. Approaches [9,15–17] fuse MF with social information to improve recommendation accuracy. On the other hand, crowdsourcing is a new possible way to fight the sparsity problem [13] in RS. Outsourcing micro-requests to the crowd, or exploiting internal user reviews as the wisdom of the crowd can help to improve recommendation accuracy. Intuitively, user reviews (comments) are important for RS in rating prediction.

The main shortcoming of the existing factorization based models is that these approaches may loss possible internal relations between users and ratings when different important factors such as social information and user reviews are considered simultaneously. In this paper, on the basis of our previous work on SocialFM [35], we propose social and crowdsourcing factorization machines, called SCFM. The advantage of our method is that social and crowd computing are combined to generate recommendations. For social influence, we fuse user trust value and user similarity value. For crowdsourcing influence, we apply LDA(Latent Dirichlet Allocation) [2] on user reviews to obtain the latent topic probabilities.

The main contributions of this paper are:

1 Propose an improved factorization machine approach that fuses social and crowd computing;
2 Propose social regularization and domain inner regularization to improve recommendation accuracy.

The rest of the paper is organized as follows. In Section 2, we discuss related work on trusted social recommendation, FMs related techniques and crowd computing for recommender systems. In Section 3, we introduce the construction of input feature vectors, describe the calculation of influence value between users, and then present the objective function, regularization terms and the learning algorithm. Experimental results are illustrated in Section 4, followed by the conclusion in Section 5.

2. Related work

2.1. Trust related social recommendation

Trust information is important for social recommendation. A simple explanation of trust relationship is that user a trusts in user b if a is willing to rely on b's activities [18]. In this case, a is the trustor, b is the trustee. Ma et al. [15] propose Social Trust Ensemble (RSTE) which is a probabilistic factor analysis approach that fuses users' preferences and their trusted friends' interests together. Jamali and Ester [9] propose SocialMF that incorporates trust propagation mechanism into base MF. Trust propagation is a crucial phenomenon in social network analysis and trust-based recommendation. Ma et al. [17] design two social regularization terms for model constraint. This approach treats friends with dissimilar preferences differently in social regularization terms. Yang et al. [34] propose Circle-based recommendation (CircleCon) that focuses on inferring category-specific social trust circles. The basic idea of CircleCon is that a user may trust friends for only specific item categories. Three variants of weigh value definition are presented: equal trust, expertise-based trust and trust splitting. Yang et al. [33] propose TrustMF which is a hybrid approach that combines both of the truster model and the trustee model. Qian et al. [22] propose a PMF based approach that fuses personal interest, interpersonal interest similarity and interpersonal influence into a unified personalized recommendation model.

2.2. Factorization machines

The FM model is a generic framework that integrates the advantages of flexible feature engineering and high-accuracy prediction of factorization models [25]. In FMs, each rating behavior with other information are integrated to generate a transaction described by vector $\mathbf{x}$ with p real-value variables. A FM model of order $d = 2$ is defined as:

$$\hat{y}(x) = w_0 + \sum_{j=1}^{p} w_j x_j + \sum_{j=1}^{p}\sum_{j'=j+1}^{p} x_j x_{j'} \sum_{f=1}^{k} v_{j,f} v_{j',f} \tag{1}$$

where w_0 represents the global bias, w_j represents the bias factor for the j-th variable. The pairwise interaction of vector x_j and $x_{j'}$ is captured by a factorized parametrization $\sum_{f=1}^{k} v_{j,f} v_{j',f}$ instead of an independent parameter, where k is the number of factors. Thus, the hyperparameters Θ include:

$$w_0 \in \Re, \ \mathbf{w} \in \Re^{p}, \ \mathbf{v} \in \Re^{p\times k}. \tag{2}$$

FMs can be extended to higher-order ($d \geqslant 3$) mode. Most of the FMs related work merely focus on second order FMs because higher-order interactions are hard to estimate due to sparse settings [24]. FMs can mimic several factorization models such as PITF [28], SVD++ [11] and BPTF [32], the complexity of FMs is proven to be in linear time $O(k\,n)$ because Eq. (1) is equivalent to:

$$\hat{y}(x) = w_0 + \sum_{j=1}^{p} w_j x_j + \frac{1}{2}\sum_{f=1}^{k}\left[\left(\sum_{j=1}^{p} v_{j,f} x_j\right)^2 - \sum_{j=1}^{p} {v_{j,f}}^2 {x_j}^2\right] \tag{3}$$

Parameters $\theta \in \Theta$ can be learned efficiently by applying alternating least-squares (ALS) [27], adaptive stochastic gradient descent (SGD) [25] and Markov Chain Monte Carlo inference inference [7].

FMs can model contextual information and provide context-aware rating predictions Rendle et al. [27]. Nguyen [21] propose Gaussian Process Factorization Machines (GPFM) to address the limitation of linear combination between contextual variables for context-aware recommendation. Qiang et al. [23] propose Ranking FM model for Microblog retrieval. Rendel [26] proposes Scaling Factorization Machines to address block structure problem and the proposed model speeds up computation. Cheng et al. [4] propose Gradient Boosting Factorization Machines Model (GBFM) to employ gradient boosting algorithms for feature selection. Rendle [25] adopts FMs to solve KDDCup2012 tasks: (Track 1) To predict which microblogger a user is following. (Track 2) To predict the click-through rate of ads. Guo et al. [8] propose PRFM which is a personalized ranking model incorporated with FM. Li et al. [14] propose DiFacto which is a distributed computing FM model with sparse memory adaptive constraints and frequency adaptive regularization. Zhou et al. [35] proposes SocialFM which constructs social information domain for FMs. The social influence propagation is estimated by taking trust value and user similarity into account. The parameters of SocialFM can be learned by using stochastic gradient descent (SGD) method. However, none of these FM related works considers using crowd computation to improve recommendation accuracy.

2.3. Crowd computing for recommender systems

Felfering et al. [6] introduce PeopleViews which employ human computation concepts to extract recommendation knowledge in a

	Feature vector $\mathbf{x}$																	Target $\mathbf{y}$	
	User			Movie				Trust Users			Other Movies rated				User				
	A	B	C	TI	NH	SW	ST	A	B	C	TI	NH	SW	ST	Topic1	Topic2	Topic3		
x^1	1	0	0	1	0	0	0	0	0.2	0.8	0	1	0	0	0.1	0.6	0.3	5	y^1
x^2	1	0	0	0	1	0	0	0	0.2	0.8	1	0	0	0	0	0	0	3	y^2
x^3	0	1	0	0	1	0	0	0	0	1	0	0	0.5	0.5	0	0	0	4	y^3
x^4	0	1	0	0	0	1	0	0	0	1	0	0.8	0	0.2	0	0	0	1	y^4
x^5	0	1	0	0	0	0	1	0	0	1	0	0.8	0.2	0	0	0	0	1	y^5
x^6	0	0	1	0	0	1	0	0	0	0	5/7	0	0	2/7	0	0	0	4	y^6
x^7	0	0	1	0	0	0	1	0	0	0	5/9	0	4/9	0	0	0	0	2	y^7
x^8	0	0	1	1	0	0	0	0	0	0	0	0	2/6	4/6	0	0	0	5	y^8

Fig. 1. An example of feature vectors construction. The *i*th row represents feature vector x^i with its corresponding target rating y^i. The first 3 columns for domain *U* represent indicator variables for the active user. The next 4 columns for domain *I* represent the indicator variables for current rated items. The next 3 columns for domain *T* represent trustees and their corresponding influence values. The next 4 columns for domain *RI* are weighted values for other rated items. The last 3 columns for domain *TR* are probability distribution on topics.

constraint-based recommendation environment. Felfering et al. [5] propose RecTurk for constraint-based recommender application. RecTurk outsources simple micro-tasks to persons without experiences. The ongoing work of collecting data for PoliMovie Nasery et al. [20] aims to provide a feature-based dataset as a benchmark for recommender systems.

3. SCFM

In this section, we describe the details of our proposed SCFM model. Section 3.1 presents input feature vector construction, Section 3.2 introduces influence value computation, Section 3.3 expounds the objective function, regularization terms and the learning algorithm. The order of SCFM is $d = 2$.

3.1. Feature vector construction

In SCFM, rating information and social relationships are transformed into feature vectors containing five categorical domains: user *U*, item *I*, trustee *T*, other rated item *RI* and topic probabilities of user review *TR*. The user domain *U* and item domain *I* are transformed into indicator value. SCFM specifies domain *T* by taking the rating value and the influence value into account. Domain *T* and *RI* are normalized and the sum of domain *TR* is 1. Domain *T* holds implicit influences of trustees, domain *RI* can be viewed as the implicit influences of other rated items, and domain *TR* reflects the probability distribution of user review in latent topics.

Here, we illustrate by a simple example. Assuming that data comes from a movie review system. The system has user-movie rating records, user trust information and people reviews. Let *U*, *I* be:

$$U = \{\textbf{A}\text{lice}, \textbf{B}\text{ob}, \textbf{C}\text{harlie}\}$$
$$I = \{\textbf{TI}\text{tanic}, \textbf{N}\text{otting}\textbf{H}\text{ill}, \textbf{S}\text{tar}\textbf{W}\text{ars}, \textbf{S}\text{tar}\textbf{T}\text{rek}\}$$

The overstriking words represent abbreviated form. Observed ratings and trust relationships with calculated influence values are given by:

$$\text{Rating Record} = \{\{A, TI, 5\}, \{A, NH, 3\}, \{B, NH, 4\}, \{B, SW, 1\}, \{B, ST, 1\}, \{C, SW, 4\}, \{C, ST, 2\}, \{C, TI, 5\}\}$$
$$\text{Trust NetWork} = \{\{A, B, 0.1\}, \{A, C, 0.4\}, \{B, C, 0.5\}\}$$
$$\text{Review} = \{\{A, TI, '\text{The movie is great! I like it very much!}'\}$$

where tuple {A,TI,5} from the Rating Record set means user A scores item 'TI' with the rating value of 5. Tuple {A,B,0.1} from the Trust Network set indicates the trust relationship and the strength of trust value that A trusts in B is 0.1. Tuple {A,B,0.1} represents user A give comment on movie 'TI' when A rates on 'TI'. Fig. 1 illustrates the constructed feature vectors $\mathbf{x}$ which are the inputs for SCFM. $\mathbf{x}^1$ is the first transaction that Alice rates movie Titanic on the score of 5, Alice trusts in Bob and Charlie with the influence value of 0.2 and 0.8 respectively. Besides the film Titanic, Alice also rated the film Notting Hill. Notice that the sum of trust users' influence values is 1. In RI domain, the value of a specific other rated movie is the weight of ratings of all the other rated movies. The *TR* domain shows that the comment of user A on item 'TI' is converted to probability distribution on latent topic 1, 2 and 3.

3.2. Influence value calculation

The influence value between friends is consist of two parts: trust value $s_{a,b}$ and user similarity $sim(a, b)$. In trust network, if user *a* trusts user *b*, then $s_{a,b} = 1$. We use the following formula to compute the trust value between users:

$$s_{a,b} = s_{a,b} \times \sqrt{\frac{d^-(\nu_b)}{d^+(\nu_a) + d^-(\nu_b)}} \tag{4}$$

where $d^+(\nu_a)$ is the outdegree of node ν_a, which indicates the number of users that user *a* trusts. $d^-(\nu_b)$ is the indegree of node ν_b, which represents the number of users who trust user *b*.

For the social network viewed as an undirected graph, each node has the same outdegree and indegree. The trust value $s_{a,b} = s_{b,a} = \sqrt{\frac{1}{2}}$ if user *a* is user *b*'s friend. The trust value cannot actually reflect user *b*'s influence on user *a*. We fuse user similarity into influence calculation together with the trust value, the intuition is that trusted friends with similar tastes may have deeper influence on each other. We apply Pearson Correlation Coefficient(PCC) [3] for similarity computing. The similarity between user *a* and user *b* calculated by PCC method is defined as follows:

$$sim_{PCC}(a, b) = \frac{\sum_{i \in I(a) \bigcup I(b)} \left(r_{a,i} - \bar{r}_a\right)\left(r_{b,i} - \bar{r}_b\right)}{\sqrt{\sum_{i \in I(a) \bigcup I(b)} \left(r_{a,i} - \bar{r}_a\right)^2} \sqrt{\sum_{i \in I(b) \bigcup I(b)} \left(r_{b,i} - \bar{r}_b\right)^2}} \tag{5}$$

where $r_{a,i}$ represents that user *a* rates on item *i*, $\bar{r}_a$ denotes the average rating value of user *a*, item $i \in I(a) \bigcup I(b)$. PCC may have negative result which reflects negative influence between users, we use the function $sim(a, b) = \frac{1}{2}(sim_{PCC}(a, b) + 1)$ to guarantee that PCC similarity is in value interval [0,1]. Then influence value between user a and user b is:

$$g(a, b) = s_{a,b} * sim(a, b) \tag{6}$$

Finally, we normalize values as $g(a, b) = g(a, b) / \sum_{b \in C_a^+} g(a, b)$, where C_a^+ denotes user a's trust user set.

3.3. The SCFM approach

3.3.1. The objective function

We define the objective function for SCFM as follows:

$$\begin{aligned} \mathrm{OPT}\left(\mathbf{D}, \mathbf{\Theta}\right) = \ & \arg\min_{\theta} \Bigg(\sum_{(\mathbf{x},y) \in \mathbf{D}} l\left(\hat{y}\left(\mathbf{x}|\theta\right) - y\right) + \sum_{\theta \in \mathbf{\Theta}} \lambda_\theta ||\theta||_F^2 \\ & + \beta \sum_{a=1}^{n} \sum_{b \in C_a^+} \sum_{f=1}^{k} \left(g(a, b) ||v_{a,f} - v_{b,f}||_F^2\right) \\ & + \alpha \sum_{j=1}^{p} \sum_{f=1}^{k} ||v_{j,f} - v_{j',f}||_F^2) \Bigg) \end{aligned} \tag{7}$$

where $j' \neq j$, j and j' are in the same domain position in U and T, or I and RI. $\mathbf{D}$ represents all training data, Θ represents hyperparameters. n is the number of users, k is the number of factors, p is the variable number of the vector $\mathbf{x}$, $||\cdot||_F^2$ denotes the Frobenius norm, $g(a, b)$ is the influence value, C_a^+ is user a's trust user set, the superscript means the profile in corresponding domain. λ_θ is the regularization parameter to avoid overfitting. The regularization structure of λ_θ is: λ^0 for w_0, λ_w^j for w_j, $\lambda_{v,j}^f$, where $j \in \{1, \ldots, p\}, f \in \{1, \ldots, k\}$.

For feature vectors, since we extend trusted users domain and calculate influence values, we add two important regularization terms: social regularization and domain inner connection regularization. For social regularization, under the assumption that a user's preference is close to his similar trustees, we impose the constraint to differently calculate influences for users' trustees and for other active users. The social regularization term handles the situation if user a and his trustee b have completely different preferences. In this case, the calculated result of small influence value between user a and b will have little effect on objective function, so if we simply compute $\sum_{a=1}^{n} \sum_{f=1}^{k} ||v_{a,f} - \sum_{b \in C_a^+} s_{a,b} v_{b,f}||_F^2$, it may lose information when users' trustees have diverse interests. We adpot the regularization term $\sum_{a=1}^{n} \sum_{b \in C_a^+} \sum_{f=1}^{k} \left(g(a, b) ||v_{a,f} - v_{b,f}||_F^2\right)$ and we employ parameter β as the weight factor. The regularization for domain inner connection handles the situation that input feature vectors are highly similar. In this case, input feature vectors constructed from the same user hold very similar domains. To alleviate this problem, we use the penalty term $\sum_{j=1}^{p} \sum_{f=1}^{k} ||v_{j,f} - v_{j',f}||_F^2$, parameter α is the weight factor. It should be noticed that a better way of measuring user similarity is to use the cosine similarity, which, however, would lead to difficulties in partial derivative calculation. Therefore we utilize distance measure instead.

3.3.2. The learning algorithm

We use a stochastic gradient descent(SGD) algorithm to optimize our objective function. The objective function is convex and the SGD algorithm is an efficient method to optimize factorization models for its low computational and storage complexity. The model parameters we need to estimate are $\Theta = (w_0, \mathbf{w}, \mathbf{v})$, where $w_0 \in \Re, \mathbf{w} \in \Re^p, \mathbf{v} \in \Re^{p \times k}$. The update rules for parameters are as follows:

$$w_0 \longleftarrow w_0 - \eta \left(\frac{\partial}{\partial w_0} l\left(\hat{y}\left(\mathbf{x}|\theta\right), y\right) + 2\lambda_0 w_0 \right) \tag{8}$$

$$w_j \longleftarrow w_j - \eta \left(\frac{\partial}{\partial w_j} l\left(\hat{y}\left(\mathbf{x}|\theta\right), y\right) + 2\lambda_w w_j \right), j \in [1, \ldots, p] \tag{9}$$

When updating $v_{j,f}$, there are two different situations. In the first situation, for active user a, we need to calculate two terms, one is the difference between a and his trustees C_a^+, the other one is the difference between a and his trustors C_a^-. The update rule of $v_{j,f}$ is as Eq. (10).

$$\begin{aligned} v_{j,f} \longleftarrow v_{j,f} - \ \eta \Bigg(& \frac{\partial}{\partial v_{j,f}} l\left(\hat{y}\left(\mathbf{x}|\theta\right), y\right) + 2\lambda_v v_{j,f} \\ & + 2\beta \left(\sum_{b \in C_a^+} g(a, b)(v_{a,f} - v_{b,f}) + \sum_{b \in C_a^-} g(b, a)(v_{a,f} - v_{b,f}) \right) \\ & + 2\alpha(v_{a,f} - v_{a',f}) \Bigg) \end{aligned} \tag{10}$$

where $a' \neq a$. In the second situation, if vector x_j and $x_{j'}$ are both from user a's rating transaction, $v_{j,f}$ is updated by:

$$v_{j,f} \longleftarrow v_{j,f} - \eta \left(\frac{\partial}{\partial v_{j,f}} l\left(\hat{y}\left(\mathbf{x}|\theta\right), y\right) + 2\lambda_v v_{j,f} + 2\alpha(v_{j,f} - v_{j',f}) \right) \tag{11}$$

where $j' \neq j$. j and j' are in the same domain position in U and T, or I and RI. Algorithm 1 describes the details.

Algorithm 1. Learning Algorithm for SCFM

Input:
Feature vector $\mathbf{x}$ with rating value y from training data D,
Regularization parameters λ_θ, learning rate η, weight factors α and β,
initialization parameter σ
Output:
Model parameters $\Theta = (w_0, \mathbf{w}, \mathbf{v})$
$w_0 \longleftarrow 0; \mathbf{w} \longleftarrow (0, \ldots, 0); \mathbf{v} \sim N(0, \sigma);$
repeat
for $(\mathbf{x}, y) \in D$ **do**
update w_0 by Eq. (8).
for $j \in [1, \ldots, p] \wedge x_j \neq 0$ **do**
update w_j by Eq. (9).
for $f \in [1, \ldots, k]$ **do**
if $j \in$ domainUandactiveuserisa **then**
update $v_{j,f}$ by Eq. (10).
else
update $v_{j,f}$ by Eq. (11).
end if
end for
end for
end for
until stopping criterion is met

3.3.3. Computational complexity

For input vectors that have the same domain U, for example, from the rating transaction of active user a, we need to calculate all the dissimilarities between user a and his trustees in set C_a^+ and the differences between a and his trustors in set C_a^-. In each step of the learning process, users' feature vectors $\mathbf{v}$ are updated by the result of influence value from their trustees and trustors, which can be viewed as an influence propagation process. When the objective function converges during the learning phase, the propagation of influences will reach a harmonic status.

Vectors $\mathbf{x}$ constructed from real-word transaction data are very sparse, the FM related models are efficient because most of the

Table 1
Description of test datasets with basic meta.

Dataset	Users	Items	Ratings	Social	Review	Density
FilmTrust	1508	2071	35497	1853	–	1.13%
Epinions	1809	2000	12,057	23,090	–	0.33%
CiaoDVD	973	1197	17,604	4221	–	1.51%
Musical instruments	900	1429	10,261	–	10,261	0.79%
Automotive	1835	2928	20,473	–	20,473	0.38%
Instant video	1685	5130	37,126	–	37,126	0.42%
Yelp	2000	1699	3103	5223	3103	0.09%

elements are zero. For SCFM, the computational complexity of evaluating each predicted rating is $O(k\bar{m})$, where $\bar{m}(x)$ is the average value of m(x) for x ∈ all transactions. The computational complexity of parameter learning is $O(|C_u|)$, where $|C_u|$ is the average number of trustees and trustors for all users.

4. Experiment

4.1. Datasets

We test the SCFM model on three different groups of datasets, which are shown in Table 1. Group one contains FilmTrust, Epinions and CiaoDVD. FilmTrust and CiaoDVD datasets are taken from https://www.librec.net/datasets.html. The Epinions dataset are selected from https://www.trustlet.org/wiki/Epinions. Group two contains selected Amazon review datasets of Musical Instruments, Automotive and Instant Video from http://jmcauley.ucsd.edu/data/amazon/. Group three is the selected data from Yelp Dataset Challenge. Datasets in group one contain social information, datasets in group two contain review information, dataset in group three contains both social and review information.

4.2. Metrics

We apply two popular metrics to evaluate the prediction quality: Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). The metrics are defined as:

$$MAE = \frac{\sum_{i,j}|r_{i,j} - \hat{r_{i,j}}|}{N}, \quad RMSE = \sqrt{\frac{\sum_{i,j}\left(r_{i,j} - \hat{r_{i,j}}\right)^2}{N}} \tag{12}$$

where $r_{i,j}$ is the observed rating value that user u rates on item i in the validation set, $\hat{r_{i,j}}$ is the predicted rating value that user u rates on unknown item i. N is the total number of ratings in the validation set. Smaller values of MAE and RMSE represent better performance.

4.3. Experimental result comparison

4.3.1. Baseline approaches

We adopt two different groups of baseline approaches. The approaches in the first group are traditional collaborative filtering algorithms that do not utilize social or crowd information.

- BiasedMF [11]: this method is a baseline estimate that fuses the global average and observed deviations of a specific user and a certain item.
- PMF [19]: this method is a well-known baseline. PMF models the Base MF method from a probabilistic perspective.
- BPMF [30]: this method presents a fully Bayesian treatment of the PMF model and BPMF is trained by using Markov chain Monte Carlo (MCMC) method.
- SVD++ [11]: this method is another well-known baseline. SVD++ is an extension of SVD-based latent factor models that integrates implicit feedback into the model.

Table 2
Baseline algorithm comparison I.

	RSTE	SoRec	SocialMF	SoReg	TrustMF	SCFM
MAE						
FilmTrust	0.630	0.631	0.638	0.672	0.627	**0.605**
Epinions	0.889	0.879	0.858	0.955	0.856	**0.815**
CiaoDVD	0.849	0.745	0.821	0.730	0.737	**0.695**
RMSE						
FilmTrust	0.811	0.812	0.837	0.878	0.808	**0.798**
Epinions	1.118	1.115	1.091	1.213	1.106	**1.055**
CiaoDVD	1.053	0.973	1.042	0.969	0.946	**0.931**

The bold values indicate that SCFM approach achieves the best result compared with baseline algorithms.

The methods in the second group are social-related algorithms that fully consider social and trust information between users.

- RSTE [15]: this method combines MF and social analysis together with the notion that the predicted rating of user u on item i should reflect the preferences of u himself and u's trustees.
- SoRec [16]: this method fuses users' social network information into the user-item rating matrix and solves the problem by using PMF.
- SocialMF [9]: this method incorporates trust propagation into base MF approach.
- SoReg [17]: this method proposes social regularizations to constrain the objective function.
- TrustMF [33]: this method proposes a hybrid model that combines the trustor model and the trustee model from the perspectives of trustors and trustees.

4.3.2. Results comparison

For each experiment, we use a 5-fold cross-validation method and take the mean as the final result. The proportion of the training set is 80%, and the rest 20% is for validation set. We adopt a grid search strategy to find optimal parameters for test algorithms.

Table 2 describes the comparison of SCFM and other baseline algorithms performed on FilmTrust, Epinions and CiaoDVD dataset. The three test datasets contain social information without review information, so the TR domain in SCFM is excluded. For SCFM, we set $\lambda_0 = -0.01$, $\lambda_w = -0.0001$ and $\lambda_{v,f} = 0.01$, learning rate $\eta = 0.003$ and the number of factor is 5. Parameters $\alpha = \beta = 0.1$. We observe that, compared with the best baseline algorithm outputs, SCFM improves 4.3% for MAE and 1.9% for RMSE on the average.

Table 3 describes the comparison of SCFM and other baseline algorithms performed in Amazon Musical Instruments, Amazon Automotive and Amazon Instant Video dataset. The three test datasets contain review information without social information,

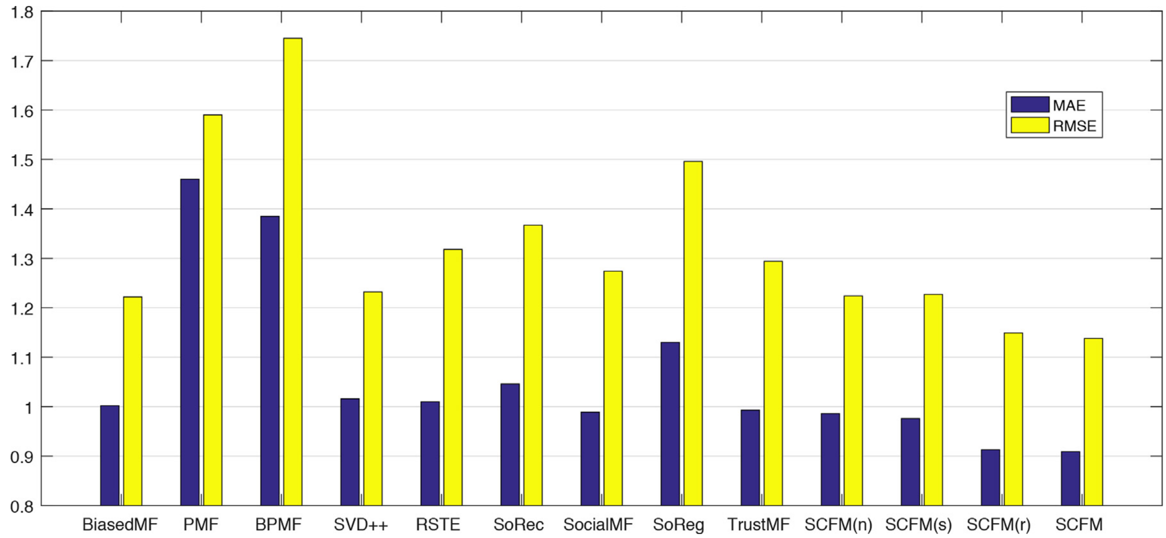
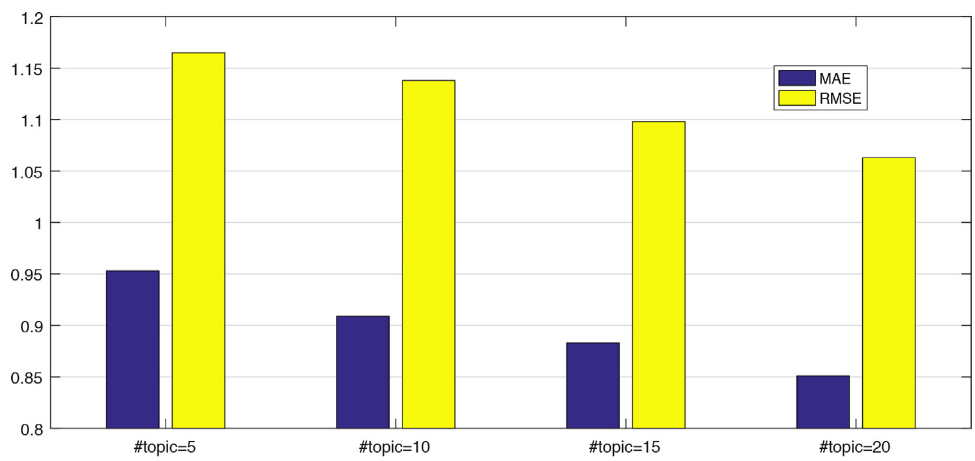
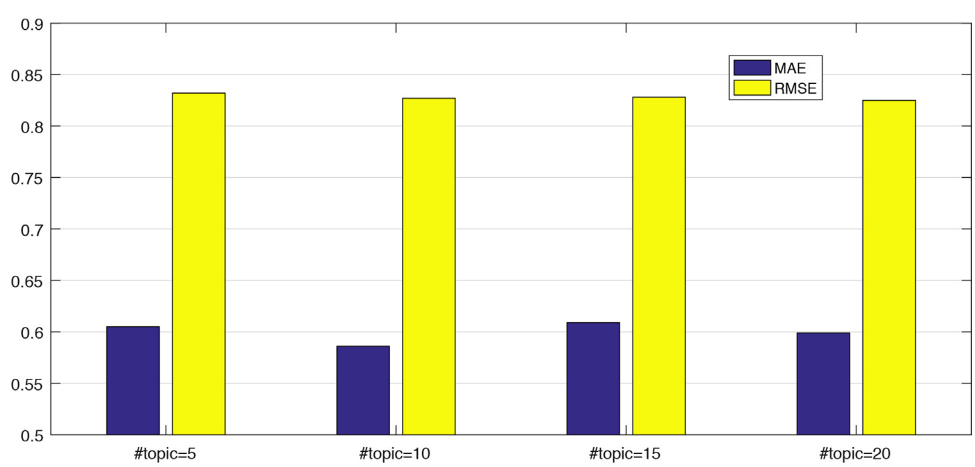


Fig. 2. Baseline algorithm comparison III.

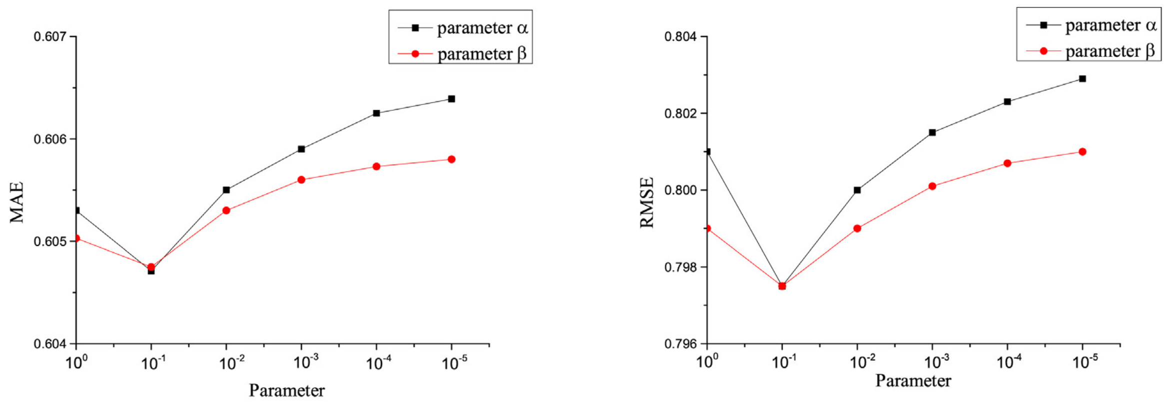


(a) Yelp



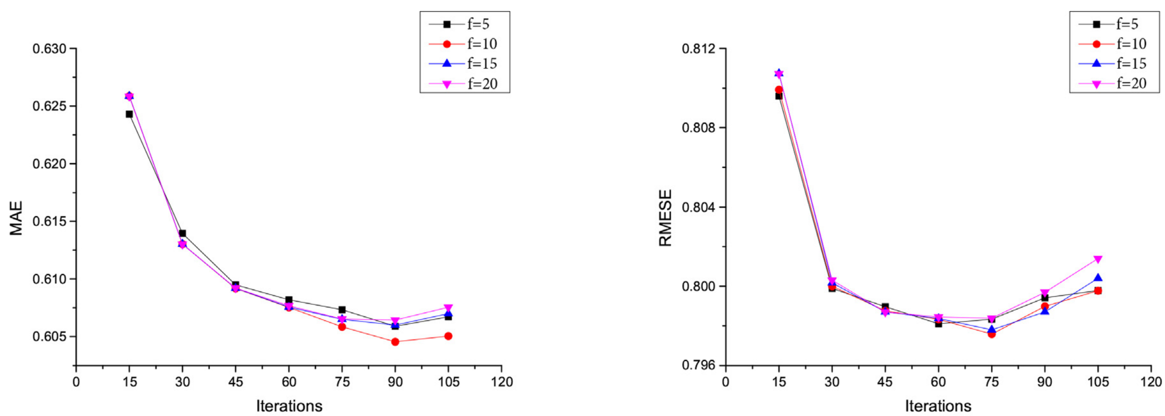
(b) Amazon Musical Instruments

Fig. 3. Impact of topics.



(a) MAE (b) RMSE

Fig. 4. Impact of parameter α and β in FilmTrust.



(a) MAE (b) RMSE

Fig. 5. Performance on FilmTrust with different value of *f*.

Table 3
Baseline algorithm comparison II.

	BiasedMF	PMF	BPMF	SVD++	SCFM
MAE					
Musical instruments	0.621	0.890	0.694	0.629	**0.586**
Automotive	0.626	0.896	0.703	0.630	**0.622**
Instant video	0.707	0.909	0.804	0.721	**0.657**
RMSE					
Musical instruments	0.876	1.165	0.989	0.874	**0.827**
Automotive	0.894	1.171	1.025	0.896	**0.857**
Instant video	0.961	1.197	1.126	0.963	**0.888**

The bold values indicate that SCFM approach achieves the best result compared with baseline algorithms.

so the T domain in SCFM is excluded and the objective function of SCFM is as Eq. (13),

$$\mathrm{OPT}\left(\mathbf{D},\boldsymbol{\Theta}\right) = \underset{\theta}{arg\,min}\left(\sum_{(\mathbf{x},y)\in\mathbf{D}} l\left(\hat{y}\left(\mathbf{x}|\theta\right)-y\right)+\sum_{\theta\in\boldsymbol{\Theta}}\lambda_\theta||\theta||_F^2 + \alpha\sum_{j=1}^{p}\sum_{f=1}^{k}||v_{j,f}-v_{j',f}||_F^2)\right) \quad (13)$$

We set parameters as follows, $\lambda_0=0.01$, $\lambda_w=0.01$ and $\lambda_{v,f}=0.01$, learning rate $\eta=0.001$, $\alpha=0.1$, the number of factor is 5 and the number of latent topic is 10. We can observe that SCFM improves 4.7% for MAE and 5.6% for RMSE on the average.

Fig. 2 illustrates the algorithms performed on Yelp dataset, the x-axis represents algorithms. Note that we evaluate SCFM in different forms. SCFM(n) represents SCFM without social and review term, then SCFM turns to be basic FM model. SCFM(s) represents SCFM with social domain only and SCFM(r) represents SCFM with review domain only. The best results of MAE and RMSE achieved from baseline algorithms are 0.993 and 1.222. SCFM achieves 0.909 on MAE and 1.138 on RMSE, which makes improvement by 8% on MAE and 6.8% on RMSE. The parameters are set as follows, $\lambda_0=-0.01$, $\lambda_w=-0.0001$ and $\lambda_{v,f}=0.01$, learning rate $\eta=0.003$, $\alpha=\beta=0.1$, the number of factor is 5 and the number of latent topic is 10. We notice that the test Yelp dataset is very sparse, each user rates 1.55 on the average and each item is rated for 1.82 times, the utilization of social and review information can greatly improve rating accuracy. We can also find that review information contributes more on rating prediction compared with social information, and the combination of social and review information achieves the best result.

Fig. 3 describes the impact of topics. We test on Yelp and Amazon Musical Instrumental datasets. Considering that reviews are short comments and the average number of words per comment is about 80, we set the maximum number of topics as 20. For Yelp dataset, MAE and RMSE decrease as the number of topics increases.

For Amazon Musical Instrumental dataset, MAE and RMSE fluctuate as the number of topics increases. It needs to be mentioned that the computation complexity rises if the number of topics increases. We investigate the relationship between accuracy and efficiency on Yelp dataset. We find that SCFM improves 13.7% for MAE and 6.5% for RMSE compared with base FM model, but the time consumption of SCFM is 2.6 times than that of base FM. It is easy to understand because the computational complexity of SCFM expands with the increase of input vector dimension. How to balance accuracy and efficiency is a problem needs to be studied in future work.

In the regularization term of SCFM, parameter α controls the weight of inner relationship of domains and parameter β controls the weight of social information, Fig. 4 illustrates the impact of α and β in FilmTrust datasets. Both of MAE and RMSE perform best when $\alpha = 0.1$ and $\beta = 0.1$. The results of MAE and RMSE have no obvious fluctuation with the wide change of α and β.

Fig. 5 illustrates the performance of RMSE and MAE when the number of factors f changes on FilmTrust dataset. When all the other parameters are fixed, we can observe that we can get the best result when $f = 10$.

4.3.3. Time cost

We run our python code on the PC with the Intel i7-6700 CPU and 8 GB memory. When the SCFM learning algorithm achieves the best MAE and RMSE results, the time cost on the Epinions dataset which has the most social connections is 587 s, the time cost on the Amazon Instant Video dataset which has the most review contents is 337 s, and the time cost on the Yelp dataset which has both social and review information is 32 s. It is clear that the social factor has great impact on computational complexity.

5. Conclusion

In this paper, we propose SCFM which is an improved factorization machine model combining social and crowd information. SCFM constructs input feature vectors by five domains: user, item, trustee, other rated items and topic. The proposed method can take advantage of social and crowd information to efficiently estimate interactions between categorical variables. SCFM can simulate typical characteristics of social network by calculating the influence between users, and apply LDA to obtain underlying topic probabilities. We impose the social regularization to handle the situation that trusted users have complete different preferences, and we build inner regularization to alleviate the situation that input vectors are highly similar. The experimental results show that our method is flexible and outperforms the state-of-the-art algorithms. However, the SCFM model has its limitations. SCFM cannot deal with cold start problem. Another shortcoming is that the SCFM approach lacks a mechanism for dealing with large-scale data. Future work may focus on the following aspects. First, we consider to design a mechanism to deal with new user and item situation. One idea is that we can choose a few representative users and items, then a new user or item can be expressed by representative users or items using randomly given weight value. Second, we consider to design a distributed algorithm for SCFM to solve the bottleneck concerning large-scale data. Last, we consider to adopt more semantic analysis methods on user review information. Since review information plays an important role in making recommendation, we would try to investigate the inner connection between review and rating to make the recommendation results explainable.

Acknowledgements

This project is supported by the National Natural Science Foundation of China (Nos. 61672340, 61472240, 61572268).

References

[1] G. Adomavicius, A. Tuzhilin, 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) (2005) 734–749.
[2] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[3] J.S. Breese, D. Heckerman, C. Kadie, Empirical analysis of predictive algorithms for collaborative filtering, Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (1998) 43–52.
[4] C. Cheng, F. Xia, T. Zhang, I. King, M.R. Lyu, Gradient boosting factorization machines, Proceedings of the 8th ACM Conference on Recommender systems (RecSys) (2014) 265–272.
[5] A. Felfernig, S. Haas, G. Ninaus, M. Schwarz, T. Ulz, M. Stettinger, Recturk: constraint-based recommendation based on human computation, CrowdRec 2014 (ACM RecSys Workshop) (2014).
[6] A. Felfernig, T. Ulz, S. Haas, M. Schwarz, S. Reiterer, M. Stettinger, Peopleviews: human computation for constraint-based recommendation, CrowdRec 2015 (ACM RecSys Workshop) (2015).
[7] C. Freudenthaler, L. Schmidt-Thieme, S. Rendle, Bayesian factorization machines, in: Workshop on Sparse Representation and Low-rank Approximation, Neural Information Processing Systems (NIPS-WS), 2011.
[8] W. Guo, S. Wu, L. Wang, T. Tan, Personalized ranking with pairwise factorization machines, Neurocomputing (2016), http://dx.doi.org/10.1016/j.neucom.2016.05.074.
[9] M. Jamali, M. Ester, A matrix factorization technique with trust propagation for recommendation in social networks, Proceedings of the fourth ACM conference on Recommender systems (RecSys) (2010) 135–142.
[10] M. Jiang, P. Cui, R. Liu, Q. Yang, F. Wang, W. Zhu, S. Yang, Social contextual recommendation, Proceedings of the 21st ACM International Conference on Information and Knowledge Management (CIKM) (2012) 45–54.
[11] Y. Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, Proceedings of the 14th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD) (2008) 426–434.
[12] Y. Koren, R. Bell, C. Volinsky, Matrix factorization techniques for recommender systems, Computer 42 (8) (2009) 30–37.
[13] M. Larson, A. Said, Y. Shi, Activating the crowd: exploiting user-item reciprocity for recommendation, CrowdRec 2013 (ACM RecSys Workshop) (2013).
[14] M. Li, Z. Liu, A.J. Smola, Y.-X. Wang, Difacto: Distributed factorization machines, WSDM '16 Proceedings of the Ninth ACM International Conference on Web Search and Data Mining (2016) 377–386.
[15] H. Ma, I. King, M.R. Lyu, Learning to recommend with social trust ensemble, 32nd International ACM Conference on Research and Development in Information Retrieval (SIGIR) (2009) 203–210.
[16] H. Ma, H. Yang, M.R. Lyu, I. King, Sorec: social recommendation using probabilistic matrix factorization, 17th ACM Conference on Information and Knowledge Management (CIKM) (2008) 931–940.
[17] H. Ma, D. Zhou, C. Liu, M.R. Lyu, I. King, Recommender systems with social regularization, The Fourth ACM International Conference on Web Search and Data Mining (2011) 287–296.
[18] R.C. Mayer, J.H. Davis, F.D. Schoorman, An integrative model of organizational trust, Acad. Manag. Rev. 20 (3) (1995) 709–734.
[19] A. Mnih, R. Salakhutdinov, Probabilistic matrix factorization, Advances in Neural Information Processing Systems (NIPS) (2007) 1257–1264.
[20] M. Nasery, M. Elahi, P. Cremonesi, Polimovie: a feature-based dataset for recommender systems, CrowdRec 2015 (ACM RecSys Workshop) (2015).
[21] T.V. Nguyen, A. Karatzoglou, L. Baltrunas, Gaussian process factorization machines for context-aware recommendations, The 37th International ACM Conference on Research & Development in Information Retrieval (SIGIR) (2014) 63–72.
[22] X. Qian, H. Feng, G. Zhao, T. Mei, Personalized recommendation combining user interest and social circle, IEEE Trans. Knowl. Data Eng. (TKDE) 26 (7) (2014) 1763–1777.
[23] R. Qiang, F. Liang, J. Yang, Exploiting ranking factorization machines for microblog retrieval, The 22nd ACM International Conference on Information & Knowledge Management (CIKM) (2013).
[24] S. Rendle, Factorization machines, IEEE 10th International Conference on Data Mining (ICDM) (2010) 995–1000.
[25] S. Rendle, Factorization machines with libfm, ACM Trans. Intell. Syst. Technol. 3 (3) (2012) 57;
S. Rendle, Learning recommender systems with adaptive regularization, The 5th ACM International Conference on Web Search and Data Mining (WSDM) (2012) 133–142;
S. Rendle, Social network and click-through prediction with factorization machines, KDD-Cup Workshop (2012).
[26] S. Rendle, Scaling factorization machines to relational data, VLDB J. 6 (5) (2013) 337–348.
[27] S. Rendle, Z. Gantner, C. Freudenthaler, L. Schmidt-Thieme, Fast context-aware recommendations with factorization machines, The 34th International ACM Conference on Research and Development in Information Retrieval (SIGIR) (2011) 635–644.
[28] S. Rendle, L. Schmidt-Thieme, Pairwise interaction tensor factorization for personalized tag recommendation, The Third ACM International Conference on Web Search and Data Mining (2010) 81–90.

[29] P. Resnick, H.R. Varian, Recommender systems, Commun. ACM 40 (3) (1997) 56–58.
[30] R. Salakhutdinov, A. Mnih, Bayesian probabilistic matrix factorization using Markov chain Monte Carlo, Proceedings of the 25th International Conference on Machine Learning (ICML) (2008) 880–887.
[31] J. Tang, X. Hu, H. Liu, Social recommendation: a review, Soc. Netw. Anal. Mining 3 (4) (2013) 1113–1133.
[32] L. Xiong, X. Chen, T.-K. Huang, J.G. Schneider, J.G. Carbonell, Temporal collaborative filtering with Bayesian probabilistic tensor factorization, SIAM International Conference on Data Mining, vol. 10 (2010) 211–222.
[33] B. Yang, Y. Lei, D. Liu, J. Liu, Social collaborative filtering by trust, The 23th International Joint Conference on Artificial Intelligence (IJCAI) (2013) 2747–2753.
[34] X. Yang, H. Steck, Y. Liu, Circle-based recommendation in online social networks, The 18th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD) (2012) 1267–1275.
[35] J. Zhou, D. Wang, Y. Ding, L. Yin, Socialfm: a social recommender system with factorization machines, in: 17th International Conference on Web-Age Information Management (WAIM), Lecture Notes in Computer Science, vol. 9658, 2016, pp. 286–297.