This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

Social Recommendation With Evolutionary Opinion Dynamics

Fei Xiong, Ximeng Wang, Shirui Pan, *Member, IEEE*, Hong Yang, Haishuai Wang, and Chengqi Zhang, *Senior Member, IEEE*

***Abstract*—When users in online social networks make a decision, they are often affected by their neighbors. Social recommendation models utilize social information to reveal the impact of neighbors on user preferences, and this impact is often described by the linear superposition of neighbor preferences or by global trust propagation. Further exploration needs to be undertaken to determine whether the influence pattern of other users from online interaction behaviors is adequately described. In this paper, we introduce evolutionary opinion dynamics from the field of statistical physics into recommender systems, characterizing the impact of other users. We propose an opinion dynamic model by evolutionary game theory. To describe online user interactions, we define the strategies during an interaction between two users, and present the payoff for each strategy in terms of errors of estimated ratings. Therefore, user behaviors are associated with their preferences and ratings. In addition, we measure user influence according to their topological roles in the social network. We incorporate evolutionary opinion dynamics and user influence into the recommendation framework for the prediction of unknown ratings. Experiment results on two real-world datasets demonstrate that our method outperforms state-of-the-art models in terms of accuracy, and it also performs well for cold-start users. Our method reduces the divergence of user preferences, in accordance with online opinion interactions. Furthermore, our method has approximate computational complexity with matrix factorization, and results in less computation than state-of-the-art models. Our method is quite general, and indicates that studies in social physics, statistics, and other research fields may be involved in recommendation to improve the performance.**

Manuscript received April 18, 2018; accepted June 29, 2018. This work was supported in part by the Fundamental Research Funds for the Central Universities under Grant 2017JBM013, and in part by the Beijing Natural Science Foundation under Grant 4184084. This paper was recommended by Associate Editor F. Wang. *(Fei Xiong and Ximeng Wang are co-first authors.) (Corresponding author: Fei Xiong.)*

F. Xiong is with the Key Laboratory of Communication and Information Systems, Beijing Municipal Commission of Education, Beijing Jiaotong University, Beijing 100044, China (e-mail: xiongf@bjtu.edu.cn).

X. Wang is with the Key Laboratory of Communication and Information Systems, Beijing Municipal Commission of Education, Beijing Jiaotong University, Beijing 100044, China, and also with the Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: wangxm@bjtu.edu.cn).

S. Pan, H. Yang, and C. Zhang are with the Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: shirui.pan@uts.edu.au; hong.yang@student.uts.edu.au; chengqi.zhang@uts.edu.au).

H. Wang is with the Department of Biomedical Informatics, Harvard University, Boston, MA 02115 USA (e-mail: haishuai_wang@hms.harvard.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TSMC.2018.2854000

***Index Terms*—Evolutionary opinion dynamics, game theory, matrix factorization (MF), recommender systems, user influence.**

I. Introduction

With the advent of Web 2.0 technologies, there has been an enormous growth in the amount of data which is constantly being generated [1]. This presents a significant challenge in terms of finding useful information. In recent years, recommender systems [2] have attracted a lot of attention as a tool for information filtering and have been used in many aspects of people's life. The common algorithms for recommender systems contain content-based recommendation and collaborative filtering. For content-based recommendation [3], a set of detailed user and item features needs to be collected; however, in many situations, features are not easy to obtain, and the authenticity and validity of the features often cannot be guaranteed. On the contrary, collaborative filtering [4] uses past review or rating data without the need for exogenous information, and therefore, it has widely been applied in real society.

Collaborative filtering can be divided into two categories, i.e., memory-based and model-based filtering. Memory-based approaches calculate similarities among users or items, and find the neighborhood for each user or item in terms of the similarities [5]. Then, missing ratings are predicted by a weighted sum of ratings from similar users or items. In contrast, model-based approaches use machine learning algorithms to build a predictive model based on existing ratings, such as matrix factorization (MF) [4]. These approaches map the user-item rating matrix into two low-rank matrices. Users and items are identified as the same dimensional vectors, and elements in these vectors express the weights of users or items on latent factors. Therefore, user factor vectors can be regarded as the preferences and opinions of users. A user's rating on an item can be inferred by the product of the user vector and item vector. Model-based approaches have been proved to have higher accuracy and scalability. Exogenous content features can also be introduced to MF. It has been proved that these features improve the performance of recommender systems [6].

Users in online social networks often interact with others. They may observe the actions of other users, and comment on these users. Therefore, they make connections with other users. The connections contain both physical links and virtual trust

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

relations [7]. Users' social relations have been combined with recommender systems [8]. Users who have connections with each other are assumed to have similar tastes. Current studies consider that user ratings are not only determined by their own opinions, but they are also influenced by the tastes of their friends [9]. Therefore, opinions of neighbors are incorporated into the product of latent vectors. In addition, social regularization is used to minimize the difference between latent vectors of each user and its neighbors [10]. The network of user relations has also been mapped into latent factor spaces in two ways, which can explicitly describe the feedback of how users affect or follow the opinions of others. Recommender systems often have a better performance than traditional recommendation algorithms due to their inclusion of social or trust information [11].

Indeed, when users in social networks make a decision on ratings or reviews, opinions and behaviors about items will be directly or indirectly affected by others [12]–[14]. Current studies often model the impact of friends by a linear combination of others' latent vectors [15]. The combination of vectors may affect the ratings or social connections among users, including both explicit and implicit influence. However, it is still unclear whether the evolution of a user's opinion follows the superposition of others when it interacts with friends. The evolutionary pattern of opinions has been widely investigated in the research field of social physics and statistical learning [16]–[18]. In the Deffuant–Weisbuch (DW) opinion model [19], when two users discuss a topic, their opinions change and become closer to each other. In the model with continuous opinions and discrete actions [20], users change their opinions under the Bayesian rule of how likely their neighbors are to be correct, after they observe the external actions (ratings or reviews) of their neighbors. This model promotes the appearance of extreme opinions and forces opinions to cluster together. In [21], users update their opinions according to the birth–death and death–birth process during interactions. These models often originate from real physical phenomena, and have been verified in the interactions of real society. Further exploration is needed to determine whether these opinion models can be applied to characterize real opinion interactions in online social networks. The impact pattern of opinions which is integrated in the MF framework greatly affects the recommendation performance. Current trust-based models may not adequately capture the essential characteristics of individual opinion evolution. The work in [22] reveals that trust-based models may be inferior to the state-of-the-art model which is merely based on user-item ratings. Therefore, a simple combination of latent user vectors may not make full and adequate use of ratings and trust information.

In this paper, we propose a recommendation model that includes opinion interactions and user influence. Evolutionary opinion dynamics are introduced to recommender systems. We characterize the impact of neighbors on user opinions by evolutionary game theory. We define the strategies during an interaction of two users, i.e., changing or keeping their opinions, and give the payoff for each strategy. Users choose a better strategy to maximize their payoffs when they discuss an item with another user. Opinion interactions are conducted with the MF framework, and therefore, user ratings are affected by the opinions of others. In addition, user influence which measures the status of a node in the network, is added to the recommendation model, so that the ratings of each user are weighted. We conduct experiments on two real-world datasets, and the results demonstrate that our method works better than state-of-the-art recommendation models. Furthermore, our method has much less computational complexity than its counterparts. This paper reveals that studies in other research fields, such as social physics and statistics, can be incorporated in recommender systems, to improve the recommendation performance. This paper makes the following contributions.

1) We model online opinion dynamics using evolutionary game theory. The payoffs of strategies during an interaction are associated with latent item factors and observed ratings. Users update their opinions to reduce rating errors and the distances between their opinions. This model considers both the dynamic process in real society and the rating prediction of recommender systems.
2) We introduce opinion dynamics and user influence to the MF framework, and improve the recommendation. During the training of MF, users update their opinions according to the payoff matrix of the game. When users make decisions on items, they are affected by others, so the opinions of others contribute to the ratings. In addition, user influence that originates from the trust network is added to the recommendation. The method which combines MF and random dynamics is general.
3) We conduct extensive experiments to evaluate the effectiveness of our method for all users and cold-start users. We compare our method with several state-of-the-art recommendation models, and analyze the computational complexity of the proposed method. Results show that our method outperforms its counterparts and encourages users to reduce the divergence of their opinions, in accordance with real dynamics.

The rest of this paper is structured as follows. Section II overviews the related work on social recommendation. Section III introduces the MF approach. Section IV proposes a recommendation method with opinion interactions and user influence. Section V presents the evaluation of the method for two sets of real social data. We close this paper in Section VI with concluding remarks.

II. Related Work

Social-based recommendation models often integrate the preferences of neighbors into the prediction of unknown ratings, so that social information is utilized in the recommendation framework [11]. Latent factors of neighbors may affect the prediction of user ratings or relations. In [23], user's social trust was combined with probabilistic MF, and the interests of users and their trusted friends were fused to make a decision on uncollected ratings. Instead of integrating social information for rating prediction, recommendation with social regularization exerts social constraints on the

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

MF framework [10]. Users may take the average preference of neighbors, or they may have similar interests with each neighbor. Both trust networks and social networks are considered. Social relations are not homogeneous among different users, and weak dependency connections exist widely in social networks. Weak dependency connections represent the relations among users in a group that have similar tastes. In [24], after community detection, social dimensions that express user tastes were exploited, and a user may be involved in different dimensions. Based on social dimensions, a recommendation framework was proposed, which incorporates the heterogeneity of social relations and weak dependency connections. Social dimensions improve the effectiveness of recommendation.

From an analysis of real-world datasets, rating data and social data in social networks are usually complementary. Guo *et al.* [6] incorporated both the explicit and implicit influence of user trust, and both trusted and trusting users were considered in the prediction of ratings for an active user. Explicit and implicit influence result in a better performance than other social-based models.

User preferences do not always remain unchanged, instead, they drift over time. Zhang *et al.* [25] inferred the latent social network from cascade data, and identified the dynamic changes of users over time using the latest updated social network. A model of implicit dynamic social recommendation was proposed to address the common existing preference drafting issues. Mining social information in time helps to improve recommendation. Tang *et al.* [26] leveraged social science theories to develop a methodology for this paper of online trust evolution. The dynamics of user preferences were exploited to reveal trust evolution. The trust evolution model can be applied for trust prediction, rating prediction, and ranking prediction.

User relations are not always positive, and social networks also contain negative links. The work in [27] exploited signed social networks for recommendation, and leveraged positive and negative links in signed social networks. The preferences of users are likely to be closer to those of their friends than those of their foes. The results proved that negative links in signed social networks were as important as positive links for recommendation.

These aforementioned studies utilized social information directly, and user relations were incorporated with latent interest vectors in recommender systems. Hu *et al.* [28] measured user influence from network topology. They distinguished different social relations among users, and latent user preferences were learned from those who have the most influence in the social network. The Shannon entropy principle was used to optimize an influence factor, and the topological distances of users were calculated for the building of influence. Zhang *et al.* [29] developed the global influential model and local influential model to find influential users. They carried out Monte-Carlo simulations to obtain an approximate result while handling large-scale user networks. Global and local influence were used as regularization terms in the MF framework. The experiment results proved that these methods which explore user influence from social relations have an advantage in terms of accuracy and stability.

III. Regularized Matrix Factorization

MF is an effective approach for recommender systems to predict missing ratings. This method assumes that user decisions are determined by a few latent factors, and a rating is estimated according to how an item meets a user's preference toward the latent factors.

We define the set of users as $\{u_1, u_2, \ldots, u_m\}$, and the set of items as $\{v_1, v_2, \ldots, v_n\}$. m denotes the number of users, and n denotes the number of items. The ratings are given by a matrix $\boldsymbol{R} \in \mathbb{R}^{m\times n}$. MF decomposes the $m\times n$ rating matrix into two low-rank matrices $\boldsymbol{U} \in \mathbb{R}^{m\times d}$ and $\boldsymbol{V} \in \mathbb{R}^{n\times d}$, obviously $d < \min(m, n)$. The rating matrix is expressed by $\boldsymbol{R} = \boldsymbol{U}\boldsymbol{V}^T$, meaning that the target matrix $\boldsymbol{R}$ can be approximated by the product of two low-rank matrices. For an accurate description, we rewrite the approximation process as

$$\boldsymbol{R} = \boldsymbol{U}\boldsymbol{V}^T + \boldsymbol{e} \tag{1}$$

where $\boldsymbol{e}$ is the error matrix. One can find suitable $\boldsymbol{U}$ and $\boldsymbol{V}$ to make the error as small as possible. Thus, we approximate the rating matrix by minimizing

$$\mathcal{L} = \frac{1}{2}\left\|\boldsymbol{R} - \boldsymbol{U}\boldsymbol{V}^T\right\|^2 \tag{2}$$

where $\|\cdot\|$ denotes the Frobenius norm. $\boldsymbol{U}$ and $\boldsymbol{V}$ are obtained from the observed ratings, and they can be utilized to predict the missing ratings. Considering the observed ratings, (2) is changed to

$$\min_{U,V} \mathcal{L} = \min_{U,V} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}\left(R_{ij} - U_i V_j^T\right)^2 \tag{3}$$

where I is a binary function standing for whether user i has rated item j. To avoid over fitting, quadratic regularization terms are added to the sum-of-squared-errors objective function as

$$\min_{U,V} \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{n} I_{ij}\left(R_{ij} - U_i V_j^T\right)^2 + \frac{\lambda}{2}\|\boldsymbol{U}\|^2 + \frac{\lambda}{2}\|\boldsymbol{V}\|^2 \tag{4}$$

where λ is the extent of regularization, and $\lambda > 0$. Stochastic gradient descent (SGD) is applied to optimize the objective function and find a local minimum. In each iteration of training, all the observed ratings are estimated by latent vectors, and the corresponding vectors are updated as follows:

$$\begin{aligned}
\frac{\partial \mathcal{L}}{\partial U_i} &= -\sum_j \left(R_{ij} - U_i V_j^T\right) V_j + \lambda U_i \\
\frac{\partial \mathcal{L}}{\partial V_j} &= -\sum_i \left(R_{ij} - U_i V_j^T\right) U_i + \lambda V_j \\
U_i &\leftarrow U_i - \gamma \frac{\partial \mathcal{L}}{\partial U_i} \\
V_j &\leftarrow V_j - \gamma \frac{\partial \mathcal{L}}{\partial V_j}
\end{aligned} \tag{5}$$

where γ denotes the learning rate. MF is one of the most popular methods in model-based collaborative filtering.

IV. Recommendation With Opinion Dynamics and User Influence

In this section, we introduce our recommendation method in detail. We define the influence of each user, and use it to weight the objective function of MF. Opinion interactions are characterized by evolutionary game theory, and they are incorporated into the SGD training of MF. In the following, first, we introduce the game theory model of opinion dynamics. Then, we describe the method of MF with user influence. We detail the whole training algorithm in Section IV-C. Last, we analyze the computational complexity of the proposed method.

A. Game Theory Model of Opinion Evolution

User i's latent vector U_i is treated as user i's opinion toward latent factors, revealing how each factor applies to the user. Opinions of users do not always remain unchanged. Users try to persuade others to adopt their opinions, and therefore, opinions are dynamic. Users in online social networks may interact with other users, and exchange opinions. On product review websites, when a user publishes a rating or comment on an item, some other users may read the comment and discuss the item with the user. After the interaction, they may change their opinions. Therefore, opinions evolve during the dynamics.

Many opinion models were proposed to characterize the process of opinion interactions. As a typical representative of continuous opinion models, the DW model describes pairwise interactions between users who have similar opinions. In each update event, two agents i and f are selected at random, and they start a conversation. Meanwhile, the assumption of bound confidence is introduced to the opinion model. When the opinions of these two agents are close enough, they will change their opinions. Therefore, if the opinions of user i and f satisfy $\|U_i - U_f\| < \varepsilon$ $(\varepsilon > 0)$, each opinion moves in the direction of the other as

$$U_i \leftarrow U_i + \mu \cdot \left(U_f - U_i\right) \tag{6}$$

$$U_f \leftarrow U_f + \mu \cdot \left(U_i - U_f\right) \tag{7}$$

where μ $(0 < \mu \leq 0.5)$ is the trust parameter of users, and ε is the tolerance threshold. In the DW model, ε and μ are constants during the evolution. For a special case in which opinions have only one dimension, i.e., $d = 1$, if $\varepsilon > 0.5$, all opinions converge to a single central one, and the system reaches consensus. If $\varepsilon < 0.5$, the system reaches a state of fragmentation, in which a final number of opinion clusters occur, scaling with the number of users. The number of clusters is in proportion to $1/\varepsilon$.

The DW model characterizes user interaction behaviors, but the impacts of item factors and observed ratings are ignored during the evolution. In addition, the tolerance threshold ε is fixed for each user; however, users in real society often have different thresholds. Now, we use evolutionary game theory to model the process of user interactions with item and rating information. Game theory investigates the process of decision making when two players struggle to maximize their own payoffs. Meanwhile, game theory can also be used to explore user behaviors in opinion dynamics.

We present the opinion dynamic model through the framework of evolutionary game theory as follows. In each interaction, two users i and f are selected at random, and are regarded as players in a game. An item j is randomly selected, and is treated as a topic. Users generally try to persuade others or reach agreement on the topic. The interaction strategies available to each player are either to change their opinions or maintain their opinions. The payoffs that the players receive depend on the strategies they implement in the game. A strategy with a higher payoff is preferred by players [30]. In real interactions, each player wants to convince the other one that its opinion is correct. Meanwhile, each player tends to adopt the strategy that can decrease errors of estimated ratings. Therefore, user opinions and ratings should be included in payoffs. Assume that in an interaction, user i changes its opinion U_i, and then its opinion will be updated to $U_{i,\text{new}}$ following (6). Considering the observed rating R_{ij} and item j's latent vector, the payoffs for the strategies are defined as follows.

1) If user i changes its opinion, the payoff that user i obtains is $|R_{ij} - U_i V_j^T| - |R_{ij} - U_{i,\text{new}} V_j^T|$. Users should adapt their opinions to reduce the errors between the observed ratings and estimated ratings. Therefore, if the error for the estimated rating decreases after the opinion update, user i will obtain a positive payoff and it is willing to change its opinion. We suppose that the payoff for the strategy, i.e., the user changing its opinion, depends on the difference between the original error and that after this strategy is adopted.
2) If user i retains its opinion, the payoff for i is $\beta \cdot |U_i V_j^T - U_f V_j^T|$ where β $(\beta > 0)$ is used to control the contribution of this strategy which represents individual stubbornness. Users generally prefer to persuade their opponents rather than changing their own opinions, since changing an opinion may incur a cost. The payoff correlates with the difference between user i's and f's estimated ratings on item j, and a large difference between ratings leads to a large cost when changing opinions. If users decide to maintain their opinions, they will receive a positive payoff.
3) If user f changes its opinion, user i receives the payoff $\beta \cdot |U_i V_j^T - U_f V_j^T|$. If a user succeeds in persuading its opponent to change an opinion, it will obtain a positive payoff.

When user i and f interact in relation to topic j, the payoffs for user i are shown in Table I.

The Nash equilibrium point of the aforementioned game is related to latent item vector V_j and rating R_{ij}. We can infer the Nash equilibrium point from Table I as follows.

1) When $|R_{ij} - U_i V_j^T| - |R_{ij} - U_{i,\text{new}} V_j^T| - \beta \cdot |U_i V_j^T - U_f V_j^T| > 0$, the Nash equilibrium strategy for user i is changing its opinion.
2) When $|R_{ij} - U_i V_j^T| - |R_{ij} - U_{i,\text{new}} V_j^T| - \beta \cdot |U_i V_j^T - U_f V_j^T| \leq 0$, the Nash equilibrium strategy for user i is maintaining its opinion.

The analogous Nash equilibrium strategy can be found for user f. For the aforementioned evolutionary game model, the condition for opinion updates varies with time. The model

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

TABLE I
PAYOFFS FOR USER i

	User f changes its opinion	User f maintains its opinion
User i changes its opinion	$\left\|R_{ij}-U_iV_j^T\right\|-\left\|R_{ij}-U_{i,new}V_j^T\right\|+\beta\cdot\left\|U_iV_j^T-U_fV_j^T\right\|$	$\left\|R_{ij}-U_iV_j^T\right\|-\left\|R_{ij}-U_{i,new}V_j^T\right\|$
User i maintains its opinion	$2\beta\cdot\left\|U_iV_j^T-U_fV_j^T\right\|$	$\beta\cdot\left\|U_iV_j^T-U_fV_j^T\right\|$

does not have a fixed tolerance threshold ε. Inserting (6) into the Nash equilibrium condition, we have

$$\left|R_{ij}-U_iV_j^T\right|-\left|R_{ij}-U_iV_j^T-\mu\cdot\left(U_f-U_i\right)V_j^T\right| - \beta\cdot\left|U_iV_j^T-U_fV_j^T\right|>0. \tag{8}$$

From (8), if user i changes its opinion in an interaction, it holds true that $U_iV_j^T<R_{ij}$ & $U_fV_j^T>U_iV_j^T$, or $U_iV_j^T>R_{ij}$ & $U_fV_j^T<U_iV_j^T$. In addition, if we do not consider the impact of observed ratings, so the condition for opinion updates in (8) reduces to $|U_iV_j^T-U_fV_j^T|<\varepsilon$. As in [31], when the system in homogeneous networks converges, the initial average value of $|U_iV_j^T-U_fV_j^T|$ over all users should be below ε. U_i and V_j are d-dimensional vectors, and each dimension in the beginning is randomly distributed from [0, 1]. The expectation of initial $U_iV_j^T$ is $d/4$. It can be inferred that the expectation of initial $|U_iV_j^T-U_fV_j^T|$ is $d/18$. A large number of latent factors d leads to a large divergence of opinions and prevents the system from converging.

In each iteration of SGD during the training process, we implement opinion interactions of users following the evolutionary game model. In multiagent opinion dynamics, a Monte-Carlo time step contains m times of opinion interactions for a population of m users, and hence, we introduce m such interactions into an iteration of SGD. In an opinion interaction, two users are selected at random, and they employ different strategies according to their payoffs in relation to a randomly selected item. In an iteration of SGD, opinion interactions are asynchronously implemented m times. When the objective function reaches convergence, for a majority of user-item pairs, the product of latent vectors $U_iV_j^T$ approaches R_{ij}, so that in the stable state, $|R_{ij}-U_i{V_j}^T|$ for these user-item pairs is small. Therefore, for most of users, the payoff of changing their opinions $|R_{ij}-U_i{V_j}^T|-|R_{ij}-U_{i,\text{new}}{V_j}^T|$ is often smaller than that of maintaining their opinions $\beta\cdot|U_i{V_j}^T-U_f{V_j}^T|$. In the stable state of the network, the strategy of maintaining one's opinion dominates in opinion interactions.

B. Matrix Factorization With User Influence

User influence represents the role of a user in a network. This influence is often regarded as a contribution in the process of information diffusion. With large influence, a user may diffuse its information to a greater number of other users, and information recommended by this user is readily accepted by neighbors. Therefore, it has a large impact on others' preferences. Some features of the underlying topology can be used to measure influence, such as the degree centrality, betweenness centrality, k-core index, average clustering coefficient, etc [32]. Here, for the sake of simplicity, we choose degree centrality as the indicator of user influence.

In the real world, users generally consult their friends before making decisions on items, since they tend to trust the preferences of their friends. From trust relations found on movie or product review websites, a trust network can be obtained and then user influence can be calculated. We define the number of users that trust user i as F_i^-. F_i^- is quite heterogeneous for different users, and therefore, we should renormalize it. User i's influence is given by

$$\varphi_i=\frac{\log\left(F_i^-/\alpha_1+\alpha_2\right)}{\log\left(\max\limits_f F_f^-/\alpha_1+\alpha_2\right)}. \tag{9}$$

The offset α_2 in the logarithmic function increases user influence to be greater than 0, as some users do not have any trust relations. The value of α_2 should be set in the interval (2, 10), since too large α_2 reduces the effect of user influence. The denominator in (9) renormalizes the influence and limits the value of φ_i in the interval (0, 1]. The parameter α_1 is used to control the decay of user influence. If a user has larger influence, its preference makes a greater contribution in the sum-of-squared-errors objective function. The objective function of (4) is rectified as

$$\min_{U,V}\frac{1}{2}\sum_{i=1}^{m}\varphi_i\sum_{j=1}^{n}I_{ij}\left(R_{ij}-U_iV_j^T\right)^2+\frac{\lambda}{2}\|\boldsymbol{U}\|^2+\frac{\lambda}{2}\|\boldsymbol{V}\|^2. \tag{10}$$

Then, the derivatives of the corresponding latent vectors in SGD are calculated as follows:

$$\frac{\partial\mathcal{L}}{\partial U_i}=-\sum_j\varphi_i\left(R_{ij}-U_iV_j^T\right)V_j+\lambda U_i \tag{11}$$

$$\frac{\partial\mathcal{L}}{\partial V_j}=-\sum_i\varphi_i\left(R_{ij}-U_iV_j^T\right)U_i+\lambda V_j. \tag{12}$$

C. Model Learning

Here, we present our recommendation method with opinion interactions and user influence. The whole training algorithm is shown in Algorithm 1. The method is based on the framework of MF, and opinion dynamics are introduced to the process of SGD. Our method comprises two steps in an iteration of SGD.

1) For each observed rating, SGD is used to update latent user vector U_i and item vector V_j. User influence given in (9) from the trust network is included.
2) Opinion interactions are implemented in each iteration of SGD. In each interaction, two users i and f are selected at random. User i randomly selects an item j

Algorithm 1: Proposed Recommendation Method With Evolutionary Opinion Dynamics

Input: List of tuples $\Omega = (users, items, ratings)$, list of tuples $SNS = (users, trusted\ users)$; the number of latent factors d, the learning rate γ, regularization parameter λ, user influence parameter α_1, α_2, trust parameter μ, and payoff parameter β

Output: Latent user matrix $\boldsymbol{U}$ and latent item matrix $\boldsymbol{V}$

Task 1: Generating user influence
for $i \leftarrow 1, 2, \ldots, m$ **do**
  Calculate φ_i according to Eq. (9)
end

Task 2: Learning user matrix $\boldsymbol{U}$ and item matrix $\boldsymbol{V}$
Initialize $\boldsymbol{U}$ and $\boldsymbol{V}$ randomly
while *not convergence* **do**
  (1) SGD training
  Calculate $\partial\mathcal{L}/\partial\boldsymbol{U}$ according to Eq. (11)
  Calculate $\partial\mathcal{L}/\partial\boldsymbol{V}$ according to Eq. (12)
  Update $\boldsymbol{U} \leftarrow \boldsymbol{U} - \gamma \cdot \partial\mathcal{L}/\partial\boldsymbol{U}$
  Update $\boldsymbol{V} \leftarrow \boldsymbol{V} - \gamma \cdot \partial\mathcal{L}/\partial\boldsymbol{V}$
  (2) Opinion interactions
  for $i \leftarrow 1, 2, \ldots, m$ **do**
    Select two users i and f at random
    Select an item j randomly that user i has rated
    if *Eq. (8) holds* **then**
      $U_i \leftarrow U_i + \mu \cdot \left(U_f - U_i\right)$
    end
  end
end

that i has rated in the training data. Then, user i interacts with f for item j according to the Nash equilibrium of the game shown in Table I. If the condition of (8) holds true, user i changes its opinion following (6). In each iteration, m interactions are implemented. Here, we do not consider the trust network, since a user in online social networks can exchange its opinion with any other users even if it does not have any trust relation with them. Users' comments and ratings on an item are accessible to all other users, so that they can have a discussion on the item.

D. Complexity Analysis

We analyze the computational complexity for the proposed method. We define the number of observed ratings in the training data as $|R|$, and the number of iterations as N. The computational complexity of SGD in MF is $O(d \cdot N \cdot |R|)$, where d is the number of latent factors. As previously mentioned, m is the number of users. Thus, the average number of observed ratings for each user is $|R|/m$. To calculate user similarities, the computational complexity $O(m^2 \cdot |R|/m) = O(m \cdot |R|)$ is required. The computational complexity for SoReg is $O(d \cdot N \cdot (|R| + 2m \cdot |f|) + m \cdot |R|)$, where $|f|$ is the average number of trusted friends for each user. Since m is often much larger than $d \cdot N$, the computation of user similarities in SoReg accounts for a greater proportion than that of SGD for MF. For TrustSVD [6], the computational complexity is $O(d \cdot N \cdot (|R| + |T|) \cdot \max(|f|, k))$, where $|T|$ is the number of observed relations and k is the average number of ratings received by an item.

We simply write our recommendation method with evolutionary opinion dynamics and user influence as REOD in the following. For REOD, the computation is mainly caused by SGD training and opinion dynamics. In the process of opinion dynamics during an iteration, m opinion interactions are implemented, each of which only contains one opinion update. An opinion update takes the computational complexity $O(d)$. Therefore, opinion dynamics result in the computational complexity $O(d \cdot N \cdot m)$. Overall, the computational complexity for REOD is $O(d \cdot N \cdot (|R| + m))$. Since $|R|$ is much larger than m, the complexity of our method approximates MF which costs $O(d \cdot N \cdot |R|)$, and REOD involves much less computation than state-of-the-art models.

TABLE II
STATISTICS OF DATASETS

	Ciao	Epinions
Users	7267	7411
Items	11,211	8728
Ratings	149,147	276,116
Social relations	110,755	52,982

V. EXPERIMENTS

In this section, we address the following questions: 1) does the proposed method with evolutionary opinion dynamics and user influence improve the accuracy of recommendation? 2) what is the contribution of opinion interactions and user influence for recommendation? and 3) how do the intrinsic parameters of opinion interactions and user influence affect the recommendation results? First, we use two real-world datasets to evaluate our method, and compare the recommendation results of our approach with other state-of-the-art recommendation models to answer the first question. Then, we investigate the effects of the components in our method to answer the second question. Last, we vary the parameters of opinion interactions to explore their effects to answer the third question.

A. Datasets and Metrics

To evaluate the proposed recommendation method, we collected two datasets, which were taken from the popular social networking websites Ciao[1] and Epinions.[2] Statistics on these datasets are presented in Table II. Users of these social networking services can rate items, browse/write reviews, discuss with others, and add trusted friends. Therefore, we can obtain rating and social relation data from these websites.

Ciao and Epinions are well-known product review websites, where users can rate a product using one of five discrete ratings

[1] http://www.ciao.co.uk/
[2] http://www.epinions.com/

TABLE III
RESULTS OF RECOMMENDATION ON MAE AND RMSE IN CIAO

		PMF	LLORMA	SocialMF	SoRec	RSTE	TrustMF	SoReg	TrustSVD	REOD
60%	MAE	0.9767	0.8592	0.7762	0.7908	0.7971	0.7883	0.7626	0.7515	**0.7359**
	RMSE	1.2401	1.2339	1.0036	1.1337	1.1097	1.0858	1.0116	**0.9844**	0.9858
70%	MAE	0.9078	0.8055	0.7702	0.7855	0.7897	0.7838	0.7539	0.7434	**0.7294**
	RMSE	1.1572	1.1251	0.9988	1.1179	1.0989	1.0685	1.0002	0.9768	**0.9766**
80%	MAE	0.8696	0.7795	0.7640	0.7809	0.7786	0.7792	0.7472	0.7376	**0.7262**
	RMSE	1.1130	1.0654	0.9919	1.1052	1.0859	1.0560	0.9899	0.9704	**0.9701**

from 1 to 5. Ratings imply the sentiment of users toward a given item. If a user is not satisfied with a product, it will give the product a rating of 1; if a user appreciates a product, it will give the product a rating of 5. Each user maintains a "trust" list which includes the user's social relations. For Ciao, we collected 7267 users, 11 211 items, and 149 147 ratings. The density of the user-item rating matrix is 0.183%. For Epinions, we collected 7411 users, 8728 items, and 276 116 ratings, and the density of the user-item rating matrix is about 0.427%.

In both datasets, F_i^- follows a power-law distribution. The power exponent in the Ciao dataset is -1.076 ± 0.023, and that in Epinions is -0.991 ± 0.021. The average and maximal values of F_i^- in Ciao are 15.2408 and 796, while those in Epinions are 7.1491 and 336, respectively.

For each dataset, we choose $x\%$ as the training set to learn the parameters and use the remaining $1 - x\%$ as the test set. We set x at 60, 70, and 80, respectively, and obtain the results. The experiments are conducted five times independently, and we give the average performance.

We use two metrics to evaluate the performance, i.e., mean absolute error (MAE) and root mean square error (RMSE). MAE is defined as

$$\text{MAE} = \frac{1}{|R_{\text{test}}|} \sum\nolimits_{R_{ij} \in R_{\text{test}}} \left| R_{ij} - U_i V_j^T \right| \tag{13}$$

where R_{test} refers to the test set, and $|R_{\text{test}}|$ refers to the number of ratings in R_{test}. RMSE is defined as

$$\text{RMSE} = \sqrt{\frac{1}{|R_{\text{test}}|} \sum\nolimits_{R_{ij} \in R_{\text{test}}} \left(R_{ij} - U_i V_j^T \right)^2}. \tag{14}$$

It has been proved that a smaller MAE or RMSE value means a better performance.

B. Baselines

In this section, to demonstrate the effectiveness of the proposed method, we compare it with the following representative recommendation models.

PMF [33]: This method only utilizes the user-item rating matrix, and performs probabilistic MF to make recommendations.

LLORMA [34]: This method relaxes the low-rank assumption, and approximates the observed matrix as a weighted sum of local low-rank matrices.

SocialMF [35]: This method introduces the mechanism of trust propagation into the model.

SoRec [36]: This method is based on probabilistic MF, and performs a co-factorization on the user-item rating matrix and user-user social relation matrix.

RSTE [23]: This method makes social recommendation by using social trust ensemble and naturally fusing the preferences of users and their trusted friends together.

TrustMF [37]: This method performs MF to map users into low-dimensional latent spaces in terms of their trust relations.

SoReg [10]: This method incorporates social regularization into MF, and social regularization represents the social constraints on recommender systems.

TrustSVD [6]: This method incorporates the explicit and implicit influence of rated items and trusted users on the prediction of items.

To focus on model evaluation and a fair comparison, for all methods, we set the same number of latent factors $d = 20$. For different parameters in baseline models, we employ cross-validation to determine the optimal values. For PMF, the parameters are $\lambda = 0.1$, $\gamma = 0.01$ in Ciao, and $\lambda = 0.1$, $\gamma = 0.005$ in Epinions. For SoRec, we set $\gamma = 0.005$, $\lambda_c = 1$, $\lambda = 0.001$ in Ciao, and $\gamma = 0.005$, $\lambda_c = 1$, $\lambda = 0.005$ in Epinions. For TrustMF, the parameters are $\gamma = 0.3$, $\lambda = 0.005$, $\lambda_t = 1$ in Ciao, and $\gamma = 0.2$, $\lambda = 0.005$, $\lambda_t = 5$ in Epinions. For TrustSVD, we set $\gamma = 0.001$, $\lambda = 0.5$, $\lambda_t = 1$ in Ciao, and $\gamma = 0.001$, $\lambda = 0.9$, $\lambda_t = 0.5$ in Epinions. For the remaining methods, we set the regularization parameter $\lambda = 0.001$ in both datasets. For LLORMA, the learning rate is $\gamma = 0.01$ in both datasets. For SocialMF, we set the parameters $\lambda_t = 5$, $\gamma = 0.1$ in Ciao, and $\lambda_t = 5$, $\gamma = 0.05$ in Epinions. For RSTE, we set $\gamma = 0.04$, and $\alpha = 0.9$ in both datasets. For SoReg, the parameters are $\gamma = 0.0005$ and $\beta = 5$. For REOD, we set the payoff parameter $\beta = 0.05$, the trust parameter $\mu = 0.12$, the learning rate $\gamma = 0.001$, and the influence offset $\alpha_2 = 6$ in both datasets. The influence decay is $\alpha_1 = 30$ in Ciao, and it is $\alpha_1 = 80$ in Epinions. Degrees of the trust network have a heavy tailed distribution, and many users have a tiny F_i^- in (9). Therefore, the ratings of these users make little contribution to the objective function, so (10) may cause over fitting. To alleviate this problem, we multiply φ_i by a positive random number with normal distribution for each iteration. Some baselines are implemented by LibRec[3] [38].

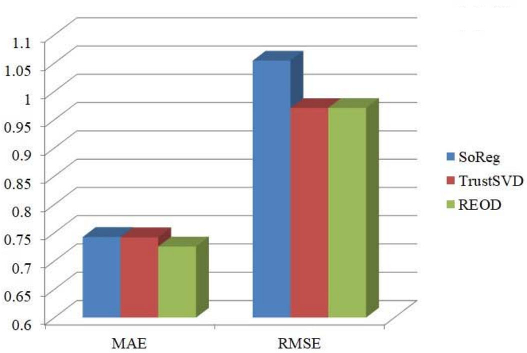
C. Performance Comparisons

Tables III and IV compare the results of the different methods for all users. More training data leads to higher

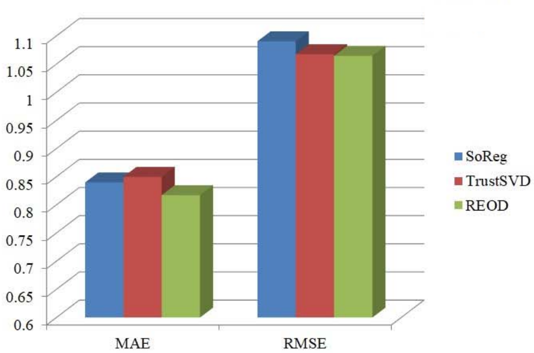
[3]https://www.librec.net/

TABLE IV
RESULTS OF RECOMMENDATION ON MAE AND RMSE IN EPINIONS

		PMF	LLORMA	SocialMF	SoRec	RSTE	TrustMF	SoReg	TrustSVD	REOD
60%	MAE	0.8969	0.8271	0.8552	0.8574	0.8689	0.8788	0.8269	0.8096	**0.7974**
	RMSE	1.1334	1.1277	1.1231	1.1066	1.1705	1.1616	1.0721	1.0425	**1.0396**
70%	MAE	0.8759	0.8119	0.8510	0.8530	0.8611	0.8747	0.8230	0.8041	**0.7938**
	RMSE	1.1129	1.0998	1.1115	1.0947	1.1558	1.1438	1.0674	1.0379	**1.0331**
80%	MAE	0.8624	0.8041	0.8487	0.8493	0.8564	0.8729	0.8211	0.8004	**0.7910**
	RMSE	1.1004	1.0862	1.1004	1.0861	1.1475	1.1313	1.0647	1.0360	**1.0308**



(a)



(b)

Fig. 1. Performance comparison of SoReg, TrustSVD, and REOD for cold-start users. (a) Ciao. (b) Epinions.

recommendation accuracy, especially in Ciao which has sparser rating data. PMF performs worse than all social recommendation models except in the case where TrustMF performs the worst when 80% of data in Epinions is used for training. The reason for this is that the dataset of Epinions has much sparser user relations. Directly factorizing the matrix of the sparse trust network may harm the prediction accuracy on unknown ratings for recommender systems. LLORMA has low accuracy in Ciao, but it performs well in Epinions and even outperforms some social recommendation methods. LLORMA obtains low-rank matrices that are limited to a local region of the observed matrix, so that it achieves a high performance in denser rating data. In Epinions, SocialMF and SoRec almost perform the same, but when more user relations are available, SocialMF has higher accuracy in the Ciao dataset. RSTE uses social trust ensemble and requires more relation data, so that it performs worse than SoRec in Epinions. SoReg has a smaller MAE and RMSE than SocialMF, SoRec, RSTE, and TrustMF, since SoReg uses better social regularization terms. TrustSVD incorporates the implicit influence of user trust and item ratings, so recommendation accuracy is improved and it performs the best of the state-of-the-art methods. Clearly, our method REOD outperforms the other models. When 60% of the training data of Ciao is used, REOD decreases MAE as high as 3.501% in contrast to SoReg, and 2.076% in contrast to TrustSVD; in Epinions, the corresponding improvement is 3.568% in contrast to SoReg, and 1.507% in contrast to TrustSVD. Although in Ciao, the RMSE of REOD approaches that of TrustSVD, REOD can obtain a better performance with sparse social connections. Therefore, we draw the conclusion that REOD improves the accuracy of recommendation.

Recommender systems often suffer from cold start problems, degrading the recommendation performance. We address the accuracy of these models for cold start users who have only rated a few items (equal to or less than five ratings). Fig. 1 shows the performance of SoReg, TrustSVD, and our approach for cold start users. The parameters are the same as above. We select cold start users and evaluate the MAE and RMSE on these users. Here, we use 80% of the data as training data, and the results are similar for different proportions of training data. Fig. 1 shows that REOD still outperforms the other methods for cold start users, although the RMSE of TrustSVD approximates our method. In both datasets, SoReg has a similar MAE with TrustSVD, but has a larger RMSE than the other two methods. The results demonstrate that incorporating evolutionary opinion dynamics can help recommender systems cope with cold start situations.

Now, we focus on the second issue of examining the effects of user influence and opinion interactions. It has been proved above that recommendation with both effects outperforms the representative recommendation models. We investigate which aspect plays a more significant role in social recommendation. We eliminate the effect of opinion interactions or user influence separately by defining the following algorithmic variants.

1) *REOD\UI:* Eliminating the effect of user influence. Evolutionary opinion dynamics are considered. The objective function of (10) reduces to that of traditional MF in (4).
2) *REOD\OP:* Eliminating the effect of opinion interactions. User influence is calculated from the trust

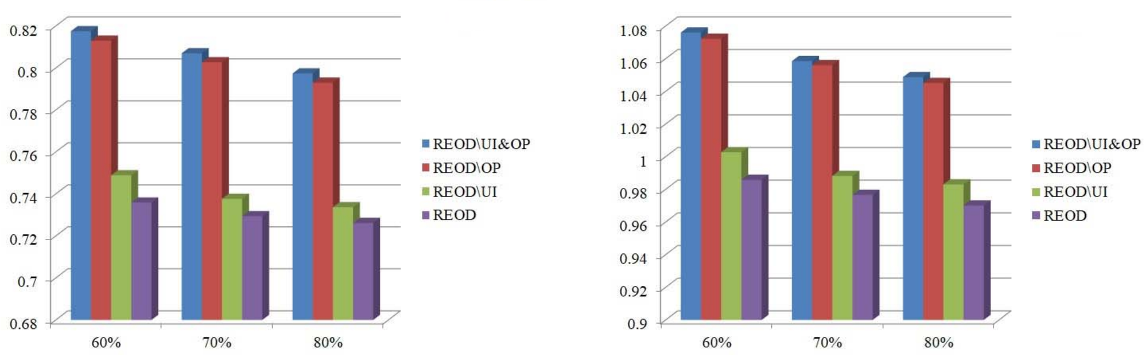


Fig. 2. Impact of user influence and opinion interactions on recommendation in the Ciao dataset. (a) MAE. (b) RMSE.

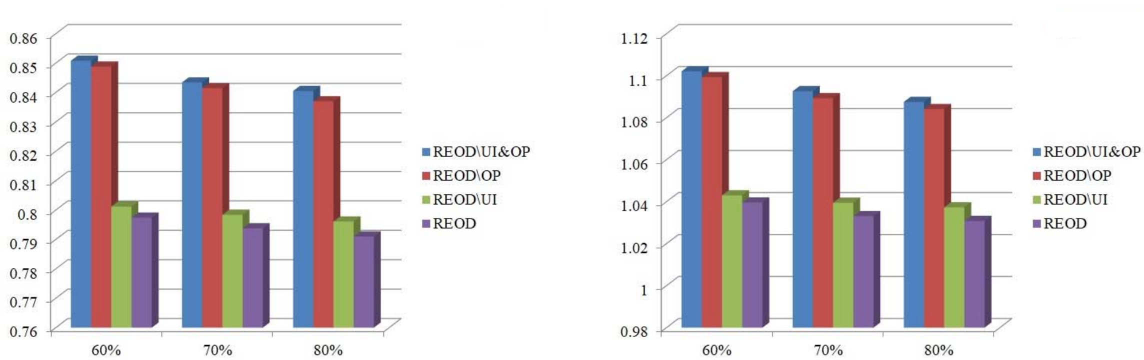


Fig. 3. Impact of user influence and opinion interactions on recommendation in the Epinions dataset. (a) MAE. (b) RMSE.

network. Opinion interactions are removed from each iteration of SGD training.

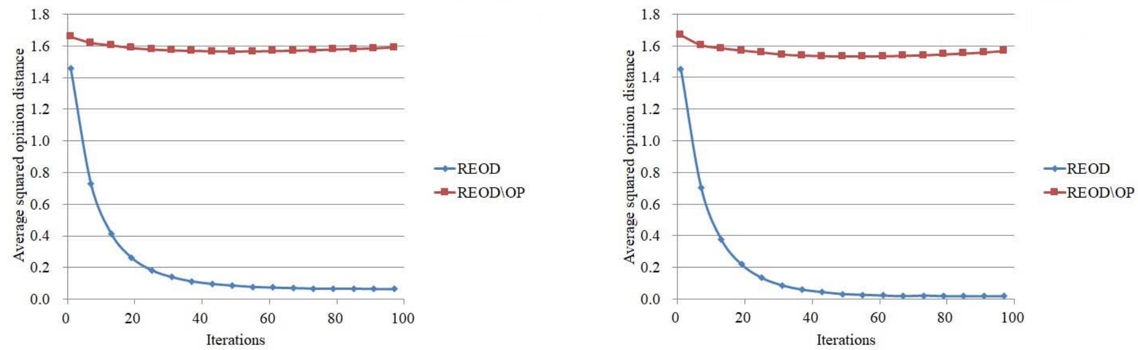
3) *REOD\UI&OP:* Eliminating both the effects of opinion interactions and user influence.

We also use cross-validation to determine the parameters of these variants. In the Ciao dataset, for REOD\UI&OP and REOD\UI, we set $\gamma = 0.0005$ and $\lambda = 0.001$. For REOD\UI, the parameters in opinion interactions are $\mu = 0.12$ and $\beta = 0.05$ in both datasets. For REOD\OP, the parameters are $\gamma = 0.001$ and $\lambda = 0.001$, $\alpha_1 = 40$ and $\alpha_2 = 8$. We find the optimal parameters for these three algorithmic variants in Epinions are the same as those in Ciao. Figs. 2 and 3 show the accuracy of these variants in Ciao and Epinions, respectively. In general, each component in our method contributes to better recommendation, and eliminating the effect of opinion interactions or user influence degrades the performance. In both datasets, opinion interactions play a far more significant role in the prediction of unknown ratings, compared with user influence. Therefore, when 60% of the data is used for training, REOD\OP has a 7.884% relative performance reduction for MAE in Ciao data, and 5.608% in Epinions data. The procedure of opinion interactions in each iteration of SGD does not need the trust network, therefore, it will not suffer from the sparsity problem of trust relations. User influence slightly reduces MAE and RMSE in both datasets whether the effect of opinion interactions is included or not. Furthermore, the improvement of the performance under the action of user influence is more obvious in Ciao data than in Epinions data, as a result of denser user relations in Ciao, especially when less training data are applied.

We are concerned about the evolution of user opinions during the SGD training of our method. We use the average squared distance of individual opinions to reflect the divergence of opinions. The average squared distance is defined as

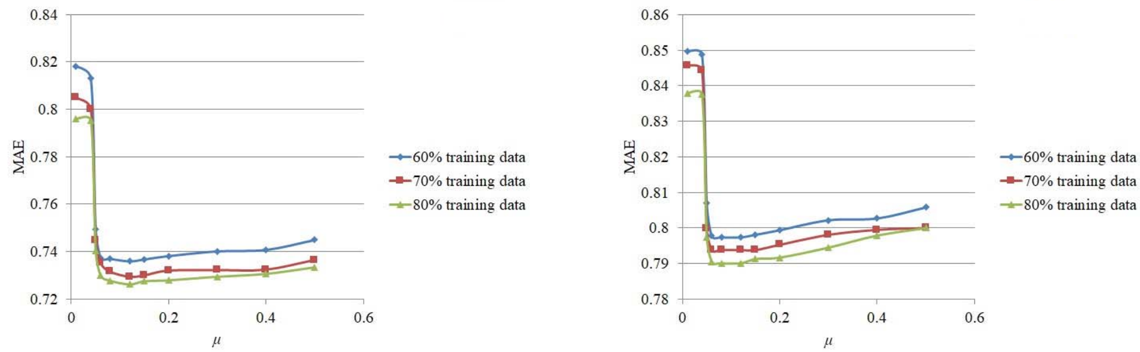
$$D(t) = \frac{\sum_i \|U_i(t) - E(U(t))\|^2}{m} \tag{15}$$

where $E(\cdot)$ means the expectation operation. A larger squared distance means more disordering exists in user opinions. Fig. 4 shows the average squared opinion distance versus the iteration number with or without opinion interactions, when 70% of the data is used for training. We also find that with different training data, the evolution of opinions is analogous. For REOD, the average squared distance drops to a very low value and gradually becomes stable in about 50 iterations. Consensus is almost achieved, especially in the Epinions data, implying very small divergence among user opinions. Due to the existence of opinion interactions, users tend to adapt their opinions so that they are close to each other. This phenomenon is in accordance with real situations in social networks [39], since users tend to persuade others to trust their opinions during opinion interactions. As a result, opinion interactions clearly improve the recommendation accuracy. However, for REOD\OP, the



(a) (b)

Fig. 4. Average squared distance of user opinions versus iteration number with 70% training data. (a) Ciao. (b) Epinions.



(a) (b)

Fig. 5. Performance variations of REOD versus the trust parameter μ. (a) Ciao. (b) Epinions.

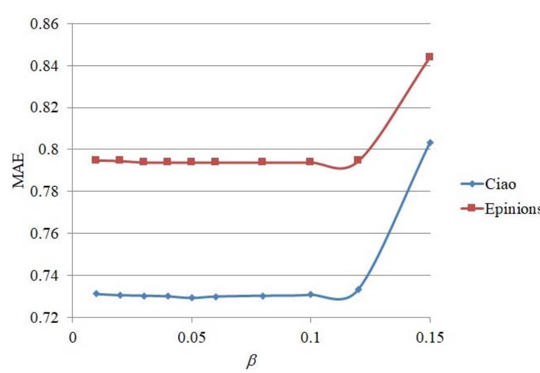


Fig. 6. Performance variations of REOD versus the payoff parameter β with 70% training data.

average squared distance only marginally decreases, and user opinions are quite divergent.

Now, we address the third issue: the effects of parameters for opinion interactions and user influence on the performance. The parameter μ determines the rate of opinion exchanges. We change the value of μ, and investigate the corresponding recommendation accuracy. Since users generally update their opinions so that they are close to their neighbors' opinions, the value of μ does not exceed 0.5. Fig. 5 shows the impact of μ with different training data. The variations of RMSE are similar to those of MAE, and therefore, we do not depict RMSE here. We can clearly observe a transition at $\mu = 0.05$ below which the method has larger MAE. MAE starts a precipitous decline around $\mu = 0.05$, and reaches a plateau in the interval $[0.05, 0.2]$. We investigate opinion evolution with different μ. For $\mu < 0.05$, the final average squared opinion distance is larger than 1.5, while that for $\mu > 0.05$ is below 0.08. Therefore, if $\mu < 0.05$, user opinions have few changes during opinion interactions, and opinion interactions do not take effect in recommendation. Then, our method reduces to REOD\OP. If $\mu > 0.2$, MAE increases slowly with μ. For large μ, the variation of an estimated rating $|\mu\cdot(U_f - U_i)V_j^T|$ may be larger than $2\cdot|R_{ij} - U_iV_j^T|$, so that we have $|R_{ij} - U_{i,\text{new}}V_j^T| > |R_{ij} - U_iV_j^T|$ and errors of estimated ratings may increase. Generally, our method achieves lower MAE in the interval $[0.05,\ 0.2]$ of μ in both datasets, irrespectively of the proportion of training data. Thus, we can typically set $\mu = 0.12$ without loss of generality. Since the impact of μ does not depend on the datasets, the complexity of our method can be reduced.

The parameter β controls the equilibrium between the strategy of changing an opinion or maintaining an opinion in the evolutionary game model. Here, we only consider one case with 70% training data. In other cases with a different amount of training data, the performance variations are similar. Fig. 6 shows the impact of β on MAE. When $0 < \beta < 0.1$, MAE remains relatively stable in both datasets as β varies. Around $\beta = 0.05$, MAE reaches the lowest value. When $\beta > 0.1$,

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

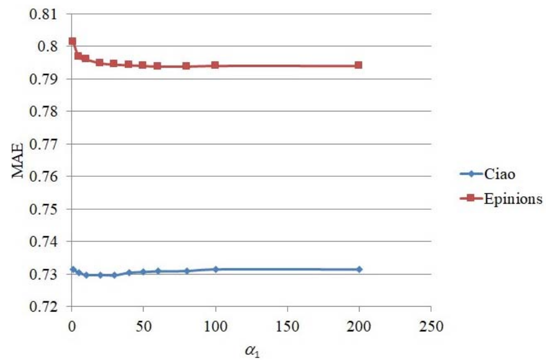


Fig. 7. Performance variations of REOD versus the influence decay parameter α_1 with 70% training data.

TABLE V
RESULTS OF RECOMMENDATION ACCURACY WITH DIFFERENT USER INFLUENCE IN CIAO

	Degree centrality	Betweenness centrality	Clustering coefficient	k-core index
MAE	0.7262	0.7325	0.7261	0.7259
RMSE	0.9701	0.9821	0.9697	0.9698

MAE increases rapidly, demonstrating that the error for the estimated rating should preferentially be considered in the evolutionary game of opinion interactions. From (8), when β approaches μ, increasing β makes users choose the strategy of maintaining their opinions, and restrains the effect of opinion interactions in recommendation. In addition, the impact of the payoff parameter β is also independent of the datasets, reducing the complexity of our method. In different datasets, we can empirically set the value of β. Fig. 7 shows the impact of the influence decay parameter α_1. It is obvious that MAE in both datasets does not have a close correlation with α_1. Although the best performance in different datasets varies with α_1, the change of MAE is small and we obtain relatively low MAE in the interval (20, 90) of α_1. Most of degrees F_i^- in both datasets are less than 50. When $\alpha_1 > 20$, the variation of user influence versus α_1 is very small. The aforementioned properties of parameters are useful from a practical point of view because they make it easier to set parameters in using our method.

We use other topological descriptors to measure user influence, such as betweenness centrality, clustering coefficients and k-core index, and incorporate the influence into recommender systems. We use 80% data as training data, and evaluate the recommendation performance with different forms of user influence. All parameters are determined by cross-validation. Results of recommendation accuracy with different user influence are shown in Tables V and VI. Although topological descriptors have different capabilities of measuring user influence, their effects on the recommendation performance are similar in both datasets. MAE and RMSE of degree centrality, clustering coefficients and k-core index are approximately the same, but the descriptor of betweenness centrality has a lower performance. The reason is that betweenness in these networks is more heterogeneous than other descriptors, so that users with large betweenness play an excessively important role in the sum-of-squared-errors objective function. Ratings of the users that have very small betweenness have limited contribution to the objective function, but these users may have many ratings and cannot be ignored in recommendation. In addition, we also measure user influence by tie strength, and incorporate user influence and social regularization into recommender systems, but the accuracy cannot be improved.

TABLE VI
RESULTS OF RECOMMENDATION ACCURACY WITH DIFFERENT USER INFLUENCE IN EPINIONS

	Degree centrality	Betweenness centrality	Clustering coefficient	k-core index
MAE	0.7910	0.8020	0.7907	0.7904
RMSE	1.0308	1.0433	1.0305	1.0302

VI. CONCLUSION

When users in online social networks interact with their friends, their opinions are influenced by others. User interactions can be applied in recommender systems to improve performance. Social recommendation models utilize social relations, and introduce neighbors' impact into the MF framework. In this paper, we investigated the impact pattern of other users on latent preferences, and studied its effect on recommendation. We proposed an evolutionary game model to characterize opinion interactions. We defined two interaction behaviors, i.e., maintaining one's opinion or changing one's opinion, and determined the payoff for each behavior according to the rating on a given item. Users choose the behavior which maximizes their payoffs. Then, we measured user influence according to node ingoing degrees in the social network. We further used user influence to weight the objective function of MF, and conducted dynamic opinion interactions during each iteration of training. Experiment results on real-world datasets demonstrated that our method performs better than state-of-the-art recommendation models for all users and cold start users. Meanwhile, our method has much less computational complexity than the other models. Opinion dynamics drive user opinions to converge and reduce the divergence, coinciding with the real situation in online interactions. Moreover, our method does not have a significant dependence of opinion interaction or user influence parameters.

We considered random opinion interactions and user influence in recommendation, but information diffusion and the implicit influence of the network were not involved. In online social networks, users diffuse information about an item, and meanwhile, they exchange their opinions. In future work, we will incorporate diffusion dynamics into recommender systems, and investigate the concurrent process of information diffusion and opinion interactions. Furthermore, we will extent the method by incorporating the implicit influence of ratings and social relations to make a more accurate recommendation.

REFERENCES

[1] J. Xuan, X. Luo, G. Zhang, J. Lu, and Z. Xu, "Uncertainty analysis for the keyword system of Web events," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 6, pp. 829–842, Jun. 2016.

[2] J. Lu, D. Wu, M. Mao, W. Wang, and G. Zhang, "Recommender system application developments: A survey," *Decis. Support Syst.*, vol. 74, pp. 12–32, Jun. 2015.

[3] T. Bansal, M. Das, and C. Bhattacharyya, "Content driven user profiling for comment-worthy recommendations of news and blog articles," in *Proc. 9th ACM Conf. Recommender Syst.*, 2015, pp. 195–202.

[4] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[5] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 143–177, 2004.

[6] G. Guo, J. Zhang, and N. Yorke-Smith, "A novel recommendation model regularized with user trust and item ratings," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 7, pp. 1607–1620, Jul. 2016.

[7] X. Wang, Y. Liu, G. Zhang, F. Xiong, and J. Lu, "Diffusion-based recommendation with trust relations on tripartite graphs," *J. Stat. Mech. Theory Exp.*, vol. 2017, no. 8, p. 083405, 2017.

[8] M. Eirinaki, M. D. Louta, and I. Varlamis, "A trust-aware system for personalized user recommendations in social networks," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 44, no. 4, pp. 409–421, Apr. 2014.

[9] X. Wang, Y. Liu, and F. Xiong, "Improved personalized recommendation based on a similarity network," *Physica A Stat. Mech. Appl.*, vol. 456, pp. 271–280, Aug. 2016.

[10] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. 4th ACM Int. Conf. Web Search Data Min.*, 2011, pp. 287–296.

[11] M. Mao, J. Lu, G. Zhang, and J. Zhang, "Multirelational social recommendations via multigraph ranking," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4049–4061, Dec. 2017.

[12] Z. Lu, Y. Wen, W. Zhang, Q. Zheng, and G. Cao, "Towards information diffusion in mobile social networks," *IEEE Trans. Mobile Comput.*, vol. 15, no. 5, pp. 1292–1304, May 2016.

[13] J. Ok, Y. Jin, J. Shin, and Y. Yi, "On maximizing diffusion speed over social networks with strategic users," *IEEE/ACM Trans. Netw.*, vol. 24, no. 6, pp. 3798–3811, Dec. 2016.

[14] Y. Hu, R. J. Song, and M. Chen, "Modeling for information diffusion in online social networks via hydrodynamics," *IEEE Access*, vol. 5, pp. 128–135, 2017.

[15] G. Guo, J. Zhang, F. Zhu, and X. Wang, "Factored similarity models with social trust for top-n item recommendation," *Knowl. Based Syst.*, vol. 122, pp. 17–25, Apr. 2017.

[16] C. Jiang, Y. Chen, and K. J. R. Liu, "Evolutionary dynamics of information diffusion over social networks," *IEEE Trans. Signal Process.*, vol. 62, no. 17, pp. 4573–4586, Sep. 2014.

[17] J. Castro, J. Lu, G. Zhang, Y. Dong, and L. Martínez, "Opinion dynamics-based group recommender systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, to be published, doi: 10.1109/TSMC.2017.2695158.

[18] C. Jiang, Y. Chen, and K. R. Liu, "Graphical evolutionary game for information diffusion over social networks," *IEEE J. Sel. Topics Signal Process.*, vol. 8, no. 4, pp. 524–536, Aug. 2014.

[19] G. Deffuant, D. Neau, F. Amblard, and G. Weisbuch, "Mixing beliefs among interacting agents," *Adv. Complex Syst.*, vol. 3, no. 01n04, pp. 87–98, 2000.

[20] A. C. R. Martins, "Continuous opinions and discrete actions in opinion dynamics problems," *Int. J. Mod. Phys. C*, vol. 19, no. 4, pp. 617–624, 2008.

[21] L. Cao and X. Li, "Mixed evolutionary strategies imply coexisting opinions on networks," *Phys. Rev. E, Stat. Phys. Plasmas Fluids Relat. Interdiscip. Top.*, vol. 77, no. 1, Art. no. 016108, 2008.

[22] Y. Koren, "Factor in the neighbors: Scalable and accurate collaborative filtering," *ACM Trans. Knowl. Disc. Data*, vol. 4, no. 1, p. 1, 2010.

[23] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proc. 32nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2009, pp. 203–210.

[24] J. Tang *et al.*, "Recommendation with social dimensions," in *Proc. 30th AAAI Conf. Artif. Intell.*, 2016, pp. 251–257.

[25] Q. Zhang *et al.*, "Inferring latent network from cascade data for dynamic social recommendation," in *Proc. 16th Int. Conf. Data Min.*, 2016, pp. 669–678.

[26] J. Tang, H. Gao, A. D. Sarma, Y. Bi, and H. Liu, "Trust evolution: Modeling and its applications," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 6, pp. 1724–1738, Jun. 2015.

[27] J. Tang, C. Aggarwal, and H. Liu, "Recommendations in signed social networks," in *Proc. 25th Int. Conf. World Wide Web*, 2016, pp. 31–40.

[28] J. Hu, B. Wang, Y. Liu, and D.-Y. Li, "Personalized tag recommendation using social influence," *J. Comput. Sci. Technol.*, vol. 27, no. 3, pp. 527–540, 2012.

[29] Q. Zhang *et al.*, "Global and local influence-based social recommendation," in *Proc. 25th ACM Int. Conf. Inf. Knowl. Manag.*, 2016, pp. 1917–1920.

[30] W. Luo, W.-P. Tay, and M. Leng, "Infection spreading and source identification: A hide and seek game," *IEEE Trans. Signal Process.*, vol. 64, no. 16, pp. 4228–4243, Aug. 2016.

[31] F. Xiong, Y. Liu, L. Wang, and X. Wang, "Analysis and application of opinion model with multiple topic interactions," *Chaos Interdiscip. J. Nonlin. Sci.*, vol. 27, no. 8, p. 083113, 2017.

[32] F. Xiong and Z.-Y. Li, "Effective methods of restraining diffusion in terms of epidemic dynamics," *Sci. Rep.*, vol. 7, no. 1, p. 6013, 2017.

[33] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1257–1264.

[34] J. Lee, S. Kim, G. Lebanon, Y. Singer, and S. Bengio, "Llorma: Local low-rank matrix approximation," *J. Mach. Learn. Res.*, vol. 17, no. 1, pp. 442–465, 2016.

[35] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. 4th ACM Conf. Recommender Syst.*, 2010, pp. 135–142.

[36] H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec: Social recommendation using probabilistic matrix factorization," in *Proc. 17th ACM Conf. Inf. Knowl. Manag.*, 2008, pp. 931–940.

[37] B. Yang, Y. Lei, J. Liu, and W. Li, "Social collaborative filtering by trust," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1633–1647, Aug. 2017.

[38] G. Guo, J. Zhang, Z. Sun, and N. Yorke-Smith, "Librec: A Java library for recommender systems," in *Proc. 23rd Conf. User Model. Adapt. Personalization Posters Demos Late Breaking Results Workshop*, 2015, pp. 1–4.

[39] L. Li, A. Scaglione, A. Swami, and Q. Zhao, "Consensus, polarization and clustering of opinions in social networks," *IEEE J. Sel. Areas Commun.*, vol. 31, no. 6, pp. 1072–1083, Jun. 2013.

Fei Xiong received the B.E. degree in communication engineering and the Ph.D. degree in communication and information systems from Beijing Jiaotong University, Beijing, China, in 2007 and 2013, respectively.

He is currently an Associate Professor with the School of Electronic and Information Engineering, Beijing Jiaotong University. From 2011 to 2012, he was a Visiting Scholar with Carnegie Mellon University, Pittsburgh, PA, USA. He has published over 60 papers in refereed journals and conference proceedings. He was a recipient of the National Natural Science Foundation of China and several other research grants. His current research interests include Web mining, complex networks, and complex systems.

Ximeng Wang received the B.E. degree in communication engineering from the Nanjing University of Posts and Telecommunications, Nanjing, China, in 2011 and the M.E. degree in software engineering from Beijing Jiaotong University, Beijing, China, in 2013. He is currently pursuing the dual Ph.D. degrees in communication and information systems with Beijing Jiaotong University and in software engineering with the University of Technology Sydney, Ultimo, NSW, Australia.

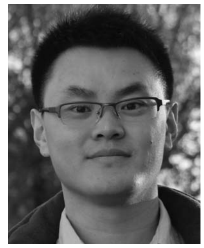
His current research interests include recommender systems, complex networks, and data mining.

Shirui Pan (M'16) received the Ph.D. degree in computer science from the University of Technology Sydney, Ultimo, NSW, Australia, in 2015.

He is a Lecturer with the Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney. He has published over 40 research papers in top-tier journals and conferences, including the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, *Pattern Recognition*, International Joint Conferences on Artificial Intelligence, IEEE International Conference on Data Engineering, IEEE International Conference on Data Mining, SIAM International Conference on Data Mining, International Conference on Information and Knowledge Management, and Pacific-Asia Conference on Knowledge Discovery and Data Mining. His current research interests include data mining and machine learning.

Hong Yang received the M.E. degree in communication and information systems from the Graduate University of Chinese Academy of Sciences, Beijing, China, in 2007. She is currently pursuing the Ph.D. degree in computer science with the Centre for Artificial Intelligence, Faculty of Engineering and Information Technology, University of Technology Sydney, Ultimo, NSW, Australia.

She was with MathWorks, Natick, MA, USA, as a Senior Training Engineer and a Leader of training teams from 2007 to 2016. Her current research interests include data mining and image processing.



Haishuai Wang received the Ph.D. degree in computer science from the Centre of Artificial Intelligence, University of Technology Sydney, Ultimo, NSW, Australia.

He was a Post-Doctoral Associate with the Department of Computer Scient and Engineering, Washington University in St. Louis, St. Louis, MO, USA. He is a Research Fellow with the Department of Biomedical Informatics, Harvard University, Cambridge, MA, USA. His current research interests include data mining and machine learning as well as its applications on bioinformatics and social networks.

Chengqi Zhang (SM'95) received the Ph.D. degree in computer science from the University of Queensland, Brisbane, QLD, Australia, in 1991 and the D.Sc. degree in computer science from Deakin University, Geelong, VIC, Australia, in 2002.

Since 2001, he has been a Professor of Information Technology with the University of Technology Sydney (UTS), Ultimo, NSW, Australia, where he is currently a Distinguished Professor and the Deputy Vice-Chancellor. He was the Director of the UTS Priority Investment Research Centre for Quantum Computation and Intelligent Systems. Since 2005, he has been the Chairman of the Australian Computer Society National Committee for Artificial Intelligence. He has published six monographs and edited 16 books, and has attracted 12 Australian Research Council grants. He has published over 300 research papers, including a number of papers in top-tier journals and conferences, such as *Artificial Intelligence*, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, IEEE TRANSACTIONS ON CYBERNETICS, IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, *ACM Transactions on Internet Technology*, *ACM Transactions on Knowledge Discovery From Data*, International Joint Conferences on Artificial Intelligence (IJCAI), IEEE International Conference on Data Mining (ICDM), SIAM International Conference on Data Mining, and International Conference on Information and Knowledge Management. His current research interest includes data mining and its applications.

Dr. Zhang has been serving as an Associate Editor for three international journals, including the IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING from 2005 to 2008. He served as the General Chair, the PC Chair, or the Organising Chair for five international conferences including *KDD*-2015, *ICDM*-2010, and *WI/IAT*-2008 and he is also the Local Arrangements Chair of *IJCAI*-2017 in Melbourne. He is a fellow of the Australian Computer Society.