Collaborative Filtering-Based Recommendation of Online Social Voting

Xiwang Yang, Chao Liang, Miao Zhao, *Member, IEEE*, Hongwei Wang, Hao Ding, Yong Liu, *Fellow, IEEE*, Yang Li, and Junlin Zhang

***Abstract*— Social voting is an emerging new feature in online social networks. It poses unique challenges and opportunities for recommendation. In this paper, we develop a set of matrix-factorization (MF) and nearest-neighbor (NN)-based recommender systems (RSs) that explore user social network and group affiliation information for social voting recommendation. Through experiments with real social voting traces, we demonstrate that social network and group affiliation information can significantly improve the accuracy of popularity-based voting recommendation, and social network information dominates group affiliation information in NN-based approaches. We also observe that social and group information is much more valuable to cold users than to heavy users. In our experiments, simple metapath-based NN models outperform computation-intensive MF models in hot-voting recommendation, while users' interests for nonhot votings can be better mined by MF models. We further propose a hybrid RS, bagging different single approaches to achieve the best top-*k* hit rate.**

***Index Terms*— Collaborative filtering, online social networks (OSNs), recommender systems (RSs), social voting.**

I. Introduction

ONLINE social networks (OSN), such as Facebook and Twitter, facilitate easy information sharing among friends. A user not only can share her updates, in forms of text, picture, and video, with her direct friends, but also can quickly disseminate those updates to a much larger audience of indirect friends, leveraging on the rich connectivity and global reach of popular OSNs. Many OSNs now offer the *social voting* function, through which a user can share with friends her opinions, e.g., *like* or *dislike*, on various subjects, ranging from user statuses, profile pictures, to games played, products purchased, websites visited, and so on. Taking like–dislike type of votings one step further, some OSNs, e.g., Sina Weibo [20], empower users to initiate their own voting campaigns, on any topic of their interests, with user-customized voting options. The friends of a voting initiator can participate in the campaign or retweet the campaign to their friends. Other than stimulating social interactions, social voting also has many potential commercial values. Advertisers can initiate votings to advertise certain brands. Product managers can initiate votings to conduct market research. E-commerce owners can strategically launch votings to attract more online customers.

The increasing popularity of social voting immediately brings forth the "information overload" problem: a user can be easily overwhelmed by various votings that were initiated, participated, or retweeted by her direct and indirect friends. It is critical and challenging to present the "right votings" to the "right users" so as to improve user experience and maximize user engagement in social votings. Recommender systems (RSs) deal with information overload by suggesting to users the items that are potentially of their interests. In this paper, we present our recent effort on developing RSs for online social votings, i.e., recommending interesting voting campaigns to users. Different from the traditional items for recommendation, such as books and movies, social votings propagate along social links. A user is more likely to be exposed to a voting if the voting was initialized, participated, or retweeted by her friends. A voting's visibility to a user is highly correlated with the voting activities in her social neighborhood. Social propagation also makes social influence more prominent: a user is more likely to participate in a voting if her friends have participated in the voting. Due to social propagation and social influence, a user's voting behavior is strongly correlated with her social friends. Social voting poses unique challenges and opportunities for RSs utilizing social trust information [14], [26], [28], [32], [34]. Furthermore, voting participation data are binary without negative samples. It is, therefore, intriguing to develop RSs for social voting.

Toward addressing these challenges, we develop a set of novel RS models, including matrix-factorization (MF)-based models and nearest-neighbor (NN)-based models, to learn user-voting interests by simultaneously mining information on user-voting participation, user–user friendship, and user-group affliction. We systematically evaluate and compare the performance of the proposed models using real social voting

Manuscript received October 22, 2014; revised August 19, 2016; accepted January 24, 2017. Date of publication March 2, 2017; date of current version March 6, 2017.

X. Yang was with New York University, New York, NY 10003, USA. He is now with Ads Delivery and Optimization at toutiao.com, Haidian, Beijing 100086, China (e-mail: xy271@nyu.edu).

C. Liang was with Alcatel-Lucent, Murray Hill, NJ 07974 USA, and also with New York University, Brooklyn, NY 11201 USA. He is now with Apple Inc., Cupertino, CA 95014 USA (e-mail: chaoliang.us@gmail.com).

M. Zhao is with the Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong (e-mail: csmiaozhao@comp.polyu.edu.hk).

H. Wang was with the Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong. He is now with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: wanghongwei55@gmail.com).

H. Ding and Y. Liu are with the Department of Electrical and Computer Engineering, New York University Tandon School of Engineering, Brooklyn, NY 11201 USA (e-mail: hd510@nyu.edu; yongliu@poly.edu).

Y. Li and J. Zhang are with Sina Weibo, Beijing 100193, China (e-mail: liyang2@staff.sina.com.cn; junlin@staff.sina.com.cn).

Digital Object Identifier 10.1109/TCSS.2017.2665122

traces collected from Sina Weibo. The contribution of this paper is threefold.

1) Online social voting has not been much investigated to our knowledge. We develop MF-based and NN-based RS models. We show through experiments with real social voting traces that both social network information and group affiliation information can be mined to significantly improve the accuracy of popularity-based voting recommendation.
2) Our experiments on NN-based models suggest that social network information dominates group affiliation information. And social and group information is more valuable to cold users than to heavy users.[1]
3) We show that simple metapath-based NN models outperform computation-intensive MF models in hot-voting recommendation, while users' interests for nonhot votings can be better mined by MF models.

The rest of this paper is organized as follows. Section II presents the related work. We provide a quick overview on the social voting function of Sina Weibo and present measurement results of our data set in Section III. In Section IV, we first develop a multichannel MF model that simultaneously mines user-voting, user–user, and user-group information. We then propose several NN models based on different metapaths in the heterogeneous information network. Experimental results are presented in Section V. This paper is concluded in Section VI.

II. Related Work

Bond *et al.* [1] conducted a 61-million-person experiment about social influence on Facebook [24] during the 2010 U.S. congressional elections. They demonstrated that strong ties in OSNs can influence people's adoption of voting activities. Different from [1], we study social influence on user's adoption of online social votings, which are initiated and propagate purely in OSNs.

Collaborative filtering-based RSs use user feedback data to predict user interests, leading to very accurate recommendations [2]–[11], [13], [29], [36], [37]. Adomavicius and Tuzhilin [2] presented a survey of RSs. Koren [4], [5] and Salakhutdinov and Mnih [7] proposed MF-based models for rating prediction. Cremonesi *et al.* [10] and Shi *et al.* [28] studied collaborative filtering for top-k recommendation. Rendle *et al.* [36] presented a generic optimization criterion Bayesian Personalized Ranking (BPR)-Optimization (Opt) derived from the maximum posterior estimator for optimal personalized ranking. Rendle *et al.* [36] proposed a generic learning algorithm LearnBPR to optimize BPR-Opt. BPR can work on top of our proposed methods, such as Weibo-MF and NN approaches to optimize their performance.

The increasingly popular OSNs provide additional information to enhance pure rating-based RSs. There are many previous studies concerning how to integrate social network information to increase recommendation accuracy, just to name a few, [28], [30]–[35], [38]–[41]. Ma *et al.* [32] proposed to factorize user-item rating matrix and user–user relationship matrix together for item rating prediction. Ma *et al.* [33] claimed that a user's rating of an item is influenced by his/her friends. A user's rating to an item consists of two parts, the user's own rating of the item and the user's friends' ratings of the item. The authors then proposed to combine the two ratings linearly to get a final predicted rating. Jamali and Ester [31] claimed that a user's interest is influenced by his/her friends. Thus, a user's latent feature is constrained to be similar to his/her friends' latent features in the process of MF. Yang *et al.* [30] claimed that a user's interest is multifacet and proposed to split the original social network into circles. Difference circles are used to predict ratings of items in different categories. Jiang *et al.* [38] addressed utilizing information from multiple platforms to understand user's needs in a comprehensive way. In particular, they proposed a semisupervised transfer learning method in RS to address the problem of cross-platform behavior prediction, which fully exploits the small number of overlapped crowds to bridge the information across different platforms. Jiang *et al.* [39] considered enriching information for accurate user-item link prediction by representing a social network as a star-structured hybrid graph centered on a social domain, which connects with other item domains to help improve the prediction accuracy. Moreover, context awareness is also an important measure to facilitate recommendation. For example, Sun *et al.* [40] proposed a collaborative nowcasting model to perform context-aware recommendation in mobile digital assistants, which models the convoluted correlation within contextual signals and between context and intent to address sparsity and heterogeneity of contextual signals. Gao *et al.* [41] studied the content information on location-based social networks with respect to point-of-interest properties, user interests, and sentiment indications, which models three types of information under a unified point-of-interest recommendation framework with the consideration of their relationship to check-in actions. In contrast, online social votings are quite different from the traditional recommendation items in terms of social propagation. Different from the existing social-based RSs, besides social relationship, our models also explore user-group affiliation information. We study how to improve social voting recommendation using social network and group information simultaneously.

One-class collaborative filtering (OCCF) deals with binary rating data, reflecting a user's action or not. In OCCF, only positive samples are observed, and there are a large number of missing entries. OCCF has been widely studied, such as [17]–[19]. This paper can also be classified into OCCF. The difference is that we are dealing with binary data from multiple channels, consisting of binary user-voting activities, user–user trust relationships, and user-group affiliations. We are the first to study recommendation of the emerging online social votings to the best of our knowledge.

NN algorithms identify the so-called neighbors of a target user. A prediction of item preferences or a list of recommended items for the target user can be produced by combining preferences of the neighbors. Jamali and Ester [26] proposed an approach, namely Trust-CF, to incorporate social network

[1] In this paper, we define users with less than five votings as *cold users* and with more than ten votings as *heavy users*. We define votings that attract no less than 1000 users as *hot votings* and less than 10 users as *cold votings*.

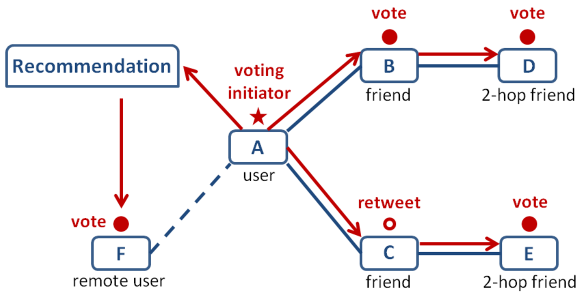


Fig. 1. Social voting propagation paradigm.

into NN-based top-k RSs. Trust-CF calculates the predicted rating for a candidate item as the weighted average of all observed ratings in the traditional CF neighborhood and social neighborhood. Trust-CF does not work with binary data set, as the weighted average of all observed items is 1. Yang *et al.* [14] proposed Trust-user latent feature space-based collaborative filtering approach (Trust-CF-ULF) to incorporate social network information into top-k RSs. *Trust-CF-ULF approach* is the combination of CF-ULF and social network-based approach. Using metapath-based approaches, we consider a wider set of neighborhoods than [14], which can be treated as a special case of our hybrid NN approaches.

Social voting as a new social network application has not been studied much in the existing literature. Compared with traditional items for recommendation, the uniqueness of online social voting lays in its social propagation along social links. Also, the purpose of initializing a voting is to engage people to express their opinions. Thus, the topics covered in online social votings are generally more engaging than other applications in OSNs. Section III presents some interesting statistics of our online social voting data trace.

III. Social Voting

Weibo [20] (the chinese word for "microblog") is a hybrid of Twitter and Facebook-like social application launched by the Sina corporation, China's biggest Web portal, in August 2009. As of 2013, it had accumulated more than 600 million registered users and over 120 million daily active users in 2016 [21]. Users on Weibo follow each other. A user can write posts (tweets) and share them with his followers. Users can also join different interest groups based on their geographic/demographic features and interests of topics.

Voting [22] is an embedded feature of Sina Weibo. More than 92 million users have participated in various votes on Weibo as of January 2013. There are more than 2.2 million ongoing votings available on Sina Weibo each day. As shown in Fig. 1, any user can initiate a voting campaign. After a voting is initiated, there are two major ways through which other users can see the voting and potentially participate. The first way is social propagation: after a user initiated or participated in a voting, all his/her followers can see the voting; a user can also choose only retweet a voting to his followers without participation. The other way is through Weibo voting recommendation list, which consists of popular votings and personalized recommendation. We have no information about Weibo's voting recommendation algorithms.

TABLE I

General Statistics of Weibo Data Set

Users	1,011,389	Social Relations	83,636,677
Votings	185,387	User-Groups	5,643,534
Groups	299,077	User with Votings	525,589
User-Votings	3,908,024	User with Groups	723,913

A. Measurement Study

We obtained user-voting logs directly from the technical team of Sina Weibo.[2] The data set covers votings from November 2010 to January 2012. The data set has detailed information about votings each user participated in, voting contents, and the end time of each voting. We only know user-voting participation, not user-voting results, i.e., we do not know which voting option a user chose. The data set also contains social connections between users and groups a user joined. The data set only contains bidirectional social links, i.e., A follows B and B follows A. Our following study is thus focused on the impact of social ties between users with more or less equal statuses. Summary statistics of the data set are shown in Table I. On average, each user has 82.7 followees, and each user has participated in 3.9 votings. If we count only users with at least one voting, the average voting number of each user is 7.4. Fig. 2 presents the distribution curves of the above-mentioned statistics. Fig. 2(a) is the distribution of the number of votings participated by a user, for all the users with at least one voting. The horizontal axis in Fig. 2(a) is ranking of users based on the number of votings they participated. The ranking starts with 1. Fig. 2(b) is the distribution of the number of participants of a voting, and the average number of participants of a voting is 21.1. The horizontal axis in Fig. 2(b) is ranking of votings based on the number of participants. The ranking starts with 1. Fig. 2(c) is the distribution of the number of followees of a user. Fig. 2(d) is the distribution of the number of groups joined by a user, for all the users joining at least one group. Among users joined at least one group, the average number of joined groups is 7.8. Fig. 2(e) is the distribution of the number of users in a group, for all group joined by at least one user. The horizontal axis in Fig. 2(e) is ranking of groups based on the number of users joining the group. The ranking starts with 1. The average number of users joining a same group is 18.9. Fig. 2(f) is the distribution of the number of votings (may contain duplicated votings) participated by the users in a group. The average number of votings participated by a group is about 56.7.

To gain more understandings about how users are connected and how social votings propagate in OSNs, we calculate the social distances, i.e., the length of shortest path in the social networks, between different types of user pairs. We consider the entire social network with 1 011 389 users as a graph and randomly select 10k users as the source vertices. We iteratively conduct breadth-first-search (BFS) to compute the shortest path distance between each of those sources and all other vertices along social graph edges. Fig. 3(a) shows the Cumulative Distribution Function (CDF) of the

[2]We would love to publish the data set for other researchers to verify and improve our results if our paper gets accepted.

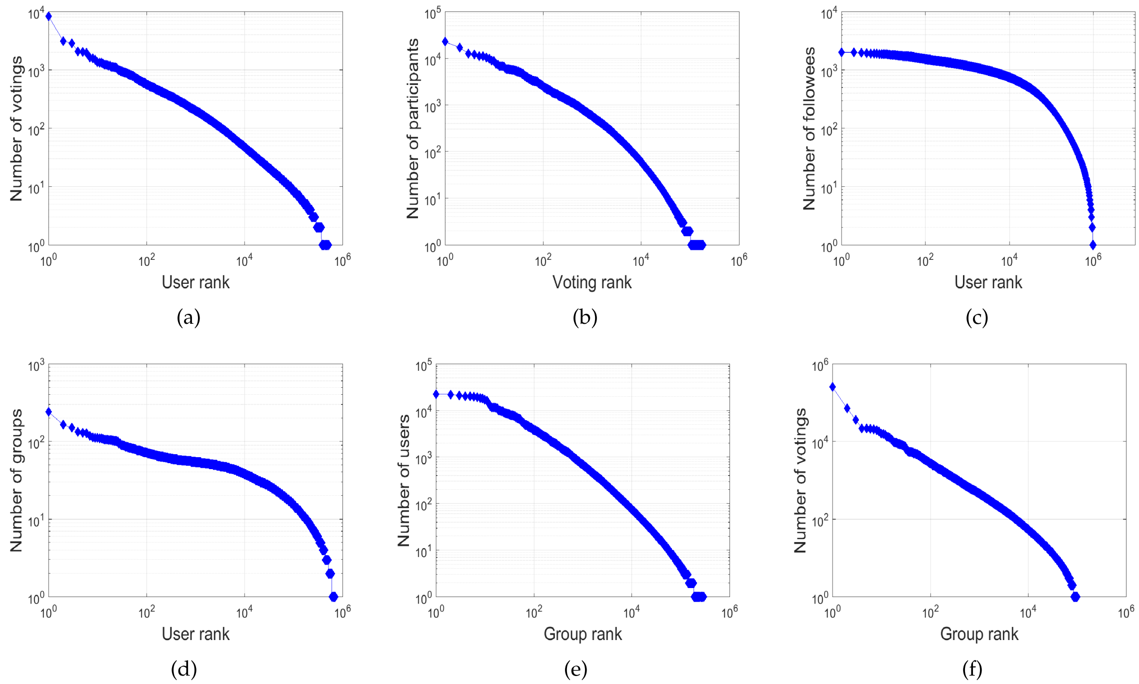


Fig. 2. (a) Distribution of the number of votings participated by a user. (b) Distribution of the number of participants of a voting. (c) Distribution of the number of followees of a user. (d) Distribution of the number of groups joined by a user. (e) Distribution of the number of users in a group. (f) Distribution of the number of votings participated by the users in a group.

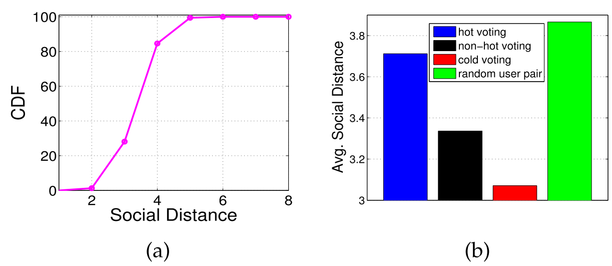


Fig. 3. (a) Distribution of random user pair's social distance. (b) Average social distance between user pairs in hot, nonhot, and cold votings versus random user pairs.

social distance among the pairs between these sources and all other users. The average social distance is 3.86 hops. Then, we calculate the social distances for users participating in the same voting. We order the votings according to the number of participants and divided them into three types: *hot votings*, with no less than 1000 users; *nonhot votings* with less than 1000 users; and *cold votings* with less than 10 users. We randomly select 100 hot votings, nonhot votings, and cold votings separately. The average social distances between users in different types of votings are shown in Fig. 3(b), and the average distances are 3.71, 3.34, and 3.07 for hot, nonhot, and cold votings, respectively. As expected, users participated in the same voting are socially closer than randomly selected users. More popular social votings propagate deeper in the underlying social network, and their participants can be farther away from each other than less popular votings.

In our Weibo data set, for a voting participated by a user, there is a 33.9% chance that at least one of his 1-hop followee has participated in the voting, and 80.6% chance that at least one of his followees within two hops has participated in the voting, and the number for followees within 3-hop is 96.4%. For comparison, the corresponding numbers for regular item adoption in the Epinions data set [25] are: 28.7%, 61.9%, and 76.9%, respectively. Epinions is a consumer opinion site where users review various items, such as cars, movies, books, software, and so on, and assign ratings to the items. Users also assign trust values (i.e., a value of 1) to other users whose reviews and/or ratings they find valuable.

It is also interesting to study the correlation between social votings and social groups. We see from the Weibo data that among users participating in a same voting, 10.40% of the user pairs joined at least one common group, while only 0.92% of randomly selected user pairs joined at least one common group. It indicates that users in a same group share similar voting interests. Thus, we will also study how much group information can improve social voting recommendation in this paper.

IV. Social Voting Recommendation

We consider *top-k voting recommendation* in OSNs. For each user, the RS has to recommend a *small* number, say k, of votings from *all available* votings. We introduce performance metrics for top-k recommendation in Section IV-A. MF methods were found to be very efficient in general top-k recommendation [10], [12]. Furthermore, social network information can be exploited to improve the accuracy of top-k recommendation [14], [26]. For this reason, we start with MF approaches using both social network information and group affiliation information. In Section IV-B, we propose a multichannel MF model, which factorizes user-voting inter-

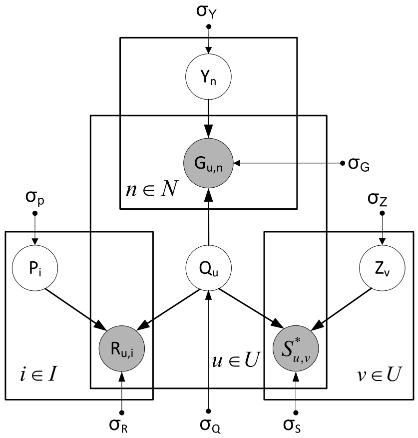


Fig. 4. Graphic model of Weibo-MF.

actions, user–user interactions, and user-group interactions simultaneously, gearing to optimize top-k hit rate. Other than MF approaches, we also consider NN approaches in Section IV-C. We first construct neighborhoods by traversing different types of metapaths in the Weibo heterogeneous information network. We then explore user neighborhoods in the latent feature space derived from MF models.

A. Performance Metrics

Recall or top-k hit rate is widely used in evaluating RSs. To compute the top-k hit rate, we rank the items $i \in I$ according to their predicted rating $\hat{R}_{u,i}$ for each user $u \in U$. An item is defined as *relevant* to a user in the test set if she/he finds it appealing or interesting, e.g., the rating value is above a certain threshold. In our experiments with Weibo data, the real rating values are binary (0, 1), and we consider 1 as relevant. The top-k hit rate or recall of user u is defined as the fraction of relevant items in the test set that appear in the top-k of the ranking list, denoted by $N(k, u)$, from among all relevant items, $N(u)$. Similar to [12], the recall over all users is computed as follows:

$$\text{recall} = \frac{\sum_u N(k, u)}{\sum_u N(u)}. \tag{1}$$

Note that a higher top-k hit rate or recall is better. We use recall as the evaluation metric in our experiments.

B. Multichannel Matrix Factorization

The social network information is represented by a matrix $S \in \mathbb{R}^{u_0 \times u_0}$, where u_0 is the number of users. The directed and weighted social relationship of user u with user v (e.g., user u trusts/knows/follows v) is represented by a positive value $S_{u,v} \in (0, 1]$. An absent or unobserved social relationship is reflected by $S_{u,v} = s_m$, where typically $s_m = 0$. The user-group affiliation information is represented by matrix $G \in \mathbb{R}^{u_0 \times n_0}$, where $G_{u,n}$ is binary and takes value 1 if user u joins group n, and 0 otherwise.

1) Weibo-MF Model: The graphic model of Weibo-MF is shown in Fig. 4. The user-voting interaction $R_{u,i}$ is determined by user latent feature Q_u and voting latent feature P_i, user-group interaction $G_{u,n}$ is determined by user latent feature Q_u and group latent feature Y_n, and user–user interaction $S^*_{u,v}$ is determined by user latent feature Q_u and factor feature Z_v. Similar to [32], we normalize the social network matrix S to incorporate local authority and local hub values

$$S^*_{u,v} = S_{u,v}\sqrt{\frac{d_v^-}{d_u^+ + d_v^-}}$$

where d_u^+ is the out-degree of user u in the social network (i.e., the number of users whom u follows/trusts), and d_v^- is the in-degree of user v in the network (i.e., the number of users who follow/trust user v). The predicted rating of user u for item i is a function of u's latent feature Q_u and i's latent feature P_i

$$\hat{R}_{u,i} = r_m + Q_u P_i^T \tag{2}$$

with matrices $P \in \mathbb{R}^{i_0 \times j_0}$ and $Q \in \mathbb{R}^{u_0 \times j_0}$, where $j_0 \ll i_0$, u_0 is the rank; and $r_m \in \mathbb{R}$ is a (global) offset. Besides the rating data, the social network information is also used in model training. The social relationships between users are predicted as follows:

$$\hat{S}^*_{u,v} = s_m + Q_u Z_v^\top \tag{3}$$

where $Z \in \mathbb{R}^{u_0 \times j_0}$ is a third matrix in this model, besides P and Q. The row vector Z_v denotes factor specific latent feature vector of user v. Ma *et al.* [32] provide more detailed description of matrix Z. Note that the matrix Z is not needed for predicting rating values, and, hence, may be discarded after the matrices P and Q have been learned.

Another matrix G is used for factorization. $G_{u,n}$ is the affinity of user u to group n. Typically, the affinity value is binary, i.e., user u belongs to a group n or not. Group affinity values are predicted as

$$\hat{G}_{u,n} = g_m + Q_u Y_n^T \tag{4}$$

with $Y \in \mathbb{R}^{n_0 \times j_0}$, where the rank $j_0 \ll n_0$; and $g_m \in \mathbb{R}$ is a (global) offset. Note that the matrix Q is shared among the three equations (2)–(4). In other words, we expect user latent feature vectors directly influence user voting, trust between users, and user-group affiliation. Predictions in (2)–(4) are combined in the training objective function (5). Study [13] showed that user's selection bias leads to lower rating values missing with higher probability. Thus, the observed user-rating tuples are not representative of the real user-item matrix (which is unknown). Therefor, analogous to [12], we modify the training function as to account for *all* items (instead of Root Mean Square Error (RMSE) on the *observed* ratings) for improved top-k hit rate on the test data

$$\begin{aligned}\sum_{\text{all } u}\sum_{\text{all } i} & W_{u,i}\big(R^{\text{o\&i}}_{u,i} - \hat{R}_{u,i}\big)^2 \\ & + \sum_{\text{all } u}\sum_{\text{all } v} W^{(S)}_{u,v}\big(S^{*o\&i}_{u,v} - \hat{S}^*_{u,v}\big)^2 \\ & + \sum_{\text{all } u}\sum_{\text{all } n} W^{(G)}_{u,n}\big(G^{o\&i}_{u,n} - \hat{G}_{u,n}\big)^2 \\ & + \lambda\big(||P||_F^2 + ||Q||_F^2 + ||Y||_F^2 + ||Z||_F^2\big)\end{aligned} \tag{5}$$

where $|| \cdot ||_F$ denotes the Frobenius norm of the matrices, and λ is the usual regularization parameter. $R^{\text{o\&i}}_{u,i}$ equals the actual rating value in the training data if observed for user u and item i; otherwise, the value $R_{u,i}^{\text{o\&i}} = r_m$ is imputed. The training weights are set as in [12]

$$W_{u,i} = \begin{cases} 1, & \text{if } R_{u,i}^{\text{obs}} \text{ observed} \\ w_m, & \text{otherwise.} \end{cases} \tag{6}$$

Note that, in deriving (5), to make the model clear and tractable, we made the simplifying assumption that user–user social relationships, user-group affiliations, and user-voting interactions are independent of each other. Similar assumptions were made in [12] and [32], which also included detailed description on how to derive the training model.

The term concerning the social network (in the second line) is analogous to the first term concerning the ratings. In particular, the absent or unobserved user–user affinity is treated analogous to the missing ratings, i.e., we impute the missing user–user affinity with value s_m and weight $w_m^{(S)}$. Like $W_{u,i}$ in (6), $W_{u,v}^{(S)}$ is defined as follows:

$$W_{u,v}^{(S)} = \gamma_s \cdot \begin{cases} 1, & \text{if } S_{u,v}^{*} \text{ observed} \\ w_m^{(S)}, & \text{otherwise} \end{cases} \tag{7}$$

where $\gamma_s \geq 0$ determines the weight of the social network information compared with the rating data. Obviously, $\gamma_s = 0$ corresponds to the extreme case where the social network is ignored when learning the matrices P and Q. As γ_s increases, the influence of the social network increases. The effect is that the latent feature vectors Q_u and Q_v of two users u and v become more similar to each other if they are friends.

Akin to missing user–user affinity, the absent or unobserved user-group affiliations are treated analogously. We impute the value g_m with weight $w_m^{(G)}$. Like $W_{u,i}$ in (6), $W_{u,v}^{(G)}$ is defined as follows:

$$W_{u,n}^{(G)} = \gamma_g \cdot \begin{cases} 1, & \text{if } G_{u,n} \text{ observed} \\ w_m^{(G)}, & \text{otherwise} \end{cases} \tag{8}$$

where $\gamma_g \geq 0$ determines the weight of the user-group interaction information compared with the rating data. Obviously, $\gamma_g = 0$ corresponds to the extreme case where the user-group interaction is ignored when learning the matrices P and Q. As γ_g increases, the influence of the user-group affiliation increases. The effect is that if two users u and v share more groups in common, their latent feature vectors Q_u and Q_v should become more similar. This objective function can be optimized using the method of alternating least squares. We manually tune the different weights to optimize the recall metric. Finally, we summarize the details of Weibo-MF model, as shown in Algorithm 1.

Algorithm 1 Algorithm of Weibo-MF Model

Data: Sina Weibo voting dataset
Result: Top-k Hit Rate

```
// Training part
Load sina weibo voting training data;
Initialize latent feature matrices Q and P;
// Update latent features by ALS
while Not Converge & Iteration Number is less than Iter_Num do
    Update Q by fixing P and minimizing Eq. (5);
    Update P by fixing Q and minimizing Eq. (5);
end
// Testing part
for each user u in Sina Weibo voting dataset for testing do
    for each voting i in test dataset for user u do
        Calculate the predicted rating of user u on voting i as R̂_{u,i} = r_m + Q_u P_i^T;
        Put R̂_{u,i} into the queue recomm_pool;
    end
    Sort recomm_pool in an decreasing order according to the value of R̂_{u,i};
    Select foremost K votings with largest R̂_{u,i} from recomm_pool as the items for recommendation;
    Calculate top-k hit rate for user u;
end
Return average top-k hit rate for entire system;
```

C. Nearest-Neighbor Methods

Other than MF approaches, NN-based recommendations have also been studied. NN methods are widely used in RSs [4], [14], [26]. Thus, it is very intriguing to study the performance of NN models on social voting recommendation problem. In NN-based approaches, the neighborhood of a user can be calculated using collaborative filtering, or it can be a set of directly or indirectly connected friends in a social network, or just a set of users with similar interests in a same group. This makes it convenient to incorporate social trust and user-group interaction into NN-based top-k recommendation. In this section, we try different approaches to construct nearest neighborhood for a target user.

1) Metapath Neighborhoods: In heterogeneous information networks, objects are of multiple types and are linked via different types of relations or sequences of relations, forming a set of metapaths [15]. Metapath is a path that connects objects of different types via a sequence of relations. Different metapaths have different semantics. Sun *et al.* [16] employ metapaths for clustering task in heterogeneous information networks. In this paper, we use metapaths for recommendation task.

In this paper, we leverage the idea of metapath [15] to construct nearest neighborhoods for target users. Different from [15], the starting object type in a metapath is user, and the ending object type is voting. Fig. 5(a) shows the schema of Weibo heterogeneous information network. It contains three types of objects, namely, user (U), voting (V), and group (G). Links exist between a user and a voting by the relation of "vote" and "voted by," between a user and a group by "join" and "joined by," between a user and another user by "follow" and "followed by." We consider a set of different metapaths for the purpose of NN voting recommendation. Fig. 5(b)–(d) shows different metapaths. The solid lines between users are social connections; the dashed lines between users and groups are user-group interactions, i.e., a user joins a group;

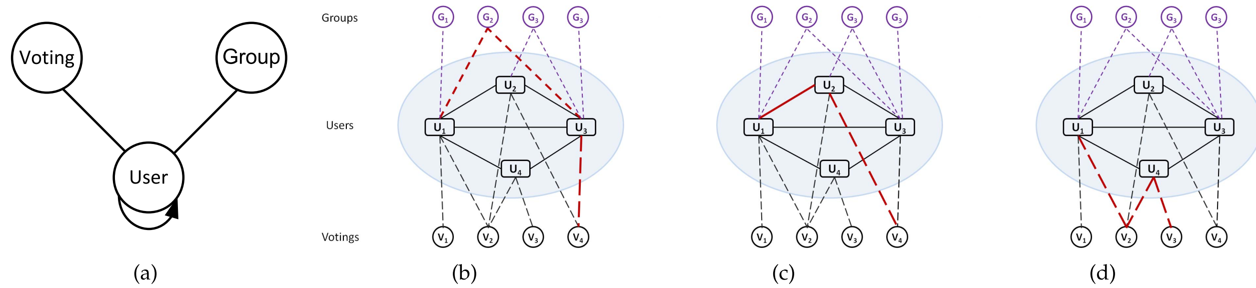


Fig. 5. (a) Weibo heterogeneous information networks. (b) Example of U-G-U-V metapath. (c) Example of U-U-V metapath. (d) Example of U-V-U-V metapath.

the dashed lines between users and votings are user-voting activities, i.e., a user participates in a voting. In Fig. 5(b)–(d), the red highlighted lines compose the metapaths, and the starting object of metapaths is U_1.

a) UGUV metapath: As shown in Fig. 5(b), the semantic of using $U - G - U - V$ metapath for recommendation is finding users that in a same group with the target user, then recommending their votings to the target user. More specifically, *UGUV* works as follows.

1) For a target user u, UGUV searches for all the groups that u has joined. Denote the set of groups as G^u.
2) For each joined group $g \in G^u$, search for all the users that belong to group g.
3) Users in group g report their relevant votings.
4) Combine the reports of all groups. The score for a candidate voting i to the target user u is computed as

$$\text{Score}_{u,i} = \sum_{g \in G^u} \sum_{v \in g} \sum_{i} w(g) \delta_{i \in I_v} \tag{9}$$

   where δ is the Kronecker delta, I_v denotes the set of user v's relevant votings, and $w(g)$ is the weight of users in group g. In our later experiments, we try $w(g)$ as a function of group size. We found that the best function of $w(g)$ is to simply set $w(g) = 1$.
5) Rank recommended votings according to their scores, and return the top-k votings.

b) UUV(m-hop) metapath: As shown in Fig. 5(c), the semantic of $U - U - V(m - hop)$ metapath-based recommendation is to recommend a target user the relevant votings of his followees within m-hops. *UUV* approach employs the BFS in social network to find users similar to the target user u. The scoring scheme is similar to the scheme employed in *UGUV*

$$\text{Score}_{u,i} = \sum_{v \in N_u^{(s)}} \sum_{i} w_s(u, v) \delta_{i \in I_v} \tag{10}$$

where $N_u^{(s)}$ is the set of neighbors of u in social networks and $w_s(u, v)$ is the weight of user v. We set $w_s(u, v) = w_s(d_v)$, where d_v is the depth of user v in the BFS tree rooted at user u. By fixing 1-hop followees' weight at $w_s(1) = 1$, we tune the weight of 2-hop users. In our later experiments, we found the best value is $w_s(2) = 0.1$. Votings are ranked according to their scores to form the recommendation list.

Algorithm 2 Algorithm of UGUV Metapath

Data: Sina Weibo voting dataset
Result: Top-k votings for recommendation

Initialization;
for *each target user u* **do**
    Find all groups g's that user u has joined and put them in a set G^u;
    for *each joined group $g \in G^u$* **do**
        Find all user v's in group g;
        for *each user v in group g* **do**
            User v reports its relevant votings and put them in a set I_v;
            for *each candidate voting $i \in I_v$* **do**
                $Score_{u,i} += w(g)$;
            end
        end
    end
    Sort $\{Score_{u,i}\}$ in a decreasing order;
    Return and recommend top k votings with highest scores to user u;
end

c) UVUV metapath: As shown in Fig. 5(d), the semantic of $U - V - U - V$ metapath-based recommendation is to find users that share votings with the target user, and then recommend their relevant votings to the target user. For a target user u, *UVUV* works as follows.

1) Find all votings that u has participated in, and denote this voting set as I_u.
2) For each of the voting $j \in I_u$, find the set of users who have participated in j. Denote the set of users as N_j.
3) Each user $v \in N_j$ reports all the votings that he has participated in.
4) Aggregate the reports of all users to assign scores to votings as follows:

$$\text{Score}_{u,i} = \sum_{j \in I_u} \sum_{v \in N_j} \sum_{i} w(v) \delta_{i \in I_v}. \tag{11}$$

In our later experiments, we set $w(v) = 1$ for all users.

Finally, we summarize the algorithm details of UGUV, UUV(m-hop), and UVUV metapath approaches in Algorithms 2–4, respectively.

2) Neighborhoods in Latent Feature Space: Other than neighborhoods visited through metapaths, we also explore

Algorithm 3 Algorithm of UUV(m-Hop) Metapath

Data: Sina Weibo voting dataset
Result: Top-k votings for recommendation
Initialization;
for *each target user* u **do**
  Find all followees v's within m-hops by BFS;
  Put all those v's in a set $N_u^{(s)}$;
  for *each user* $v \in N_u^{(s)}$ **do**
    User v reports its relevant votings and put them in a set I_v;
    Set weight parameter $w_s(u, v)$ according to the depth of user v in the BFS tree rooted at user u;
    for *each voting* $i \in I_v$ **do**
      $Score_{u,i} += w_s(u, v)$;
    end
  end
  Sort $\{Score_{u,i}\}$ in a decreasing order;
  Return and recommend top k votings with highest scores to user u;
end

Algorithm 4 Algorithm of UVUV Metapath

Data: Sina Weibo voting dataset
Result: Top-k votings for recommendation
Initialization;
for *each target user* u **do**
  Find all votings j's that user u has participated;
  Put all those voting j's into a set I_u;
  for *each voting* $j \in I_u$ **do**
    Find all users v's who ever participated in voting j and put them in a set N_j;
    for *each user* $v \in N_j$ **do**
      Find all votings i's that user v has participated and put them in a set I_v;
      for *each voting* $i \in I_v$ **do**
        $Score_{u,i} += w(v)$;
      end
    end
  end
  Sort $\{Score_{u,i}\}$ in a decreasing order;
  Return and recommend top k votings with highest scores to user u;
end

neighborhoods in the user latent feature space derived from MF models. Note that, previous works show that *PureSVD* [10] and *AllRank* [12] perform better than neighborhood-based approaches in user-item space directly when used in top-k recommendation. Yang *et al.* [14] shows that neighborhood in latent feature space approach is comparable with *AllRank*; therefore, we study neighborhood in latent feature space in this section.

a) UNN: *UNN* uses MF (i.e., *AllRank* [12]) to obtain the user latent features. Users are then clustered in the user latent feature space using the Pearson correlation coefficient. Users nearest to the source user u are identified and denoted as N_u. The relevant votings of these nearest users are scored and ranked to form the top-k recommendation list. The score of a candidate voting i is calculated as follows:

$$\text{Score}_{u,i} = \sum_{v\in N_u}\sum_{i} \text{sim}(u, v)\delta_{i\in I_v} \tag{12}$$

where N_u is the set of NNs of user u in the user latent feature space, and the NNs of user u are weighted according to their similarity $sim(u, v)$ with user u, measured in terms of the Pearson correlation between user u and v.

b) VNN: This approach works similarly as UNN, except we cluster votings in the voting latent feature space

$$\text{Score}_{u,i} = \sum_{x\in I_u}\sum_{i} sim(x, i)\delta_{i\in N_x} \tag{13}$$

where I_u is the set of votings participated by user u and N_x is the set of NNs of voting x in the voting latent feature space.

3) Combined Neighborhoods: Hybrid Approach is the combination of UGUV, UUV(m-hop), UVUV, and UNN approaches. We integrate the four recommenders by combining their voting results. Basically, for a target user u, we consider a set of neighboring users that either share the same group with u, or have short social distances to u, or share similar tastes in votings. The score of a potential relevant voting i for user u is calculated as

$$\begin{aligned}\text{Score}_{u,i} = {}& \rho_1 \times \sum_{g\in G^u}\sum_{v\in g}\sum_{i} w(g)\delta_{i\in I_v} \\ &+ \rho_2 \times \sum_{v\in N_u^{(s)}}\sum_{i} w_s(u, v)\delta_{i\in I_v} \\ &+ \rho_3 \times \sum_{j\in I_u}\sum_{v\in N_j}\sum_{i} w(v)\delta_{i\in I_v} \\ &+ \rho_4 \times \sum_{v\in N_u}\sum_{i} sim(u, v)\delta_{i\in I_v}\end{aligned} \tag{14}$$

where ρ_1, ρ_2, ρ_3, and ρ_4 are the weights of UGUV, UUV(m-hop), UVUV, and UNN approaches, respectively.

V. Experiments

In this section, we evaluate the proposed MF models and NN models using Sina Weibo voting data set.

A. Methodology

We evaluate the performance of a set of voting RSs using the same trace. We use a simple popularity-based RS as the baseline model.

- *MostPop:* This RS recommends the most popular items to users, i.e., the votings that have been voted by the most numbers of users.

For the Weibo-MF model proposed in (5), we evaluate several variants by setting different weights for social and group information.

1) *Voting-MF:* By setting $\gamma_s = 0$ and $\gamma_g = 0$ in (5), we only consider user-voting matrix and ignore social and group information. Note that Voting-MF is essentially the same as AllRank model, which is proposed in [12].

TABLE II

Top-k Hit Rate OF MF METHODS ($j_0 = 10$). THE PERCENTAGE NUMBERS IN EACH CELL ARE THE RELATIVE IMPROVEMENTS OVER THE MOSTPOP BASELINE. THE STANDARD DEVIATIONS OF THE RESULTS ARE WITHIN 0.006

Top-K	10	20	50	100
MostPop	0.032	0.050	0.087	0.127
Voting-MF	0.045 40.6%	0.076 52.0%	0.137 57.5%	0.204 60.6%
Voting+Social-MF	0.049 53.1%	0.079 58.0%	**0.148** 70.1%	0.217 70.9%
Voting+Group-MF	0.048 50.0%	0.078 56.0%	0.145 66.7%	**0.218** 71.7%
Weibo-MF	**0.050** 56.3%	**0.080** 60.0%	**0.148** 71.3%	**0.218** 71.7%

AllRank was found to be the best model of optimizing top-k hit ratio on various data sets according to [10] and [12].

2) *Voting + Social-MF:* By setting $\gamma_s > 0$ and $\gamma_g = 0$, we additionally consider social network information on top of Voting-MF.
3) *Voting + Group-MF:* By setting $\gamma_s = 0$ and $\gamma_g > 0$, we additionally consider user-group matrix information on top of Voting-MF.
4) *Weibo-MF:* By setting $\gamma_s > 0$ and $\gamma_g > 0$, we add both social and group information to Voting-MF.

For NN-based RSs, we evaluate UGUV metapath and UUV(m-hop) metapath (with $m = 1, 2$) described in Section IV-C1; UNN, VNN described in Section IV-C2; and the hybrid approach described in Section IV-C3 by setting different weights in (14).

We randomly choose 80% of the data set as training set and the remaining 20% as test set. The random selection was carried out five times independently, and we report the average statistics. We conducted our experiments on a Linux server with four E5640 Intel Xeon CPUs. Each CPU has four cores with 2.67 GHz, and each core has 12.3-MB cache. The shared memory size is 36 GB.

B. MF-Based Approaches

We tune the regularization constant λ and the optimal value is 0.5. For the dimensionality, we choose $j_0 = 10$. We tune the remaining parameters to optimize top-20 hit rate. The performance of MF-based RSs is compared in Table II. In Voting-MF model, the parameters that lead to the best top-20 hit rate are: $w_m = 0.01$ and $r_m = 0$. As expected, *Voting-MF significantly outperforms the naive popularity-based RS*. Since user-voting data are binary, impute the missing value of user-voting as $r_m < 1$, leading to the same result as $r_m = 0$.

In Voting + Group-MF, the optimal parameters are $\gamma_g = 0.1, w_m^{(G)} = 0.001$, and $g_m = 0$. In Voting + Social-MF, the optimal parameters are $\gamma_s = 0.1, w_m^{(S)} = 0.00005$, and $s_m = 0$. Due to the computation constraints, we only present the results of $j_0 = 10$ for all different MF models here.

It is evident that Weibo-MF outperforms all other MF-based approaches, since more information used in the model leads to more prediction power. Regarding the results between Voting-MF and Voting + Social-MF, it is noticed that Voting-MF model is good to represent and mine the data with 40.6%–60.6% relative improvement over MostPop. Adding social information to Voting-MF leads to additional ten plus percent relative gain, which validates that explicitly reinforcing the social influence in MF model can further improve the performance at certain level. Another interesting observation is that Voting + Group-MF and Weibo-MF almost cannot or can only bring limited improvement over Voting + Social-MF approach. This implies that *group information is dominated by social information in social voting recommendation*. This is because votings propagate via social links not via groups as described in Section III.

TABLE III

Top-k Hit Rate COMPARISON FOR VOTING-MF AND NEIGHBORHOOD-BASED METHODS. THE PERCENTAGE NUMBERS IN EACH CELL ARE THE RELATIVE IMPROVEMENTS OVER THE MOSTPOP BASELINE. THE STANDARD DEVIATIONS OF THE RESULTS ARE WITHIN 0.007

Top-K	10	20	50	100
MostPop	0.032	0.050	0.087	0.127
Voting-MF($j_0 = 80$)	0.094 193.8%	0.133 166.0%	0.204 134.5%	0.270 112.6%
UGUV	0.063 96.9%	0.094 88.0%	0.152 74.7%	0.210 65.4%
UUV(1-hop)	0.070 118.8%	0.100 100.0%	0.151 73.6%	0.196 54.3%
UUV(2-hop)	0.071 121.9%	0.107 114.0%	0.176 102.3%	0.246 93.7%
UVUV	0.093 190.6%	0.128 156.0%	0.192 120.7%	0.255 100.8%
UNN	**0.113** 253.1%	**0.155** 210.0%	**0.227** 160.9%	**0.291** 129.1%
VNN	0.041 28.1%	0.061 22.0%	0.097 11.5%	0.132 3.9%
UGUV +UUV(2-hop)	0.076 137.5%	0.114 128.0%	0.185 112.6%	0.256 101.6%
UGUV+UNN	0.117 265.6%	0.165 230.0%	0.248 185.1%	0.323 154.3%
UGUV+UVUV	0.096 200.0%	0.134 168.0%	0.201 131.0%	0.269 111.8%
UUV(2-hop) +UNN	0.125 290.6%	0.174 248.0%	0.262 201.1%	0.342 169.3%
UUV(2-hop) +UVUV	0.099 209.4%	0.139 178.0%	0.209 140.2%	0.279 119.7%
UNN+UVUV	0.120 275.0%	0.164 228.0%	0.242 178.2%	0.315 148.0%
UGUV+UNN +UUV(2-hop)	0.125 290.6%	0.175 250.0%	0.261 200.0%	0.341 168.5%
UGUV+UVUV +UUV(2-hop)	0.099 209.4%	0.138 176.0%	0.209 140.2%	0.279 119.7%
UUV(2-hop) +UNN+UVUV	**0.127** 296.9%	**0.177** 254.0%	**0.264** 203.4%	**0.345** 171.7%

C. NN-Based Approaches

Table III shows the top-k hit rate for neighborhood-based methods. The percentage numbers in each cell are the relative improvements over the MostPop method. Among which UNN is based on user latent features obtained by Voting-MF at $j_0 = 80$. The detailed performance of UNN at different neighborhood sizes is shown in Fig. 6.

In Table III, we can see that UGUV + UNN outperforms UNN, and UGUV + UVUV outperforms UVUV.

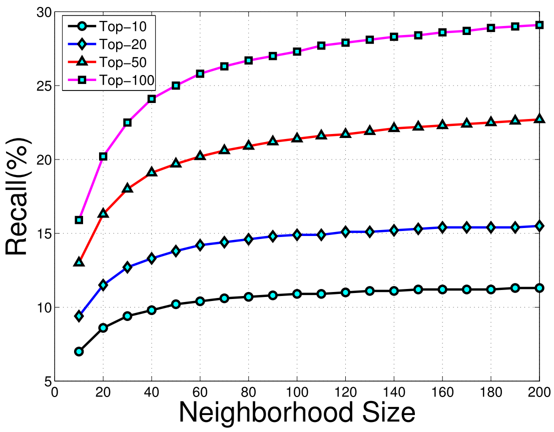


Fig. 6. Recall versus neighborhood size of UNN.

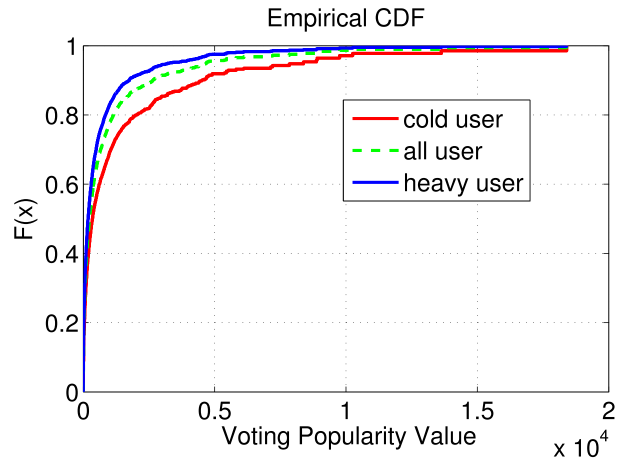


Fig. 7. Comparison of cold users', heavy users', and overall users' tendencies for popular votings.

This suggests that group information is helpful for social voting recommendation. Meanwhile, UGUV + UUV(2-hop) + UNN performs almost the same as UUV(2-hop) + UNN, with top-20 hit rate of 0.175 versus 0.174; and UGUV + UUV (2-hop) + UVUV performs almost the same as UUV(2-hop) + UVUV, with a top-20 hit rate of 0.138 versus 0.139. Thus, we can conclude that *social network information dominates user-group information in social voting recommendation*.

The top performers in Table III all employ UNN. *This suggests that metapath-based neighborhoods alone are not sufficient in achieving high hit rate.*

We can also see from Table III that UUV(2-hop) + UNN + UVUV performs the best, and outperforms MostPop by 254% in terms of top-20 recommendation. The optimal weights of UUV(2-hop) + UNN + UVUV hybrid method are $\rho_2 = 1$, $\rho_4 = 5$, and $\rho_3 = 0.02$.

We find that VNN performs the worst among all approaches except MostPop. It suggests that cluster votings in the latent feature space are not a good idea. It might due to the fact that voting adoption behavior is more social propagation-oriented, and users prefer to vote because of his/her followees vote. We also try adding VNN into the hybrid model described in Section IV-C3, which bring no further improvements.

D. Different Views

To gain more insights, we present experiment results of four different views: cold users, who have less than five votings; heavy users, who have more than ten votings; hot votings, which attract no less than 1000 users; and nonhot votings, which attract less than 1000 users. We calculate top-k hit rate for each view separately. The results are presented in Tables IV and V.

From Table IV, we make the following observations. For hot votings, among all the single methods, UVUV performs the best. It might be because UVUV tends to recommend hot votings, and/or UVUV returns more accurate hot votings. We will look into this issue more in Section V-E. For nonhot votings, UNN performs the best among all single methods, and Voting-MF ranks the second. Thus, we see here MF-based approaches (UNN and Voting-MF) not only learn users' mainstream interests (hot votings), but also learn users' nonmainstream interests (nonhot votings) efficiently.

Comparing UUV(1-hop) with UUV(2-hop), we can see that UUV(1-hop) is much better in nonhot voting and UUV(2-hop) is much better in hot voting. It can be explained that nonhot votings are local events, and cannot propagate far away in social networks, and thus do not need users far away for recommendation. To the contrary, hot votings can propagate widely in social network, so friends within 2-hops can be useful in recommending hot votings. *Intuitively, with more samples from a larger neighborhood, recommendation of hot votings can be made more accurate.*

Comparing UGUV + UNN with UNN, UGUV + UVUV with UVUV, UUV(2-hop) + UNN with UNN, and UUV(2-hop) + UVUV with UVUV, we can see that both group information and social information can improve the hit rate of hot votings, while nonhot voting hit rate stays the same or even slightly worse. This suggests that while larger neighborhood can promote hot votings better, it might have negative impact on nonhot votings.

From Table V, for all approaches, hit rates for cold users performs are higher than heavy users, which seems counterintuitive. After investigating into the data set, we find that cold users' tendency to participate in hot votings is rather strong. The CDFs of the popularity values of votings participated by different types of users are shown in Fig. 7. The average number of participants of cold users' votings is 1564, and the median value is 349, while for heavy users, the average and median values are only 727 and 183, respectively. *It is obvious that cold users are more likely to participate in hot votings, while heavier users are more likely to try less popular votings.* From Table IV, it is always easier to recommend hot votings. This explains why hit rates for cold users are consistently higher than those for heavy users cross all the approaches.

From UUV(2-hop) versus UUV(1-hop), UUV(2-hop) + UNN versus UNN, UUV(2-hop) + UVUV versus UVUV, UGUV + UNN versus UNN, and UGUV + UVUV versus UVUV, we can see that in all cases, cold user's recommendation accuracy has been improved significantly, while heavy

TABLE IV

HOT VOTINGS VERSUS NONHOT VOTINGS. THE STANDARD DEVIATIONS OF THE RESULTS ARE WITHIN 0.009

Approach	Top-10		Top-20		Top-50		Top-100	
	hot	non-hot	hot	non-hot	hot	non-hot	hot	non-hot
MostPop	0.145	0	0.223	0	0.390	0	0.568	0
UGUV	0.188	0.027	0.274	0.043	0.419	0.075	0.547	0.113
UUV(1-hop)	0.142	0.049	0.197	0.072	0.277	0.114	0.334	0.155
UUV(2-hop)	0.215	0.029	0.316	0.047	0.483	0.088	0.629	0.135
Voting-MF	0.231	0.053	0.292	0.085	0.376	0.153	0.443	0.217
UNN	0.219	**0.082**	0.288	**0.116**	0.385	**0.181**	0.458	**0.242**
UVUV	**0.284**	0.038	**0.388**	0.053	**0.552**	0.087	**0.675**	0.134
UGUV+UUV(2-hop)	0.234	0.030	0.341	0.048	0.514	0.089	0.665	0.138
UGUV+UNN	0.263	0.075	0.361	0.109	0.503	0.174	0.616	0.239
UGUV+UVUV	0.304	0.036	0.416	0.052	0.593	0.088	0.728	0.137
UUV(2-hop)+UNN	0.278	0.080	0.377	0.115	0.534	0.183	0.654	0.252
UUV(2-hop)+UVUV	0.316	0.037	0.433	0.053	0.616	0.092	0.752	0.143
UNN+UVUV	0.271	0.076	0.363	0.106	0.504	0.166	0.614	0.228
UUV(2-hop)+UNN +UVUV	0.302	0.076	0.413	0.108	0.578	0.173	0.707	0.241

TABLE V

COLD USERS VERSUS HEAVY USERS. THE STANDARD DEVIATIONS OF THE RESULTS ARE WITHIN 0.008

Approach	Top-10		Top-20		Top-50		Top-100	
	cold	heavy	cold	heavy	cold	heavy	cold	heavy
MostPop	0.061	0.017	0.088	0.029	0.140	0.055	0.193	0.086
UGUV	0.087	0.049	0.123	0.077	0.180	0.133	0.233	0.192
UUV(1-hop)	0.108	0.050	0.143	0.077	0.192	0.127	0.230	0.177
UUV(2-hop)	0.114	0.048	0.162	0.078	0.241	0.141	0.310	0.208
Voting-MF	0.096	0.076	0.130	0.114	0.177	0.194	0.213	0.273
UNN	0.114	0.096	0.143	0.140	0.180	0.226	0.207	0.310
UVUV	0.111	0.072	0.146	0.104	0.201	0.167	0.250	0.236
UGUV+UUV(2-hop)	0.119	0.052	0.170	0.084	0.249	0.148	0.322	0.218
UGUV+UNN	0.144	0.091	0.190	0.137	0.258	0.225	0.313	0.311
UGUV+UVUV	0.128	0.072	0.173	0.104	0.242	0.167	0.304	0.236
UUV(2-hop)+UNN	0.163	0.093	0.215	0.139	0.299	0.228	0.366	0.316
UUV(2-hop)+UVUV	0.140	0.072	0.190	0.104	0.271	0.168	0.341	0.237
UNN+UVUV	0.128	0.099	0.164	0.143	0.219	0.230	0.262	0.316
UUV(2hop)+UNN +UVUV	0.165	0.096	0.219	0.141	0.304	0.228	0.371	0.317

user's performance almost stays the same. *Thus, we conclude that group and social information are valuable for cold users but not helpful for heavy users in social voting recommendation.*

From UUV(2-hop) + UNN versus UGUV + UNN and UUV(2-hop) + UVUV versus UGUV + UVUV, we can see that social information helps improve recommendation accuracy for cold users more than group information does. This again suggests that *social information is more useful* to cold users than group information, echoing our findings in Sections V-B and V-C.

E. Hot-Voting-Only Recommendation

As mentioned in Section III, it is very intriguing to study hot-voting recommendation as it propagates through both social networks and global channels, such as headline news. In this section, we focus on recommending hot votings only.

To study hot-voting recommendation, we filter out a hot-voting-data set that only contains hot votings. We choose votings with no less than 1000 participants as hot votings. In the training set, we pick out all the hot votings and only keep hot-voting related tuples. In the testing set, we only keep hot-votings related tuples for testing. We further get rid of users in the testing set who do not appear in the training set. In the resulting hot-voting data set, there are 290 184 users and 329 votings, 700 628 user-voting tuples in the training set, and 138 682 user-voting tuples in the testing set.

In hot-voting experiments, we report top-5 to top-50 results. We tune all parameters to optimize top-10 hit rate. The optimal regularization constant λ is 0.5. We try different value of dimensionality and the best value is $j_0 = 20$. As we can see that, due to much less number of votings, the optimal j_0 is much smaller than in the whole data set. In Voting-MF model, the optimal parameters are: $w_m = 0.06$ and $r_m = 0$. The optimal parameters of UUV(2-hop) are $w_s(2) = 0.1$, the same as the whole data set. The optimal weights of UUV(2-hop) + UNN are $\rho_2 = 1$ and $\rho_4 = 1$. The optimal weights of UUV(2-hop) + UVUV are $\rho_2 = 1$ and $\rho_3 = 0.1$.

In Table VI, Voting-MF is better than UNN, and UVUV is better than Voting-MF. From Section V-D, we know that UVUV favors hot-voting recommendation. It might just because UVUV tends to recommend more hot votings than other methods. From these hot-voting-only experiments, we can see that *UVUV can indeed recommend hot votings more accurately than other methods, even if all the methods are only focused on hot votings.* One explanation is that UVUV approach's neighborhood size is very large. Through a hot voting, a user is connected to more than 1000 other

TABLE VI
Top-k Hit Rate OF HOT VOTINGS. THE STANDARD DEVIATIONS OF THE RESULTS ARE WITHIN 0.009

Top-K	View	5	10	20	50
MostPop	all	0.077	0.131	0.207	0.373
	cold	0.089	0.151	0.227	0.396
	heavy	0.051	0.093	0.163	0.321
Voting-MF	all	0.187	0.273	0.371	0.542
	cold	0.183	0.258	0.354	0.504
	heavy	0.169	0.272	0.396	0.639
UGUV	all	0.137	0.209	0.310	0.482
	cold	0.151	0.223	0.320	0.481
	heavy	0.105	0.174	0.274	0.473
UUV(1-hop)	all	0.129	0.196	0.280	0.384
	cold	0.136	0.203	0.283	0.372
	heavy	0.114	0.180	0.285	0.418
UUV(2-hop)	all	0.145	0.231	0.344	0.544
	cold	0.161	0.252	0.363	0.553
	heavy	0.117	0.198	0.325	0.550
UNN	all	0.168	0.232	0.315	0.446
	cold	0.140	0.186	0.243	0.323
	heavy	0.188	0.285	0.421	0.661
UVUV	all	**0.212**	**0.296**	**0.406**	**0.598**
	cold	0.217	0.302	0.406	0.588
	heavy	0.167	0.252	0.376	0.616
UUV(2-hop) +UNN	all	0.210	0.311	0.438	0.643
	cold	0.212	0.314	0.436	0.630
	heavy	0.180	0.282	0.427	0.662
UUV(2-hop) +UVUV	all	0.215	0.308	0.424	0.628
	cold	0.221	0.317	0.432	0.625
	heavy	0.177	0.261	0.388	0.624

users. UVUV can quickly sample voting popularity among a large number of users who share common voting interest with the target user. From Section V-D, we know that *with more samples from a larger neighborhood, recommendation of hot votings can be made more accurate.*

Comparing UUV(2-hop) + UVUV with UVUV, social network information cannot improve much over UVUV metapath method. Comparing UUV(2-hop) + UNN with UNN, we see that UUV(2-hop) improves the performance of UNN, especially for cold users.

VI. CONCLUSION AND FUTURE WORK

In this paper, we present a set of MF-based and NN-based RSs for online social voting. Through experiments with real data, we found that both social network information and group affiliation information can significantly improve the accuracy of popularity-based voting recommendation, especially for cold users, and social network information dominates group affiliation information in NN-based approaches. This paper demonstrated that social and group information is much more valuable to improve recommendation accuracy for cold users than for heavy users. This is due to the fact that cold users tend to participate in popular votings. In our experiments, simple metapath-based NN models outperform computation-intensive MF models in hot-voting recommendation, while users' interests for nonhot votings can be better mined by MF models. This paper is only our first step toward thorough study of social voting recommendation. As an immediate future work item, we would like to study how voting content information can be mined for recommendation, especially for cold votings. We are also interested in developing voting RSs customized for individual users, given the availability of multichannel information about their social neighborhoods and activities.

REFERENCES

[1] R. M. Bond *et al.*, "A 61-million-person experiment in social influence and political mobilization," *Nature*, vol. 489, pp. 295–298, Sep. 2012.

[2] G. Adomavicius and 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.*, vol. 17, no. 6, pp. 734–749, Jun. 2005.

[3] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques," *Adv. Artif. Intell.*, vol. 2009, Aug. 2009, Art. no. 421425, doi: 10.1155/2009/421425.

[4] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. ACM KDD*, 2008, pp. 426–434.

[5] Y. Koren, "Collaborative filtering with temporal dynamics," in *Proc. KDD*, Paris, France, 2009, pp. 447–456.

[6] A. Paterek, "Improving regularized singular value decomposition for collaborative filtering," in *Proc. KDDCup*, 2007, pp. 39–42.

[7] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. NIPS*, vol. 20. 2008, pp. 1257–1264.

[8] K. Yu, A. Schwaighofer, V. Tresp, X. Xu, and H. P. Kriegel, "Probabilistic memory-based collaborative filtering," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 1, pp. 56–69, Jan. 2004.

[9] R. H. Keshavan, A. Montanari, and S. Oh, "Matrix completion from noisy entries," *J. Mach. Learn. Res.*, vol. 11, pp. 2057–2078, Jul. 2010.

[10] P. Cremonesi, Y. Koren, and R. Turrin, "Performance of recommender algorithms on top-N recommendation tasks," in *Proc. ACM RecSys*, 2010, pp. 39–46.

[11] Y. Zhang, B. Cao, and D.-Y. Yeung, "Multi-domain collaborative filtering," in *Proc. 26th Conf. Uncertainty Artif. Intell. (UAI)*, Catalina Island, CA, USA, 2010, pp. 725–732.

[12] H. Steck, "Training and testing of recommender systems on data missing not at random," in *Proc. ACM KDD*, 2010, pp. 713–722.

[13] B. Marlin and R. Zemel, "Collaborative prediction and ranking with non-random missing data," in *Proc. ACM RecSys*, 2009, pp. 5–12.

[14] X. Yang, H. Steck, Y. Guo, and Y. Liu, "On top-k recommendation using social networks," in *Proc. ACM RecSys*, 2012, pp. 67–74.

[15] Y. Sun, J. Han, X. Yan, P. S. Yu, and T. Wu, "PathSim: Meta path-based top-k similarity search in heterogeneous information networks," *Proc. VLDB Endowment*, vol. 4, no. 11, pp. 992–1003, 2011.

[16] Y. Sun, B. Norick, J. Han, X. Yan, P. S. Yu, and X. Yu, "Integrating meta-path selection with user-guided object clustering in heterogeneous information networks," in *Proc. ACM KDD*, 2012, pp. 1348–1356.

[17] R. Pan *et al.*, "One-class collaborative filtering," in *Proc. ICDM*, 2008, pp. 502–511.

[18] Y. Li, J. Hu, C. Zhai, and Y. Chen, "Improving one-class collaborative filtering by incorporating rich user information," in *Proc. ACM CIKM*, 2010, pp. 959–968.

[19] V. Sindhwani, S. S. Bucak, J. Hu, and A. Mojsilovic, "One-class matrix completion with low-density factorizations," in *Proc. IEEE ICDM*, Dec. 2010, pp. 1055–1060.

[20] *Weibo*. [Online]. Available: http://weibo.com/

[21] *By the Numbers: 58 Amazing Weibo Statistics*. (May 2016). [Online]. Available: http://expandedramblings.com/index.php/weibo-user-statistics/

[22] *Sina Vote*, accessed on 2013. [Online]. Available: http://vote.weibo.com/

[23] Twitter, accessed on 2017. [Online]. Available: http://twitter.com

[24] Facebook, accessed on 2013. [Online]. Available: http://www.facebook.com

[25] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in *Proc. ACM KDD*, 2002, pp. 61–70.

[26] M. Jamali and M. Ester, "Using a trust network to improve top-N recommendation," in *Proc. ACM RecSys*, 2009, pp. 181–188.

[27] K. Järvelin and J. Kekäläinen, "Cumulated gain-based evaluation of IR techniques," *ACM Trans. Inf. Syst.*, vol. 20, no. 4, pp. 422–446, 2002.

[28] S.-H. Yang, B. Long, A. Smola, N. Sadagopan, Z. Zheng, and H. Zha, "Like like alike: Joint friendship and interest propagation in social networks," in *Proc. WWW*, 2011, pp. 537–546.

[29] Y. Shi, A. Karatzoglou, L. Baltrunas, M. Larson, N. Oliver, and A. Hanjalic, "CLiMF: Learning to maximize reciprocal rank with collaborative less-is-more filtering," in *Proc. ACM RecSys*, 2012, pp. 139–146.

[30] X. Yang, H. Steck, and Y. Liu, "Circle-based recommendation in online social networks," in *Proc. ACM KDD*, 2012, pp. 1267–1275.

[31] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. ACM RecSys*, 2010, pp. 135–142.

[32] H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec: Social recommendation using probabilistic matrix factorization," in *Proc. ACM CIKM*, 2008, pp. 931–940.

[33] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proc. SIGIR*, 2009, pp. 203–210.

[34] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. ACM WSDM*, 2011, pp. 287–296.

[35] J. Noel *et al.*, "New objective functions for social collaborative filtering," in *Proc. 21st Int. Conf. World Wide Web (WWW)*, 2012, pp. 859–868.

[36] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian personalized ranking from implicit feedback," in *Proc. 25th Conf. Uncertainty Artif. Intell. (UAI)*, 2009, pp. 452–461.

[37] R. Jäschke, L. Marinho, A. Hotho, L. Schmidt-Thieme, and G. Stumme, "Tag recommendations in folksonomies," in *Proc. 11th Eur. Conf. Principles Practice Knowl. Discovery Databases (PKDD)*, 2007, pp. 506–514.

[38] M. Jiang, P. Cui, N. J. Yuan, X. Xie, and S. Yang, "Little is much: Bridging cross-platform behaviors through overlapped crowds," in *Proc. 13th AAAI Conf. Artif. Intell.*, 2016, pp. 13–19.

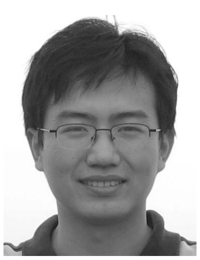
[39] M. Jiang, P. Cui, X. Chen, F. Wang, W. Zhu, and S. Yang, "Social recommendation with cross-domain transferable knowledge," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 11, pp. 3084–3097, Nov. 2015.

[40] Y. Sun, N. J. Yuan, X. Xie, K. McDonald, and R. Zhang, "Collaborative nowcasting for contextual recommendation," in *Proc. 25th Int. World Wide Web Conf. (WWW)*, 2016, pp. 1407–1418.

[41] H. Gao, J. Tang, X. Hu, and H. Liu, "Content-aware point of interest recommendation on location-based social networks," in *Proc. 29th AAAI Conf. Artif. Intell. (AAAI)*, 2015, pp. 1721–1727.

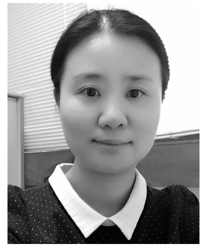
Xiwang Yang received the B.S. degree from the Department of Electronic Science and Technology, University of Science and Technology of China, Hefei, China, in 2008, and the Ph.D. degree in electrical and computer engineering from New York University, Brooklyn, NY, USA, in 2013.

He was a Research Scientist with Facebook Inc., Menlo Park, CA, USA. He is currently with Ads Delivery and Optimization at toutiao.com, Beijing, China. His current research interests include machine learning and data mining, focused on developing models and algorithms for explanatory and the predictive analysis of largescale online user behavior and their application in ads optimization.



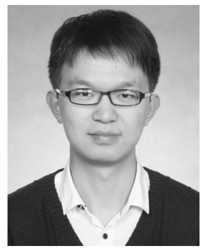
Chao Liang received the B.E. and M.E. degrees from the Department of Electronic and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, and the Ph.D. degree from the Department of Electrical and Computer Engineering, Tandon School of Engineering, New York University, Brooklyn, NY, USA.

His current research interests include network optimization, data analytics, and recommender systems.



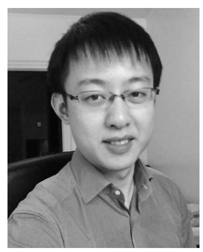
Miao Zhao (M'10) received the B.E. and M.E. degrees from the Department of Electronic and Information Engineering, Huazhong University of Science and Technology, Wuhan, China, and the Ph.D. degree from Electrical and Computer Engineering Department, State University of New York, Stony Brook, NY, USA.

She has been an Assistant Professor with the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, since 2016. Her current research interests include data mining, recommender systems, and multimodal deep learning for various applications.



Hongwei Wang received the B.Sc. degree in computer science from Shanghai Jiao Tong University, China. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China.

His current research interests include data mining, recommender systems, and deep learning.



Hao Ding received the B.S. degree in physics from the University of Science and Technology of China, Hefei, China. He is currently pursuing the Ph.D. degree in electrical and computer engineering with Tandon School of Engineering, New York University, Brooklyn, NY 11201 USA.

His current research interests include machine learning and data mining, with an emphasis on online social network-based recommender system.

Yong Liu (F'17) received the bachelor's and master's degrees in automatic control from the University of Science and Technology of China, Hefei, China, in 1994 and 1997, respectively, and the Ph.D. degree from Electrical and Computer Engineering Department, University of Massachusetts Amherst, Amherst, MA, USA, in 2002.

He was with New York University, Brooklyn, NY, USA, as an Assistant Professor, in 2005, where he is currently an Associate Professor with the Electrical and Computer Engineering Department, New York University Tandon School of Engineering. His current research interests include modeling, design, and analysis of communication networks, multimedia networking, network measurement and analytics, software-defined networks, online social networks, and recommender systems.

Dr. Liu is a member of ACM.

Yang Li, photograph and biography not available at the time of publication.

Junlin Zhang received the Ph.D. degree from the Institute of Software, Chinese Academy of Science, Beijing, China, in 2004.

He was as a Senior Expert in Sina Weibo, China, Baidu, Beijing, and Alibaba, Hangzhou, China, in past ten years. His current research interests include natural language processing, deep learning, and recommender systems.