

Received March 22, 2018, accepted April 25, 2018, date of publication May 21, 2018, date of current version July 19, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2832185

Recommendation With Social Roles

DUGANG LIU, JIE HUANG, AND CHEN LIN[✉], (Member, IEEE)

Department of Computer Science, School of Information Science and Technology, Xiamen University, Xiamen 361005, China

Corresponding author: Chen Lin (chenlin@xmu.edu.cn)

This work was supported by the Natural Science Foundation of China under Grant 61472335 and Grant 61472333.

ABSTRACT Social recommender is an active research area. Most previous social recommenders adopt existing social networks to augment recommendations which are based on user preferences. In this contribution, we propose to simultaneously infer the social influence network and the user preferences in a matrix factorization framework. Furthermore, we assume that the influence strength is dependent on the social roles of users. We present an incremental clustering algorithm to detect dynamic social roles. Comprehensive experiments on real data sets demonstrate the efficiency and effectiveness of our model to generate precise recommendations.

INDEX TERMS Social role, recommender system, social recommendation, matrix factorization.

I. INTRODUCTION

Recommender Systems (RS) seek to predict user ratings on items by learning user preferences in historical consumptions. Since the early stage of RS, researchers have believed in the “word of mouth” effect and adopted social networks to augment predictions which are purely based on individuals’ preferences. This type of social network enhanced recommender systems is usually called “social recommenders”. Social recommender is an active research area. Many studies [1], [6]–[10], [13]–[17], [20], [21] have shown that, social recommenders can produce better recommendations, deal with cold start recommendations, and are more robust to fraud.

In the literature of social recommenders, most works utilize an existing social network and adopt a two-step approach to augment recommendations, i.e. modeling of social networks and recommendations are considered separately. In the first step, the social network is constructed as a graph, with nodes representing peers, and ties representing *social influences*. In the second step, the social ties are incorporated in the recommender framework either as binary variables, i.e. trust or not trust [6], [7], [9], [13], [14], [16], [20]; or with heuristically assigned weights [15], [17], [21]. A common sense is that social influences are of greatly varying reliability and strength which could not be accurately determined by simple heuristics. Therefore a few recent works [1], [10] attempt to automatically learn the weights for each tie between two peers. However, such a two-step approach can not take full advantage of the user’s preference information, so that the model is often suboptimal.

Furthermore, inferring peer-to-peer strength might raise two challenges. On one hand, communities and social groups are universally observed in human societies. A peer-to-peer social network per se could not explain the formation of communities and social groups. On the other hand, with the availability of very large social networks online, it is a hurdle for us to efficiently learn the weights for each tie.

Social scientists develop the *social role* theory to explain communities and social groups, by “presuming that persons are members of social positions and hold expectations of their own behaviors and those of other persons” [2]. In recent results Zhao *et al.* [26] indicates social role is the part that a person plays as a member of a particular society and people behave differently in social situations because they carry different latent social roles. It is possible to capture different social roles in various social networks by observing historical behavioral features. Social influence “is rooted in social roles”, since social influence occurs “when we expect people within certain social roles to act according to our expectations for those roles” [5].

Recommendation systems can benefit from the theoretic foundation of social roles and social influences. Let’s take the academic recommender for researchers as an example. Suppose we are given the social network of several researchers with their respective social roles, i.e. researchers *a, b, c, d, e* are students and *f* is an advisor. As shown in Fig. 1, the strength of a social tie (social inference) is dependent on the social roles of both ends. For example, teachers may expect their students to work on research domains that they are interested, thus teachers have stronger influences on

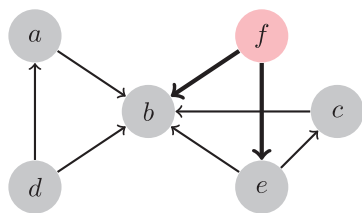


FIGURE 1. Illustrative social network. Black nodes are students, red nodes are teachers, thickness of lines indicates strength of influence.

TABLE 1. Illustrative topic distribution for users in Fig. 1. Blond columns for topic “recommender systems”, blue columns for topic “data mining”, celadon columns for mixture of topics.

user	recommend	matrix	rank	skyline	frequent
a	1	2		5	4
b	4	5	4	1	1
c	1		1	3	5
d			4	5	5
e		1			
f	5	5	3		

the students’ research topics. Tab. 1 illustrates the hidden researcher preferences over a set of keywords in two domains. In order to predict the preferences for researcher *b*, we must take into account the strength of influence. Though he has more collaborators in domain “data mining”, his research interest is dominantly influenced by his teacher and thus focuses on domain “recommender system”. In a word, social role “advisor” has more influence than social role “students” to *b* and we can use role specific influence instead of the original peer-to-peer influence.

In this work, we enhance recommender systems by incorporating role specific influence. Human behaviors are complex and dynamic. As a natural consequence, the social role for one individual evolves persistently. Our first step is to identify the social roles for each individual at each timestamp. Given a stream of social network snapshots, we propose to incrementally cluster the roles of peers in each network snapshot.

We then assume that each user-item rating is generated as a weighted combination of users’ own preferences and preferences of role specific influence sources. We propose a novel single-step approach to simultaneously infer the social influence network and the user preferences in a matrix factorization framework. The user preferences are learnt while role specific influence is inferred simultaneously by iteration. Our model can be interpreted as a cyclic model. We show that if the roles are static for a period of time, the user preferences inferred in our model are limiting approximation on average ratings given the time lag approaches to zero.

The inferred role specific influence strengths can be treated as a skeleton network. Given a social network of U users, suppose we have detected K roles, the computational complexity and storage costs at each iteration are significantly reduced from $O(U^2)$ to $O(K^2)$. For very big social networks, the skeleton network alone is capable to produce recommendations.

Our experiment on real academic data sets shows that our model outperforms state-of-the-art social recommenders. Furthermore, we experimentally demonstrate that the skeleton network significantly cuts down the number of iterations required to converge. The inferred skeleton network is a good abstract of the original network, as in the experiments it produces comparative performance for graph based recommender systems, compared with a full network.

The rest of this paper is organized as follows. Section II briefly summarizes related researches. Section III and Section IV introduce the methodology. Section V presents our experimental results and analysis. Section VI gives our conclusion of this contribution and future work.

II. RELATED WORK

A. SOCIAL RECOMMENDATION

Most state-of-the-art social recommendation systems are within the framework of matrix factorization [12] or its probabilistic variants [18]. In general, there are three manners to exploit the social network information. The first style is to allow the connected users to contribute in the ratings. For example, STE [14] presumes that a rating is generated not only by the user’s own interest but also his/her trusted friends interests. A similar linear combination scheme is also presented in [1] and RoRec [21].

Another type of methods treats the social links as observations. In a similar way to the rating matrix, the binary link matrix is decomposed to hidden user spaces. However, the definition of hidden user spaces could be different. Sorec [16] employs the user preference vector to factorize the link matrix. TrustMF [20], on the contrary, constructs the links from truster-specific and trustee-specific feature vectors, which are then fused as user preference vectors to predict missing ratings.

The third category of research incorporates a regularization term in the original MF optimization objective. According to the social network structure, there would be various forms of regularization terms. SocialMF [9], SoReg [17] and RoRec [21] fall into this category and apply L2 regularization. The model can be easily expanded to distrust relations, while the intuition is naturally converted to maximize the difference between a user and his/her distrusted neighbors, e.g. RWT and RWD [15]. Other attempts in this category include ranking alike regularization terms inspired by MR-BPR [13]. Hinge loss is adopted in PushTrust [6] and MF+TD [7].

B. SOCIAL ROLE

Social Role is one of the basic theories of sociology. It is a set of connected behaviours, rights, obligations, beliefs, and norms as conceptualized by people in a social situation [2]. In recent years, some works have been proposed to utilize the social role information as a new component to enhance the performance of recommender systems. Most of them assume that social roles can be obtained directly or easily from social networks [27]–[30]. Others presume that part of

the social roles are labeled and thus can be solved within the classification framework [26]. Only a few adopt unsupervised learning, such as community detection [31], [32] to infer a user's social role.

III. IDENTIFY SOCIAL ROLES

We are handling a stream of social network snapshots $G = G(0), G(1), \dots, G(T)$, where each network snapshot is represented by a set of nodes (users) and edges (social influences) $G(t) = \langle V(t), E(t) \rangle$. With some pre-processing steps, we represent each user $v(t) \in V(t)$ as a feature vector. We use the same notation $v(t)$ for both the node and its corresponding feature vector, whenever their use is not ambiguous. We define the social role identification problem as a clustering problem. Formally, given $V = V(0), V(1), \dots, V(T)$, a specified number K of clusters (roles), our goal is to learn the cluster label $b(v(t))$ for each user $v(t) \in V(t)$. Following the conventions used in the corresponding research literature, we use one-of- K coding, where $b_k(v(t)) \in \{0, 1\}$ and $\sum_{k=1}^K b_k(v(t)) = 1$.

Many clustering algorithms are distance based. For example, by introducing a cluster centroid, we can assign the user at each timestamp $v(t)$ to its nearest role cluster. The assignment is implemented by computing the distance between any given node to any *centroid cluster vector*.

Definition 1: For each cluster k , we define a **centroid cluster vector** as $c_k = \frac{\sum_{v,t} b_k(v(t))v(t)}{\sum_{v,t} b_k(v(t))}$.

Distance based clustering algorithms are simple and powerful. However, they are not the best choice for clustering dynamic roles. Social roles are continuously evolving. One can reasonably assume that the change is smooth, so that the social role for one user will remain static in a certain period of time. However, a distance based clustering algorithm might wrongly split such a time segment.

To solve this problem, we present an incremental algorithm. For each user at each timestamp $v(t)$, the algorithm favors to keep to the previous cluster assignment, unless it is too close to another cluster and it deviates from its previous patterns. It is necessary to maintain statistics for each user. Therefore we define a *user specific cluster centroid* as follows.

Definition 2: For each user v , we define a **user specific cluster centroid** as $\bar{v}_k = \frac{\sum_t b_k(v(t))v(t)}{\sum_t b_k(v(t))}$.

Intuitively, a large social network, regardless of the timestamp, consists of all possible social roles. In the beginning snapshot of the stream $V(0)$, we conduct a K-means clustering algorithm to initialize K clusters. Once the initial clusters are established, we compute the centroid cluster vectors and user specific cluster centroids. As shown in Alg. 1, for a new $v(t)$, we find its nearest cluster. If the nearest cluster does not match its previous cluster assignment, we evaluate distances to both cluster centroid and previous user specific cluster centroid. If the new node is much closer to another cluster, or if it is much far from its previous centroid, we assign it to a new cluster. We use a distance threshold ϵ to control the evaluation. For example, in Fig. 2, black dashed lines

are cluster boundaries by conventional clustering algorithms, therefore $v(3)$ will be assigned to cluster 2. However, since it is not close enough to the second cluster, $l_{3,1} - l_{3,2} < \epsilon$ and it is not far from previous patterns $d_{3,1} < \epsilon$, we assign $v(3)$ to its previous cluster. $v(5)$ also lies near the boundary. But since it deviates from previous patterns $d_{5,2} > \epsilon$, we assign it to the nearest cluster.

Algorithm 1 Incremental Role Clustering

Require: Initialized clusters $C = \{c_k\}, b(V) = \{b(v)\}, V_{b(v)} = \{v_{b(v)}\}$

while $t! = T$ **do**

for all $v(t) \in V(t)$ **do**

$k \leftarrow \arg_k \min \text{distance}(v(t), c_k)$

if $k \neq b(v)$ **then**

$l_{k,t} \leftarrow \text{distance}(c_k, v(t))$

$l_{b(v),t} \leftarrow \text{distance}(c_{b(v)}, v(t))$

$d_{b(v),t} \leftarrow \text{distance}(v_{b(v)}, v(t))$

if $l_{b(v),t} - l_{k,t} > \epsilon$ **then**

$b(v) \leftarrow k$

else

if $d_{b(v),t} > \epsilon$ **then**

$b(v) \leftarrow k$

end if

end if

end if

$b_{b(v)}(v(t)) = 1$

end for

 update $C, b(V), V_{b(v)}$

end while

IV. ROLE SPECIFIC SOCIAL RECOMMENDER

Let the input be a stream of ratings $r_{i,j,t}$, which is the rating given by user i on item j at time t . We allow the items to be rated for multiple times. For example, in keyword recommendation, each keyword could be repeatedly adopted at different timestamps; in venue recommendation, each user is likely to publish papers on the same venue at different timestamps. Suppose we are given the graph stream G and detected roles of each user at different timestamp $b(v(t))$. Intuitively, if the user's social role changes at a certain time point, i.e. from student to teacher, it is necessary to distinguish the two roles. We divide the rating stream for each user according to the value of $b(v(t))$. For a user with a modified role, we create a new pseudo user. We collect all the ratings in the time period with a consistent role and compute the average rating within this period. Thus we construct a rating matrix $R \in \mathcal{R}^{|U| \times |Q|}$, where $|U|$ is the number of pseudo users, $|Q|$ the number of items, the element $r_{u,q} \in R$ is the average rating which (pseudo) user u gives to item q in a period of time T . We also build a social network based on the cumulative G , where user u is linked to a set of neighbors $N(u) = N(u)^o + N(u)^i$. $N(u)^o$ is the set of out nodes, and $N(u)^i$ represents the in nodes. There are K different roles, the role of user u is denoted by $g(u) \in \{1, \dots, K\}$. Therefore $N(u)_k^i$ is the set of in nodes with role k for user u .

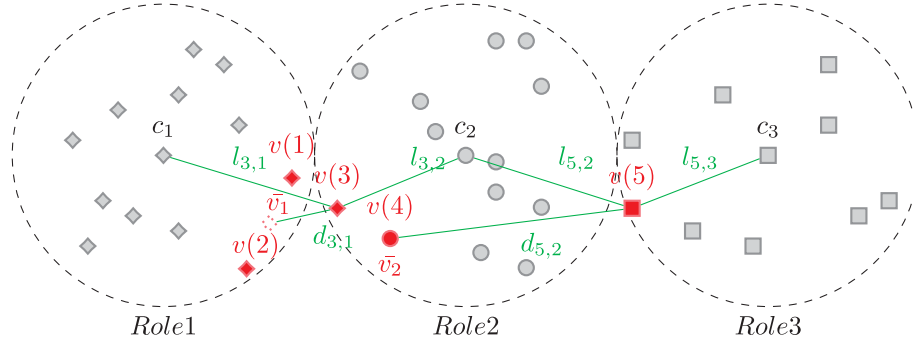


FIGURE 2. An illustrative example of role identification.

Suppose each user is represented by a preference vector $u \in \mathcal{R}^P$ over P topics, and each item $q \in \mathcal{R}^P$ is factorized to P topics. $W \in \mathcal{R}^{K \times K}$, $w_{k,k'}$ represents the strength of influence from role k to role k' . The Role Specific Social Recommender (RSSR) model aims to minimize the following objective function.

$$\begin{aligned} \min_{U, Q, W} \quad & \sum_{r_{u,q} \neq 0} \{r_{u,q} - \alpha(u^T q) \\ & - (1 - \alpha)(\sum_k \sum_{n \in N(u)^i, g(n)=k} \frac{w_{k,g(u)}}{|N(u)_k^i|} r_{n,q})\}^2 \\ & + \lambda_u \|U\|_2 + \lambda_v \|V\|_2 \\ \text{w.r.t.} \quad & \forall w, \forall k, i, w_{k,i} \geq 0, \sum_k w_{k,i} = 1. \quad (1) \end{aligned}$$

Equ. 1 learns the user preference matrix $U \in \mathcal{R}^{|U| \times P}$, the item factor matrix $Q \in \mathcal{R}^{|Q| \times P}$. One possible interpretation of the RSSR model is to assume the user preference at time t is affected by himself/herself and a combination of influencers. $u(t) = \alpha u(t-1) + (1 - \alpha)(\sum_k \sum_{n \in N(u)^i, g(n)=k} \frac{w_{k,u}}{|N(u)_k^i|} n(t-1))$. In matrix form, let's denote $U(t) = AU(t)$, where $A \in \mathcal{R}^{|U| \times |U|}$ is the cyclic adjustment matrix. For a period of time $0 \sim T$, we know from construction that the role of each user is static. In other words, A is static over the period. We have the following theorem.

Theorem 1: Given the cyclic adjustment matrix A as defined above, $\bar{U}(t) = \sum_{k=1}^K U(kt)$. If $U(t) = AU(t)$, then $\lim_{t \rightarrow 0} \bar{U}(t) = A\bar{U}(t)$

Proof: Because $U(t) = AU(t)$, $\bar{U}(t) = A \sum_{k=0}^{K-1} U(kt) = A\bar{U}(t) + A(U(0) - U(T))t$. Then $\bar{U}(t) - A\bar{U}(t) = \sum_j K_j \lambda_j \eta_j$, where K_j s are constants depending on $U(0)$, and λ_j are the characteristic values and η_j are the corresponding characteristic vectors. Since $\rho(A) \leq |A| = 1$, we know that $\lim_{t \rightarrow 0} \bar{U}(t) = A\bar{U}(t)$.

The inference of Equ. 1 is based on a simplex projection algorithm in Alg. 2.

Complexity Analysis: One time-consuming job in algorithm 2 is the updates U and V . We need to invert a $P \times P$ matrix which usually takes $O(P^3)$ operations. Another time-consuming job in Algorithm 2 is to update W and project W on a simplex. The complexity of updates W is $O(K^2)$, as it is shown that there are at most K candidates which can

Algorithm 2 Inference

Require: Randomly initialize U, V, W

while not converge **do**

for all u, v, w **do**

$$Ru_j = r_{u,j} - (1 - \alpha)(\sum_g \frac{g_{g(u)}}{|N(i)_g^i|} \sum_{n \rightarrow i, g(n)=g} r_{n,j})$$

$$Rv_i = r_{i,v} - (1 - \alpha)(\sum_g \frac{g_{g(i)}}{|N(i)_g^i|} \sum_{n \rightarrow i, g(n)=g} r_{n,v})$$

$$Rgu_{g,j} = \frac{\sum_{n \rightarrow u, g(n)=g} r_{n,j}}{|N(u)_g^i|}$$

$$Rg_g = \frac{\sum_{g(u)=w} \sum_{r_{u,v} \neq 0} (r_{u,v} - \alpha uv - (1 - \alpha) \frac{\sum_{n \rightarrow u, g(n)=g} r_{n,v}}{|N(u)_g^i|})}{\sum_{g(u)=w} \sum_{r_{u,v} \neq 0} (r_{u,v} - \alpha uv - (1 - \alpha) \frac{\sum_{n \rightarrow u, g(n)=g} r_{n,v}}{|N(u)_g^i|})}$$

$$u = (\alpha^2 V^u V^u T + \lambda_u I)^{-1} (\alpha V^u R u)$$

$$v = (\alpha^2 U^v U^v T + \lambda_v I)^{-1} (\alpha U^v R v)$$

$$w = w + \eta \{ (1 - \alpha)^2 [\sum_{g(u)=w} Rgu Rgu^T] W_s - (1 - \alpha) Rg \}$$

sort elements in $w : W_{(1)} \leq W_{(2)} \dots W_{(K)}$

for $i = K - 1, i \geq 1, i + +$ **do**

$$t_i = \frac{\sum_{j=i+1}^K W_{(j)}^{-1}}{K - i}$$

if $t_i \geq W_{(i)}$ **then**

$$\hat{t} = t_i$$

break

else

$$i \leftarrow i - 1$$

end if

$$\hat{t} = \frac{\sum_{j=1}^K W_{(j)}^{-1}}{K}$$

$$W = (W - \hat{t})_+$$

end for

end for

end while

be computed explicitly [33]. The overall time complexity is $O((U + Q)P^3 + UQP^2)$. By contrast, the complexity of the original peer-peer algorithm is determined by $\max(O((U + Q)P^3 + UQP^2), O(U^2))$. Our algorithm is more efficient especially when U is large, which is common in real social networks.

V. EXPERIMENT

A. EXPERIMENTAL SETUP

We use a common benchmark for academic recommendations. We use the DBLP publication data set with co-author

TABLE 2. Statistics of dataset.

Dataset	#authors	#papers	#keywords	#coauthor relationships
DBLP	1,712,433	2,092,356	415,555	4,258,615
dm2000	10,628	5,684	7,450	39,102
dm2006	27,130	13,237	13,617	115,938
gr2000	7,563	4,808	6,517	31,828
gr2006	16,688	10,101	10,934	85,230

network available in [24]. For eliminating possible noise brought by cross domain co-authorships and suiting to different experimental tasks, we select proceedings and journals related to two research domains, namely data mining and graphics. We then extract publications on these domains during different time periods and construct four subsets: (1)dm2000: papers published on data mining related conferences and journals between 2000 to 2005; (2)dm2006: papers published on data mining related conferences and journals between 2006 to 2011; (3)gr2000: papers published on graphics related conferences and journals between 2000 to 2005; (4)gr2006: papers published on graphics related conferences and journals between 2006 to 2011. The statistics of the dataset are shown in Tab. 2.

Preprocessing: The keywords are extracted from the title field of each paper. We segment the keywords by filtering the English markers, removing stop-words and stemming. The dataset is constructed into a set of three tuples $\{user_id, keywords_id, ratings/frequency\}$. The features used in role identification include (1)in-degree, (2) out-degree, (3) a vector of average degree of its neighbors, (4) number of ratings and (5) its Pagerank value at the current snapshot. In our algorithm, $P = 20, \alpha = 0.7, K = 15, \lambda_u = \lambda_v = 0.005$. Convergence is achieved when maximal iteration number achieves 400. The comparison algorithms are implemented by LibRec [34] and using the default parameters. The experiments are executed on a server with an Intel Xeon E5620 Processor and main memory 80GB.

B. PARAMETER TUNNING

Two parameters might affect the model performance: (1) α : the combination coefficient which controls to what degree the user preference is dependent on its neighbors; (2) K : the number of social roles in the social network. We tune the parameters based on a small set of publications which contains 200 authors with their frequently adopted keywords. We plot the performance trends (in RMSE) for $\alpha = 0.1, 0.3, 0.5, 0.7, 0.9$ and $K = 3, 5, 10, 15, 20$ in Fig. 3. It shows that best results are obtained when $\alpha = 0.7$ and $K = 15$. The mediate value of α suggests that users are more likely to inherit their own preferences, but they will also be influenced by other people. The difference in RMSE for various values of K is small, which suggests that we can set the estimated number of roles to a small number when the performance loss is affordable

C. SOCIAL RECOMMENDATION

We first evaluate the performance of recommendation. The task is to predict author’s ratings on keywords. Each

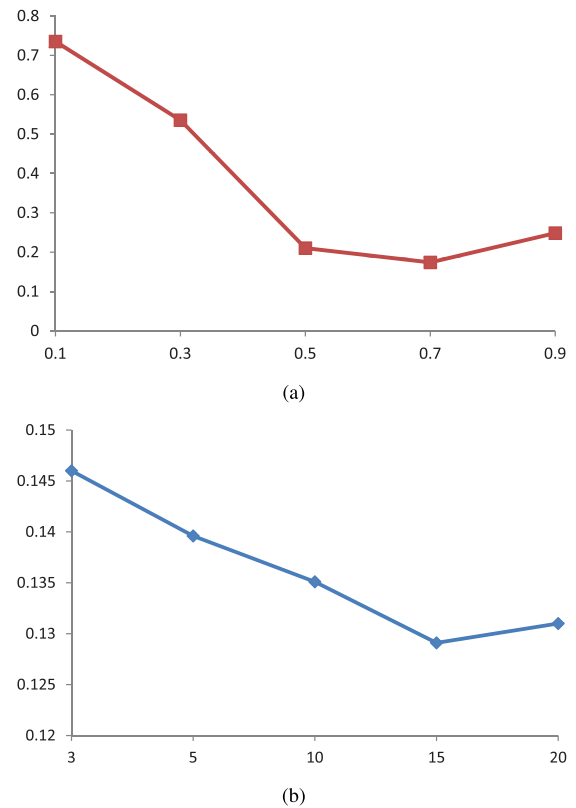


FIGURE 3. RMSE performance over various values of parameters. (a) α . (b) K .

rating is computed as the average normalized frequency over papers during the period of the detected period where the author’s social role is static by Sec. III. The comparative methods include SocialMF [9], BPMF [25] and STE [14], TrustMF [20] and SoRec [16]. We use the standard metrics to measure recommender performances: the mean absolute error (MAE) and the root mean square error (RMSE), as defined below.

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{r}_i - r_i|$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (\hat{r}_i - r_i)^2}$$

We report average MAE and RMSE results in 5-fold validation. As shown in Tab. 3, our proposed model outperforms state-of-the-art recommenders in all subsets. We observe that the increase of performance is significant. In most data sets, MAE performance is boosted by 40%. The enhancement in RMSE is less. The underlying reason might be that for less prestigious authors (users with lower publication frequencies), our algorithm produces more accurate predictions; while for highly popular authors (users with higher publication frequencies), the improvement by our algorithm is smaller.

D. SOCIAL ROLE

We next compare the recommendation performance with different role clustering methods. The comparative methods

TABLE 3. Comparative performance of recommendation.

Evaluation	SocialMF	BPMF	STE	TrustMF	SoRec	RSSR
dm2000						
MAE	0.0989	0.1327	0.1128	0.1943	0.1029	0.0611
RMSE	0.1495	0.2007	0.1457	0.2680	0.1505	0.1312
dm2006						
MAE	0.0949	0.1346	0.0916	0.1976	0.0986	0.0480
RMSE	0.1419	0.1972	0.1381	0.2635	0.1457	0.1237
gr2000						
MAE	0.1134	0.1817	0.1184	0.2129	0.1192	0.0568
RMSE	0.1677	0.2755	0.1674	0.2821	0.1679	0.1659
gr2006						
MAE	0.0997	0.1691	0.0998	0.1889	0.1027	0.0634
RMSE	0.1479	0.2498	0.1476	0.2693	0.1543	0.1457

TABLE 4. Comparative performance of role identification.

Evaluation	Kmeans	Spectral clustering	Community detection	RoIX	Our
dm2000					
MAE	0.0612	0.0610	0.0600	0.0599	0.0610
RMSE	0.1324	0.1321	0.1318	0.1315	0.1312
dm2006					
MAE	0.0571	0.0569	0.0561	0.0557	0.0480
RMSE	0.1238	0.1202	0.1197	0.1200	0.1237
gr2000					
MAE	0.0765	0.0765	0.0760	0.0758	0.0568
RMSE	0.1668	0.1663	0.1668	0.1668	0.1659
gr2006					
MAE	0.0642	0.0641	0.0634	0.632	0.0634
RMSE	0.1461	0.1463	0.1461	0.1459	0.1457

TABLE 5. Performance of graph based recommendation on social network.

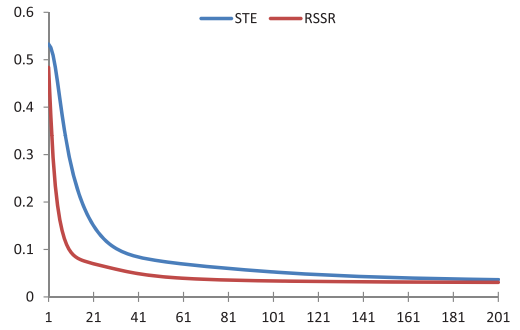
Network	Binary	Heuristic1	Heuristic2	Block	Peer-to-Peer	RSSR
Overall	2.3311	0.3285	0.8176	0.8346	0.6975	0.3465
Cold-start	2.1437	0.3854	0.8704	0.9964	0.8412	0.3767
Old User	2.3855	0.3093	0.8008	0.7784	0.6472	0.3367

include community detection [31] and RoIX [32], K-means and spectral clustering. As shown in Tab. 4, our incremental clustering algorithm is competitive with respect to state-of-the-art algorithms. This verifies that our algorithm achieves both effectiveness and efficiency. We found that the model performance on dm2000 and dm2006 would be slightly worse. One possible explanation is that DM is an interdisciplinary domain, thus to extract accurate social roles, one must use comprehensive feature information as in [31] and RoIX [32].

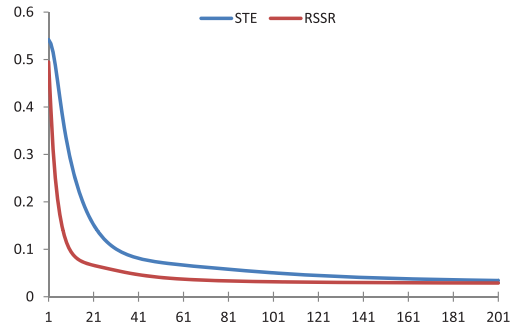
E. THE SKELETON NETWORK

We next testify whether the abstract network inferred in our model can replace the original network within the context of recommender systems. We implement a naive graph-based recommendation algorithm on different networks. For each target user, the graph-based recommendation algorithm propagates its neighbors' ratings to itself, $r_{i,j} = \sum_{n \in N(i)} w_{n,i} r_{n,j}$.

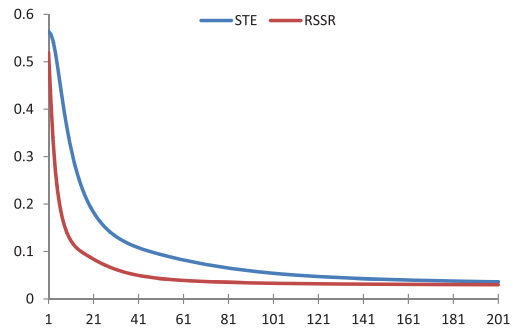
The comparative networks are: (1) Binary: original network with binary edge weighting, where $w_{n,i} = 1$ indicates that a coauthor relationship exists; (2) Heuristic1: undirected coauthor network with heuristically assigned weight $w_{n,i} = \frac{1}{2} \frac{1}{|N(i)|} + \frac{1}{|N(n)|}$; (3) Heuristic2: directed coauthor network with heuristically assigned weight $w_{n,i} = \frac{|N(n)^o|}{|N(n)^o + N(i)^i}$; (4) Block: inferred network with com-



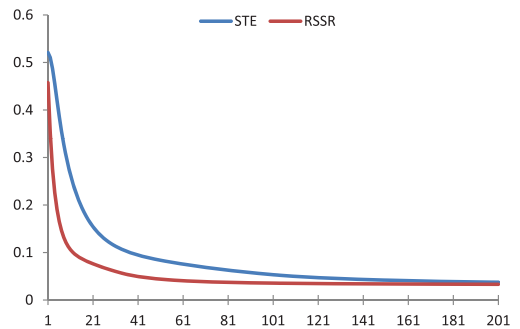
(a)



(b)



(c)



(d)

FIGURE 4. RMSE versus number of iterations for different models. (a) gr2000. (b) dm2006. (c) dm2000. (d) gr2006.

munity specific weights in [10]; (5) Peer-to-Peer: inferred network with peer-to-peer weights in [1].

The evaluation metric is RMSE. As shown in Tab. 5, the inferred network in our model generates RMSE performance which is significantly better than other inferred

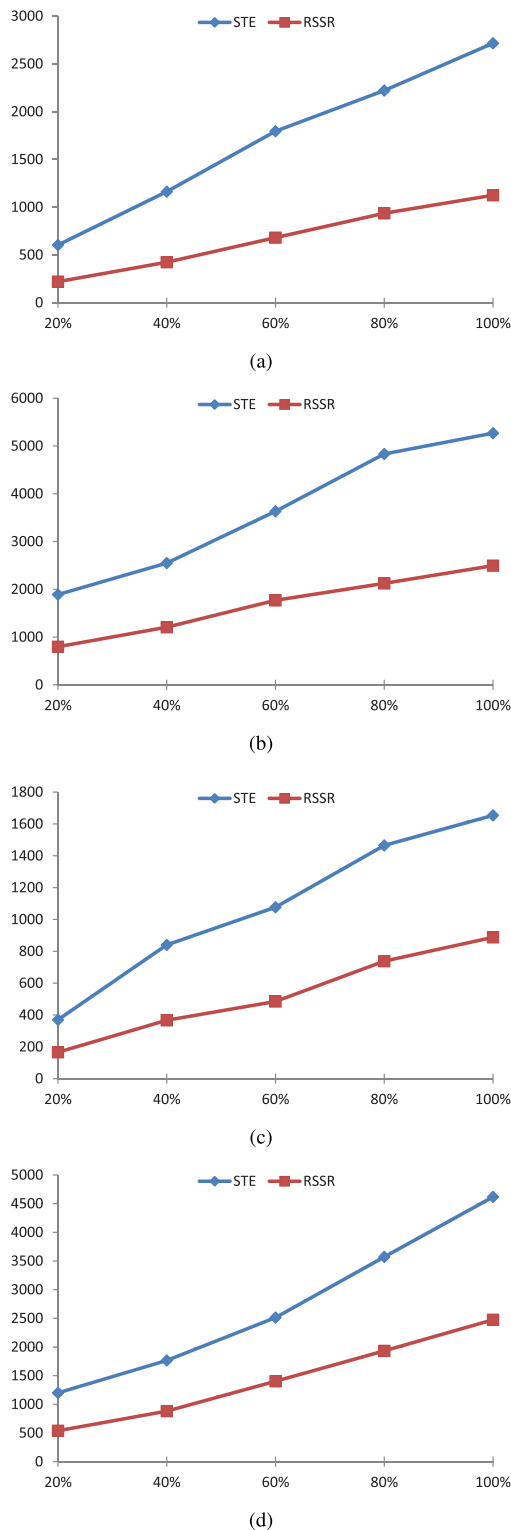


FIGURE 5. Number of iterations for convergence versus the size of data set. (a) gr2000. (b) dm2006. (c) gr2000. (d) gr2006.

networks. It produces best results on cold-start users. The best results overall are obtained by a heuristic weighted undirected network. But our model achieve comparable results. We would like to point out here that the best performance is obtained on a “full” network, while our performance is

achieved on a “skeleton” network with remarkably reduced storage costs.

F. EFFICIENCY

The efficiency study is implemented on domain specific networks to eliminate possible noise brought by cross domain co-authorships. We first plot the change of RMSE on training set at each iteration. The comparative method is STE [14], which also models rating as a combination of user’s own preference and influencers’ preferences. Unlike us, STE uses peer-to-peer influence. From Fig. 4, we can see that our model converges faster than STE on all domains. It suggests that, despite of the reduced storage costs and computation costs at each iteration, our model is more efficient in term of the total running time.

We shrink the set of publication on each domain to 20%, 40%, 60%, 80% and plot the number of iterations required to achieve the difference between $RMSE < 0.01$ in Fig. 5. We can see that the number of iterations demanded to converge increases as the size of data sets increases. The number for STE increases at a faster rate, while that for our model increases linearly at a lower rate. It demonstrates the scalability of our model for large data sets.

VI. CONCLUSION

Role theory is an important perspective in sociology, which considers human behavior to be the acting out of social roles. Though valid and intuitive in explaining the formation of cultural and social norms, its effectiveness in recommender systems is undetermined. In this contribution, we propose to incorporate social roles into the matrix factorization framework for recommendation. We first define the problem of social role detection and present an incremental clustering algorithm. Most previous social recommenders adopt a two-step approach to augment recommendations. Such a two-step approach can not take full advantage of the user’s preference information. We next model the user preference at each timestamp being a combination of users’ own previous preferences and influences’ previous preferences. We present a single-step algorithms to simultaneously infer preferences and influence strength, based on the assumption that social role is static within a time window. Comprehensive experiments on real data sets demonstrate the efficiency and effectiveness of our model to generate precise recommendations. In the future, it is worthy to investigate the dynamic nature of social roles. Also, we would like to model multiple roles and infer the semantics of social roles for better interpretability.

REFERENCES

- [1] C.-M. A. Yeung and T. Iwata, “Strength of social influence in trust networks in product review sites,” in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, 2011, pp. 495–504.
- [2] B. J. Biddle, “Recent development in role theory,” *Annu. Rev. Sociol.*, vol. 12, no. 1, pp. 67–92, 1986.
- [3] L. Charlin, R. Ranganath, J. McInerney, and D. M. Blei, “Dynamic Poisson factorization,” in *Proc. 9th ACM Conf. Rec. Syst.*, 2015, pp. 155–162.

- [4] W. Chen, W. Hsu, and M. L. Lee, "Modeling user's receptiveness over time for recommendation," in *Proc. 36th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2013, pp. 373–382.
- [5] K. J. Duff, *THINK Social Psychology*. Boston, MA, USA: Allyn & Bacon, 2012.
- [6] R. Forsati, I. Barjasteh, F. Masrouf, A.-H. Esfahanian, and H. Radha, "PushTrust: An efficient recommendation algorithm by leveraging trust and distrust relations," in *Proc. 9th ACM Conf. Rec. Syst.*, 2015, pp. 51–58.
- [7] R. Forsati, M. Mahdavi, M. Shamsfard, and M. Sarwat, "Matrix factorization with explicit trust and distrust side information for improved social recommendation," *ACM Trans. Inf. Syst.*, vol. 32, no. 4, p. 17, Oct. 2014.
- [8] M. Jamali and M. Ester, "TrustWalker: A random walk model for combining trust-based and item-based recommendation," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 397–406.
- [9] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in social networks," in *Proc. 4th ACM Conf. Rec. Syst.*, 2010, pp. 135–142.
- [10] M. Jamali, T. Huang, and M. Ester, "A generalized stochastic block model for recommendation in social rating networks," in *Proc. 5th ACM Conf. Rec. Syst.*, 2011, pp. 53–60.
- [11] Y. Koren, "Collaborative filtering with temporal dynamics," in *Proc. 15th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2009, pp. 447–456.
- [12] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *IEEE Comput.*, vol. 42, no. 8, pp. 30–37, Aug. 2009.
- [13] A. Krohn-Grimberghe, L. Drumond, C. Freudenthaler, and L. Schmidt-Thieme, "Multi-relational matrix factorization using Bayesian personalized ranking for social network data," in *Proc. 5th ACM Int. Conf. Web Search Data Mining*, 2012, pp. 173–182.
- [14] H. Ma, I. King, and M. R. Lyu, "Learning to recommend with social trust ensemble," in *Proc. 32th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2009, pp. 203–210.
- [15] H. Ma, M. R. Lyu, and I. King, "Learning to recommend with trust and distrust relationships," in *Proc. 3rd ACM Conf. Rec. Syst.*, 2009, pp. 189–196.
- [16] H. Ma, H. Yang, M. R. Lyu, and I. King, "SoRec: Social recommendation using probabilistic matrix factorization," in *Proc. 17th ACM Conf. Inf. Knowl. Manage.*, 2008, pp. 931–940.
- [17] 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 Mining*, 2011, pp. 287–296.
- [18] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1257–1264.
- [19] L. Xiang *et al.*, "Temporal recommendation on graphs via long- and short-term preference fusion," in *Proc. 16th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2010, pp. 723–732.
- [20] B. Yang, Y. Lei, D. Liu, and J. Liu, "Social collaborative filtering by trust," in *Proc. 23rd Int. Joint Conf. Artif. Intell.*, 2013, pp. 2747–2753.
- [21] W. Yao, J. He, G. Huang, and Y. Zhang, "Modeling dual role preferences for trust-aware recommendation," in *Proc. 37th Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 2014, pp. 975–978.
- [22] Q. Yuan, L. Chen, and S. Zhao, "Factorization vs. regularization: Fusing heterogeneous social relationships in top-n recommendation," in *Proc. 5th ACM Conf. Rec. Syst.*, 2011, pp. 245–252.
- [23] J. Zhang, C. Wang, J. Wang, and J. X. Yu, "Inferring continuous dynamic social influence and personal preference for temporal behavior prediction," *Proceedings VLDB Endowment*, vol. 8, no. 3, pp. 269–280, 2014.
- [24] J. Tang, J. Zhang, L. Yao, J. Li, L. Zhang, and Z. Su, "ArnetMiner: Extraction and mining of academic social networks," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 990–998.
- [25] R. Salakhutdinov and A. Mnih, "Bayesian probabilistic matrix factorization using Markov chain Monte Carlo," in *Proc. 25th Int. Conf. Mach. Learn.*, 2008, pp. 880–887.
- [26] Y. Zhao, G. Wang, P. S. Yu, S. Liu, and S. Zhang, "Inferring social roles and statuses in social networks," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2013, pp. 695–703.
- [27] X. Zhou *et al.*, "Social stream organization based on user role analysis for participatory information recommendation," in *Proc. Ubi-Media Comput. Workshops (UMEDIA)*, 2014, pp. 105–110.
- [28] B. Wu, X. Zhou, and Q. Jin, "Participatory information search and recommendation based on social roles and networks," *Multimedia Tools Appl.*, vol. 74, no. 14, pp. 5173–5188, 2015.
- [29] Y. Wang and V. Varadarajan, "Role-based recommendation and trust evaluation," in *Proc. E-Commerce Technol., 4th IEEE Int. Conf. Enterprise Comput., E-Commerce, E-Services*, Jul. 2007, pp. 278–288.
- [30] S. Huang, J. Zhang, L. Wang, and X.-S. Hua, "Social friend recommendation based on multiple network correlation," *IEEE Trans. Multimedia*, vol. 18, no. 2, pp. 287–299, Feb. 2016.
- [31] A. Mislove, B. Viswanath, K. P. Gummadi, and P. Druschel, "You are who you know: Inferring user profiles in online social networks," in *Proc. 3rd ACM Int. Conf. Web Search Data Mining*, 2010, pp. 251–260.
- [32] K. Henderson *et al.*, "RoIX: Structural role extraction & mining in large graphs," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 1231–1239.
- [33] Y. Chen and X. Ye. (Feb. 2011). "Projection onto a simplex." [Online]. Available: <https://arxiv.org/abs/1101.6081>
- [34] G. Guo *et al.*, "LibRec: A java library for recommender systems," in *Proc. UMAP Workshops*, vol. 4. 2015, pp. 1–4.



DUGANG LIU was born in Quanzhou, Fujian, China in 1993. He received the B.S. degree in information and computing science from Fujian Normal University, Fuzhou, China, in 2016. He is currently pursuing the M.S. degree with the School of Information Science and Technology, Xiamen University, China. His research interests include recommender systems and natural language processing.



JIE HUANG was born in Putian, Fujian, China, in 1991. He received the B.S. degree in information and computing science from Xiamen University, Xiamen, China, in 2014, where he is currently pursuing the M.S. degree with the School of Information Science and Technology. His research interests include recommender systems.



CHEN LIN (M'18) was born in Xiamen, Fujian, China, in 1982. She received the B.S. and Ph.D. degrees in computer science from Fudan University, China, in 2004 and 2010, respectively.

From 2010 to 2013, she was an Assistant Professor with the Department of Computer Science, Xiamen University, China. Since 2013, she has been an Associate Professor with the School of Information Science and Technology, Xiamen University. She was a Visiting Researcher with Simon Fraser University, Canada, from 2015 to 2016. Her research interests include text mining, recommender system, and social network analysis.