A Community-based Collaborative Filtering Method for Social Recommender Systems

Bin Liang*§, Bo Xu†, Xiaowei Wu*, Dong Wu*, Deqing Yang‡§, Yanghua Xiao*§¶, and Wei Wang*§

*School of Computer Science, Fudan University, Shanghai, China

†School of Computer Science and Technology, Donghua University, Shanghai, China

‡School of Data Science, Fudan University, Shanghai, China

§Shanghai Key Laboratory of Data Science, Shanghai, China

¶Shanghai Insitute of Intelligent Electroics & Systems, Shanghai, China

liangbin@fudan.edu.cn, xubo@dhu.edu.cn, {14212010020, dongwu14, yangdeqing, shawyh, weiwang1}@fudan.edu.cn

***Abstract*—Recommender systems have become indispensable for recommending items of interest to users and have been successfully deployed in a wide range of real-world applications. In this paper, we exploit the community influence of users in social networks to improve the recommendation accuracy. Depending on the community in which the user is located, the user's preferences are defined more accurately. Specifically, we propose a community-based collaborative filtering method for social recommender systems, which makes full use of the rich link/community structure within a social network. We first group users in a social network into overlapping communities. Then we explicitly incorporate the community preference into the latent factor model. Compared with seven state-of-the-art methods on four real-world datasets, our method achieves the best performance.**

I. Introduction

Recommender systems have become indispensable information filtering system for recommending items of interest to users and have been successfully deployed in a wide range of real-world applications such as movie recommendations, product recommendations and music recommendations.

One method widely used in recommender systems is collaborative filtering, which is based on the historical user's behavior (e.g., purchase records or ratings) as well as similar decisions made by other users and predict items (or ratings for items) that the user may have an interest in. However, in practice, due to the cold start problem, the user-item matrix used for collaborative filtering could be extremely large and sparse, which results in poor recommendation.

In this paper, we exploit the community influence of users in social networks to improve the recommendation accuracy. We argue that *individual user preferences are complex and diverse, but community preferences are simpler depending on the community in which the user is located, the user's preferences can be defined more accurately*. For example in Figure 1, different communities like different types of movies. User u belongs to community c_1 and c_2, so he/she tends to like action and war movies, while user v belongs to community c_3 and c_4, so he/she tends to like drama and romantic movies. Although user u and user v are friends, they don't share the same preferences.

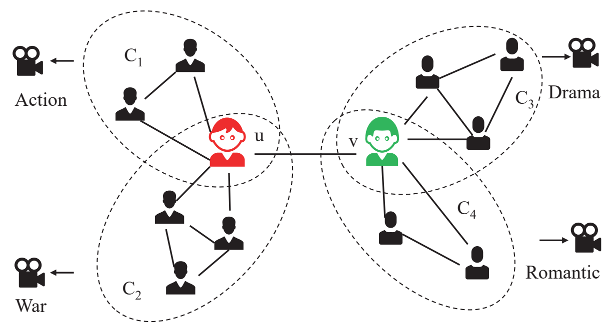

Figure 1. An example of community influence of users in social networks. There are four communities (c_1, c_2, c_3, c_4) in this social network, and different communities like different types of movies (Action, War, Drama and Romantic). User u and v are friends, but they don't share the same preferences.

The main contributions of our paper are summarized as follows:

- Firstly, we exploit the community influence of users for social recommender systems to tackle the data sparsity problem.
- Secondly, we propose a novel recommendation framework ConSVD, which explicitly incorporates the community preference into a latent factor model rather than as a regularization term into the objective function.
- Finally, compared with seven state-of-the-art methods on four real-world datasets, our method achieves the best performance.

II. Method

In this section, we first formulate our problem, and then introduce the framework of our system: ConSVD.

† Bo Xu is corresponding author.

A. Problem Definition

The task of a social recommender system is to predict an unknown rating for users on items based on the user-item ratings and user friendships.

Specifically, let $U = \{u_1, u_2, \cdots, u_n\}$ be the set of users, $I = \{i_1, i_2, \cdots, i_m\}$ be the set of items, $R = [r_{ui}]_{n \times m}$ be the rating matrix between users and items, $G = (U, E)$ be the social network graph, where each node $u \in U$ represents an user and an edge $e \in E$ represents the social relationships between users. Our goal is to predict an unknown rating r_{ui} for user u on item i based on the user-item ratings R and user social network graph G.

B. Framework

We introduce our proposed method to predict user ratings by taking users' community influence into account. The goal of our work is to gain the rich link/community structure information within a social network. There are already many related studies. Our approach is motivated by studies in social influence and conformity theory [1], which suggest that when faced with conformity pressure, individuals tend to adjust their opinions, choices or preferences to obtain social approval and belonging from others in a group.

Our proposed method, called ConSVD, consists of two stages. In the first stage, we run the overlapping community detection algorithm BigCLAM on the social network G, which is based on Nonnegative Matrix Factorization. This algorithm is designed to return C overlapping communities, so that we can obtain the propensity of user u in community c: $\alpha_{uc} \geq 0$.

Next, for each community $c = 1, \ldots, C$, we assume that there is a latent community preference vector $\tilde{p}_c \in R^d$ that represents the overall preference pattern of community c. Since community influence is a group phenomenon, we model each u's feature vector as $p_u + \sum_{c:u \in c} \alpha_{uc}\tilde{p}_c$ by considering external group influence and the degree of conformity to different groups. It is essential that the inherent user interest p_u adjusted by a weighted average of community preference vectors, where the weight α_{uc} is used to measure the conformity of user u to community c. By incorporating such community influence phenomenon into the latent factor model, we obtain the following prediction rule of the rating for user u on item i:

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T(p_u + \sum_{c:u \in c} \alpha_{uc}\tilde{p}_c). \tag{1}$$

Although both SoDimRec and ConSVD make use of overlapping community structure in a network, the key difference is that we explicitly incorporate the community preference into the prediction rule (1) rather than a regularization term into the objective function.

We believe our approach is more intuitive and interpretable. Compared with TrustSVD [4] where each user has two latent feature vectors, the number of parameters of our ConSVD is pruned thanks to the fact that the number of communities is much less than the number of users.

The model parameters $\Theta = \{b_u, b_i, p_u, \tilde{p}_c, q_i\}$ are estimated by minimizing the following regularized objective function:

$$\begin{aligned}\mathcal{L} = & \sum_{(u,i) \in R} (r_{ui} - \hat{r}_{ui})^2 + \lambda(\sum_u b_u^2 + \sum_i b_i^2 \\ & + \sum_u ||p_u||^2 + \sum_{c \in C} ||\tilde{p}_c||^2 + \sum_i ||q_i||^2)\end{aligned} \tag{2}$$

where $\hat{r}_{ui}$ is defined in equation (1), and λ is a parameter to control model complexity and to avoid overfitting.

III. Experiments

In this section, we evaluate the effectiveness of the proposed ConSVD framework using four real-world data sets. The experiments are set up as the following.

A. Data Sets and Evaluation Metrics

Four publicly available data sets are used in our experiments: Filmtrust [3], Epinions [9], Ciao [2] and Flixster [5].

All four data sets contain both ratings and social links. Although the social links are directed in Epinions, Filmtrust and Ciao, we treat all these links as undirected when performing overlapping community detection. The statistics of four data sets are shown in Table I.

We use 5-fold cross-validation for training and testing. Each data set is randomly divided into five folds. In each round, four folds are used as training set and the remaining fold is used as testing set. The average of the five rounds is reported as the final result.

Two well-known metrics, Mean Absolute Error (MAE) and Root Mean Square Error (RMSE), are used to evaluate prediction performance. They are defined as follows: $MAE = \frac{\sum_{u,i} |r_{ui} - \hat{r}_{ui}|}{N}, RMSE = \sqrt{\frac{\sum_{u,i} (r_{ui} - \hat{r}_{ui})^2}{N}}$

where N is the number of ratings in the test set.r_{ui} denotes the observed rating in the testing data, and $\hat{r}_{ui}$ is the predicted rating. RMSE increases the penalty for inaccurate item ratings (the penalty for squaring items), thus making the evaluation of the system more critical.

Table I
STATISTICS OF THE DATA SETS.

Features	Filmtrust	Epinions	Ciao	Flixster
#Users	1,508	40,163	7,375	47,109
#Items	2,071	139,738	105,114	27,457
#Ratings	35,497	664,824	282,650	1,675,545
Density$_1$	1.140%	0.012%	0.037%	0.130%
#Users	1,642	40,163	7,375	47,109
#Trusts	1,853	442,979	111,781	839,277
Density$_2$	0.069%	0.028%	0.206%	0.038%

Table II
PERFORMANCE COMPARISON IN ALL VIEW

Methods	d	Metrics	PMF	SoRec	SoReg	TrustMF	TrustSVD	MFC$_b$	SoDimRec	ConSVD
Filmtrust	5	RMSE	0.944	0.821	0.886	0.810	h**0.791**	0.869	0.859	0.797
		MAE	0.698	0.658	0.681	0.630	0.611	0.670	0.661	**0.607**
	10	RMSE	0.945	0.821	0.864	0.817	**0.789**	0.838	0.839	0.793
		MAE	0.700	0.621	0.662	0.627	0.609	0.662	0.652	**0.605**
	20	RMSE	0.950	0.830	0.863	0.865	0.792	0.835	0.833	**0.791**
		MAE	0.705	0.655	0.664	0.648	0.615	0.640	0.639	**0.607**
Epinions	5	RMSE	1.624	1.131	1.371	1.082	1.054	1.280	1.297	**1.044**
		MAE	1.186	0.927	1.070	0.829	0.809	0.994	1.013	**0.796**
	10	RMSE	1.588	1.143	1.176	1.125	1.053	1.206	1.155	**1.043**
		MAE	1.174	0.912	0.915	0.836	0.807	0.951	0.916	**0.795**
	20	RMSE	1.483	1.222	1.120	1.302	1.112	1.162	1.094	**1.044**
		MAE	1.121	0.917	0.840	0.883	0.810	0.894	0.842	**0.795**
Ciao	5	RMSE	1.882	0.993	1.481	0.977	0.956	1.228	1.249	**0.950**
		MAE	1.395	0.768	1.135	0.749	0.723	0.959	0.972	**0.716**
	10	RMSE	1.751	1.010	1.194	0.999	0.957	1.073	1.092	**0.949**
		MAE	1.323	0.768	0.909	0.750	0.723	0.836	0.852	**0.715**
	20	RMSE	1.530	1.077	1.097	1.111	0.962	1.010	1.027	**0.948**
		MAE	1.187	0.778	0.834	0.767	0.732	0.768	0.784	**0.714**
Flixster	5	RMSE	0.969	0.899	0.966	0.982	0.893	0.957	0.984	**0.877**
		MAE	0.708	0.676	0.725	0.775	0.655	0.737	0.750	**0.653**
	10	RMSE	0.955	0.908	0.927	1.027	0.908	0.939	0.955	**0.876**
		MAE	0.702	0.683	0.698	0.806	0.666	0.723	0.732	**0.652**
	20	RMSE	0.939	0.907	1.035	1.181	0.938	0.916	0.944	**0.875**
		MAE	0.697	0.683	0.755	0.914	0.688	0.694	0.719	**0.651**

Table III
PERFORMANCE COMPARISON IN COLD-START VIEW

Methods	d	Metrics	PMF	SoRec	SoReg	TrustMF	TrustSVD	MFC$_b$	SoDimRec	ConSVD
Filmtrust	5	RMSE	1.698	0.896	1.189	0.901	0.888	1.238	1.198	**0.886**
		MAE	1.237	0.712	0.978	0.700	0.685	1.021	0.936	**0.676**
	10	RMSE	1.442	0.889	1.086	0.916	0.893	0.983	1.024	**0.912**
		MAE	1.129	**0.661**	0.831	0.691	0.687	0.810	0.808	0.682
	20	RMSE	1.384	0.941	1.060	0.976	0.894	0.906	0.903	**0.912**
		MAE	1.084	**0.657**	0.727	0.732	0.704	0.597	0.641	0.682
Epinions	5	RMSE	2.214	1.163	2.148	1.137	1.116	1.673	1.888	**1.110**
		MAE	1.778	0.937	1.897	0.902	0.878	1.363	1.615	**0.855**
	10	RMSE	2.233	1.188	1.535	1.182	1.119	1.377	1.491	**1.111**
		MAE	1.827	0.872	1.312	0.855	0.880	1.066	1.272	**0.853**
	20	RMSE	2.157	1.310	1.233	1.357	1.120	1.276	1.217	**1.106**
		MAE	1.787	0.856	0.862	0.861	0.886	0.841	0.862	**0.849**
Ciao	5	RMSE	2.089	1.030	2.799	1.042	0.979	2.118	1.995	**0.978**
		MAE	1.797	0.842	2.610	0.854	0.724	1.927	1.795	**0.701**
	10	RMSE	2.197	0.991	2.336	1.045	0.986	1.492	1.681	**0.989**
		MAE	1.948	0.715	2.104	0.739	0.726	1.311	1.489	**0.706**
	20	RMSE	2.090	1.220	1.169	1.226	0.980	1.154	1.137	**0.977**
		MAE	1.878	0.781	0.949	0.724	0.726	0.841	0.736	**0.706**
Flixster	5	RMSE	1.972	1.152	1.504	1.159	1.169	1.411	1.647	**1.055**
		MAE	1.473	0.913	1.258	0.941	0.949	1.114	1.404	**0.819**
	10	RMSE	1.858	1.167	1.286	1.165	1.147	1.318	1.363	**1.056**
		MAE	1.396	0.910	1.054	0.925	0.897	1.013	1.334	**0.817**
	20	RMSE	1.697	1.201	1.275	1.278	1.199	1.247	1.241	**1.055**
		MAE	1.293	0.932	1.011	0.990	0.938	0.956	0.972	**0.815**

B. Competing Methods

To evaluate the performance of **ConSVD**, we compare our methods with the following baseline methods: **PMF** [10], **SoRec** [7], **SoReg** [8], **TrustMF** [12], **TrustSVD** [4], **MFC$_b$** [6], **SoDimRec** [11].

We use the implementations of SoRec, SoReg, TrustMF, and TrustSVD provided by LibRec[1]. The codes of the SoDimRec and MFC$_b$ are not publicly available, so we use our implementations.

We choose the optimal parameter settings according to either previous work or experiments. Specifically, we set the dimension of latent factors d to 5, 10, and 20.

Both *All* and *Cold Start* views are created for testing. All ratings are used as the test set in the *All* view, while only the users who rate less than five ratings are used as the test set in the *Cold Start* view. The performance measures are

[1]http://www.librec.net

shown in Table II and Table III, respectively.

Table II (in the testing view of *All*) shows that ConSVD outperforms all competing methods across almost all the settings except on FilmTrust when RMSE@$d = 5$ and RMSE@ $d = 10$, for which TrustSVD performs best.

In the testing view of *Cold Start*, as shown in Table III, ConSVD achieves almost best performance for most settings. In some cases, SoRec performs best in terms of MAE (on FilmTrust when $d = 10$ and 20). Like our models, both MFC$_b$ and SocDimRec leverage the community structure when building the recommendation models. However, both MFC$_b$ and SocDimRec use the community's preference vectors in the regularization terms, that limits their prediction capacity. Through explicitly exploiting the community's preference vectors in the prediction rule, the improved performance of ConSVD demonstrates the advantage of our designs.

In general, SoRec, TrustMF, and TrustSVD achieve good performance when $d = 5$. However, their prediction errors in terms of RMSE grow when the latent dimension d increases, which indicates that these methods could become overfitting with more parameters. In contrast, the performance of ConSVD is more robust with different latent dimensions.

Obviously, our method is better than other benchmark methods in practical effect and stability.

IV. Conclusion

In this paper, we exploit the community influence of users in social networks to improve the recommendation accuracy. Specifically, we propose a community-based collaborative filtering method for social recommender systems. We first group users in a social network into overlapping communities. Then we explicitly incorporate the community preference into the latent factor model. Compared with seven state-of-the-art methods on four real-world datasets, our method achieves the best performance.

Acknowledgment

This paper was sponsored by National Key R&D Program of China (2016YFC0801003), by Shanghai Sailing Program No.19YF1402300, by the Initial Research Funds for Young Teachers of Donghua University (No. 112-07-0053019), by National Key R&D Program of China (No.2017YFC1201200 and 2017YFC0803700), by National Key Basic Research Program of China under No.2015CB358800, by the National NSFC(No.61472085, No.U1509213,No.U1636207), by Shanghai Municipal Science and Technology Commission foundation key project under No.15JC1400900, by Shanghai Municipal Science and Technology project(No.16511102102, No.16JC1420401), Shanghai STCSM's R&D Program under Grant(No.16JC1420400)

References

[1] R. B. Cialdini and N. J. Goldstein. Social influence: Compliance and conformity. *Annual Review of Psychology*, 55:591–621, 2004.

[2] Guibing Guo, Jie Zhang, Daniel Thalmann, and Neil Yorke-Smith. Etaf: An extended trust antecedents framework for trust prediction. In *Proceedings of the 2014 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, pages 540–547. IEEE Press, 2014.

[3] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. A novel bayesian similarity measure for recommender systems. In *Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

[4] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence, January 25-30, 2015, Austin, Texas, USA.*, pages 123–129, 2015.

[5] Mohsen Jamali and Martin Ester. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 135–142. ACM, 2010.

[6] Hui Li, Dingming Wu, Wenbin Tang, and Nikos Mamoulis. Overlapping community regularization for rating prediction in social recommender systems. In *Proceedings of the 9th ACM Conference on Recommender Systems, RecSys 2015, Vienna, Austria, September 16-20, 2015*, pages 27–34, 2015.

[7] Hao Ma, Haixuan Yang, Michael R. Lyu, and Irwin King. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM 2008, Napa Valley, California, USA, October 26-30, 2008*, pages 931–940, 2008.

[8] Hao Ma, Dengyong Zhou, Chao Liu, Michael R. Lyu, and Irwin King. Recommender systems with social regularization. In *Proceedings of the Forth International Conference on Web Search and Web Data Mining, WSDM 2011, Hong Kong, China, February 9-12, 2011*, pages 287–296, 2011.

[9] Paolo Massa and Paolo Avesani. Controversial users demand local trust metrics: An experimental study on epinions. com community. In *AAAI*, pages 121–126, 2005.

[10] R. Salakhutdinov and A. Mnih. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2008.

[11] Jiliang Tang, Suhang Wang, Xia Hu, Dawei Yin, Yingzhou Bi, Yi Chang, and Huan Liu. Recommendation with social dimensions. In *AAAI*, pages 251–257, 2016.

[12] Bo Yang, Yu Lei, Dayou Liu, and Jiming Liu. Social collaborative filtering by trust. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, pages 2747–2753, 2013.