Integrating User Embedding and Collaborative Filtering for Social Recommendations

Junliang Yu[1,2], Min Gao[1,2](✉), Yuqi Song[1,2], Qianqi Fang[1,2],
Wenge Rong[3], and Qingyu Xiong[1,2]

[1] School of Software Engineering, Chongqing University, Chongqing 400044, China
{yu.jl,gaomin,songyq,fqq0429,xiong03}@cqu.edu.cn
[2] Key Laboratory of Dependable Service Computing in Cyber Physical Society,
Chongqing University, Ministry of Education, Chongqing 400044, China
[3] School of Computer Science and Engineering, Beihang University,
Beijing 100191, China
w.rong@buaa.edu.cn

Abstract. Social recommendation has attracted increasing attention over the years due to the potential value of social relations, which can be harnessed to mitigate the dilemma of data sparsity in traditional recommender systems. However, recent studies show that social recommenders fail in the practical use in industry for the reason that some problems in social relations, such as the noise, lead to a degradation in recommendation quality. To solve the problem, in this paper, a social recommender, SocialEM, which integrates the neural user embedding and collaborative filtering is proposed. Enlightened by the factorization of the word co-occurrence matrix which is equivalent to the skip-gram model in word2vec, SocialEM will jointly decomposes the user-item rating matrix and the user-user co-occurrence matrix with shared user latent factors. For each pair of users, the co-occurrence matrix encodes the number of being trusted together by other users in the social relation network. Experiments conducted on the real-world datasets have shown that the side effect of social relations can be diminished by tuning parameters for SocialEM. And compared with previous studies, our method significantly improves the quality of recommendations.

Keywords: Social recommendations · User embedding
Collaborative filtering · Matrix factorization

1 Introduction

Nowadays, the recommender system is playing an important role in information filtering. The basic technique of recommender systems is collaborative filtering, which depends on the collective historical ratings to predict items that will be positively rated by the active user [10]. However, traditional rating-based only

I. Romdhani et al. (Eds.): CollaborateCom 2017, LNICST 252, pp. 470–479, 2018.
https://doi.org/10.1007/978-3-030-00916-8_44

recommenders may suffer from the problem of data sparsity since most users rate few of the millions of available items. With the prosperity of social platforms, abundant social information can be harnessed to mitigate the problem, for the reason that the users' preferences can be inferred from those of their friends [18]. Based on this startup idea, social recommender systems [15] emerged and have attracted increasing attention over the years. Nevertheless, recent reports [1,2] show that social recommenders fail in the practical use in industry. Key findings from negative experiences in applying social recommender systems include: (1) social relations are too noisy and may have a negative impact; (2) social relations have different interpretations in different contexts [20]. Therefore, utilizing the explicit social relations without further processing will lead to a degradation in recommendation quality.

To diminish the aforementioned negative influence, the ways which can reveal the implicit nature of the social relations should be explored. Most existing social recommenders are based on matrix factorization [7], a CF model which usually decomposes the observed user-item rating matrix into user and item latent factors. But almost all of them utilize the explicit social relations directly. Enlightened by the recent success of word embedding models such as word2vec [16], we consider that users in recommender systems can be embedded likewise. The prior work [8] proves that the skip-gram model of word2vec is equivalent to the implicit factorization for the word co-occurrence matrix. Based on this work and CF, our model, SocialEM is proposed.

In the model of SocialEM, the user-item rating matrix and the user-user co-occurrence matrix are jointly decomposed with shared user latent factors. For each pair of users, the co-occurrence matrix encodes the number of being trusted together by other users in the social relation network. With the help of the user co-occurrence embedding, the recommender can further expand its model capacity and eliminate the side effect of explicit social relations. Firstly, learned with a context, the embeddings of users can uncover more interactions among users. Secondly, by tuning the minimal co-occurrence counts, users with few attention will be regarded as the anomalies which should be filtered. The experiments conducted on real-world datasets show that SocialEM is superior to state-of-the-art methods in terms of recommendation accuracy. At last, we also provide exploratory analysis to better understand the effectiveness of SocialEM.

The remainder of this paper is organised as follows. In Sect. 2, the preliminaries are briefly introduced. Section 3 will focus on the proposed method - SocialEM. Section 4 reports the experimental results. In Sect. 5, the related studies are introduced. Finally, in Sect. 6 we conclude this paper and point out some potential future work.

2 Preliminaries

In this section, we will briefly introduce matrix factorization and word embedding, which are the pillars of the SocialEM model.

Matrix Factorization. Generally, most model-based CF methods are based on matrix factorization [7], which maps both users and items into a low-dimensional latent-factor space such that user-item interactions are modeled as inner products in that space and user and item latent factors are denoted by $p_u \in \mathbb{R}^k$ and $q_i \in \mathbb{R}^k$, respectively. The resulting dot product, $p_u^T q_i$, can capture the interaction between user u and item i, which is used to predict the missing ratings.

Let I_{ui} be an indicator of whether a rating has been observed in the user-item rating matrix R,

$$I_{ui} = \begin{cases} 1 \; if \; r_{ui} \text{ is observed} \\ 0 \; if \; r_{ui} \text{ is missing} \end{cases} \tag{1}$$

Matrix factorization optimizes the following objective:

$$\mathcal{L} = \frac{1}{2} \sum_{u,i} I_{ui}(r_{ui} - p_u^T q_i)^2 + \lambda(\|p_u\|^2 + \|q_i\|^2), \tag{2}$$

where r_{ui} denotes the rating expressed by user u on item i and the algorithmic parameter λ controls the magnitudes of the latent factors.

Word Embedding. Word embedding models [5,6,16] have caught much attention in a variety of NLP tasks recently for its extraordinary performance. The skip-gram model trained with negative sampling value of k in word2vec [16] has been proven to be equivalent to the implicit factorization for the point-wise mutual information (PMI) matrix shifted by k [8]. PMI between a word w and its context word c can be empirically estimated as:

$$PMI(i,j) = \log \frac{\#(w,c) \cdot |D|}{\#(w) \cdot \#(c)}, \tag{3}$$

where $\#(w,c)$ is the number of times word w appears in the context of word c, $\#(w) = \sum_c \#(w,c)$, $\#(c) = \sum_w \#(w,c)$, and $|D|$ is the total number of the pairs in the corpora.

When PMI has been constructed, the distributed representation of word w is obtained by factorizing the shifted positive point-wise mutual information matrix (SPPMI) which is defined as:

$$SPPMI_k(w,c) = \max(PMI(w,c) - \log k, 0), \tag{4}$$

where k is the number of the negative sampling that controls the sparsity of the SPPMI matrix here.

3 SocialEM: Integrating User Embedding and CF for Social Recommendations

In this section, we propose a **S**ocial recommender that combines user **E**mbedding and **M**atrix factorization, called **SocialEM**.

User Embedding. In word2vec, given a center word, the sequence of words surrounded are defined as the context of the center word. Likewise, in social recommender systems we can define the context of a user u as other users who are trusted together with u by a same user. For example, both u_1 and u_2 are trusted by u_3, therefore, u_1 and u_2 are contexts of each other. Then, the user co-occurrence SPPMI matrix $T \in \mathbb{R}^{m\times m}$ is constructed by computing $\#(i,j)$ that denotes the number of times both user i and user j are trusted by other users. After that, we can obtain user embedding by factorizing T. In contrast to the explicit trust or relation, co-occurrences can express the trust degree or affection in a finer grained view. Factorizing SPPMI matrix further reveals the implicit interactions among users in the social network. In addition, by tuning the value of k, noises in the social relations can be neglected whereas available connections are preserved.

Training Procedure. SocialEM has two main components: factorization for the user-item rating matrix and the user-user co-occurrence matrix. To fusing both the rating and social information into the latent factors, we jointly decompose the two matrices with a shared user latent space. Factorizing the rating matrix encodes users' preferences and items' characteristics; factorizing the co-occurrence matrix can extract the implicit connection pattern among the user network. Sharing a same user latent space ensures that both preferences and the pattern are included in the user latent factors. After training, low dimensional shared user latent factors $P \in \mathbb{R}^{m\times d}$, item latent factors $Q \in \mathbb{R}^{n\times d}$, and the context embedding $G \in \mathbb{R}^{m\times d}$ are obtained. Then, missing ratings can be predicted by the dot products of shared user latent factors and item latent factors.

Model Formulation. The objective function of SocialEM is stated as:

$$\mathcal{L} = \overbrace{\frac{1}{2}\sum_{u,i} I_{ui}(r_{ui} - P_u^T Q_i - b_u - b_i)^2}^{\text{Factorization for the rating matrix}} + \overbrace{\frac{\alpha}{2}\sum_{t_{uj}\neq 0}(t_{uj} - P_u^T G_j - w_u - c_j)^2}^{\text{Social user embedding}}$$
$$+ \overbrace{\frac{\lambda}{2}(\sum_{u}(P_u^2 + G_u^2 + b_u^2 + w_u^2 + c_u^2) + \sum_{i}(Q_i^2 + b_i^2))}^{\text{Regularization}}. \quad (5)$$

As can be seen in Eq. 5, the objective function has three parts. In the part of factorization for the rating matrix, the errors between observed ratings and predicted ratings are quadratically cumulated. b_u and b_i indicate the deviations of user u and item i. In the part of social user embedding, the user co-occurrence matrix is factorized. The user latent factors connect these two parts and account for both user-item interactions and user-user co-occurrence. w_u and c_j are the biases of the user and context. In the last part, all model parameters are regularized. Besides, to balance the effect of the rating information and social information, we introduce a hyper-parameter α. The impact of α will be investigated in Sect. 4.

In [8], several optimizers are available to identify good solutions for factorizing the word co-occurrence matrix. In our work, we use stochastic gradient descent because it works very efficiently in the case of redundant data. A local minimum of the objective function given by Eq. 5 can be found by performing gradient descent in all model parameters:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial b_u} &= \lambda b_u - e_{ui} & \frac{\partial \mathcal{L}}{\partial b_i} &= \lambda b_i - e_{ui} \\ \frac{\partial \mathcal{L}}{\partial b_u} &= \lambda w_u - e_{uj} & \frac{\partial \mathcal{L}}{\partial c_j} &= \lambda c_j - e_{uj} \\ \frac{\partial \mathcal{L}}{\partial Q_i} &= \lambda Q_i - e_{ui} P_u & \frac{\partial \mathcal{L}}{\partial G_j} &= \lambda G_j - e_{uj} P_u \\ \frac{\partial \mathcal{L}}{\partial P_u} &= \lambda P_u - e_{ui} Q_i - \alpha e_{uj} g_j \end{aligned} \tag{6}$$

where e_{ui} denotes the error between the observed rating and the predicted rating, e_{uj} denotes the error between co-occurrences and the dot products of user latent factors and context embedding, coupled with biases.

Complexity. With regard to complexity, compared with matrix factorization, SocialEM needs to learn only an additional context embedding matrix with $m \times d$ elements. Since d is a small number, which is consistent with the dimensionality of the latent factors of users and items, the cost is not be computationally expensive. However, SocialEM computes the user-user co-occurrence with a $O(n^2)$ complexity at the beginning, which is time-consuming. To save time, the user-user co-occurrence should be computed off-line. Fortunately, the computation for co-occurrence is new-user-friendly. When a new user enters, SPPMI merely has few entries which are related to the new user need to be updated.

4 Experimental Results

In this section, we present the experimental results. Two types of experiments are conducted: (1) the recommendation quality of the previous models and SocialEM is compared; (2) the impact of the parameters α and k is investigated.

Two common real-world datasets, Ciao and Epinions, are used in our experiments. The statistics of the datasets released by [15,19] are shown in Table 1. The mean absolute error (MAE) and root mean square error (RMSE) are chosen to measure the prediction error for all methods. A lower MAE or RMSE indicates that missing ratings are predicted more precisely.

Table 1. Dataset statistics

Dataset	#Users	#Items	#Ratings	#Relations	Rating scale
Ciao	7,375	105,114	284,086	111,781	1.0 – 5.0
Epinions	40,163	139,738	664,823	487,182	1.0 – 5.0

To demonstrate the performance improvement of SocialEM, we compare our method with the following methods: PMF [17], SoRec [13], SocialMF [4], and RSTE [11].

To tune the methods, we use 80% of the data as the training set, from which we randomly select 10% as the validation set. For the remaining 20% of the data, we consider the users and items that appear in the training and validation sets to obtain the test set. We record the best parameters of these methods according to their performance. Afterwards, all the experiments are performed with 5-fold cross validation (Table 2).

Table 2. The MAE and RMSE for all methods

	Ciao						
Metric	UserMean	ItemMean	PMF	SoRec	SocialMF	RSTE	**SocialEM**
MAE	0.780	0.767	0.772	0.760	0.753	0.757	**0.736**
Improve	5.64%	4.04%	4.66%	3.15%	2.25%	2.77%	
RMSE	1.030	1.034	1.029	1.020	1.013	1.022	**0.975**
Improve	5.33%	5.70%	5.24%	4.41%	3.75%	4.60%	
	Epinions						
Metric	UserMean	ItemMean	PMF	SoRec	SocialMF	RSTE	**SocialEM**
MAE	0.927	0.856	0.839	0.834	0.822	0.831	**0.811**
Improve	12.51%	5.25%	3.33%	2.75%	1.33%	2.40%	
RMSE	1.198	1.152	1.135	1.127	1.091	1.117	**1.077**
Improve	10.18%	6.41%	5.11%	4.44%	1.28%	3.58%	

4.1 Performance for Predicting Missing Ratings

In this section, we compare the performance of SocialEM with that of the baselines. We set the step size $\gamma = 0.05$, $\alpha = 0.5$ on Ciao and 1 on Epinions, and $k = 5$ for SocialEM. The dimensionality d of the latent factors is set to 20, and the regularization parameter λ is set to 0.01 for all the methods.

As can be seen in Table 1, SocialEM is superior to the corresponding methods which utilize explicit social relations directly, along with two baselines. The main findings are summarized as follows: (1) SocialEM beats PMF by a convincing margin, demonstrating the effectiveness of incorporating social relations; (2) SocialEM outperforms social-related methods by an average improvement of 2.44% on MAE, and 3.84% on RMSE, which proves that adopting user embeddings to reveal the implicit interactions among users is promising. In particular, both SocialEM and SoRec jointly decompose the user-item rating matrix and the relation matrix with a shared user latent space. The difference on performance further validates the positive effect of user embeddings.

4.2 Impact of Parameters α and k

In SocialEM, two hyper-parameters, α and k, are introduced. α is the trade-off used to balance the effect of ratings and relations; k controls the sparsity of the user co-occurrence matrix. First, we fix $k = 5$ and increase the value of η from 0 to 10 with different intervals to observe the variation in the performance. Then, we fix $\alpha = 0.1$ to observe the influence of changing k. The experiments are conducted on both datasets, and the other settings are the same as those in Sect. 4.2.

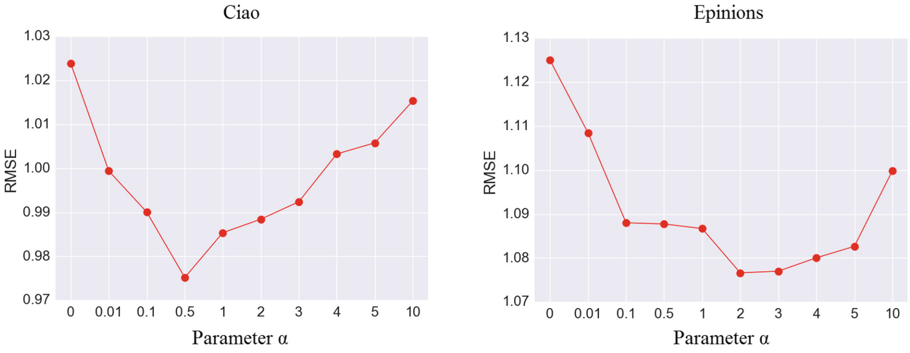

Fig. 1. Impact of parameter α

From Fig. 1, we can witness the fluctuation. Initially, SocialEM has poor performances when $\alpha = 0$, which illustrates the importance of considering social relations. Then the curves decline until that $\alpha = 0.5$ on Ciao and 2 on Epinions where the best performance is reached. And then the curves climb back to a poor level with the increase of α. The observed changes once again emphasize the effectiveness of social relations and also exhibit a proper range of the trade-off on different datasets.

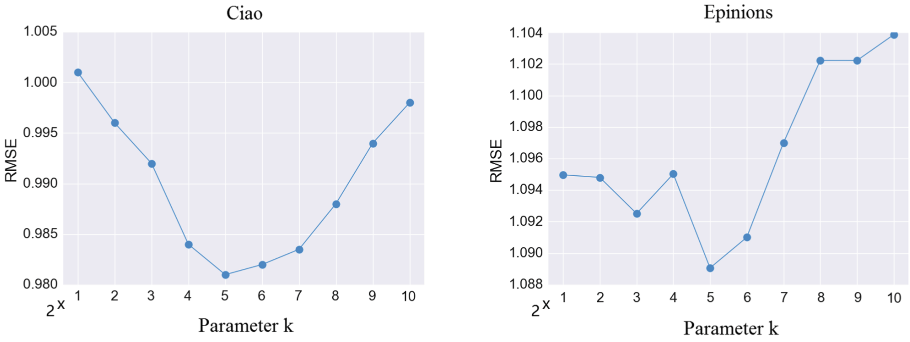

Fig. 2. Impact of parameter α

As can be seen in Fig. 2, SocialEM reaches its best performance when $k = 2^5$. Neither larger values nor smaller values lead to a performance degradation.

Specifically, larger k value means a more sparse SPPMI matrix, otherwise a more dense one. The result demonstrates that user pairs with few co-occurrence times can be the noises which negatively impact the accuracy. Because the few co-occurrences are likely to derive from the coincidence, they can not uncover the real interactions among users. In addition, a more sparse SPPMI matrix leaves out finer grained interactions and only considers user pairs with many co-occurrence times. Therefore, the predictions for users with few relations may fail. According to the result, we can draw a conclusion that finding a proper k helps to relieve the problem of noises in social relations.

5 Related Work

Complementing matrix factorization with further data, e.g., implicit feedback, temporal information or predefined metadata, has widely been accepted to increase algorithmic accuracy. Because of the convenience of matrix factorization for incorporating social information, several social recommenders based on matrix factorization have been proposed recently [3,4,11–14,21].

In [11], the authors proposed an ensemble method that considers the preference of users to be determined by their own tastes and their friends' tastes. Therefore, the missing ratings of a given user are predicted as a linear combination of the ratings of the user and his friends. [13] proposed a model to co-factorize the rating matrix and the relation matrix into three low-rank matrices. In this model, social information and rating information are connected through the shared user latent feature space. In [21], two trust models were proposed. In contrast to other social recommenders, this work considers that both the trustors and trustees can affect users' preferences. The authors of [4] proposed a method named SocialMF. The idea behind SocialMF is that a user's preferences should be similar to those of his friends. Thus, for a given user, the method forces the latent factors to be close to that of his friends. In [12], a method was proposed that takes both trust and distrust into consideration. The authors deemed that each user's latent factors should be similar to those of his friends and differ widely from those of distrusted users. However, all these methods use social relations directly, which makes these methods less reliable.

Among the existing research, the principle of [9,22] is similar to that of our method. In [9], the authors investigated the interactions among items. It is the earliest work which combines word embeddings with CF, but it ignores the potential value in social relations. [22] adopts the skip-gram model of word2vec to embeds the social network structure. Then it forces the user latent factors to approximate the top-k friends embedding. This model has been prove to be effective, however, it is time-consuming for the construction of network embeddings.

6 Conclusion

This paper is motivated by the recent success of the word embedding model, word2vec, which can be interpreted as the implicit factorization for the word co-occurrence matrix. Based on this work, we proposed a novel social recommender called SocialEM that combines user embedding and CF. For each pair of users, the user co-occurrence matrix encodes the number of being trusted together by other users in the social relation network. Then SocialEM jointly decomposes the user-item rating matrix and the user co-occurrence matrix with shared latent user factors, and next uses the dot products of user latent factors and item latent factors to predict missing ratings. Experiments conducted on real-word datasets show that SocialEM outperforms state-of-the-art methods in terms of recommendation accuracy.

In contrast to general CF-based recommenders, models which integrates user embeddings can further reveal implicit interactions among users. Therefore, the extensions of our model can be applied to other fields related to recommender systems. For instance, spammers who are known as shilling attackers can insert malicious ratings to manipulate the recommendations, which results in different user profiles. User embeddings can uncover the nuance between normal users and spammers, which will contribute to the spammer detection.

Acknowledgment. This research is supported by the National Key Basic Research Program of China (973) (2013CB328903), the National Natural Science Foundation of China (61472021), and the Graduate Scientific Research and Innovation Foundation of Chongqing (cys17035).

References

1. Ibm: Ibm's black friday report says twitter delivered 0 percent of referral traffic and facebook sent just 0.68 percent (2012). http://t.cn/ROA8BeQ
2. Quora: Why does the startup idea of social recommendations consistently fail? (2012). http://t.cn/ROA8gPS
3. Guo, G., Zhang, J., Yorkesmith, N.: TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings, pp. 123–129 (2015)
4. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks, pp. 135–142 (2010)
5. Ji, S., Yun, H., Yanardag, P., Matsushima, S., Vishwanathan, S.V.N.: WordRank: learning word embeddings via robust ranking. In: Conference on Empirical Methods in Natural Language Processing, pp. 658–668 (2016)
6. Joulin, A., Grave, E., Bojanowski, P., Mikolov, T.: Bag of tricks for efficient text classification (2016)
7. Koren, Y., Bell, R.M., Volinsky, C.: Matrix factorization techniques for recommender systems. IEEE Comput. **42**(8), 30–37 (2009)
8. Levy, O., Goldberg, Y.: Neural word embedding as implicit matrix factorization. In: Advances in Neural Information Processing Systems, vol. 3, pp. 2177–2185 (2014)
9. Liang, D., Altosaar, J., Charlin, L., Blei, D.M.: Factorization meets the item embedding: regularizing matrix factorization with item co-occurrence. In: ACM Conference on Recommender Systems, pp. 59–66 (2016)
10. Lu, J., Wu, D., Mao, M., Wang, W., Zhang, G.: Recommender system application developments: a survey. Decis. Support Syst. **74**, 12–32 (2015)

11. Ma, H., King, I., Lyu, M.R.: Learning to recommend with social trust ensemble, pp. 203–210 (2009)
12. Ma, H., Lyu, M.R., King, I.: Learning to recommend with trust and distrust relationships, pp. 189–196 (2009)
13. Ma, H., Yang, H., Lyu, M.R., King, I.: SoRec: social recommendation using probabilistic matrix factorization. In: Proceedings of the 17th ACM Conference on Information and Knowledge Management, pp. 931–940. ACM (2008)
14. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: Proceedings of the Fourth ACM International Conference on Web Search and Data Mining, pp. 287–296. ACM (2011)
15. Massa, P., Avesani, P.: Trust-aware recommender systems. In: Proceedings of the 2007 ACM Conference on Recommender Systems, pp. 17–24. ACM (2007)
16. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. Computer Science (2013)
17. Salakhutdinov, R., Mnih, A.: Probabilistic matrix factorization, pp. 1257–1264 (2007)
18. Sinha, R., Swearingen, K.: Comparing recommendations made by online systems and friends. In: Proceedings of the DELOS-NSF Workshop on Personalization and Recommender Systems in Digital Libraries (2001)
19. Tang, J., Gao, H., Liu, H.: mTrust: discerning multi-faceted trust in a connected world. In: International Conference on Web Search and Web Data Mining, WSDM 2012, Seattle, WA, USA, pp. 93–102, February 2012
20. Tang, J., Hu, X., Liu, H.: Social recommendation: a review. Soc. Netw. Anal. Min. **3**(4), 1113–1133 (2013)
21. Yang, B., Lei, Y., Liu, J., Li, W.: Social collaborative filtering by trust. IEEE Trans. Pattern Anal. Mach. Intell. **39**(8), 1633–1647 (2017)
22. Zhang, C., Yu, L., Wang, Y., Shah, C., Zhang, X.: Collaborative user network embedding for social recommender systems (2017)