Contextual Attention Model for Social Recommendation

Hongfeng Bao, Le Wu(✉), and Peijie Sun

Hefei University of Technology, Hefei, China
lewu@hfut.edu.cn

Abstract. Recently, with the emergence of a large number of social platforms, more and more works have been explored for social recommendation. On a social platform, social scientists converged that there exists social influence among users. Thus, accurately modeling the social influence could alleviate the data sparsity issue in Collaborative Filtering (CF). Most of the methods simply define the social influence with the normalized constant weights. However, this is not accuracy enough, which requires more reliable modeling. Besides, many studies have adopted neural network with CF in various recommendation tasks due to the effective ability of neural network for representation. In this paper, we attempt to apply attention mechanism based neural network structure for social recommendation. Specifically, social attention can weigh the contribution of social influence in the form of scores from each neighbor, and then generates each user's social context. Finally, extensive experimental results confirm the feasibility and effectiveness of our proposed model.

Keywords: Deep learning for recommender system
Attention model · Social network

1 Introduction

With the rapid development of the Internet and the explosive growth of information, recommender systems have been increasingly applied to help users extract the information they need efficiently. Collaborative Filtering is one of the most commonly used method due to its making recommendation based on the collaborative behaviors of users without requiring the explicit user and item profiles [1]. However, CF also has several weaknesses, especially it achieves poor performance when the user-item interaction data is very sparse. Thus, many researchers have explored additional auxiliary information to alleviate this issue of CF such as social information.

It is well known that users in a social network are correlated, and users' interests are similar to or influenced by his social connections apart from their own interests [2]. Therefore, many works are proposed to analyse how to learn social influence from social network effectively, and then incorporate it in the classical latent factor based models [3–5]. Although several methods successfully used

R. Hong et al. (Eds.): PCM 2018, LNCS 11165, pp. 630–641, 2018.
https://doi.org/10.1007/978-3-030-00767-6_58

the social information and solved the data sparsity problem of CF, they modeled the social influence between one user and his neighbor with a normalized constant weight. This is obviously unreasonable, thus requiring a more appropriate approach to represent social influence.

The wide population of deep learning techniques gives us more chances to further to enhance the recommendation performance for social recommendation. The neural network has a strong ability of feature representation to better represent users' preferences and items' contents from available data. Hence, several researchers actively explored the approach to combining deep learning techniques with CF models for various recommendation tasks [6–8]. These efforts applied deep neural networks to learn the complex user-item relationship from training data and make tremendous progress toward improving the effectiveness of recommender system.

In this paper, we make an attempt to model social influence through an attention based neural network. Attention mechanism can be regarded as a resource allocation model, which has a wide range of applications in many fields of machine learning [9–11]. In our model, we use the attention mechanism to quantize the social influence between a user and his neighbors in the form of probability, and then create a context. As a consequence, our objective prediction function is composed of a classical latent factor based model and a social context generated by an attention model, which models the users' interests and social influence respectively. Finally, we conduct extensive experiments on two real-world datasets to demonstrate the superiority of our proposed model.

2 Related Work

2.1 Collaborative Filtering and Social Recommendation

CF aims at making recommendations or predictions of the unknown preferences for a group of users by learning the known other users' historical behavior [1]. So far many CF techniques have been developed and the latent factor based models are undoubtedly the most popular one among them due to their relatively high performance [12,13]. In some real world applications, only binary data can be available, which reflect a user's action or inaction. Such recommendation with implicit feedback is called the one-class problem, where only positive feedback is available [14]. In this paper, we use common approach of sample based learning samples negative feedback from the missing data [7,13,14]. Since there are many challenges to CF systems such as data sparsity [1,15], many efforts were done for social recommendation. Several researchers completed a series of attempts of fusing users' preference with social links and achieve desirable results [4,5,16,17]. In addition, more and more deep learning techniques are applied to social recommendation now [18]. Some of them processed analysed user-provided content in social media to understand social information [19]. These models make better use of social information thus showing improved performance over classical recommendation models.

2.2 Deep Learning and Attention Mechanism

Recently, many data scientists have put their focus on deep learning in the field of recommender systems [20]. Several researches used autoencoder to learn hidden user representations from the rating matrix and improved the result of recommendation based on collaborative filtering techniques [6,21–23]. NeuMF combines a generalization of matrix factorization and a multilayer perceptron [7]. Except for these, attention model is also increasingly applied to the framework of neural network for recommendation and successfully captures various contextual information [24]. The main idea of attention mechanism is that human recognition only focuses on selective parts of a target scenario instead of the whole perception space. IARN learns different scores from all time steps in user-item interaction history to capture user and item dynamics jointly [25]. Besides, Chen et al. [26] and Seo et al. [27] proposed a hierarchical attention network and a dual attention network respectively for recommendation. Our work extends attention mechanism in social network. Specifically, we score social influence of each user's all connections through attention model for more reliable modeling.

3 The Proposed Model

In a social recommender system, we have a userset U with N users and an itemset I with M item. Let $\mathbf{S} \in \mathbb{R}^{N\times N}$ denotes social network matrix and S_u represents all the followees of user u, i.e., if $S_{uv} > 0$, then $v \in S_u$. Since the social contextual information is made up of social influence, the social context C_{S_u} can be computed by social influence from each user u's social neighbors S_u. Besides, The ratings expressed by users on items are given in a user-item interaction matrix $\mathbf{R} \in \mathbb{R}^{N\times N}$ with $R_{ui} = 1$ if user u shows preference for item i, otherwise it equals 0. Given the sparse social matrix $\mathbf{S}$ and user-item interaction matrix $\mathbf{R}$, our goal is to recommend the potential top-N items each user may like based on various social contextual information.

3.1 General Framework

In this subsection, we describe our proposed Contextual Attention Model for Social Recommendation. Figure 1 shows the overall architecture of our model, which is composed of two parts. Attention model part models user's preference score with respect to the social contextual information from social influence. Given a user u, we use a_{uv} to denote the degree of a user is influenced by his neighbors S_u. Besides, in the part of social embedding, we apply an autoencoder to learn the social embedding of each user with the social network structure. Then we take it as another input of attention model, which makes the training of attention model consider richer social information.

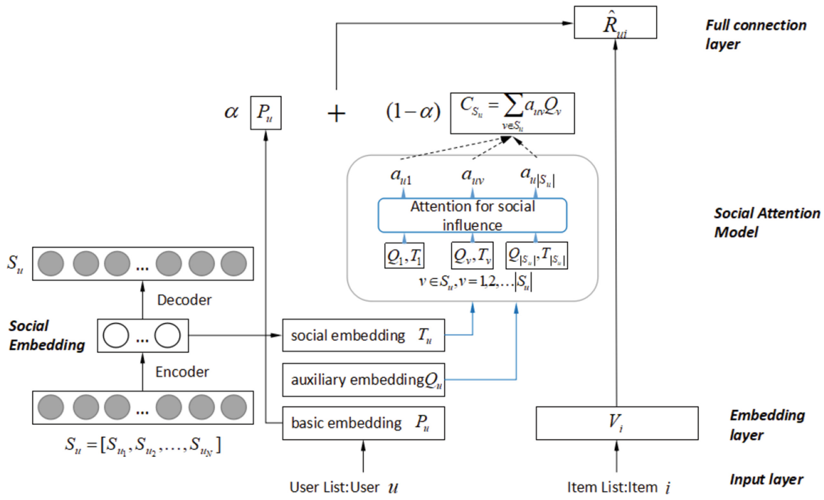


Fig. 1. The overall architecture of the proposed model.

Objective Prediction Function. In a social recommender system, a user u's preference score on item i can be predicted by the user-item interaction and social contextual information. The former is interpreted by the user u's favor on item i, and the latter represents the user u's potential favor on item i affected by his social neighbors S_u's favors. For the input, we parameterize each user u with a base embedding P_u and each item i with a base embedding V_i as general latent factor based models. The base embedding denotes a user's base latent interest or the basic item latent vector of an item. In the other part, we use attention mechanism to estimate the social influence strength of one user to another. Then we get:

$$\hat{R}_{ui} = (\alpha P_u + (1 - \alpha)\, C_{S_u}) \times V_i \tag{1}$$

In the above function, the basic latent factor based model and the social context are smoothed by the parameter α, which controls the effects of social neighbors on the predicted rating. Specifically, the part of the social context C_{S_u} is an attention subnetwork and use learned attentive weights to model social influence. The realization of the attention part is stated in detail in the following parts. In addition, if we replace all attentive weights with normalized weights $\frac{1}{|S_u|}$, our model turns to a RSTE model [4].

Social Network Embedding. To train the attention model by making more use of social information, we add the social embedding for each user in the alignment model. The social embedding part aims at gaining another representation for each user from another different low-dimensional space. In this paper, we choose the autoencoder to construct such a social embedding space directly among many useful methods [28,29].

Autoencoder is one of the feed-forward neural network. Its basic idea is compressing the high-dimensional input data into hidden representation and then reconstructing the input. The whole structure of autoencoder consists of an encoder part and a decoder part, which is shown in the left part in Fig. 1. In our model, we let a sparse social matrix $\mathbf{S}$ as the input and expect to gain a low-dimensional social embedding space $T \in \mathbb{R}^{N \times D}$ from two parts:

$$\begin{aligned} T_u &= f\left(\mathbf{W_1} \times S_u + b\right), \\ \hat{S}_u &= f'\left(T_u\right) = f'\left(\mathbf{W_1}' \times T_u + b'\right), \end{aligned} \tag{2}$$

where $f(x)$ and $f'(x)$ are both nonlinear functions. $\hat{S}_u$ a reconstructed vector and $S_u \in \mathbb{R}^N, T_u \in \mathbb{R}^D$. Its parameter set is $\theta_1 = \{\mathbf{W_1} \in \mathbb{R}^{D \times N}, b \in \mathbb{R}^D, \mathbf{W_1}' \in \mathbb{R}^{N \times D}, b' \in \mathbb{R}^N\}$.

Social Attention Model. In this part, we describe how to use attention mechanism to learn influence weights between users from social network matrix $\mathbf{S}$. Specifically, we use e_{uv} to denote the social influence strength of v to u if user u follows v. Then, the social attention model generates a social contextual vector in accordance with the contribution of social influence from each user's social connections. The attention model can be simply realized by a single multilayer perceptron such that:

$$e_{uv} = w_3 \times \sigma\left(\mathbf{W_2} \times [P_u, P_v, T_u, T_v]\right), \tag{3}$$

where we choose the sigmoid function $\sigma(x)$ as activation function. $\theta_2 = [\mathbf{W_2}, w_3]$ are the parameters in the social attention model, with $\mathbf{W_2}$ denotes the matrix parameter and w_3 is the parameters of the activation function. P_u and P_v denote the base embeddings of the user u and one of his neighbor v respectively. Similarly, T_u and T_v are the social embeddings of them which represent their own social network structures.

Then, we obtain the final attentive social influence scores a_{uv} by normalizing the above attentive scores using Softmax function:

$$a_{uv} = \frac{exp\left(e_{uv}\right)}{\sum_{v \in S_u} exp\left(e_{uv}\right)}. \tag{4}$$

The normalized attentive social influence scores a_{uv} above can be interpreted as the contribution of the preference of user u's all neighbors to user u, so we calculate the social context C_{S_u} of user u as:

$$C_{S_u} = \sum_{v \in S_u} a_{uv} P_v. \tag{5}$$

Here we make an improvement of adding an auxiliary embedding vector Q_u to distinguish the user embedding in two parts as showed in Fig. 1:

$$R_{ui} = \left(\alpha P_u + (1-\alpha) \sum_{v \in S_u} a_{uv} Q_v \right) \times V_i,$$
$$\text{where } a_{uv} = \frac{exp\left(e_{uv}\right)}{\sum_{v \in S_u} exp\left(e_{uv}\right)}, e_{uv} = w_3 \times \sigma\left(\mathbf{W_2} \times \left[Q_u, Q_v, T_u, T_v\right]\right). \quad (6)$$

In the new network, the base embedding of user P_u only participates in the first half of equation, and the social influence score in the part of attention model is calculated by the auxiliary embeddings and the social embeddings. With this change, the model is better trained, thus our results ending up with a lot of improvement.

3.2 Model Learning

As similar as most of the latent factor based models [12], we apply the regularized squared loss to the optimization function:

$$\arg\min_{P,V,\theta_2} \sum_{(u,i) \in D} \left(R_{ui} - \hat{R}_{ui} \right)^2 + \lambda \left(\|P\|^2 + \|V\|^2 \right), \quad (7)$$

where P and V denote the base embedding matrices and θ_2 is the parameters of social attention model. D represents the set that the users have the observed ratings on items in training data. λ controls the strength of regularization.

During actual model training, the social embedding part was pretrained. Besides, our datasets are processed for implicit feedback of users, which has no negative values. Therefore, for each observed positive value, we randomly sample 5 negative values from missing data at each iteration of the training process [14]. At last, we choose TensorFlow to train our model using mini-batch Adam.

4 Experiments

4.1 Experimental Settings

Datasets. We choose two publicly accessible datasets to complete our experiments: Flixster [3] and Douban. At first, due to high sparsity of datasets, we filter the datasets by retaining users that have at least 5 rating records and 5 social links. We also filter out items that have been rated less than 5 times. Then, since we are concerned about recommendation with implicit feedback, we relabel the rating matrix with 1 or 0 indicating whether the user has rated the item [7,26]. The characteristics of the two datasets after adjustment are showed in Table 1. In data splitting process, we adopt the widely used leave-one-out procedure in [7,13,26]. Specifically, for each user, we hold-out his latest rating record as the test data, and the remaining data are used for training, 5% of which form the validation dataset.

Table 1. Statistics of the two evaluation datasets.

Dataset	Users	Items	Ratings	Links	Rating density
Douban	6,739	17,902	840,828	201,014	0.697%
Flixster	9,874	10,978	283,503	120,306	0.262%

Evaluation Metrics. We select two widely adopted metrics for top-K ranking performance in several recent works for recommendation: the Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) [7,26]. HR measures the percentage of top-K ranked items that are rated by users in the test data. NDCG gives a higher score when the hit items are ranked higher in a ranking list. As it is too time-consuming to complete top-K ranking evaluation for all users, we follow the common strategy: for each test user, we randomly select 1000 unrated items in the training data. Then we mix them with the records in both validation and test data to generate top-K results, and we repeat this process for 10 times and average the results [7,26].

Baselines. We choose the following baselines compared with our model:

- *PMF.* It is a basic competitive method for recommendation [12].
- *SocialMF.* This model incorporates the social information into the matrix factorization framework. It models the social influence with the equal normalized weights, which is the improvement of RSTE model [3,4].
- *SR.* It is another state-of-the-art model for social recommendation [5].
- *AutoRec.* It provides an autoencoder based neural network structure for recommendation [6].
- *NeuMF.* It is an excellent method as a competitive neural network based model for recommendation [7].

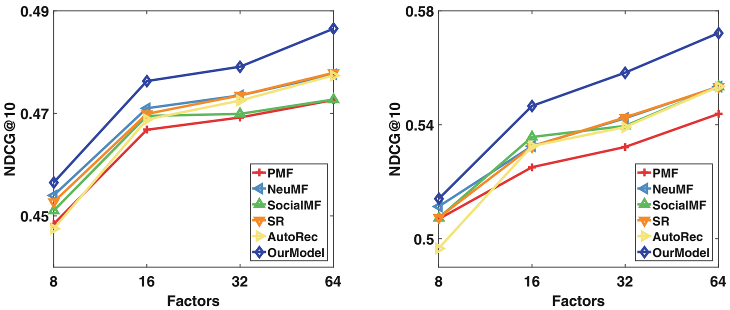
Parameter Settings. In view of the effectiveness and efficiency, we finally set the batch size to 2048 and the learning rate to 0.001. Besides, the latent feature dimension is set as 16 and the parameter α is set as optimal value in the following parts if not specified. For the initialization of these methods, we initialized these models with the parameters of PMF model.

4.2 Performance Comparison

Figure 2 only shows the performance of NDCG@10 with respect to the number of latent factors for page limit. We can observe that our proposed model always achieves the best performance on both datasets as the number of predictive factors grows. This means that attention mechansim can capture more accurate social influence to improve social recommendation significantly. What's more, we find that most models have the largest increases from the factor of 8 to 16.

Thus, we set the number of predictive factors as 16 in the following experiments, although our proposed model has better results with a larger factor.

Next, the performance of Top-K recommendation lists on the two datasets are presented in Fig. 3. We can see that our proposed method always performs the best on both datasets across all positions. In particular, our model is greatly superior to socialMF, which indicates that social influence can be modeled more reasonably with attentive weights than the equal normalized weights. Besides, other four baselines outperform PMF. This shows that encoding social network information into the latent factor based model or applying deep learning techniques really help to enhance the performance of recommendation.



(a) Douban-NDCG@10 (b) Flixster-NDCG@10

Fig. 2. Performance of HR@10 and NDCG@10 w.r.t. the number of predictive factors on the two datasets.

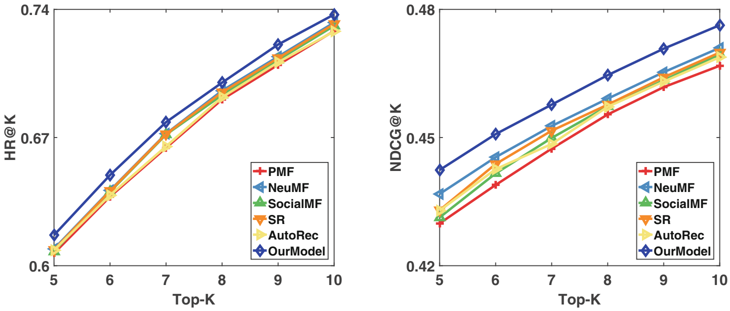
4.3 Impact of Parameter α

In our proposed model, parameter α balances how much information from the users' own characteristics and their neighbors' preference can be utilize in social recommendation, similar as that in RSTE [4]. If $\alpha = 1$, it turns to a PMF model, which does not encode any social information. If $\alpha = 0$, we train the recommendation model only by social information learned from attention mechanism.

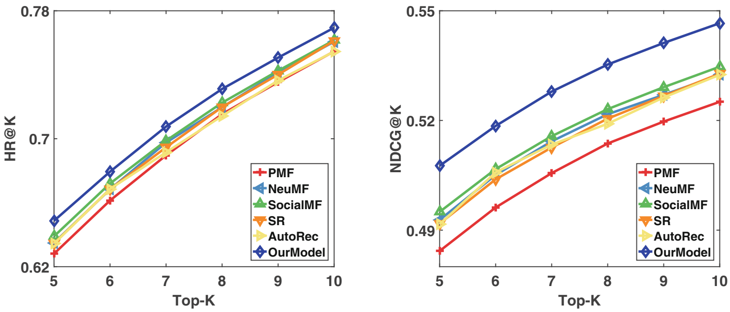
Figure 4 shows the metric of NDCG@10 on two datasets with different values of α. To show the difference clearly, we remove the minimum value when $\alpha = 0$ here. We observe that two curves have a similar trend, and the optimal value of α is around 0.9 and 0.7 respectively. This means that our method learns effective social information to improve the performance of recommendation.

4.4 Effect on Attention Input

Since our model introduces many parameters to learn social influence from social matrix, it inevitably leads to overfitting. In this paper, dropout is applied to

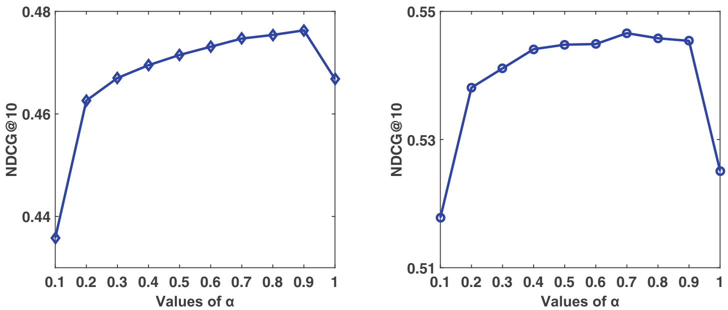


(a) Douban-HR@K (b) Douban-NDCG@K



(c) Flixster-HR@K (d) Flixster-NDCG@K

Fig. 3. Performance of Top-K item recommendation where K ranges from 5 to 10 on the two datasets.



(a) Douban-NDCG@10 (b) Flixster-NDCG@10

Fig. 4. Impact of parameter α

reduce overfitting to enhance the performance of model [30]. Table 2 shows the results with different dropout ratio ρ. We choose the dropout ratio ρ as 0.2 and 0.3 respectively for their best performances. Compared with the performance when $\rho = 0$, there are great improvements on both datasets.

In Table 3, we show whether the addition of auxiliary embeddings for users can make the model better trained. We observe that the model with two different embeddings for users performs better on both datasets. Especially, on Flixster, the second way relies more on social information with a smaller value of α, which means it learns more reliable social influence.

Table 2. HR@10 and NDCG@10 with different dropout ratio.

Dropout ratio	0	0.1	0.2	0.3	0.4	0.5
Douban						
HR	0.7325	0.7369	**0.7372**	0.7333	0.7310	0.7289
NDCG	0.4722	0.4756	**0.4763**	0.4736	0.4714	0.4691
Flixster						
HR	0.7612	0.7634	0.7673	**0.7694**	0.7682	0.7659
NDCG	0.5348	0.5377	0.5428	**0.5466**	0.5450	0.5411

Table 3. The best results of our model with two ways for users.

Model	Optimal α	Dropout ratio	HR	NDCG
Douban				
$P = Q$	0.9	0.2	0.7341	0.4735
$P \neq Q$	0.9	0.2	0.7369	0.4763
Flixster				
$P = Q$	0.9	0.3	0.7663	0.5449
$P \neq Q$	0.7	0.3	0.7694	0.5466

5 Conclusion

In this paper, we proposed an attention mechanism based neural network architecture for social recommendation. Specifically, we attempted to capture social influence from social network through attention mechanism and learn more real attentive weights to generate social context. Obviously, the attentive weights learned are not all the same between one user and different neighbors. Then, we encoded the learned social information in a general matrix factorization model. The extensive experiments were conducted on two real-world datasets, which demonstrated the effectiveness of our proposed model. After analysing the results of comparative experiments, it is proved that attention model can learn social influence more effectively than those models with the simple modeling likes the normalized constant weights for social recommendation.

Acknowledgments. This work was supported in part by the National Key Research and Development Program of China (Grant No. 2017YFC0820604), the National Natural Science Foundation of China (Grant No. 61602147, No. 61632007), the Open Project Program of the National Laboratory of Pattern Recognition (NLPR Grant No. 201700017), and the Fundamental Research Funds for the Central Universities (Grant No. JZ2018HGTB0230).

References

1. Su, X., Khoshgoftaar, T.M.: A survey of collaborative filtering techniques. Adv. Artif. Intell. **2009**(4), 2 (2009). https://doi.org/10.1155/2009/421425
2. Anagnostopoulos, A., Kumar, R., Mahdian, M.: Influence and correlation in social networks. In: SIGKDD, pp. 7–15. ACM, New York (2008). https://doi.org/10.1145/1401890.1401897
3. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: RecSys, pp. 135–142. ACM, New York (2010). https://doi.org/10.1145/1864708.1864736
4. Ma, H., King, I., Lyu, M.R.: Learning to Recommend with Social Trust Ensemble. In: SIGIR, pp. 203–210. ACM, New York (2009). https://doi.org/10.1145/1571941.1571978
5. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: WSDM, pp. 287–296. ACM, New York (2011). https://doi.org/10.1145/1935826.1935877
6. Sedhain, S., Menon, A.K., Sanner, S., Xie, L.: Autorec: autoencoders meet collaborative filtering. In: WWW, pp. 111–112. ACM, New York (2015). https://doi.org/10.1145/2740908.2742726
7. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.-S.: Neural collaborative filtering. In: WWW, pp. 173–182, Perth, Australia (2017). https://doi.org/10.1145/3038912.3052569
8. Cheng, H.-T., Koc, L., Harmsen, J., et al.: Wide & deep learning for recommender systems. In: Proceedings of the 1st Workshop on Deep Learning for Recommender Systems, pp. 7–10. ACM, New York (2016). https://doi.org/10.1145/2988450.2988454
9. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv:1409.0473 (2014)
10. Xu, K., et al.: Show, attend and tell: Neural image caption generation with visual attention. In: ICML, pp. 2048–2057. JMLR.org (2015)
11. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical attention networks for document classification. In: Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 1480–1489 (2016)
12. Mnih, A., Salakhutdinov, R.: Probabilistic matrix factorization. In: NIPS, pp. 1257–1264. Curran Associates Inc., New York (2007)

13. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: BPR: Bayesian personalized ranking from implicit feedback. In: UAI, pp. 452–461. AUAI Press Arlington, Virginia (2009)
14. Pan, R., et al.: One-class collaborative filtering. In: ICDM, pp. 502–511. IEEE, New York (2008). https://doi.org/10.1109/ICDM.2008.16
15. Adomavicius, G., Tuzhilin, A.: Toward the next gen- eration of recommender systems: a survey of the state-of-the-art and possible extensions. TKDE **17**(6), 734–749 (2005). https://doi.org/10.1109/TKDE.2005.99
16. Ma, H., Yang, H., Lyu, M.R., King, I.: SoRec: social recommendation using probabilistic matrix factorization. In: CIKM, pp. 931–940. ACM, New York (2008). https://doi.org/10.1145/1458082.1458205
17. Wu, L., et al.: Modeling the evolution of users preferences and social links in social networking services. TKDE **29**(6), 1240–1253 (2017). https://doi.org/10.1109/TKDE.2017.2663422
18. Deng, S., Huang, L., Xu, G., Wu, X., Wu, Z.: On deep learning for trust-aware recommendations in social networks. TNNLSS **28**(5), 1164–1177 (2017). https://doi.org/10.1109/TNNLS.2016.2514368
19. Li, Z., Tang, J.: Weakly Supervised deep matrix factorization for social image understanding. IEEE Trans. Image Process. **26**(1), 276–288 (2017). https://doi.org/10.1109/TIP.2016.2624140
20. Zhang, S., Yao, L., Sun, A.: Deep learning based recommender system: a survey and new perspectives. arXiv:1707.07435 (2017)
21. Vincent, P., Larochelle, H., Lajoie, I., Bengio, Y., Manzagol, P.-A.: Stacked denoising autoencoders: learning useful representations in a deep network with a local denoising criterion. JMLR **11**(3), 3371–3408 (2010)
22. Wu, Y., DuBois, C., Zheng, A.X., Ester, M.: Collaborative denoising autoencoders for top-n recommender systems. In: WSDM, pp. 153–162. ACM, New York (2016). https://doi.org/10.1145/2835776.2835837
23. Wang, H., Wang, N., Yeung, D.-Y.: Collaborative deep learning for recommender systems. In: SIGKDD, pp. 1235–1244. ACM, New York (2015). https://doi.org/10.1145/2783258.2783273
24. Sun, P., Wu, L., Wang, M.: Attentive recurrent social recommendation. In: SIGIR, pp. 185–194. ACM, New York (2018). https://doi.org/10.1145/3209978.3210023
25. Pei, W., Yang, J., Sun, Z., Zhang, J., Bozzon, A., Tax, D.M.J.: Interacting attention-gated recurrent networks for recommendation. In: CIKM, pp. 1459–1468. ACM, New York (2017). https://doi.org/10.1145/3132847.3133005
26. Chen, J., Zhang, H., He, X., Nie, L., Liu, W., Chua, T.-S.: Attentive collaborative filtering: multimedia recommendation with item-and component-level attention. In: SIGIR, pp. 335–344. ACM, New York (2017). https://doi.org/10.1145/3077136.3080797
27. Seo, S., Huang, J., Yang, H., Liu, Y.: Interpretable convolutional neural networks with dual local and global attention for review rating prediction. In: RecSys, pp. 297–305. ACM, New York (2017). https://doi.org/10.1145/3109859.3109890
28. Chang, S., Han, W., Tang, J., Qi, G.-J., Aggarwal, C.C., Huang, T.S.: Heterogeneous network embedding via deep architectures. In: SIGKDD, pp. 119–128. ACM, New York (2015). https://doi.org/10.1145/2783258.2783296
29. Wang, D., Cui, P., Zhu, W.: Structural deep network embedding. In: SIGKDD, pp. 1225–1234. ACM, New York (2016). https://doi.org/10.1145/2939672.2939753
30. Srivastava, N., Hinton, G.E., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. JMLR **15**(1), 1929–1958 (2014)