



Modeling Users' Multifaceted Interest Correlation for Social Recommendation

Hao Wang^{1(✉)}, Huawei Shen², and Xueqi Cheng²

¹ Baidu Inc., Beijing, China
way_wh@yeah.net

² CAS Key Lab of Network Data Science and Technology,
Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China
{shenhuawei, cxq}@ict.ac.cn

Abstract. Recommender systems suggest to users the items that are potentially of their interests, by mining users' feedback data on items. Social relations provide an independent source of information about users and can be exploited for improving recommendation performance. Most of existing recommendation methods exploit social influence by refining social relations into a *scalar indicator* to either directly recommend friends' visited items to users or constrain that friends' embeddings are similar. However, a scalar indicator cannot express the multifaceted interest correlations between users, since each user's interest is distributed across multiple dimensions. To address this issue, we propose a new embedding-based framework, which exploits users' multifaceted interest correlation for social recommendation. We design a *dimension-wise attention* mechanism to learn a *correlation vector* to characterize the interest correlation between a pair of friends, capturing the high variation of users' interest correlation on multiple dimensions. Moreover, we use friends' embeddings to smooth a user's own embedding with the correlation vector as weights, building the elaborate unstructured social influence between users. Experimental results on two real-world datasets demonstrate that modeling users' multifaceted interest correlations can significantly improve recommendation performance.

Keywords: Recommendation systems · Social influence · Interest correlation · Attention mechanism · POI recommendation

1 Introduction

Recommender systems suggest to users the items that are potentially of their interests [22] by mining users' feedback data on items [23]. Real-world recommender systems often allow users to build social relations [26], and such social relations provide an independent source of information about users beyond the

H. Wang—This work is done when Hao Wang was a Ph.D. candidate in Institute of Computing Technology, Chinese Academy of Sciences.

feedback information [13]. Social correlation theories [18], such as homophily and social influence, indicate that there are correlations between two socially connected users [2], which can potentially be used to exploit social relations for improving recommendation accuracy [25].

Many methods have been proposed for social recommendation in recent years, and these methods can be mainly grouped into two categories: (1) memory-based methods [1, 12, 14] use social relation as an indicator that filters relevant users and directly recommend friends' visited items to a user; (2) model-based methods [4, 5, 9, 10, 22, 27, 29, 31] integrate social relation into factorization methods to constrain that friends share similar interest embeddings. Moreover, feedback-based similarities are utilized to weigh friends' interest relevance in memory-based methods [3] or embedding coherence in model-based methods [11, 17]. In sum, existing methods refine two users' social relation into a *scalar indicator* to build their interest correlation.

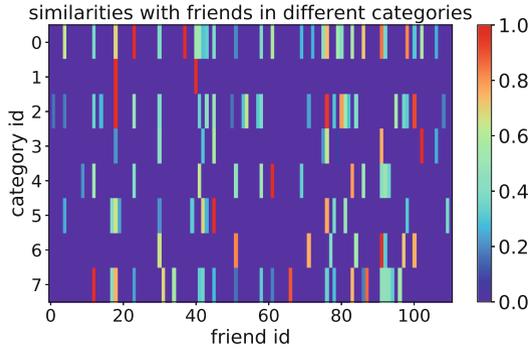


Fig. 1. Feedback-based similarities of 8 different categories between a random user and his 111 friends. We normalize similarities by the largest value in each category.

However, each user's interest is differently distributed across multiple dimensions, and the consistency in one dimension does not mean consistency in other dimensions. As Fig. 1 shows, a user's interest similarities with his friends vary greatly in different categories of items. When the user needs suggestions on one category, he may refer to friends with strong correlations on that category, and suggestions of friends with strong correlations on other categories are not useful. Therefore, a global scalar indicator used in existing methods cannot express the multifaceted interest correlations between friends. Unfortunately, there exists no explicit evidence to refine social networks into the elaborate correlation, and simply distinguishing items' categories would make the problem of data sparsity even more serious, which is not conducive to the learning of model parameters.

In this paper, we propose a new embedding-based social recommendation method (Fig. 2). We propose to use a *correlation vector*, instead of a scalar value, to characterize the interest correlation between each pair of friends, and design

a dimension-wise attention mechanism with the social network as input to learn it. The correlation vector has the same dimension with user’s embedding, thus can sufficiently capture the high variation in users’ interest correlations on each fine-grained dimension. Moreover, we smooth a user’s embedding by his friends’ embeddings, with the correlation vector into consideration. The combination of the dimension-wise attention mechanism and the smoothing operation can impose strong and delicate unstructured correlations on users’ embeddings while making interactions between users and items. Such an end-to-end framework allow the proposed method to learn the unstructured correlations in a fully data-driven manner.

We evaluate the proposed method by extensive experiments on two real-world datasets collected from Gowalla and Epinions respectively. Experimental results show that the proposed method outperforms the state-of-the-art social recommendation methods.

2 Related Work

Recommender systems normally utilize the user-item rating information for recommendation. Collaborative filtering [7] has become one of the most popular technologies, which achieves good recommendation results.

Social Recommender Systems leverage the social network information to enhance traditional recommendation methods [6, 21, 30, 32]. According to the nature of the existing social recommendation techniques, we classify them into two main categories: memory-based methods [1, 12, 14] which normally directly or indirectly recommend users items that their friends like, and model-based methods [4, 5, 9–11, 22, 27, 29, 31] use users’ social relations to constrain that friends share similar embeddings.

In sum, existing methods use a scalar value to build friends’ interest correlation, which cannot sufficiently express their multifaceted interest correlations. Although Yang et al. [28] integrate items’ category information to train a matrix factorization model for each category of items, they make data too sparse to learn parameters, and they cannot utilize correlations among different categories.

Attention Mechanism has recently been used in recommendation tasks [15, 16, 24]. For example, Sun et al. [16] use attention to model the dynamic social influence for recommendation. However, they still express users’ interest correlation by a scalar value, which cannot sufficiently capture the high variation of users’ interest correlation.

3 Model

3.1 Problem Formulation

For ease of description, we first formalize variables used and the problem dealt with in this paper. We denote with U and I the set of users and the set of items respectively. For a user u and an item i , we denote with r_{ui} u ’s feedback to

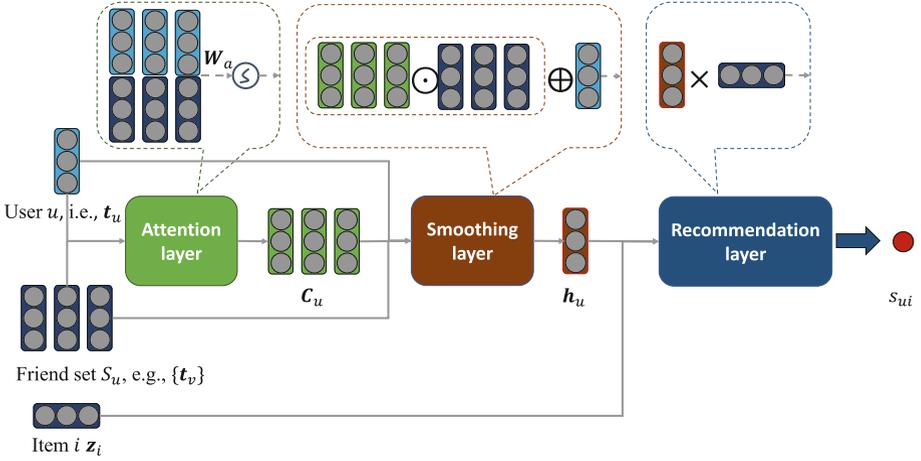


Fig. 2. Architecture of the proposed Dimension-wise Attention model for Social Recommendation, i.e., DASR. DASR infers a user u 's preference to a candidate item i , with embeddings of the user u (t_u), user u 's each friend v (t_v) and the item i (z_i) as the model input. DASR includes three parts, an attention layer for learning user u ' correlation vectors with his friends, a smoothing layer for smoothing user u 's embedding with embeddings of his friends, and a recommendation layer matching the preference between user u and item i .

item i . We use S to represent the social network over users in U . S_u represents user u 's friend set and $S_{uv} = 1(0)$ indicates whether there exists social relation between user u and user v . In our model, we learn a preference vector t_u for each user u and a preference vector z_i for each item I .

Item Recommendation: Given a set of users U with social relations S , a set of items I and U 's feedback over items I , item recommendation recommends for each target user $u \in U$ a list of items $\{i|i \in I\}$ consisting of items that the target user is potentially interested in and has not interacted with them up to the recommendation.

Next, we present the proposed Dimension-wise Attention model for Social Recommendation, i.e., DASR in Fig. 2.

3.2 Model Architecture

Users' interest is often differently distributed across multiple dimensions. To accurately capture friends' influence on user's preference, one needs to model the multi-dimensional interest correlation between users. Fortunately, attention mechanism seems to provide a feasible solution, since it can automatically models and selects pertinent piece of information with the attentive weights from a set of inputs, where higher (lower) weights indicate the corresponding inputs more informative to generate the outputs. To accommodate our problem, we further design a dimension-wise attention mechanism and use it to learn a

correlation vector for each pair of friends, building their multi-dimensional interest correlation for social recommendation.

Figure 2 shows the architecture of DASR, which includes an attention layer, a smoothing layer and a recommendation layer. We learn a preference vector for each user and item, namely, \mathbf{t}_u and \mathbf{z}_i , and use DASR to infer a target user u 's preference to a candidate item i with social influence of u 's friends into consideration. Instead of directly performing inner product between \mathbf{t}_u and \mathbf{z}_i in the recommendation layer, we first use embeddings of user u 's friends to smooth user u 's own embedding in the smoothing layer, and the smoothing weights are the correlation vector learned in the attention layer. With these designs, we can learn strong and delicate unstructured correlations of users' embeddings in a fully data-driven manner and provide better item recommendation.

3.3 Attention Layer

We input the embeddings of the target user u and users in his friend set S_u to the attention layer, and compute the interest correlation vector between user u and each friend v .

We first use a weight matrix \mathbf{W}_a to perform self-attention on user u and friend v as follows:

$$\mathbf{e}_{uv} = \text{LeakyReLU}(\mathbf{W}_a^T(\mathbf{t}_u || \mathbf{t}_v)), \quad (1)$$

where \mathbf{t}_u and \mathbf{t}_v are embeddings of user u and friend v . T represents transposition and $||$ denotes the concatenation operation. $\text{LeakyReLU}(x) = \max(0, x) + \beta \min(0, x)$ acts as non-linear activation function with β as the negative slope. \mathbf{e}_{uv} is the attention coefficient that indicates the importance of friend v 's features to user u . To ensure that \mathbf{e}_{uv} can express the correlation of each dimension in user's embeddings, we design \mathbf{W}_a as a weight matrix with dimension $2d * d$.

For user u , we get an interest correlation matrix \mathbf{C}_u , and each column of \mathbf{C}_u represents the correlation vector between user u and one of his friends. To make coefficients easily comparable across different friends, we normalize \mathbf{C}_u 's each row across all choices of v . We denote with α_{uv} the normalized interest correlation vector between user u and friend v :

$$\alpha_{uv} = \frac{\exp(\mathbf{e}_{uv})}{\sum_{k \in S_u} \exp(\mathbf{e}_{uk})}. \quad (2)$$

3.4 Smoothing Layer

For each friend v in user u 's friend set S_u , we obtain a normalized correlation vector α_{uv} to represent dimension-wise interest correlation between user u and user v .

We then smooth user u 's embedding by adding each friend v 's embedding with the correlation vector α_{uv} serving as smoothing weight.

$$\mathbf{h}_u = \sigma\left(\sum_{k \in S_u} \alpha_{uk} \odot \mathbf{t}_k + \mathbf{t}_u\right), \quad (3)$$

where \mathbf{h}_u is user u 's smoothing embedding. \odot is the element-wise Hadamard product and $\sigma(z) = \frac{1}{1+e^{-z}}$ offers nonlinearity.

\mathbf{h}_u consists of both user u 's and his friends' embeddings, allowing the smoothing embedding not only to retain user u 's own unique interest, but also to integrate his friends' interest. In this way, we can learn different patterns of each user's interest correlation, e.g., some users barely refer to their friends, while some users often refer to a few friends' suggestions, etc.

3.5 Recommendation Layer

In the recommendation layer, we use user's smoothing embedding, i.e., \mathbf{h}_u , to make recommendation.

Denote p_{ui} as user u 's preference to item i , and we compute p_{ui} as follows:

$$p_{ui} = \mathbf{h}_u^T \mathbf{z}_i, \quad (4)$$

where \mathbf{z}_i denotes item i 's embedding.

We define two types of objective functions according to the feedback type, including implicit feedback, e.g., users' check-in counts at POIs, and explicit feedback, e.g., users' rating scores to items. First, we define the objective function in a ranking manner. For each positive feedback (u, i) , we randomly select c negative samples from item set I with item i excluded and denote the set of negative samples as $NEG(i)$. The objective function is defined as follows:

$$L(u, i) = - \sum_{j \in NEG(i)} \sigma(p_{ui} - p_{uj}). \quad (5)$$

The ranking-based objective function can be applied to both explicit feedback and implicit feedback.

Second, we define a square error-based objective function for explicit feedback only, in order to predict a user's rating score to an item:

$$L(u, i) = (p_{ui} - y_{ui})^2, \quad (6)$$

where y_{ui} is the user u 's true rating score to item i .

Finally, for user u , we compute his preference to each item in I according to Eq. 4, and take top n items as the recommendation list.

4 Experiments

4.1 Datasets

We use two real-world datasets collected from Gowalla [20] and Epinions [12] respectively for evaluation.

Gowalla is a Location-Based Social Network (LBSN), and we utilize users' check-in at Point-of-Interests (POIs) and the social network to make POI recommendation [19]. There are 1,196,248 check-ins generated by 18,737 users over

32,510 POIs in the Gowalla dataset. The total number of users' friendship records is 86,985. Epinions is a general consumer review site where users can review items. Different from Gowalla with a two-way connections between users, users' social relationship in Epinions is their Web of Trust, which is a one-way connection, like Twitter followings. We adapt our model to this different structure, and utilize users' rating histories and trust network to make item recommendation [12]. The Epinions dataset consists of 49,290 users who rated a total of 139,738 different items at least once. The total number of reviews is 664,824. The total number of issued trust statements is 487,181.

For each user u , we partition his feedback set into three parts, i.e., 70% as training data, 15% as validation data, and 15% as testing data.

4.2 Evaluation Metrics

To evaluate the recommendation performance, we use two widely-used metrics on both datasets, namely, $precision@n$ and $recall@n$, where n is the number of items in the recommendation list. They are computed as follows:

$$precision@n = \frac{1}{|U|} \sum_{u=1}^{|U|} \frac{|P_u^n \cap T_u|}{|P_u^n|}, \quad (7)$$

$$recall@n = \frac{1}{|U|} \sum_{u=1}^{|U|} \frac{|P_u^n \cap T_u|}{|T_u|}, \quad (8)$$

where P_u^n is the set of top n items in user u 's recommendation list, and T_u is user u 's ground truth set of items. $|x|$ denotes the cardinality of set x . For each metric, we consider 4 values (i.e., 1, 5, 10, 20) of n in our experiments.

For Epinions, we also evaluate the prediction on users' explicit rating scores with MAE , and it is computed as follows:

$$MAE = \frac{1}{|U|} \sum_{u=1}^{|U|} \frac{\sum_{i \in T_u} abs(p_{ui} - y_{ui})}{|T_u|}, \quad (9)$$

where y_{ui} is the true ratings given by user u for item i , and $abs(\cdot)$ is the absolute value function.

4.3 Baselines

Many existing methods are available for POI recommendation, and it is impossible to list all of them as baselines. Here, we select the baselines which serve as representative works of memory-based and model-based social recommendation methods. The baselines include:

- SoCF [3]: SoCF is a social-based collaborative filtering method, which recommend friends' visited items to users.

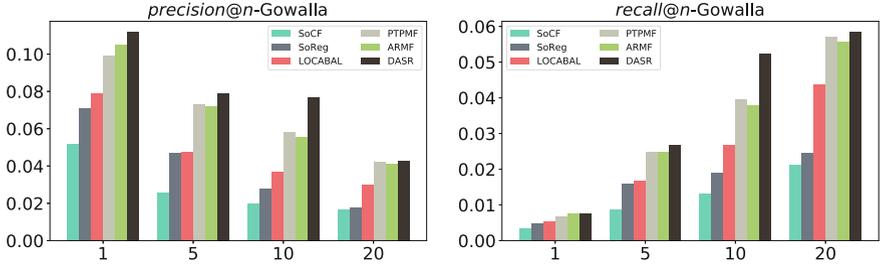


Fig. 3. $precision@n$ and $recall@n$ of DASR and baselines on the Gowalla dataset.

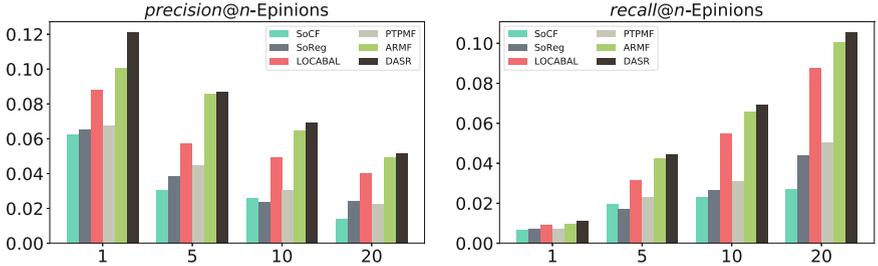


Fig. 4. $precision@n$ and $recall@n$ of DASR and baselines on the Epinions dataset.

- SoReg [11]: SoReg defines individual-based regularization with Pearson Correlation Coefficient (PCC) in traditional matrix factorization model. The PCC-version regularization achieves the best performance, compared with other variants, as reported in [11].
- LOCABAL [17]: LOCABAL takes advantage of both local friends and users with high reputations for social recommendation.
- PTPMF [22]: PTPMF is a probabilistic matrix factorization model that incorporates the distinction of strong and weak ties.
- ASMF/ARMF [8]: ASMF and ARMF argument user-item matrix using friends' visited items as potential items. ASMF optimizes a square-loss based matrix factorization model with potential items being assigned a score lower than a user's own visited items. ARMF optimizes a ranking-based matrix factorization model which assumes that users' preference to different items are: visited items > friends' visited (potential) items > unvisited items.

4.4 Experimental Settings

In the experiments, we add a L_2 regularization term to the users' and items' embeddings when performing optimization, and the regularization coefficient is set as 0.001. We set the negative slope β of the *LeakyReLU* function as 0.2. For all latent vectors, we set their dimension as $n = 128$. We set the negative count c as 10. The learning rate decreases from an initial value of 1.0 with the increase of iterations, and the decay factor is set as 0.5.

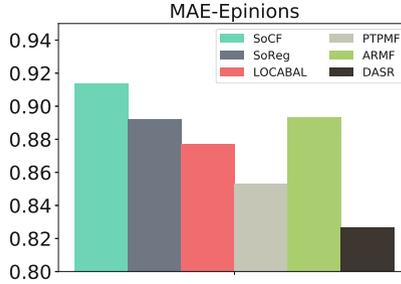


Fig. 5. MAE of DASR and baselines on the Epinions dataset.

4.5 Recommendation Results

Figure 3 and Fig. 4 present the *precision@n* and *recall@n* of all methods in comparison on the Gowalla dataset and the Epinions dataset respectively. It can be observed that the proposed DASR method achieves the best performance under different settings of n on both datasets and both metrics, which demonstrates the superiority of our method to these state-of-the-art methods.

We take Fig. 3 as an example to make a detailed discussion. Specifically, among these methods, SoCF is the only memory-based method, which directly recommend friends' items to users. Performance of SoCF is worse than other model-based methods, which learn users' and items' embeddings. SoReg integrates social relations as regularization term in matrix factorization model with feedback-based similarities as regularization coefficients, leading to that friends share similar embeddings. It achieves a good result. Besides social relations as local context, LOCABAL exploits extra social influence, i.e., users with high reputation as global social context. This makes LOCABAL outperform SoReg. PTPMF splits social relations as strong ties and weak ties, and distinguish the different influence of the two types of social ties. We can observe that the differentiation in PTPMF model benefit the recommendation performance. Since ARMF's performance is better than ASMF, we present ARMF only for comparison. It is observed ARMF is the best baseline method. This may profit from that ARMF introduce friends' visited items as potential items and it optimizes users' preference to items in a ranking manner. The proposed method, i.e., DASR, learns rich correlation patterns between users' interest by a correlation vector and finally beats these baselines.

Different from Fig. 3, we can observe that LOCABAL is better than PTPMF on the Epinions dataset in Fig. 4. This indicates that weak ties in Epinions dataset do not provide valuable suggestions for users.

4.6 Rating Prediction

Feedback in the Epinions dataset is users' rating scores for items. We also present rating prediction results of different methods in comparison, as shown in Fig. 5.

Note that, we use ASMF, rather than ARMF, since ASMF is a square loss-based method and ARMF focuses on item ranking.

It is observed that the proposed DASR achieves the best *MAE* metric on the Epinions dataset. By comparing all methods, we can find that the results of rating prediction is similar to those of recommendation results. The difference lies in that ARMF occupies the second best position in the comparison of recommendation results, while it is slightly better than SoCF only in the comparison of rating prediction. The reason may be: the value assigned to potential items cannot accurately express users' true preference and factorization on these potential values makes parameter learning deviate from a better direction.

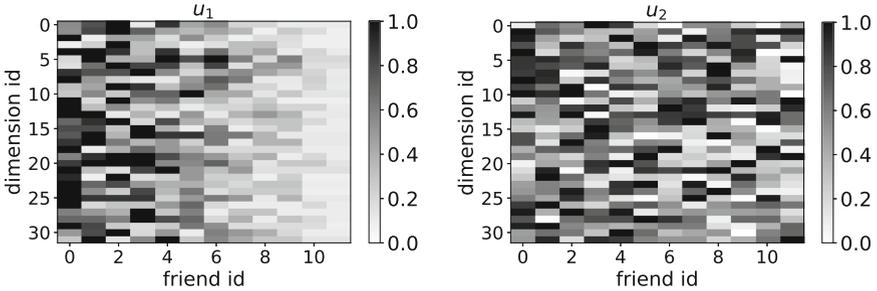


Fig. 6. The attention weights of user u_1 (left) and user u_2 (right).

4.7 Case Study: Correlation Vectors

We learn an interest correlation vector for each pair of users with social relations. Each dimension in the correlation vector represents the interest correlation in the same dimension of users' embeddings. In what follows, we study the interest correlation patterns between different pair of friends. For ease of exhibition, we set the dimension of users' embeddings as 32 and train a new DASR model to get the attention weights between each user and his friends. We select two users from our Gowalla dataset, and both them have 11 friends. Denote the two users as u_1 and u_2 respectively, we draw the heat map of the weights of attention vectors with their friends. Figure 6 shows u_1 's (left) and u_2 's (right) attention weights. For each user, we compute the norms of his correlation vectors with friends, and rearrange friend id in the descending order of the norm.

We have the following observations: (1) Most dimensions of the attention vector between u_1 and friend 0 are large values, which indicate that they are very similar and we can recommend friend 0's visited items to u_1 . (2) Each dimension of the attention vector between u_1 and friend 11 is a small value, which indicates they have no similar interest and recommending friend 11's visited items to u_1 cannot achieve a good performance. (3) By comparing u_1 and u_2 , we can find that each user's interest correlations with his friends have a specific patterns. u_1 may mainly refer to suggestions of several friends with very strong interest correlation, while u_2 may refer to suggestions of each friend dispersedly.

These observations demonstrate that the proposed method outperforms baselines, and modeling users' multi-dimensional interest correlation can significantly improve recommendation performance.

5 Conclusions

In this paper, we propose a new embedding-based social recommendation method. We use a correlation vector to characterize the high variation of users' interest correlations on all dimensions, and design a dimension-wise attention mechanism to learn the correlation vector. Moreover, we use a user's friends' embeddings to smooth the user's embedding with the correlation vector as weights, and build strong and delicate unstructured social influence. Experimental results on two real-world datasets collected from Gowalla and Epinions respectively demonstrate the superiority of our method to state-of-the-art methods.

Acknowledgments. This work was funded by the National Key Research and Development Program of China under grant number 2017YFB0803302, and the National Natural Science Foundation of China under grant numbers 61425016 and 91746301. Huawei Shen is also funded by Beijing Academy of Artificial Intelligence (BAAI) and K.C. Wang Education Foundation.

References

1. Bedi, P., Kaur, H., Marwaha, S.: Trust based recommender system for the semantic web. In: IJCAI, pp. 2677–2682 (2007)
2. Chen, J., Feng, Y., Ester, M., Zhou, S., Chen, C., Wang, C.: Modeling users' exposure with social knowledge influence and consumption influence for recommendation. In: CIKM, pp. 953–962 (2018)
3. Guy, I., Zwerdling, N., Carmel, D., Ronen, I., Uziel, E., Yogev, S., Ofek-Koifman, S.: Personalized recommendation of social software items based on social relations. In: RecSys, pp. 53–60 (2009)
4. Huang, J., Cheng, X.Q., Guo, J., Shen, H.W., Yang, K.: Social recommendation with interpersonal influence. In: ECAI, pp. 601–606 (2010)
5. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: RecSys, pp. 135–142 (2010)
6. Jiang, M., Cui, P., Chen, X., Wang, F., Zhu, W., Yang, S.: Social recommendation with cross-domain transferable knowledge. *TKDE* **27**(11), 3084–3097 (2015)
7. Jin, R., Chai, J.Y., Si, L.: An automatic weighting scheme for collaborative filtering. In: SIGIR, pp. 337–344 (2004)
8. Li, H., Ge, Y., Hong, R., Zhu, H.: Point-of-interest recommendations: learning potential check-ins from friends. In: KDD, pp. 975–984 (2016)
9. Ma, H., King, I., Lyu, M.R.: Learning to recommend with social trust ensemble. In: SIGIR, pp. 203–210 (2009)
10. Ma, H., Yang, H., Lyu, M.R., King, I.: SoRec: social recommendation using probabilistic matrix factorization. In: CIKM, pp. 931–940 (2008)
11. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: WSDM, pp. 287–296 (2011)

12. Massa, P., Avesani, P.: Trust-aware recommender systems. In: *RecSys*, pp. 17–24 (2007)
13. Meng, Y., Chen, G., Li, J., Zhang, S.: PsRec: social recommendation with pseudo ratings. In: *RecSys*, pp. 397–401 (2018)
14. O'Donovan, J., Smyth, B.: Trust in recommender systems. In: *IUI*, pp. 167–174 (2005)
15. Song, W., Xiao, Z., Wang, Y., Charlin, L., Zhang, M., Tang, J.: Session-based social recommendation via dynamic graph attention networks. In: *WSDM*, pp. 555–563 (2019)
16. Sun, P., Wu, L., Wang, M.: Attentive recurrent social recommendation. In: *SIGIR*, pp. 185–194 (2018)
17. Tang, J., Hu, X., Gao, H., Liu, H.: Exploiting local and global social context for recommendation. In: *IJCAI*, pp. 2712–2718 (2013)
18. Tang, J., Hu, X., Liu, H.: Social recommendation: a review. *SNAM* **3**(4), 1113–1133 (2013)
19. Wang, H., Ouyang, W., Shen, H., Cheng, X.: ULE: learning user and location embeddings for POI recommendation. In: *DSC*, pp. 99–106 (2018)
20. Wang, H., Shen, H., Ouyang, W., Cheng, X.: Exploiting POI-specific geographical influence for point-of-interest recommendation. In: *IJCAI*, pp. 3877–3883 (2018)
21. Wang, H., Wang, J., Zhao, M., Cao, J., Guo, M.: Joint topic-semantic-aware social recommendation for online voting. In: *CIKM*, pp. 347–356 (2017)
22. Wang, X., Hoi, S.C., Ester, M., Bu, J., Chen, C.: Learning personalized preference of strong and weak ties for social recommendation. In: *WWW*, pp. 1601–1610 (2017)
23. Wang, X., Lu, W., Ester, M., Wang, C., Chen, C.: Social recommendation with strong and weak ties. In: *CIKM*, pp. 5–14 (2016)
24. Wang, X., Zhu, W., Liu, C.: Social recommendation with optimal limited attention. In: *KDD*, pp. 1518–1527 (2019)
25. Wen, Y., Guo, L., Chen, Z., Ma, J.: Network embedding based recommendation method in social networks. In: *WWW*, pp. 11–12 (2018)
26. Wu, C., Wu, F., An, M., Huang, J., Huang, Y., Xie, X.: NPA: neural news recommendation with personalized attention. In: *KDD*, pp. 2576–2584 (2019)
27. Yang, B., Lei, Y., Liu, D., Liu, J.: Social collaborative filtering by trust. In: *AAAI*, pp. 2747–2753 (2013)
28. Yang, X., Steck, H., Liu, Y.: Circle-based recommendation in online social networks. In: *KDD*, pp. 1267–1275 (2012)
29. Yuan, Q., Chen, L., Zhao, S.: Factorization vs. regularization: fusing heterogeneous social relationships in top-n recommendation. In: *RecSys*, pp. 245–252 (2011)
30. Zhang, Q., Wu, J., Zhang, Q., Zhang, P., Long, G., Zhang, C.: Dual influence embedded social recommendation. *WWW* **21**(4), 849–874 (2018)
31. Zhao, T., McAuley, J., King, I.: Leveraging social connections to improve personalized ranking for collaborative filtering. In: *CIKM*, pp. 261–270 (2014)
32. Zhao, Z., Lu, H., Cai, D., He, X., Zhuang, Y.: User preference learning for online social recommendation. *TKDE* **28**(9), 2522–2534 (2016)