Social Attentional Memory Network: Modeling Aspect- and Friend-level Differences in Recommendation

Chong Chen
DCST, Tsinghua University
Beijing, China
cc17@mails.tsinghua.edu.cn

Min Zhang*
DCST, Tsinghua University
Beijing, China
z-m@tsinghua.edu.cn

Yiqun Liu
DCST, Tsinghua University
Beijing, China
yiqunliu@tsinghua.edu.cn

Shaoping Ma
DCST, Tsinghua University
Beijing, China
msp@tsinghua.edu.cn

ABSTRACT

Social connections are known to be helpful for modeling users' potential preferences and improving the performance of recommender systems. However, in social-aware recommendations, there are two issues which influence the inference of users' preferences, and haven't been well-studied in most existing methods: First, the preferences of a user may only partially match that of his friends in certain aspects, especially when considering a user with diverse interests. Second, for an individual, the influence strength of his friends might be different, as not all friends are equally helpful for modeling his preferences in the system. To address the above issues, in this paper, we propose a novel Social Attentional Memory Network (SAMN) for social-aware recommendation. Specifically, we first design an attention-based memory module to learn user-friend relation vectors, which can capture the varying aspect attentions that a user share with his different friends. Then we build a friend-level attention component to adaptively select informative friends for user modeling. The two components are fused together to mutually enhance each other and lead to a finer extended model. Experimental results on three publicly available datasets show that the proposed SAMN model consistently and significantly outperforms the state-of-the-art recommendation methods. Furthermore, qualitative studies have been made to explore what the proposed attention-based memory module and friend-level attention have learnt, which provide insights into the model's learning process.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; • **Computing methodologies** → **Neural networks**;

KEYWORDS

Recommender Systems, Collaborative Filtering, Attention, Memory Networks, Social Connections

ACM ISBN 978-1-4503-5940-5/19/02…$15.00
https://doi.org/10.1145/3289600.3290982

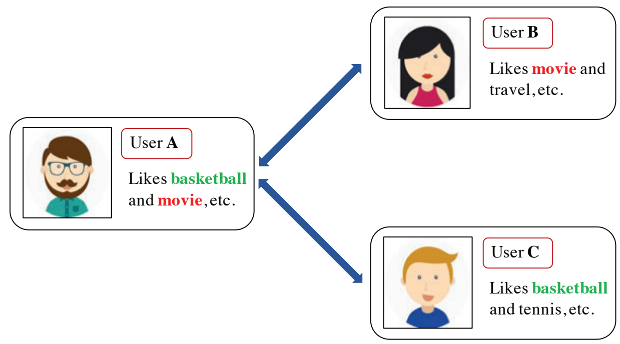

Figure 1: A possible illustration of user relationships in social network. User A is a friend of user B and user C, but the common preference aspects are distinct (movie between A and B while basketball between A and C).

ACM Reference Format:
Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2019. Social Attentional Memory Network: Modeling Aspect- and Friend-level Differences in Recommendation. In *The Twelfth ACM International Conference on Web Search and Data Mining (WSDM '19), February 11–15, 2019, Melbourne, VIC, Australia.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3289600.3290982

1 INTRODUCTION

Users expect personalized products and information in modern E-commerce, entertainment and social media platforms. In this case, recommender systems are designed to generate personalized item recommendations and deal with the information overload problem. Many recommendation methods are based on Collaborative Filtering (CF) [14, 17, 18, 26], which mainly makes use of users' historical records such as ratings, clicks, and purchases.

Recently, owing to the prevalence of social media, many E-commerce sites have become popular social platforms that help users discuss and select items [25], such as Delicious, Ciao and Epinions. In these social applications, users like to spread their preferences of items to their social connections, and a user's preferences can not only be inferred from the items he bought and clicked, but also can be inferred from his social connections. Generally, there are two types of social connections: friends in undirected social networks and followers in directed social networks. Since we do not

*Corresponding author

focus on the differences between the two types, for the convenience of description, we refer to the user's social connections as **friends** in this paper. As shown in previous studies on social-aware recommender systems, the social behavior of users and their interactions with items are positively correlated [22, 25, 37, 42]. By considering users' social connections, social-aware methods can utilize a much larger volume of data to tackle the data sparsity issue, and further improve the performance of recommender systems.

However, in social-aware recommendations, there are two issues which influence the inference of users' preferences, and haven't been well-studied in most existing methods.

The first one is **aspect-level differences**. Generally, users and their friends only have the same preferences in certain aspects. It is well recognized that the preference of a user can be used to infer his friends' preference and vice versa, which could be denoted as an influence vector. Existing methods like [19, 29, 38] assume that this vector keeps the same when facing different friends. However, a user may pay the most attention to one aspect for a friend but focus on another aspect for a different friend. In Figure 1 we show an example that is common in social relationships. User A is a friend of user B and user C, but the reasons are distinct: user A and B are friends because they are both interested in movie, while A and C are friends because they both like basketball. The aspect-level differences should be considered when building a social-aware recommender system. However, since users' preference inference is usually complex and non-linear, the aspect differences are hard to be captured by traditional latent factor based models.

The second issue is **friend-level differences**. For a user, the influence strength of his friends should be different and dynamic. Each user is associated with a set of friends in social networks, but it does not necessarily indicate that every friend has equal influence strength on his behaviors. For example, when buying basketball shoes, a user will follow the advice of his friends who play basketball, but when it comes to a trip, he will turn to those who love traveling. In previous work, the social influence strength is usually set equally for the social connections [15], or relied on a predefined static function [9, 16]. These settings are not robust in real life. To better characterize a user's preferences, the model requires different attentions on the set of the user's friends.

Motivated by the above observations, we propose to model both aspect-level differences and friend-level differences for improving the performance of social-aware recommendation. In this paper, we present a Social Attentional Memory Network (or **SAMN** for short), which utilizes the recent advances in memory networks [23, 31] and neural attention mechanisms [2, 3, 34]. Specifically, we first design an attention-based memory module to learn the user-friend specific relation vectors, and then employ friend-level attention to automatically select informative friends for user preference modeling. The memory component allows reading and writing operations to encode complex user and friend relations. An associative attention-based addressing scheme places higher weights on aspects in which user and his friend share similar preferences. The attention-based memory module is controlled by the user-friend interaction, making the learned relation vector corresponds to each user-friend pair. In the friend-level attention modeling process, a two-layer attention network is adopted to model the influence strength among users' friends in a distant supervised manner. Then, the two components are fused together to mutually enhance each other via an end-to-end training process. We evaluate SAMN extensively on three real-world datasets. Experimental results show that our model consistently outperforms the state-of-the-art methods, and also verify the effectiveness of our designed attention component and memory network.

The main contributions of this work are summarized as follows.

(1) We propose a new model for social-aware recommender systems, which considers both aspect-level differences among user-friend co-preferences and friend-level differences on social influence strength.
(2) To the best of our knowledge, we are the first to employ an attention-based memory module to construct user-friend specific relation vector. We also introduce the friend-level attention to adaptively measure the social influence strength among users' friends. These two parts are fused in a unified framework and can be learned through efficient end-to-end training.
(3) Through extensive experiments conducted on three benchmark datasets, we show that SAMN consistently outperforms the state-of-the-art models.

2 RELATED WORK

2.1 Traditional Collaborative Filtering

Among the various collaborative filtering methods, matrix factorization (MF) is the most popular one, and is also the basis of many effective recommender models [27, 30]. Popularized by the Netflix Challenge, early MF methods [18] were designed to model users' explicit feedback by mapping users and items to a latent factor space, such that user-item relationships (ratings) can be obtained by their latent factors' dot product.

Later on, some researchers found that a well-designed MF model in rating prediction may not perform well in Top-K recommendation, and called on recommendation research to focus more on the ranking task [6]. In this case, Rendle et al. [26] first proposed a pairwise learning method BPR, which is a sample-based method that optimizes the model based on the relative preference of a user over pairs of items. Then, the pairwise learning strategy has been widely used to optimize recommender models [3, 34, 39–41]. and become a dominant technique in recommendation. In our work, we also adopt BPR as our basic learning model because of its effectiveness in exploiting the unobserved user-item feedback.

2.2 Social-aware Recommendation

In the last few years, there is a large literature exploiting users' social connections for improving the recommendation performance [25]. Most studies assumed that a user's decision can be affected by his friends' opinions and behaviors. E.g., in [42] , the authors assumed that users are more likely to have seen items consumed by their friends, and exploited this effect to extend BPR [26] by changing the negative sampling strategy. Jamali et al. designed a social influence propagation based model in latent based recommendation models (SocialMF) [15]. Based on a generative influence model, the work [38] exploits social influence from friends for item recommendation by leveraging information embedded in the user social network. However, many existing methods [19, 29, 38] assume that users' influence vectors stay the same for different friends, thus the aspect-level differences are not well-studied.

In social-aware recommendation, social influence strength modeling is a central problem [8, 9]. E.g., Goyal et al. [9] designed a model to calculate influence strength from users' historical behaviors. However, in most of the existing methods, the social influence strength is assumed equal among friends [15] or with a simple metric from other sources [9, 16] (e.g., the strength between their interactions in the past).

To the best of our knowledge, few has explored the neural networks for modeling aspect-level differences and friend-level differences in social-aware recommendation, which is the main concerns of our work.

2.3 Deep Learning in Recommendation

Recently, deep learning has yielded an immense success in many fields like computer vision, speech recognition and natural language processing [20]. Some researchers also tried to exploit different neural network structure for improving the performance of recommendations. In [11]. He et al. presented a Neural Collaborative Filtering (NCF) framework to address implicit feedback by jointly learning a matrix factorization and a feedforward neural network, NCF is also the state-of-art recommendation method for using only user-item historical records. Later, Neural Factorization Machines (NFM) [10] was developed to enhance FM by modeling higher-order and non-linear feature interactions. More recently, [32] presents an attentive recurrent network for temporal social-aware recommendation (ARSE). There are two major differences between our work and ARSE: (1) We focus on a more general problem while ARSE focuses on temporal recommendations via Recurrent Neural Network (RNN) and attention mechanisms. (2) Our work introduces memory network to address the problem of aspect-level differences between users and their friends.

Attention mechanism has been shown effective in many machine learning tasks such as image captioning and machine translation [1, 28]. In the field of recommendation, [3] introduced both component-level and item-level attention into a CF framework for multimedia recommendation. [36] improved FM by learning the importance of different feature interactions via a neural attention network. Recently, Chen et al. [2] proposed to learn the "usefulness" of reviews with the help of attention mechanism for improving the performance and explainability of the recommender system.

Memory networks are recently introduced frameworks that combine reasoning, attention and memory for solving tasks in the areas of language understanding and dialogue. It generally consist of two components: an external memory typically a matrix and a controller which perform operations on the memory (e.g., read, write). The memory component increases model capacity independent of the controller while providing an internal representation of knowledge to track long-term dependencies and perform reasoning. The controller usually manipulates these memories with content-based addressing, which finds a scoring function between the given query and a passage of text [23, 31, 35]. For recommendations, [34] utilized memory module to learn the relationships between user-item interactions for extending Collaborative Metric Learning [13]. In [35], the authors proposed Collaborative Memory Network, while

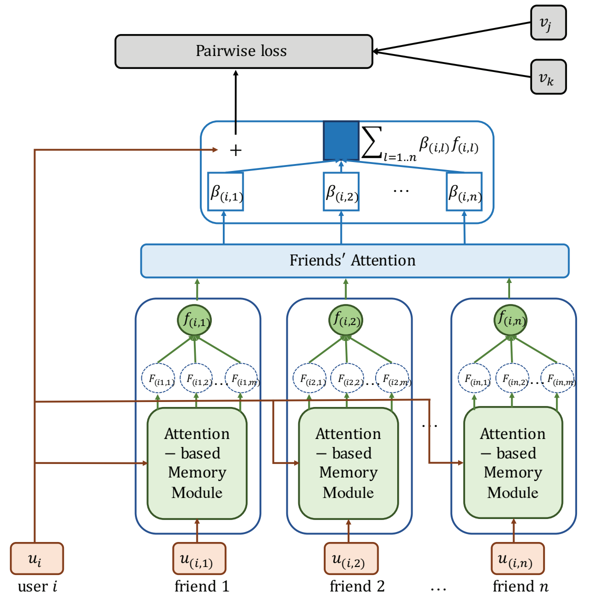

Figure 2: The architecture of our proposed Social Attentional Memory Network (SAMN). Our model contains two major components, which are the attention-based memory module and the friend-level attention component.

the associative addressing scheme of the memory module acts as a nearest neighborhood model identifying similar users.

3 SOCIAL ATTENTIONAL MEMORY NETWORK (SAMN)

In this section, we introduce our Social Attentional Memory Network (SAMN). First, we will present the general architecture of SAMN. Then, we will show the detailed formulations of our proposed attention-based memory module and friend-level attention respectively, which are the main concerns in this paper. Lastly we will go through the optimization details of SAMN.

3.1 Overview of SAMN

The goal of our model is to make recommendations based on implicit feedback and social networks. Both aspect-level and friend-level differences are considered for improving the model performance and generalization. The architecture of the proposed model is shown in Figure 2. From the figure, we can make a simple high-level overview of our model:

(1) Users and items are converted to dense vector representations using an Embedding Layer. u and v are the user and item vectors respectively.
(2) The model contains two major components, which are the attention-based memory module and the friend-level attention component. The attention-based memory module is designed for addressing the problem caused by aspect-level differences. The friend vector $f_{(i,l)}$ is generated using a neural attention mechanism over an augmented memory matrix M. It dependents on user and friend, and is learned to represent the preference relationship between user and his friend. The friend-level

attention is used to select informative friends for better inferring user preference.

(3) The model is optimized by pairwise ranking and negative sampling strategy.

3.2 Attention-based Memory Module

In most cases, users and their friends only have the same preferences in certain aspects, especially when considering a user with diverse interests. However, explicit relations between user-friend pairs are not available in implicit data (we don't know what their shared interest aspects are). Motivated by the recent advance in memory network and attention mechanisms, we designed a new attention-based memory module to learn the relation vectors between users and their friends. The structure of the module is shown in Figure 3. The memory matrix of the module is represented as $M \in \mathbb{R}^{N \times d}$, where d is the dimension of the user and item embeddings and N is the memory size. In matrix M, each slice is noted $M_j \in \mathbb{R}^d$ as a memory slice. The input of the module is a user-friend pair $(u_i, u_{(i,l)})$, where u_i denotes user i, and $u_{(i,l)}$ denotes the l-th friend of user i. The module returns the vector $f_{(i,l)}$, which represents the relationship between u_i and $u_{(i,l)}$.

3.2.1 ***Joint Embedding.*** Given the user-friend pair $(u_i, u_{(i,l)})$, the module first apply the following operation to learn a joint embedding of users and their friends. The denominator added in Eq. (1) is used to normalize and make the generated vectors have the same scale.

$$s = \frac{u_i \odot u_{(i,l)}}{\|u_i\| \left\|u_{(i,l)}\right\|} \tag{1}$$

where $\odot$ denotes the element-wise product of vectors. The generated vector $s \in \mathbb{R}^d$ is of the same dimension of u_i and $u_{(i,l)}$. Note that other functions like the multi-layered perceptron (MLP) or just element-wise product without normalization can also be adopted, but we found that the method of Eq. (1) performs better.

3.2.2 ***Key Addressing.*** After we obtain the joint embedding vector s, the attention vector is learned from a key matrix $K \in \mathbb{R}^{N \times d}$. Each element of the attention vector α is defined as:

$$\alpha_j^* = s^T K_j \tag{2}$$

where $K_i \in \mathbb{R}^d$ and the generated vector $\alpha \in \mathbb{R}^N$. Then the final attention scores are obtained by normalizing α using the softmax function:

$$\alpha_j = \frac{exp(\alpha_j^*)}{\sum_k exp(\alpha_k^*)} \tag{3}$$

3.2.3 ***Generating Friend Vector.*** In this step, the friend embedding $u_{(i,l)}$ is first extended to a matrix via the memory matrix M:

$$F_j = u_{(i,l)} \odot M_j \tag{4}$$

where $\odot$ denotes the element-wise product of vectors. The matrix $F \in \mathbb{R}^{N \times d}$ can be interpreted as a storage of conceptual building blocks that used to describe the friend preferences in different latent aspects (N can be seen as the number of latent aspects).

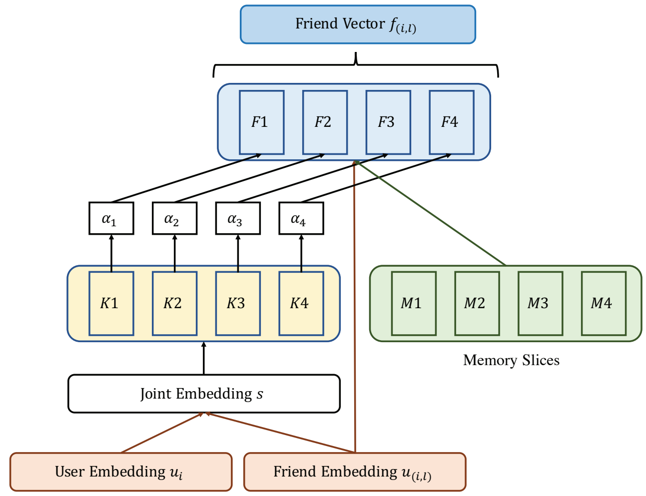

Figure 3: Illustration of our designed attention-based memory module. The module is characterized by its key-addressed end-to-end architecture which learns user-friend specific relation vectors. The memory size is set to 4 in this example.

Finally, to generate the friend vector, we use the attention scores to calculate a weighted representation of F:

$$f_{(i,l)} = \sum_j \alpha_j F_j \tag{5}$$

The output is a specific relation vector $f_{(i,l)}$, which can be seen as the influence vector of user i's l-th friend to user i's preferences.

Let $f_{(i,1)}, f_{(i,2)}, \ldots f_{(i,n)}$ be the the relation vectors of user i's friends generated by the attentional memory module. As we have mentioned in section 1, generally not every friend has the same importance for inferring a user's preference in real life. To address this problem, we introduce friend-level attention into our model, which can help to adaptively learn the influence strength of each friend.

3.3 Friend-level Attention

Attention mechanism has been widely adopted in many fields, such as computer vision [4], machine translation [1] and recommendation [2, 3, 36]. The goal of the friend-level attention is to assign non-uniform weights to users' friends, and the weights are varied when the user interacts with different items. Intuitively, if a friend has more expertise on an item (or items of the similar type), he should have a larger influence on the user's choice on the item. Formally, a two-layer network is applied to compute the attention score $\beta_{(i,l)}$ with user (u_i), current item (v_j) and friend vector $(f_{(i,l)})$ as inputs:

$$\beta^*_{(i,l)} = h^T ReLU(W_1 u_i + W_2 v_j + W_3 f_{(i,l)} + b) \tag{6}$$

where $W_1 \in \mathbb{R}^{d \times k}$, $W_2 \in \mathbb{R}^{d \times k}$, $W_3 \in \mathbb{R}^{d \times k} b \in \mathbb{R}^k$, $h \in \mathbb{R}^k$ are model parameters, k denotes the dimension of attention network, and $ReLU$ [24] is a nonlinear activation function.

Then, the final friend-level attention is normalized with a softmax function, which is a common practice in neural attention network. It makes the attention network a probabilistic interpretation, which can also deal with the problem that users may have different number of friends:

$$\beta_{(i,l)} = \frac{exp(\beta^*_{(i,l)})}{\sum_{j \in S_i} exp(\beta^*_{(i,j)})} \quad (7)$$

where S_i denotes all friends that user i has in the social network.

After we obtain the attention weight of each friend, the final representation of user i is through the sum:

$$U_i = u_i + \sum_{l \in S_i} \beta_{(i,l)} f_{(i,l)} \quad (8)$$

which considers both the user's own preference and the influence of his friends. Note that there have been a lot of work exploring the strategies of combine different features, such as concatenation, addition, or element-wise product. In this work, we adopt the addition fusion method, which has been applied in RBLT [33], NARRE [2] and A^{3}NCF [5] and achieves good performance. It is worth mentioning that we also tried to add a fully-connected neural layer after the fusion step. However, it leads to inferior performance duo to the overfitting problem.

3.4 Learning

3.4.1 ***Prediction***. After completing the model training process, the recommendation task is reduced to a ranking problem among all the items in the dataset based on estimated score $\bar{R}_{ij}$. Our prediction part is built on Matrix Factorization (MF), which is state-of-the-art for rating prediction as well as modeling implicit feedback [18]:

$$\bar{R}_{ij} = \left(u_i + \sum_{l \in S_i} \beta_{(i,l)} f_{(i,l)} \right)^T v_j \quad (9)$$

The items (unclicked/ uncomsumed) are ranked in descending order of $\bar{R}_{ij}$ to provide the Top-K item recommendation list.

3.4.2 ***Optimization***. Our objective is to study implicit feedback which is more pervasive in practice and can be collected automatically (e.g. clicks, comsumes). To this end, we opt BPR pairwise learning objective [26], a commonly used objective function in many previous studies [3, 34, 39–41]. For each positive user-item pair$< u_i, v_j >$, we randomly sample a negative item from the unobserved items of the user, which is denoted as v_k. The pairwise ranking loss is as follows:

$$L_{BPR} = \sum_{(i,j,k) \in \mathcal{D}} -ln\sigma\left(\bar{R}_{ij} - \bar{R}_{ik}\right) + \lambda_\Theta(\|\Theta\|^2) \quad (10)$$

where $\sigma(x) = 1/(1 + exp(-x))$ is the logistic sigmoid function, $\mathcal{D}$ denotes the set of pairwise training instances. and λ_Θ controls the strength of regularization, which is a L_2 norm to prevent overfitting.

To optimize the objective function, we adopt mini-batch Adagrad [7] as the optimizer. Its main advantage is that the learning rate can be self-adapted during the training phase, which eases the pain of choosing a proper learning rate and leads to faster convergence than the vanilla SGD.

Table 1: Statistical details of the evaluation datasets. "Interaction" means user-item historical records, and "Social link" denotes user-friend connections in social network.

	#User	#Item	#Interaction	#Social Link
Delicious	1,521	1,202	8,397	10,401
Ciao	7,267	11,211	157,995	111,781
Epinions	38,089	23,585	488,917	433,416

4 EXPERIMENTS

4.1 Experimental Settings

4.1.1 ***Datasets***. In our experiments, we used three publicly accessible datasets to evaluate the performance of our model, which are **Delicious**[1], **Ciao**[2], and **Epinions**[3]. We briefly introduce the three datasets:

- **Delicious**: This dataset contains social connections, book-marking, and tag information from a set of 2K users from Delicious Social Bookmarking System. In this paper we only use the social connections and book-marking records to train our model and baseline methods.
- **Ciao**: This dataset contains users' ratings to the items they have purchased and the social connections between users. Since we focus on the implicit feedback, we transform the detailed ratings into a value of 0 or 1 indicating whether the user has rated the item.
- **Epinions**: Epinions is a who-trust-whom directed online social network that provides product rating and review service. The corresponding rating is also assigned to a value of 1 (as implicit feedback) in our experiments.

All the datasets were preprocessed to make sure that all items have at least five ratings. The statistical details of these datasets are presented in Table 1.

4.1.2 ***Baselines***. To evaluate the performance of Top-K recommendation, we compare SAMN with the following methods. Note that all models are learned by optimizing the same pairwise ranking loss of BPR (cf Eqn. (10)) for a fair comparison.

- **BPR** [26]: This method optimizes MF with the BPR objective function. It is a highly competitive method for implicit feedback based recommendation.
- **SBPR** [42]: This is a ranking model that considers social relationships in the learning process, assuming that users tend to assign higher ranks to items that their friends prefer.
- **SocialMF** [15]: This is a classical model for social-aware recommendation. It incorporate the social influence among users into classical latent factor models, where the influence strength is simply set equally for all social connections.
- **NCF** [11]: This is a recently proposed state-of-the-art deep learning based framework that combines matrix factorization (MF) with a multilayer perceptron model (MLP) for item ranking. Since NCF shows superior performance over the traditional weighting methods WMF [14] and eALS [12] according to [11], we do not further compare with the performance of WMF and eALS.

[1] https://grouplens.org/datasets/hetrec-2011/
[2] http://www.jiliang.xyz/trust.html
[3] https://alchemy.cs.washington.edu/data/epinions/

Table 2: Performance comparison on three datasets for all methods. Best performance is in boldface and second best is underlined. SAMN achieves best performance on all datasets, outperforming many strong neural baselines. * and ** denote the statistical significance for p < 0.05 and p < 0.01, respectively, compared to the best baseline.

Delicious	Recall@10	Recall@20	Recall@50	NDCG@10	NDCG@20	NDCG@50
BPR	0.1301	0.1601	0.2151	0.0930	0.1013	0.1136
SBPR	0.1358	0.1715	0.2534	0.0888	0.1024	0.1216
SocialMF	0.1337	0.1688	0.2433	0.0952	0.1044	0.1204
NCF	0.1280	0.1632	0.2401	0.0887	0.1002	0.1165
SNCF	<u>0.1473</u>	<u>0.1846</u>	<u>0.2705</u>	<u>0.0971</u>	<u>0.1086</u>	<u>0.1274</u>
NFM	0.1422	0.1789	0.2612	0.0955	0.1075	0.1147
SAMN	**0.1624****	**0.2033****	**0.2837****	**0.1034****	**0.1151****	**0.1325****

Ciao	Recall@10	Recall@20	Recall@50	NDCG@10	NDCG@20	NDCG@50
BPR	0.0644	0.0994	0.1625	0.0452	0.0547	0.0711
SBPR	0.0651	0.1011	0.1645	0.0462	0.0572	0.0721
SocialMF	0.0657	0.1004	0.1638	0.0469	0.0568	0.0717
NCF	0.0677	0.1013	0.1634	0.0477	0.0581	0.0729
SNCF	<u>0.0722</u>	<u>0.1051</u>	<u>0.1725</u>	<u>0.0511</u>	<u>0.0619</u>	<u>0.0792</u>
NFM	0.0717	0.1034	0.1697	0.0509	0.0611	0.0778
SAMN	**0.0747****	**0.1083****	**0.1753****	**0.0533****	**0.0634****	**0.0806****

Epinions	Recall@10	Recall@20	Recall@50	NDCG@10	NDCG@20	NDCG@50
BPR	0.0546	0.0836	0.1399	0.0361	0.0451	0.0596
SBPR	0.0557	0.0846	0.1411	0.0365	0.0456	0.0598
SocialMF	0.0542	0.0822	0.1387	0.0354	0.0447	0.0585
NCF	0.0553	0.0840	0.1404	0.0363	0.0454	0.0597
SNCF	0.0561	<u>0.0852</u>	0.1415	0.0366	0.0457	<u>0.0603</u>
NFM	<u>0.0564</u>	0.0848	<u>0.1417</u>	<u>0.0371</u>	<u>0.0459</u>	0.0601
SAMN	**0.0575***	**0.0869****	**0.1440****	**0.0380***	**0.0469***	**0.0617****

- **SNCF**: NCF [11] is designed suitable for recommendation with side information. To adjust NCF for modeling social relations, we plug user's friends into the input feature vector and concatenate the feature vector with user ID embedding, dubbed this enhanced model as SNCF.
- **NFM** [10]: This is a recently proposed Neural Factorization Machine. It is one of the state-of-the-art deep learning methods, which uses Bi-Interaction Layer to integrate both features and historical feedback information. In our experiments, we treat users' social connections as features. Since the original NFM is designed for regression, we changed the optimize function to BPR (cf Eqn. (10)) to fit our task.

To the best of our knowledge, few has explored the neural networks for social-aware recommendation. Thus we compare with traditional social-aware methods SBPR and SocialMF, content-based neural method NFM, and the extended neural model SNCF. We leave out the comparison with ARSE [32], which is designed for temporal social-aware recommendation, because the performance difference may be caused by the temporal information.

4.1.3 ***Evaluation Metrics.*** To evaluate the performance of all algorithms, we calculate *Recall@K* and *NDCG@K* [21, 39]. Intuitively, the *NDCG@K* metric is a position-aware ranking metric while *Recall@K* metric considers whether the ground truth is ranked among the top K items. When K is fixed, the Precision is only determined by true positives while Recall is determined by both true positives and positive samples [39]. To give a more comprehensive evaluation, we exhibit Recall rather than Precision and F1-score (F1-score is almost determined by Precision since Precision is much smaller than Recall in our experiments). For each user, these metrics are computed as follows:

$$Recall@K = \frac{\sum_{j=1}^{K} rel_j}{min(K, |y_u^{test}|)};$$
$$DCG@K = \sum_{j=1}^{K} \frac{2^{rel_j} - 1}{log_2(j+1)}; \quad NDCG@K = \frac{DCG@K}{IDCG@K} \tag{11}$$

Where $rel_j = 1/0$ indicates whether the item at rank j in the Top-K recommendation list is in the test set, $|y_u^{test}|$ denotes the number of items rated by user u in the test set. The notion *IDCG* means the maximum possible *DCG* through ideal ranking. Each metric is the average for all users.

4.1.4 ***Experiments Details.*** We randomly split the dataset into training (70%), validation (20%), and test (10%) sets. The validation set was used for tuning hyper-parameters and the final performance comparison was conducted on the test set. The parameters for baseline methods were initialized as in the corresponding papers, and were then carefully tuned to achieve optimal performances. The learning rate for all models are tuned amongst [0.005, 0.01, 0.02, 0.05]. The batch size was tested in [64, 128, 256] and the latent factor number was tested in [32, 64, 128]. To prevent overfitting, we turned the margin λ_Θ in [0.001, 0.005, 0.01, 0.02]. For SAMN, the number of memory slices in M is tuned amongst [8, 16, 32, 64]. After the turning process, we set the latent factor number d=128, learning rate lr=0.05, λ_Θ=0.01. The memory size N is set 8 for Delicious and 16 for Ciao and Epinions. To evaluate on

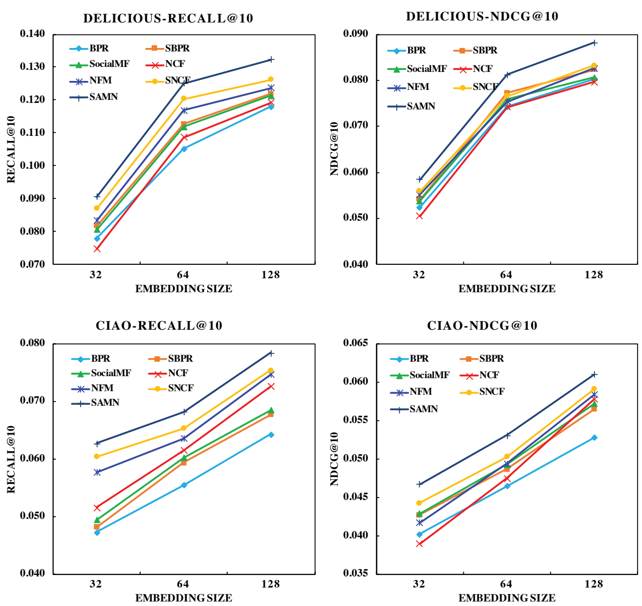

Figure 4: Performance comparison on two datasets w.r.t. different embedding sizes (validation sets).

different recommendation lengths, we set K = 10, 20 and 50 in our experiments.

4.2 Comparative Analysis on Overall Performances

The empirical results of our proposed SAMN and baselines on three datasets are given in Table 2. From the results, we can make the following observations:

Firstly, methods utilizing social information generally perform better than those without social information. For example, in Table 2, the performance of SBPR is better than BPR, SNCF performs better than NCF, and SAMN performs better than BPR and NCF. This is not surprising, since the social information is complementary to users' historical records, it can help to increase the learning accuracy of user preference.

Secondly, our method SAMN consistently and significantly outperforms all the baselines including neural methods NCF, SNCF and NFM. Specifically, SAMN outperforms the best baseline by performance gains about 6.49% on Delicious, 4.31% on Ciao and 2.43% on Epinions on the NDCG@10 metric. The performance gains on other metrics are also similarly high.

Thirdly, since SAMN share the same loss function with other baseline methods, we can attribute the performance increase to the proposed attention-based memory module and friend-level attention. In our model, the aspect-level differences and friend-level differences are considered, which allow the social information to be modeled with a finer granularity and thus lead to a better performance.

We also conduct experiments to test the influence of latent factor size d on validation sets. The results are shown in Figure 4. Duo to the space limitation, we only show the results of Delicious and Ciao datasets on Recall@10 and NDCG@10 metrics. The result of Epinions is similar to that of Ciao. As can be seen from this figure, our model outperforms all the other models with different values of d for the two ranking metrics on two datasets. Moreover, as the latent dimension size increases, the performance of all models increase. This indicates that a larger dimension could capture more hidden factors of users and items, which is beneficial to Top-K recommendation due to the increased modeling capability.

Table 3: Comparison of the variant models of SAMN. u_i is the id embedding of user i, $|S_i|$ is the number of user i's friends, $f_{(i,l)}$ is generated by the attention-based memory module (cf. Section. 3.2), and $\beta_{(i,l)}$ is generated by the friend-level attention (cf. Section. 3.3).

Variants	Representation of user	Social	Aspect-level differences	Friend-level differences
BPR	u_i	\	\	\
SE	$u_i + \sum_{l \in S_i} \frac{1}{\lvert S_i \rvert} u_l$	√	\	\
SAM	$u_i + \sum_{l \in S_i} \frac{1}{\lvert S_i \rvert} f_{(i,l)}$	√	√	\
SFA	$u_i + \sum_{l \in S_i} \beta_{(i,l)} u_l$	√	\	√
SAMN	$u_i + \sum_{l \in S_i} \beta_{(i,l)} f_{(i,l)}$	√	√	√

4.3 Effect of Attention-based Memory Module and Friend-level Attention

The key characteristics in our proposed model SAMN are the two newly designed components for social information modeling: the attention-based memory module that captures the user-friend specific relationship, and the friend-level attention that models social influence strength by adaptively learning the weight of each friend. In this subsection, we discuss the effect of attention-based memory module and friend-level attention. Specifically, we compare the effect of each component by constructing the following variants of SAMN:

- **BPR**: This is our basic collaborative filtering model without any social information. It is added as a baseline of other variants.
- **SE**: This is a variant model utilizing social information only by social embedding.
- **SAM**: This is a variant model utilizing social information with the attention-based memory module only.
- **SFA**: This is a variant model utilizing social information with the friend-level attention only.
- **SAMN**: This is our proposed model utilizing social information with both the attention-based memory module and the friend-level attention.

The characteristics of the variant models are listed in Table 3.

Figure 5 shows the performance of different variants. Due to the space limitation, we also only show the results of Delicious and Ciao datasets on Recall@10 and NDCG@10 metrics. As shown in the figure, BPR performs the worst since no social relationships are utilized to provide the extra information. With the social information of friend embedding, SE performs better, but still worse than SAM and SFA. Because just friend embedding is too crude and can provide very limited social information to help infer users' preferences. The performances of SAM and SFA are significantly better than SE (p<0.05), which shows that both the attention-based memory module and the friend-level attention can help to better utilize social information by considering aspect-level differences

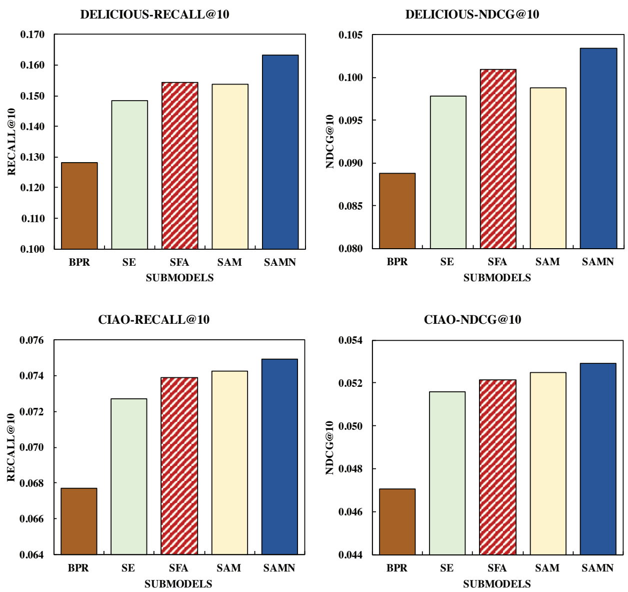

Figure 5: Effect of different components in SAMN. Note that the significant improvements for p <0.05 are achieved (SAM vs SE, SFA vs SE, SAMN vs SAM and SFA, test sets).

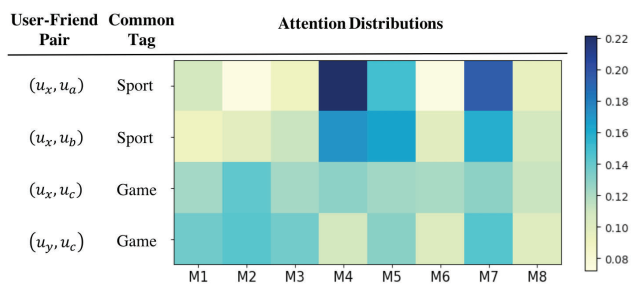

Figure 6: Attention distributions for user-friend pairs of different tags on Delicious. The color scale indicates the intensities of the weights, where a darker color indicates a higher value and a lighter color indicates a lower value. Note that the tag information is not used during the training process.

and friend-level differences respectively. Moreover, generally SAM performs better than SFA, this may because that SAM can capture user-friend specific relation vectors through the advantage of memory network, which is more flexible than SFA that only uses the friend-level attention. Lastly, our proposed SAMN, which utilizes both the attention-based memory module and the friend-level attention, performs best (significantly better than SAM and SFA for p<0.05). The reason is that these two components are not conflict with each other and can be used to model users' social influence collaboratively.

4.4 Attention Analysis

The attention weights reflect how the model learned and recommend. In this subsection we conducted experiments to show how the attention-based memory module and friend-level attention component work.

Table 4: Case studies of friend-level attention of a sampled user from Ciao. The friend weights of the user for positive items (Item #130, #212, #1258) and negative items (Item #29, #1105, #3367) are shown.

	Friend #782	Friend #1391	Friend #1446	Friend #1505
Item #130	0.212	0.057	0.319	**0.412**
Item #212	0.145	0.086	**0.517**	0.252
Item #1258	**0.533**	0.121	0.079	0.267
Item #29	0.286	0.103	**0.315**	0.296
Item #1105	**0.434**	0.187	0.315	0.064
Item #3367	0.308	0.075	**0.382**	0.235

We first focus on the attention-based memory module, which is designed to model the aspect-level differences between user-friend relationships. Ideally, if a user-friend pair has the same interests in certain aspects, it can be reflected through the attention distributions of memory slices. To better understand the attention results, we make use of users' tag information contained in Delicious dataset, which can be taken as users' explicit preferences. We take some examples to show how the attention weights identify different co-preferences between users and their friends in Figure 6. From the figure, we can see that for user-friend pair (u_x, u_a) and (u_x, u_b), the attention distributions of memory slices are similar, which means that u_a and u_b share similar interest aspects with u_x. Since they all have the tag "sport", we can imagine that the shared interest aspects may be sports-related matters. For pair (u_x, u_c), the distribution is different from the above two pairs. The reason may because that the shared interest aspects of u_x and u_c are games, not sports. With the help of attention mechanisms, our model can capture users' attention weights on different aspects of each friend, and thus could achieve more accurate predictions (cf. Section. 4.3).

Apart from the attention-based memory module, another key advantage of SAMN is its ability in adaptively measuring the influence strength of users' friends. To show this, we randomly selected a user who has four friends (#782, #1391, #1446, and #1505) in social network form Ciao dataset. We then randomly picked three items (#130, #212, and #1258) which have been purchased by the user, and three negative items (#29, #1105, and #3367) which have not been purchased. Table 4 shows the attention weights of the user's friends for the randomly selected items. We have the following observations: (1) For different item, the attention weights of the user's friends vary significantly. For example, when predicting the user's preference on item #1258, the attention weight of friend #782 are relatively high. The reason may because that the friend #782 has purchased item #1258 (according to the dataset), and thus he has a bigger influence strength for the user's purchasing behavior on this item. (2) The attention weights can also somehow reflect the richness of friend users' feedback information. For example, the friend #1391 has only purchased 2 items according to the dataset. For both positive items and negative items, his attention weights are relatively low, which means that his information is not rich enough to provide significant influence for the user's purchasing behaviors.

5 CONCLUSION

Social information plays a very important role for improving the performances of recommender systems. However, there are two differences: aspect-level differences among user-friend co-preferences and friend-level differences on social influence strength, have not been well-studied in existing methods. In this paper, we proposed a new model, which unifies the strengths of memory networks and attention mechanisms to address the problems in social-aware recommendations. To the best of our knowledge, we are the first to employ the attention-based memory module to construct user-friend specific relation vector. We also design the friend-level attention to adaptively measure the social influence strength among users' friends. Extensive experiments have been made on three real-life datasets. The proposed SAMN (Social Attentional Memory Network) consistently and significantly outperforms the state-of-the-art recommendation models on different evaluation metrics. Moreover, we performed qualitative analyses of the attentional memory module and the friend-level attention, which helps understand what the model has learnt and prove the rationality of our model.

In the future, we are interested in exploring the differences between the two types of social connections: friends in undirected social networks and followers in directed social networks. Moreover, since the social information can be used to explain the recommendation results, we also like to investigate and improve the explainability of our model.

ACKNOWLEDGMENTS

We would like to thank professor Maarten de Rijke for his valuable suggestions. Thanks also to our anonymous WSDM reviewers for the helpful comments and suggestions for making this paper better. This work is supported by the Natural Science Foundation of China under Grant No.: 61672311 and 61532011.

REFERENCES

[1] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473* (2014).

[2] Chong Chen, Min Zhang, Yiqun Liu, and Shaoping Ma. 2018. Neural Attentional Rating Regression with Review-level Explanations. In *Proceedings of WWW*. 1583–1592.

[3] Jingyuan Chen, Hanwang Zhang, Xiangnan He, Wei Liu, Wei Liu, and Tat Seng Chua. 2017. Attentive Collaborative Filtering: Multimedia Recommendation with Item- and Component-Level Attention. In *Proceedings of SIGIR*. 335–344.

[4] Long Chen, Hanwang Zhang, Jun Xiao, Liqiang Nie, Jian Shao, Wei Liu, and Tat-Seng Chua. 2016. Sca-cnn: Spatial and channel-wise attention in convolutional networks for image captioning. *arXiv preprint arXiv:1611.05594* (2016).

[5] Zhiyong Cheng, Ying Ding, Xiangnan He, Lei Zhu, Xuemeng Song, and Mohan S Kankanhalli. 2018. Aˆ 3NCF: An Adaptive Aspect Attention Model for Rating Prediction.. In *Proceedings of IJCAI*. 3748–3754.

[6] Paolo Cremonesi, Yehuda Koren, and Roberto Turrin. 2010. Performance of recommender algorithms on top-n recommendation tasks. In *Proceedings of RecSys*. 39–46.

[7] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12, Jul (2011), 2121–2159.

[8] Eric Gilbert and Karrie Karahalios. 2009. Predicting tie strength with social media. In *Proceedings of SIGCHI*. 211–220.

[9] Amit Goyal, Francesco Bonchi, and Laks VS Lakshmanan. 2010. Learning influence probabilities in social networks. In *Proceedings of WSDM*. 241–250.

[10] Xiangnan He and Tat-Seng Chua. 2017. Neural factorization machines for sparse predictive analytics. In *Proceedings of SIGIR*. 355–364.

[11] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of WWW*. 173–182.

[12] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proceedings of SIGIR*. 549–558.

[13] Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. 2017. Collaborative metric learning. In *Proceedings of WWW*. 193–201.

[14] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative filtering for implicit feedback datasets. In *Proceedings of ICDM*. 263–272.

[15] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of RecSys*. 135–142.

[16] Meng Jiang, Peng Cui, Rui Liu, Qiang Yang, Fei Wang, Wenwu Zhu, and Shiqiang Yang. 2012. Social contextual recommendation. In *Proceedings of CIKM*. 45–54.

[17] Yehuda Koren. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of SIGKDD*. 426–434.

[18] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 42, 8 (2009).

[19] Artus Krohn-Grimberghe, Lucas Drumond, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2012. Multi-relational matrix factorization using bayesian personalized ranking for social network data. In *Proceedings of WSDM*. 173–182.

[20] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436.

[21] Dawen Liang, Laurent Charlin, James McInerney, and David M Blei. 2016. Modeling user exposure in recommendation. In *Proceedings of WWW*. 951–961.

[22] Hao Ma. 2014. On measuring social friend interest similarities in recommender systems. In *Proceedings of SIGIR*. 465–474.

[23] Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. *arXiv preprint arXiv:1606.03126* (2016).

[24] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings ICML*. 807–814.

[25] Zhaochun Ren, Shangsong Liang, Piji Li, Shuaiqiang Wang, and Maarten de Rijke. 2017. Social collaborative viewpoint regression with explainable recommendations. In *Proceedings of WSDM*. 485–494.

[26] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. 452–461.

[27] Francesco Ricci, Lior Rokach, and Bracha Shapira. 2011. Introduction to recommender systems handbook. In *Recommender systems handbook*. 1–35.

[28] Alexander M Rush, Sumit Chopra, and Jason Weston. 2015. A neural attention model for abstractive sentence summarization. *arXiv preprint arXiv:1509.00685* (2015).

[29] Suvash Sedhain, Aditya Krishna Menon, Scott Sanner, Lexing Xie, and Darius Braziunas. 2017. Low-Rank Linear Cold-Start Recommendation from Social Data.. In *Proceedings of AAAI*. 1502–1508.

[30] Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence* 2009 (2009), 4.

[31] Sainbayar Sukhbaatar, Jason Weston, Rob Fergus, et al. 2015. End-to-end memory networks. In *Proceedings of NIPS*. 2440–2448.

[32] Peijie Sun, Le Wu, and Meng Wang. 2018. Attentive Recurrent Social Recommendation. In *Proceedings of SIGIR*. 185–194.

[33] Yunzhi Tan, Min Zhang, Yiqun Liu, and Shaoping Ma. 2016. Rating-Boosted Latent Topics: Understanding Users and Items with Ratings and Reviews.. In *Proceedings of IJCAI*. 2640–2646.

[34] Yi Tay, Anh Tuan Luu, and Siu Cheung Hui. 2018. Latent Relational Metric Learning via Memory-based Attention for Collaborative Ranking. In *Proceedings of WWW*. 729–739.

[35] Bin Shen Travis Ebesu and Yi Fang. 2018. Collaborative Memory Network for Recommendation Systems. *arXiv preprint arXiv:1804.10862* (2018).

[36] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. *arXiv preprint arXiv:1708.04617* (2017).

[37] Bo Yang, Yu Lei, Jiming Liu, and Wenjie Li. 2017. Social collaborative filtering by trust. *IEEE transactions on pattern analysis and machine intelligence* 39, 8 (2017), 1633–1647.

[38] Mao Ye, Xingjie Liu, and Wang-Chien Lee. 2012. Exploring social influence for recommendation: a generative model approach. In *Proceedings of SIGIR*. 671–680.

[39] Wenhui Yu, Huidi Zhang, Xiangnan He, Xu Chen, Li Xiong, and Zheng Qin. 2018. Aesthetic-based Clothing Recommendation. In *Proceedings of WWW*. 649–658.

[40] Fajie Yuan, Guibing Guo, Joemon M Jose, Long Chen, Haitao Yu, and Weinan Zhang. 2016. Lambdafm: learning optimal ranking with factorization machines using lambda surrogates. In *Proceedings of CIKM*. 227–236.

[41] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. 2016. Collaborative knowledge base embedding for recommender systems. In *Proceedings of SIGKDD*. 353–362.

[42] Tong Zhao, Julian McAuley, and Irwin King. 2014. Leveraging social connections to improve personalized ranking for collaborative filtering. In *Proceedings of CIKM*. 261–270.