Session-based Social Recommendation via Dynamic Graph Attention Networks

Weiping Song
School of EECS, Peking University
songweiping@pku.edu.cn

Zhiping Xiao
School of EECS, UC Berkeley
patricia.xiao@berkeley.edu

Yifan Wang
School of EECS, Peking University
yifanwang@pku.edu.cn

Laurent Charlin
HEC Montreal & Montreal Institute for Learning Algorithms
laurent.charlin@hec.ca

Ming Zhang
School of EECS, Peking University
mzhang_cs@pku.edu.cn

Jian Tang*
HEC Montreal & Montreal Institute for Learning Algorithms
jian.tang@hec.ca

ABSTRACT

Online communities such as Facebook and Twitter are enormously popular and have become an essential part of the daily life of many of their users. Through these platforms, users can discover and create information that others will then consume. In that context, recommending relevant information to users becomes critical for viability. However, doing recommendation in online communities is a challenging problem: 1) users' interests are dynamic, and 2) users are influenced by their friends. Moreover, the influencers may be context-dependent. That is, different friends may be relied upon for different topics. Modeling both signals is therefore essential for recommendations.

We propose a recommender system for online communities based on a dynamic-graph-attention neural network. We model dynamic user behaviors with a recurrent neural network, and context-dependent social influence with a graph-attention neural network, which dynamically infers the influencers based on users' current interests. The whole model can be efficiently fit on large-scale data. Experimental results on several real-world data sets demonstrate the effectiveness of our proposed approach over several competitive baselines including state-of-the-art models.

CCS CONCEPTS

• **Information systems** → *Social recommendation*; • **Computing methodologies** → *Ranking*; *Learning latent representations*;

KEYWORDS

Dynamic interests; social network; graph convolutional networks; session-based recommendation

ACM Reference Format:
Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. 2019. Session-based Social Recommendation via Dynamic Graph Attention Networks. 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, 10 pages. https://doi.org/10.1145/XXXXXX.XXXXXX

*M. Zhang and J. Tang are corresponding authors.

WSDM '19, February 11–15, 2019, Melbourne, VIC, Australia

ACM ISBN 978-1-4503-5940-5/19/10…$15.00
https://doi.org/10.1145/XXXXXX.XXXXXX

1 INTRODUCTION

Online social communities are an essential part of today's online experience. Platforms such as Facebook, Twitter, and Douban enable users to create and share information as well as consume the information created by others. Recommender systems for these platforms are therefore critical to surface information of interest to users and to improve long-term user engagement. However, online communities come with extra challenges for recommender systems.

First, user interests are dynamic by nature. A user may be interested in sports items for a period of time and then search for new music groups. Second, since online communities often promote sharing information among friends, users are also likely to be influenced by their friends. For instance, a user looking for a movie may be influenced by what her friends have liked. Further, the set of influencers can be dynamic since they can be context-dependent. For instance, a user will trust a set of friends who like comedies when searching for funny films; while she could be influenced by another set of friends when searching for action movies.

Motivating Example. Figure 1 presents the behavior of Alice and her friends' in their online community. Behaviors are described by a sequence of actions (e.g., item clicks). To capture users' dynamic interests, their actions are segmented into sub-sequences or *sessions*. We are therefore interested in *session-based recommendations* [27]: within each session, we want to recommend the next item Alice should consume based on the items she has consumed thus far in the session. Figure 1 presents two sessions: session (a) and (b). In addition, the items consumed by Alice's friends are also available. We would like to use them to provide even better recommendations. We are thus in a *session-based social recommendation* setting.

In session (a), Alice browses sports items. Two of her friends: Bob and Eva, are notorious sports fans (long-term interests). Further, they are also browsing sports' items (short-term interests). Considering both facts, Alice may be influenced by the two and, e.g., decide to learn more about Ping Pong next. In session (b), Alice is interested in "literature & art" items. The situation is different than in session (a) since none of her friends have consumed such items recently but David is generally interested in this topic (long-term interests). In this case, it would make sense for Alice to be

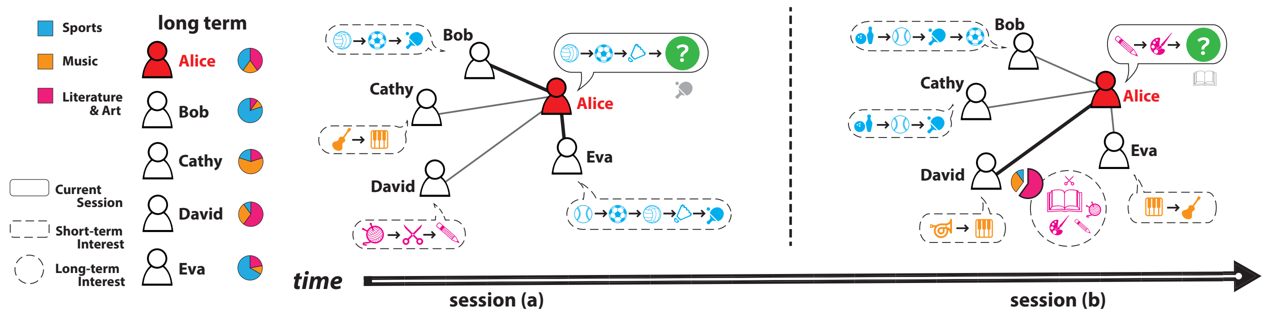


Figure 1: An illustration of Alice's social influences in two sessions. Alice's interests might change across different sessions, while she may be influenced by her friends, by either their short-term preferences or long-term preferences at different times.

influenced by David, and say, be recommended a book that David enjoyed. These examples show how a user's current interests combined with the (short- and long-term) interests of different friends provide session-based social recommendations. In this paper, we present a recommendation model based on both.

The current recommendation literature has modeled either users' dynamic interests or their social influences but, as far as we know has never combined both (like in the example above). A recent study [13] models session-level user behaviors using recurrent neural networks but social influences are not considered. Others studied social influences [5, 23, 40]. For example, Ma et al. [23] explores the social influence of friends' long-term preferences on recommendations. However, the influences from different users are static, they do not change according to users' current interests.

We propose an approach to model both users' session-based interests as well as dynamic social influences. That is, which subset of a user's friends influence her (the influencers) according to her current session. Our recommendation model is based on dynamic-graph-attention networks. Our approach first models user behaviors within a session using a recurrent neural network (RNN) [8]. According to users' current interests—captured by the hidden representation of the RNN—we capture the influences of friends using the graph-attention network [32]. To provide session-level recommendations, we distinguish the model of friends' short-term preferences from the long-term preferences one. The influence of each friend given the user's current interests is then determined automatically using an attention mechanism [2, 39].

We conduct extensive experiments on data sets collected from several online communities (Douban, Delicious, and Yelp). Our proposed approach outperforms many competitive baselines by modeling both users' dynamic behaviors and dynamic social influences.

To summarize, we make the following contributions:

- We propose to study both dynamic user interests and context-dependent social influences for the recommendation in online communities.
- We propose a novel recommendation approach based on dynamic-graph-attention networks for modeling both dynamic user interests and context-dependent social influences. The approach can effectively scale to large datasets.
- We conduct extensive experiments on real-world data sets. Experimental results demonstrate the effectiveness of our model over strong and state-of-the-art baselines.

Organization. §2 discusses related works. In §3 we give a formal definition of the session-based social recommendation problem. Our session-based social recommendation approach is described in §4. §5 presents the experimental results, followed by concluding remarks in §6.

2 RELATED WORK

We discuss three lines of research that are relevant to our work: 1) recommender systems that model the dynamic user behaviors, 2) social recommender systems that take social influence into consideration, and 3) recent progress of convolutional network developed for graph-structured data.

2.1 Dynamic Recommendation

Modeling user interests that change over time has already received some attention [6, 19, 38]. For example, Xiong et al. [38] learned temporal representations by factorizing the (user, item, time) tensor. Koren [19] developed a similar model named timeSVD++. Charlin et al. [6] developed similar ideas using Poisson factorization [10]. However, these approaches assume that the interest of users changes slowly and smoothly over long-term horizons (typically on the order of months or years). Afterwards, Wu et al. [36] used two separate RNNs to capture the dynamics of both users and items based on temporal observations. Beutel et al. [3] also built an RNN-based recommender while considering auxiliary context information. Different from these works, we are interested in capturing *session-level* preferences typical of online communities. Recent works use RNN to model user *sessions*, which consist of sequences of items consumed by a user within a given length of time (e.g., one hour or one week) [13, 20]. These models assume that items exhibit coherence within a session. We use a similar approach to model session-based user interests.

2.2 Social Recommendation

Modeling the influence of friends on user interests has also received attention [15, 16, 22–24]. Most proposed models are (also) based on Gaussian or Poisson matrix factorization. For example, Ma et al. [23] studied social recommendations by regularizing latent user factors such that the factors of connected users are close by. Chaney et al. [5] weighted the contribution of friends on a user's recommendation using a learned "trust factor". Zhao et al. [40] proposed an approach to leverage social networks for active learning. Xiao et al. [37] framed the problem as one of transfer learning between the social domain and the recommendation domain. These approaches can model social influences but assume that the influences are uniform across friends and independent from user's preferences. Tang et al. [30] and Tang et al. [29] proposed to model multi-facet trust relations, which relies on additional side information (e.g., item category) to define facets. Wang et al. [34] and Wang et al. [33] distinguished strong and weak ties among users for the social recommendation. However, they ignore user's short-term behaviors and integrate context-independent social influences. Our proposed approach models dynamic social influences by modeling the dynamic user interests, and context-dependent social influences.

2.3 Graph Convolutional Networks

Convolutional neural networks (CNNs) have achieved great success in computer vision and several other applications. CNNs are mainly developed for data with 2-D grid structures such as images. Recent work focuses on modeling more general graph-structure data using CNNs [4, 7, 11, 12, 18]. Specifically, Kipf and Welling [18] proposed graph-convolutional networks for semi-supervised graph classification. The goal is to learn node representations by leveraging both the node attributes and the graph structure. The model is composed of multiple *graph-convolutional layers*, each of which updates node representations using a combination of the current node's representation and the representations of its neighbors. Through this process, the dependency between nodes is captured. However, in the original formulation, all neighbors are given the same "weight" when updating the node representations. Velickovic et al. [32] addressed this problem by proposing *graph-attention networks*. They weight the contribution of neighbors differently using an attention mechanism [2, 39].

We propose a dynamic-graph-attention network. Compared to previous work, we focus on a different application (modeling the context-dependent social influences for recommendations). Besides, we model a dynamic graph, where the features of nodes evolve over time, and the attention between nodes also changes over time based on the current context.

3 PROBLEM DEFINITION

Recommender systems suggest relevant items to their users according to their historical behaviors. In classical recommendation models (e.g., matrix factorization [25]), the order in (or time at) which a user consumes items is ignored. However, in online communities, user preferences change very quickly, and the order of user preference behaviors must be considered in order to model users' dynamic interests. In practice, since users' entire history record can be extremely long (e.g., certain online communities have existed for years) and users' interests change quickly, a common approach is to segment user preference behaviors into different sessions (e.g., using timestamps and consider each user's behavior within a week as a session) and provide recommendations at the session level [13]. We define this problem as follows:

DEFINITION 1. **(Session-based Recommendation)** Let U denote the set of users and I be the set of items. Each user u is associated with a set of sessions by the time step T, $I_T^u = \{\vec{S}_1^u, \vec{S}_2^u, \ldots, \vec{S}_T^u\}$, where $\vec{S}_t^u$ is the t_{th} session of user u. Within each session, $\vec{S}_t^u$ is consisted of a sequence of user behaviors $\{i_{t,1}^u, i_{t,2}^u, \ldots, i_{t,N_{u,t}}^u\}$, where $i_{t,p}^u$ is the p_{th} item consumed by user u in t_{th} session, and $N_{u,t}$ is the number of items in the session. For each user u, given a new session $\vec{S}_{T+1}^u = \{i_{T+1,1}^u, \ldots, i_{T+1,n}^u\}$, the goal of *session-based recommendation* is to recommend a set of items from I that the user is likely to be interested in during the next step $n+1$, i.e., $i_{T+1,n+1}^u$.

In online communities, users' interests are not only correlated to their historical behaviors but are also commonly influenced by their friends. For example, if a friend watches a movie, I may also be interested in watching it. This is known as social influence [31]. Moreover, the influences from friends are *context-dependent*. In other words, the influences from friends vary from one situation to another. For example, if a user wants to buy a laptop, she will be more likely to turn to friends who are keen on high-tech devices; while she may be influenced by photographer friends when shopping for a camera. According to Figure 1, a user can be influenced by both her friends' short- and long-term preferences.

To provide an effective recommendation to users in online communities, in this paper, we propose to model both users' dynamic interests and context-dependent social influences. We define the resulting problem as follows:

DEFINITION 2. **(Session-based Social Recommendation)** Let U denote the set of users, I be the set of items, and $G = (U, E)$ be the social network, where E is the set of social links between users. Given a new session $\vec{S}_{T+1}^u = \{i_{T+1,1}^u, \ldots, i_{T+1,n}^u\}$ from user u, the goal of *session-based social recommendation* is to recommend a set of items from I that u is likely to be interested in during the next time step $n+1$ by utilizing information from both her dynamic interests (i.e., information from $\cup_{t=1}^{T+1}\vec{S}_t^u$) and the social influences (i.e., information from $\cup_{k=1}^{N(u)} \cup_{t=1}^{T} \vec{S}_t^k$, where $N(u)$ is the set of friends of user u).

4 DYNAMIC SOCIAL RECOMMENDER SYSTEMS

As discussed in previous sections, users are not only guided by their current preferences but also by their friends' preferences. We propose a novel dynamic graph attention model Dynamic Graph Recommendation (DGREC) which models both types of preferences.

DGREC is composed of four modules (Figure 2). First (§4.1), a recurrent neural network (RNN) [8] models the sequence of items consumed in the (target) user's current session. Her friends' interests are modeled using a combination of their short- and long-term preferences (§4.2). The short-term preferences, for example, items in their most recent session, are also encoded using an RNN. Friends' long-term preferences are encoded with a learned individual embedding. The model then combines the representation of the current user with the representations of her friends using a graph-attention network (§4.3). This is a key part of our model and contribution: our proposed mechanism learns to weigh the influence of each friend based on the user's current interests. As a final step (§4.4), the

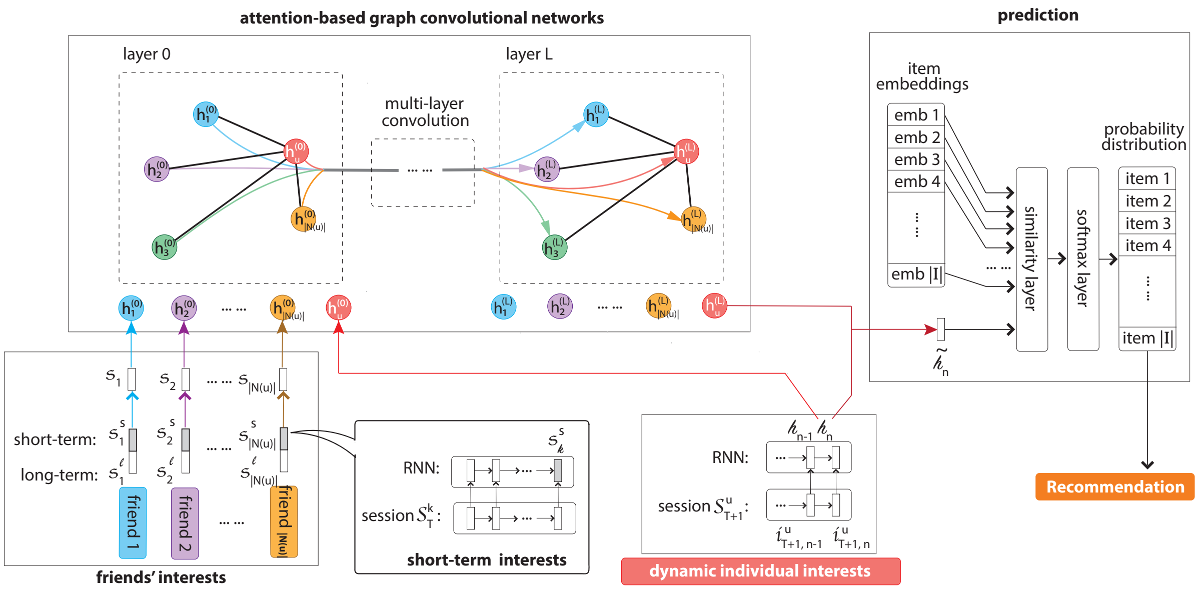


Figure 2: A schematic view of our proposed model for dynamic social recommendation.

model produces recommendations by combining a user's current preferences with her (context-dependent) social influences.

4.1 Dynamic Individual Interests

To capture a user's rapidly changing interests, we use an RNN to model the actions (e.g., clicks) of the (target) user in the current session. RNN is standard for sequence modeling and has recently been used for modeling user (sequential) preference data [13]. The RNN infers the representation of a user's session $\vec{S}^u_{T+1} = \{i^u_{T+1,1}, \ldots, i^u_{T+1,n}\}$, token by token by recursively combining the representation of all previous tokens with the latest token, i.e.,

$$h_n = f(i^u_{T+1,n}, h_{n-1}), \tag{1}$$

where h_n represents a user's interests and $f(\cdot,\cdot)$ is a non-linear function which combines both sources of information. In practice, the long short-term memory (LSTM) [14] unit is often used as the combination function $f(\cdot,\cdot)$:

$$\begin{aligned}
x_n &= \sigma(\mathbf{W}_x[h_{n-1}, i^u_{T+1,n}] + b_x) \\
f_n &= \sigma(\mathbf{W}_f[h_{n-1}, i^u_{T+1,n}] + b_f) \\
o_n &= \sigma(\mathbf{W}_o[h_{n-1}, i^u_{T+1,n}] + b_o) \\
\tilde{c}_n &= \tanh(\mathbf{W}_c[h_{n-1}, i^u_{T+1,n}] + b_c) \\
c_n &= f_n \odot c_{n-1} + x_n \odot \tilde{c}_n \\
h_n &= o_n \odot \tanh(c_n),
\end{aligned} \tag{2}$$

where σ is the sigmoid function: $\sigma(x) = (1 + \exp(-x))^{-1}$.

4.2 Representing Friends' Interests

We posit that in online communities users are likely most influenced by the recent interests of their friends. For that reason, we model friends' short- and long-term interests differently. Short-term interests are modeled using the sequence of items most recently consumed (e.g., a friend's latest online session). Long-term interests represent a friend's average interest and are modeled using an individual embedding.

Short-term preference: For a target user's current session $\vec{S}^u_{T+1}$, her friends' short-term interests are represented using their sessions right before session $T+1$ (our model generalizes beyond single session but this is effective empirically). Each friend k's actions $\vec{S}^k_T = \{i^k_{T,1}, i^k_{T,2}, \ldots, i^k_{T,N_{k,T}}\}$ are modeled using an RNN. In fact, here we reuse the RNN for modeling the target user's session (§ 4.1). In other words, both RNNs share the same weights. We represent friend k's short-term preference s^s_k by the final output of the RNN:

$$s^s_k = r_{N_{k,T}} = f(i^k_{T,N_{k,T}}, r_{N_{k,T-1}}). \tag{3}$$

Long-term preference: Friends' long-term preferences reflect their average interests. Since long-term preferences are not time-sensitive, we use a single vector to represent them. Formally,

$$s^l_k = \mathbf{W}_u[k,:], \tag{4}$$

where friend k's long-term preference s^l_k is the k_{th} row of the user embedding matrix $\mathbf{W}_u$.

Finally, we concatenate friends' short- and long-term preferences using a non-linear transformation:

$$s_k = ReLU(\mathbf{W}_1[s^s_k; s^l_k]), \tag{5}$$

where $ReLU(x) = max(0, x)$ is a non-linear activation function and $\mathbf{W}_1$ is the transformation matrix.

4.3 Context-dependent Social Influences

We described how we obtain representations of target user (§ 4.1) and her friends (§ 4.2). We now combine both into a single representation that we then use downstream (§4.4). The combined representation is a mixture of the target user's interest and her friends' interest.

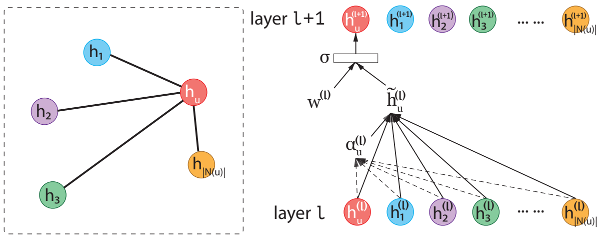


Figure 3: The graphical model of the single convolutional layer using attention mechanism, where the output conditioned on current interest is interpreted as context-dependent social influences.

We obtain this combined representation using a novel graph-attention network. First, we encode the friendship network in a graph where nodes correspond to users (i.e., target users and their friends) and edges denote friendship. In addition, each node uses its corresponding user's representation (§4.1 & §4.2) as (dynamic) features. Second, these features are propagated along the edges using a message-passing algorithm [9]. The main novelty of our approach lies in using an attention mechanism to weigh the features traveling along each edge. A weight corresponds to the level of a friend's influence. After a fixed number of iterations of message passing, the resulting features at the target user's node are the combined representation.

Below we detail how we design the node features as well as the accompanying graph-attention mechanism.

4.3.1 Dynamic feature graph. For each user, we build a graph where nodes correspond to that user and her friends. For target user u with $|N(u)|$ friends, the graph has $|N(u)| + 1$ nodes. User u's initial representation h_n is used as node u's features $h_u^{(0)}$ (the features are updated each time u consumes a new item in $\vec{S}^u_{T+1}$). For a friend k, the corresponding node feature is set to s_k and remains unchanged for the duration of time step $T+1$. Formally, the node features are $h_u^{(0)} = h_n$ and $\{h_k^{(0)} = s_k, k \in N(u)\}$.

4.3.2 Graph-Attention Network. With the node features defined as above, we then pass messages (features) to combine friends' and the target user's interests. This procedure is formalized as inference in a graph convolutional network [18].

Kipf and Welling [18] introduce graph convolutional networks for semi-supervised node representation learning. In these networks, the convolutional layers "pass" the information between nodes. The number of layers L of the networks corresponds to the number of iterations of message passing.[1] However, all neighbors are treated equally. Instead, we propose a novel dynamic graph attention network to model context-dependent social influences.

The fixed symmetric normalized Laplacian is widely used as a propagation strategy in existing graph convolutional networks [7, 18]. In order to distinguish the influence of each friend, we must break the static propagation schema first. We propose to use an attention mechanism to guide the influence propagation. The process is illustrated in Figure 3. We first calculate the similarity between the target user's node representation $h_u^{(l)}$ and all of its neighbors' representations $h_k^{(l)}$:

$$\alpha_{uk}^{(l)} = \frac{exp(f(h_u^{(l)}, h_k^{(l)}))}{\sum_{j \in N(u) \cup \{u\}} exp(f(h_u^{(l)}, h_j^{(l)}))}, \tag{6}$$

where $h_u^{(l)}$ is the representation of node/user u at layer l, and $f(h_u^{(l)}, h_k^{(l)}) = h_u^{(l)\top} h_k^{(l)}$ is the similarity function between two elements. Intuitively, $\alpha_{uk}^{(l)}$ is the *level of influence* or weight of friend k on user u (conditioned on the current context $h_u^{(l)}$). Note that we also include a self-connection edge to preserve a user's revealed interests. $\alpha_{u:}^{(l)}$ then provide the weights to combine the features:

$$\tilde{h}_u^{(l)} = \sum_{k \in N(u) \cup \{u\}} \alpha_{uk}^{(l)} h_k^{(l)}, \tag{7}$$

where $\tilde{h}_u^{(l)}$ is a mixture of user u's friends' interests at layer l, followed by a non-linear transformation: $h_u^{(l+1)} = ReLU(\mathbf{W}^{(l)} \tilde{h}_u^{(l)})$. $\mathbf{W}^{(l)}$ is the shared and learnable weight matrix at layer l. We obtain the final representation of each node by stacking this attention layer L times.[2] The combined (social-influenced) representation is denoted by $h_u^{(L)}$.

4.4 Recommendation

Since a user's interest depends on both her recent behaviors and social influences, her final representation is obtained by combining both using a fully-connected layer:

$$\hat{h}_n = \mathbf{W}_2[h_n; h_u^{(L)}], \tag{8}$$

where $\mathbf{W}_2$ is a linear transformation matrix, and $\hat{h}_n$ is the final representation of the user u's current interest.

We then obtain the probability that the next item will be y using a softmax function:

$$p(y | i^u_{T+1,1}, \ldots, i^u_{T+1,n}; \{\vec{S}^k_T, k \in N(u)\}) = \frac{\exp(\hat{h}_n^\top z_y)}{\sum_{j=1}^{|I|} \exp(\hat{h}_n^\top z_j)}, \tag{9}$$

where $N(u)$ are user u's set of friends according to the social network G, z_y is the embedding of item y, and $|I|$ is the total number of items.

4.5 Training

We train the model by maximizing the log-likelihood of the observed items in all user sessions:

$$\sum_{u \in U} \sum_{t=2}^{T} \sum_{n=1}^{N_{u,t}-1} \log p(i^u_{t,n+1} | i^u_{t,1}, \ldots, i^u_{t,n}; \{\vec{S}^k_{t-1}, k \in N(u)\}). \tag{10}$$

This function is optimized using gradient descent.

[1] We propagate information on a graph that also contains higher-order relationships (e.g., friends of friends of friends) in practice. In the l^{th} layer of the network, the target user then receives information from users that are l edges away.

[2] We also tested our model with two popular context-independent propagation strategies that do not use an attention mechanism: a) averaging friends' interests and; b) element-wise max-pooling over their interests—similar to techniques for aggregating word-level embeddings [35]. Mean aggregation outperforms the latter, but both are inferior to our proposed attention model.

5 EXPERIMENTS

We now study the effectiveness of our DGRec using real-world data sets. We highlight the following results:

- DGRec significantly outperforms all seven methods it is compared to under all experimental settings.
- Ablation studies demonstrate the usefulness of the different components of DGRec.
- Exploring the fitted models shows that attention contextually weighs the influences of friends.

5.1 Experimental Setup

5.1.1 Data Sets. We study all models using data collected from three well-known online communities. Descriptive statistics for all data sets are in Table 1.

	Douban	Delicious	Yelp
# Users	32,314	1,650	141,804
# Items	14,109	4,282	17,625
# Events	3,493,821	296,705	1,200,503
# Social links	331,315	15,328	6,818,026
Start Date	01/12/2008	08/12/2009	01/01/2009
End Date	07/22/2016	07/01/2016	10/15/2010
Avg. friends/user	10.25	9.00	48.08
Avg. events/user	108.12	179.82	8.47
Avg. session length	4.38	4.30	3.63

Table 1: Descriptive statistics of our three data sets.

Douban.[3] A popular site on which users can review movies, music, and books they consume. We crawled the data using the identity of the users in the movie community to obtain every movie they reviewed along with associated timestamps. We also crawled the users' social networks. We construct our dataset by using each review as evidence that a user consumed an item. Users tend to be highly active on Douban and we segment users' behaviors (movie consumption) into week-long sessions.

Delicious.[4] An online bookmarking system where users can store, share, and discover web bookmarks and assign them a variety of semantic tags. The task we consider is personalized tag recommendations for bookmarks. Each session is a sequence of tags a user has assigned to a bookmark (tagging actions are timestamped). This differs from the usual definition of sessions as a sequence of consumptions over a short horizon.

Yelp.[5] An online review system where users review local businesses (e.g., restaurants and shops). Similar as for Douban, we treat each review as an observation. Based on the empirical frequency of the reviews, we segment the data into month-long sessions.

We also tried different segmentation strategies, and preliminary results showed that our method consistently outperformed Session-RNN and NARM for other session lengths. A systematic method for optimizing session length is left for future work.

5.1.2 Train/valid/test splits. We reserve the sessions of the last d days for testing and filter out items that did not appear in the training set. Due to the different sparseness of the three data sets, we choose $d = 180, 50$ and 25 for *Douban*, *Yelp* and *Delicious* datasets respectively. We randomly and equally split the held out sessions into a validation and a test set.

5.1.3 Competing Models. We compare DGRec to three classes of recommenders: (A) classical methods that utilize neither social nor temporal factors; (B) social recommenders, which take context-independent social influences into consideration; and (C) session-based recommendation methods, which model user interests in sessions. (Below, we indicate a model's class next to its name.)

- ItemKNN [21] (A): inspired by the classic KNN model, it looks for items that are similar to items liked by a user in the past.
- BPR-MF [26] (A): matrix factorization (MF) technique trained using a ranking objective as opposed to a regression objective.
- SoReg [23] (B): uses the social network to regularize the latent user factors of matrix factorization.
- SBPR [40] (B): an approach for social recommendations based on BPR-MF. The social network is used to provide additional training samples for matrix factorization.
- TranSIV [37] (B): uses shared latent factors to transfer the learned information from the social domain to the recommendation domain.
- RNN-Session [13] (C): recent state-of-the-art approach that uses recurrent neural networks for session-based recommendations.
- NARM [20] (C): a hybrid model of both session-level preferences and the user's "main purpose", where the main purpose is obtained via attending on previous behaviors within the session.

5.1.4 Evaluation Metrics. We evaluate all models with two widely used ranking-based metrics: Recall@K and Normalized Discounted Cumulative Gain (NDCG).

Recall@K measures the proportion of the top-K recommended items that are in the evaluation set. We use $K = 20$.

NDCG is a standard ranking metric. In the context of next-item recommendation, it is formulated as: NDCG $= \frac{1}{\log_2(1+\text{rank}_{pos})}$, where rank_{pos} denotes the rank of a positive item. We report the average value of NDCG over all the testing examples.

5.1.5 Hyper-parameter Settings. For RNN-Session, NARM and our models, we use a batch size of 200. We use Adam [17] for optimization due to its effectiveness with $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1e^{-8}$ as suggested in TensorFlow [1]. The initial learning rate is empirically set to 0.002 and decayed at the rate of 0.98 every 400 steps. For all models, the dimensions of the user (when needed) and item representations are fixed to 100 following Hidasi et al. [13]. We cross-validated the number of hidden units of the LSTMs and the performance plateaued around 100 hidden units. The neighborhood sample sizes are empirically set to 10 and 15 in the first and second convolutional layers, respectively. We tried to use more friends in each layer but observed no significant improvement. For our models, dropout [28] with rate 0.2 is used to avoid overfitting.

5.1.6 Implementation Details. We implement our model using TensorFlow [1]. Training graph attention networks on our data with

[3]http://www.douban.com
[4]Data set available from https://grouplens.org/datasets/hetrec-2011/
[5]Data set available from https://www.yelp.com/dataset

Model Class	Model	Douban		Delicious		Yelp	
		Recall@20	NDCG	Recall@20	NDCG	Recall@20	NDCG
Classical	ItemKNN [21]	0.1431	0.1635	0.2729	0.2241	0.0441	0.0989
	BPR-MF [26]	0.0163	0.1110	0.2775	0.2293	0.0365	0.1190
Social	SoReg [23]	0.0177	0.1113	0.2703	0.2271	0.0398	0.1218
	SBPR [40]	0.0171	0.1059	0.2948	0.2391	0.0417	0.1207
	TranSIV [37]	0.0173	0.1102	0.2588	0.2158	0.0420	0.1187
Temporal	RNN-Session [13]	0.1643	0.1854	0.3445	0.2581	0.0756	0.1378
	NARM [20]	0.1755	0.1872	0.3776	0.2768	0.0765	0.1380
Social + Temporal (Ours)	DGRec	**0.1861**	**0.1950**	**0.4066**	**0.2944**	**0.0842**	**0.1427**

Table 2: Quantitative Results of Different Algorithms. We highlight that DGREC outperforms all other baselines across all three data sets and both metrics. Further analysis is provided in §5.2.

mini-batch gradient descent is not trivial since node degrees have a large range. We found the neighbor sampling technique proposed in [11] to be effective. Further, to reduce the computational cost of training DGREC, we represent friends' short-term interests using only their most recent sessions.

5.2 Quantitative Results

The performance of different algorithms is summarized in Table 2. ItemKNN and BPR-MF (both standard methods) perform very similarly, except on *Douban*. A particularity of Douban is that users typically only consume each item once (that's different for Delicious and Yelp). MF-based methods tend to recommend previously consumed items which explain BPR-MF's poor performance. By modeling social influence, the performance of social recommenders improves compared to BPR-MF in most cases. However, the improvement is marginal because these three algorithms (B) only model context-independent social influence. By modeling dynamic user interests, RNN-Session significantly outperforms ItemKNN and BPR, which is consistent with the results in Hidasi et al. [13]. Further, NARM extends RNN-Session by explicitly modeling user's main purpose and becomes the strongest baseline. Our proposed model DGREC achieves the best performance among all the algorithms by modeling both user's dynamic interests and context-dependent social influences. Besides, the improvement over RNN-Session and NARM is more significant compared to that of SoReg over BPR-MF, which shows the necessity of modeling context-dependent social influences.

5.3 Variations of DGREC

To justify and gain further insights into the specifics of DGREC's architecture, we now study and compare variations of our model.

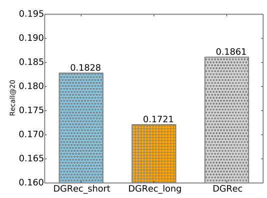
5.3.1 Self v.s. Social. DGREC obtains users' final preferences as a combination of user's consumed items in the current session and context-dependent social influences (see Eq. 8). To tease apart the contribution of both sources of information, we compare DGREC against two submodels: a) (DGREC$_{\text{self}}$) a model of the user's current session only (Eq. 8 without social influence features $h_u^{(L)}$) and; b)

Data Sets	Models	Recall@20	NDCG
Douban	DGREC$_{\text{self}}$	0.1643	0.1854
	DGREC$_{\text{social}}$	0.1185	0.1591
	DGREC	0.1861	0.1950
Delicious	DGREC$_{\text{self}}$	0.3445	0.2581
	DGREC$_{\text{social}}$	0.3306	0.2516
	DGREC	0.4066	0.2944
Yelp	DGREC$_{\text{self}}$	0.0756	0.1378
	DGREC$_{\text{social}}$	0.0690	0.1356
	DGREC	0.0842	0.1427

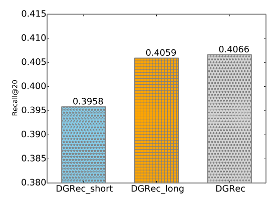
Table 3: Ablation study comparing the performance of the complete model (DGREC) with two variations.

(DGREC$_{\text{social}}$) a model using context-dependent social influence features only (Eq. 8 without individual features h_n). Note that when using individual features only, DGREC$_{\text{self}}$ is identical to RNN-Session (hence the results are reproduced from Table 2). Table 3 reports the performance of all three models on our datasets. DGREC$_{\text{self}}$ consistently outperforms DGREC$_{\text{social}}$ across all three data sets, which means that overall users' individual interests have a higher impact on recommendation quality. Compared to the full model DGREC, the performance of both DGREC$_{\text{self}}$ and DGREC$_{\text{social}}$ significantly decreases. To achieve good recommendation performance in online communities, it is, therefore, crucial to model both a user's current interests as well as her (dynamic) social influences.

5.3.2 Short-term v.s. Long-term. DGREC provides a mechanism for encoding friends' short- as well as long-term interests (see § 4.2). We study the impact of each on the model's performance. Similar to above, we compare using either short- or long-term interests to the results of using both. Figure 4 reports that for *Douban*, the predictive capability of friends' short-term interests outperforms that of friends' long-term interest drastically, and shows comparable performance compared to the full model. This is reasonable considering that the interests of users in online communities (e.g., Douban) change quickly, and exploiting users' short-term interests should be able to predict user behaviors more quickly. Interestingly, on the data set *Delicious*, different results are observed. Using



(a) Douban



(b) Delicious

Figure 4: Performance w.r.t. friends' short-term and friends' long-term preferences on different datasets. The result of *Yelp* data set is similar to *Douban* hence omitted.

Data Sets	Conv. Layers	Recall@20	NDCG
Douban	1	0.1726	0.1886
	2	0.1861	0.1950
	3	0.1793	0.1894
Delicious	1	0.4017	0.2883
	2	0.4066	0.2944
	3	0.4037	0.2932
Yelp	1	0.0760	0.1387
	2	0.0842	0.1427
	3	0.0846	0.1423

Table 4: Performance of our model w.r.t. different numbers of convolution layers.

long-term interests yield more accurate predictions than short-term interests. This is not surprising since, on *Delicious* website, users tend to have static interests.

5.3.3 Number of Convolutional Layers. DGRec aggregates friends' interests using a multi-layer graph convolutional network. More convolutional layers will yield influences from higher-order friends. In our study so far we have used two-layer graph convolutional networks. To validate this choice we compare the performance to one- and three-layer networks but maintain the number of selected friends to 10 and 5 in the first and third layer, respectively. Table 4 shows a significant decline in performance when using a single layer. This implies that the interests of friends' friends (i.e. using 2 layers) is important for recommendations.

Next, we test our model using three convolutional layers to explore the influences of even higher-order friends. The influence of the third layer on the performance is small. There is a small improvement for *Yelp* but a slightly larger drop in performance for both *Douban* and *Delicious*, which may be attributed to model overfitting or noises introduced by higher-order friends. This confirms that two convolutional layers are enough for our data sets.

5.4 Exploring Attention

DGRec uses an attention mechanism to weigh the contribution of different friends based on a user's current session. We hypothesized that while friends have varying interests, user session typically only explores a subset of these interests. As a consequence, for a target user, different subsets of his friends should be relied upon in different situations. We now explore the results of the attention learned by our model.

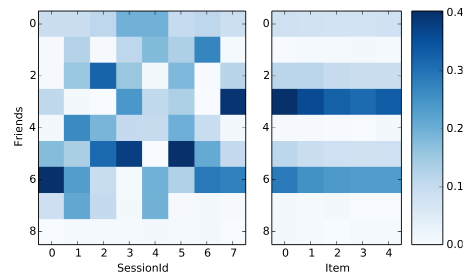


Figure 5: The heat map of the attention weights across different sessions (left) and within a session (right). For both plots, the y-axis represents friends of the target user. The x-axis represents (1) eight sessions of the target user on the left and (2) the item sequence within session #7 on the right.

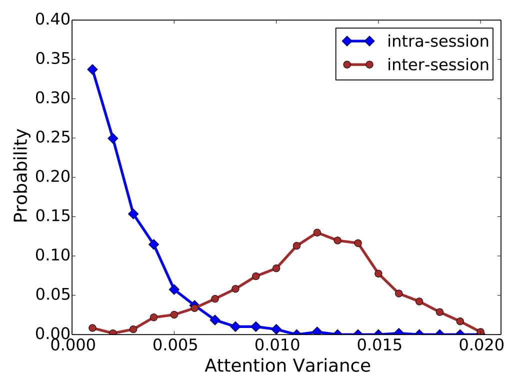


Figure 6: Attention variance distribution of DGRec for both inter-session and intra-session. Variance values are discretized into 20 intervals.

First, we randomly select a *Douban* user from those who have at least 5 test sessions as well as 5 friends and plot her attention weights (Eq. 6) within and across session(s) in Figure 5. For the inter-session level plot (left), we plot the average attention weight of a friend within a session. For intra-session level plot (right), her attention weights within one session (i.e. SessionId=7) are presented. We make the following observations. First, the user allocates her attention to different friends across different sessions. This indicates that social influence is indeed conditioned on context (i.e., target user's current interests). Further, friend #8 obtains little attention in all sessions, which means that social links do not necessarily correspond to the shared interest. Second, the distribution of attention is relatively stable within a single session. This confirms that the user's behaviors are coherent in a short period and suitable to be processed in a session manner.

As a second exploration of the behavior of the attention mechanism we take a macro approach and analyze the attention across all users (as opposed to a single user across friends). We use the attention levels inferred on the *Douban* test set. Figure 6 reports the empirical distributions of the inter-session (brown) and intra-session (blue) attention variance (i.e., how much does the attention weights vary in each case). The intra-session variance is on average lower. This agrees with our assumption that users' interests tend to be focused within a short time so that the same set of friends are attended to for the duration of a session. On the contrary, a user is more likely to trust different friends in different sessions, which further validates modeling context-dependent social influences via attention-based graph convolutional networks.

6 CONCLUSIONS

We propose a model based on graph convolutional networks for the session-based social recommendation in online communities. Our model first learns individual user representations by modeling users' current interests. Each user's representation is then aggregated with her friends' representations using a graph convolutional networks with a novel attention mechanism. The combined representation along with the user's original representation is then used to form item recommendations. Experimental results on three real-world data sets demonstrate the superiority of our model compared to several state-of-the-art models. Next steps involve exploring user and item features indicative of preferences and further improving the performance of recommender systems for online communities.

7 ACKNOWLEDGEMENT

This paper is partially supported by Beijing Municipal Commission of Science and Technology under Grant No. Z181100008918005 as well as the National Natural Science Foundation of China (NSFC Grant Nos.61772039, 61472006 and 91646202). We would like to thank Haoran Shi for collecting Douban data used in this paper.

REFERENCES

[1] Martín Abadi and et al. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. (2015). https://www.tensorflow.org/ Software available from tensorflow.org.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

[3] Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H Chi. 2018. Latent Cross: Making Use of Context in Recurrent Recommender Systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 46–54.

[4] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. 2014. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations*.

[5] Allison JB Chaney, David M Blei, and Tina Eliassi-Rad. 2015. A probabilistic model for using social networks in personalized item recommendation. In *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM, 43–50.

[6] Laurent Charlin, Rajesh Ranganath, James McInerney, and David M Blei. 2015. Dynamic poisson factorization. In *Proceedings of the 9th ACM Conference on Recommender Systems*. ACM, 155–162.

[7] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*. 3844–3852.

[8] Jeffrey L Elman. 1990. Finding structure in time. *Cognitive science* 14, 2 (1990), 179–211.

[9] Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural Message Passing for Quantum Chemistry. In *International Conference on Machine Learning*. 1263–1272.

[10] Prem Gopalan, Jake M Hofman, and David M Blei. 2015. Scalable Recommendation with Hierarchical Poisson Factorization. In *UAI*. 326–335.

[11] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*. 1024–1034.

[12] Mikael Henaff, Joan Bruna, and Yann LeCun. 2015. Deep convolutional networks on graph-structured data. *arXiv preprint arXiv:1506.05163* (2015).

[13] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. 2016. Session-based recommendations with recurrent neural networks. In *International Conference on Learning Representations*.

[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.

[15] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *Proceedings of the fourth ACM conference on Recommender systems*. ACM, 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 the 21st ACM international conference on Information and knowledge management*. ACM, 45–54.

[17] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*.

[18] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*.

[19] Yehuda Koren. 2010. Collaborative filtering with temporal dynamics. *Commun. ACM* 53, 4 (2010), 89–97.

[20] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. 2017. Neural Attentive Session-based Recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. 1419–1428.

[21] Greg Linden, Brent Smith, and Jeremy York. 2003. Amazon. com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing* 7, 1 (2003), 76–80.

[22] Hao Ma, Haixuan Yang, Michael R Lyu, and Irwin King. 2008. Sorec: social recommendation using probabilistic matrix factorization. In *Proceedings of the 17th ACM conference on Information and knowledge management*. ACM, 931–940.

[23] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *Proceedings of the fourth ACM international conference on Web search and data mining*. ACM, 287–296.

[24] Paolo Massa and Paolo Avesani. 2007. Trust-aware recommender systems. In *Proceedings of the 2007 ACM conference on Recommender systems*. ACM, 17–24.

[25] Andriy Mnih and Ruslan R Salakhutdinov. 2008. Probabilistic matrix factorization. In *Advances in neural information processing systems*. 1257–1264.

[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*. AUAI Press, 452–461.

[27] J Ben Schafer, Joseph Konstan, and John Riedl. 1999. Recommender systems in e-commerce. In *Proceedings of the 1st ACM conference on Electronic commerce*. ACM, 158–166.

[28] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research* 15, 1 (2014), 1929–1958.

[29] Jiliang Tang, Huiji Gao, and Huan Liu. 2012. mTrust: discerning multi-faceted trust in a connected world. In *Proceedings of the fifth ACM international conference on Web search and data mining*. ACM, 93–102.

[30] Jiliang Tang, Huiji Gao, Huan Liu, and Atish Das Sarma. 2012. eTrust: Understanding trust evolution in an online world. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 253–261.

[31] Jie Tang, Jimeng Sun, Chi Wang, and Zi Yang. 2009. Social influence analysis in large-scale networks. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 807–816.

[32] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2018. Graph Attention Networks. In *International Conference on Learning Representations*.

[33] Xin Wang, Steven CH Hoi, Martin Ester, Jiajun Bu, and Chun Chen. 2017. Learning personalized preference of strong and weak ties for social recommendation. In *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 1601–1610.

[34] Xin Wang, Wei Lu, Martin Ester, Can Wang, and Chun Chen. 2016. Social recommendation with strong and weak ties. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 5–14.

[35] Jason Weston, Sumit Chopra, and Keith Adams. 2014. #tagspace: Semantic embeddings from hashtags. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 1822–1827.

[36] Chao-Yuan Wu, Amr Ahmed, Alex Beutel, Alexander J Smola, and How Jing. 2017. Recurrent recommender networks. In *Proceedings of the tenth ACM international conference on web search and data mining*. ACM, 495–503.

[37] Lin Xiao, Zhang Min, Zhang Yongfeng, Liu Yiqun, and Ma Shaoping. 2017. Learning and transferring social and item visibilities for personalized recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*. ACM, 337–346.

[38] Liang Xiong, Xi Chen, Tzu-Kuo Huang, Jeff Schneider, and Jaime G Carbonell. 2010. Temporal collaborative filtering with bayesian probabilistic tensor factorization. In *Proceedings of the 2010 SIAM International Conference on Data Mining*. SIAM, 211–222.

[39] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. 2015. Show, attend and tell: Neural image caption generation with visual attention. In *International Conference on Machine Learning*. 2048–2057.

[40] Tong Zhao, Julian McAuley, and Irwin King. 2014. Leveraging social connections to improve personalized ranking for collaborative filtering. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*. ACM, 261–270.