SoRecGAT: Leveraging Graph Attention Mechanism for Top-N Social Recommendation

Vijaikumar M✉, Shirish Shevade, and M N Murty

Department of Computer Science and Automation
Indian Institute of Science, Bangalore, India
{vijaikumar,shirish,mnm}@iisc.ac.in

Abstract. Social recommendation systems typically combine extra information like a social network with the user-item interaction network in order to alleviate data sparsity issues. This also helps in making more accurate and personalized recommendations. However, most of the existing systems work under the assumption that all socially connected users have equal influence on each other in a social network, which is not true in practice. Further, estimating the quantum of influence that exists among entities in a user-item interaction network is essential when only implicit ratings are available. This has been ignored even in many recent state-of-the-art models such as SAMN (Social Attentional Memory Network) and DeepSoR (Deep neural network model on Social Relations). Many a time, capturing a complex relationship between the entities (users/items) is essential to boost the performance of a recommendation system. We address these limitations by proposing a novel neural network model, SoRecGAT, which employs multi-head and multi-layer graph attention mechanism. The attention mechanism helps the model learn the influence of entities on each other more accurately. The proposed model also takes care of heterogeneity among the entities seamlessly. SoRecGAT is a general approach and we also validate its suitability when information in the form of a network of co-purchased items is available. Empirical results on eight real-world datasets demonstrate that the proposed model outperforms state-of-the-art models.

Keywords: Social recommendation · Graph attention mechanism.

1 Introduction

In the last few years, collaborative filtering (CF) has been successful in building powerful recommendation systems. Given a partially filled implicit rating matrix (for example, a matrix representing likes or clicks), the idea of a top-N recommendation system is to come up with the highly probable list of items that a user may like in future. A common and popular approach is to use a latent factor model to learn low dimensional latent representations for entities (users and items) and use the similarity between the entities to predict the unknown ratings. Matrix factorization (MF) [17, 11, 22] remains one of the successful baselines in such tasks. In practice, however, a user typically interacts with a very small set of available items. This results in data sparsity issues which is a big challenging factor in designing better recommendation systems.

The recent explosive growth in online services and mobile technologies have provided tons of useful information. For example, Yelp[1] has friendship connections amongst users, Amazon[2] has co-purchased network associated with products or Epinion[3] has trust relationships associated with users. Crucially, *social networks*[4] associated with users and items play a pivotal role in recommending products and services to the end users. That is, users are typically influenced by social neighbours in a social network. Therefore, one expects social neighbours to have similar opinions regarding products. By the same argument, co-purchased items are expected to have a strong influence on each other. Thus, social connections associated with users and/or items can be effectively leveraged to alleviate data sparsity issues that exist in traditional recommendation systems to boost their performance.

There have been some works that leverage matrix factorization techniques for social recommendation [8, 10, 15, 31]. While [8] models both implicit and explicit influence of trust, [15] introduces the concept of a social regularizer to represent the social constraints on recommendation systems. These approaches either treat all social relations equally [8, 10, 31] or make use of a predefined similarity function [15]. Either case may result in the performance degradation of the recommendation system as users with strong ties are more likely to have similar preferences than those with weak ties [25]. Some recently proposed neural network based models which utilize external social networks [5, 21] also have the same drawback. Further, in the applications where only implicit ratings are available, it is important to learn the quantum of influence that the different entities have on each other. This would help in getting better latent representations of the entities and better performance of the recommendation system.

A few attempts have been made [3, 32] to learn the influence of entities in a network by employing an attention mechanism for recommendations. In particular, Chen *et al.* [3] presented a social attentional memory network which utilizes an attention-based memory module to learn the relation vectors for user-friend pairs. This is combined with the friend level attention mechanism to measure the influence strength among users' friends. Further, [32] proposed ATRank which models heterogeneous user behaviour using an attention model and captures the interaction between users using self-attention. However, the key challenges here are to design a unified model that exploits the influence of entities from both user-item interaction network and social network together, and to capture the complex relationships that exist among entities across networks.

Contributions. Motivated by the success of Graph Attention Networks (GAT) [26], we propose SoRecGAT – a Graph ATtention based framework for top-N Social Recommendation problem. The proposed framework is illustrated in Fig. 1. We represent the user-item interaction network as a graph with nodes representing

[1] www.yelp.com [2] www.amazon.com [3] www.epinion.com [4] Throughout this paper, we refer to a user-user network (or connection) or a co-purchased item network (or connection) as a social network (or connection).

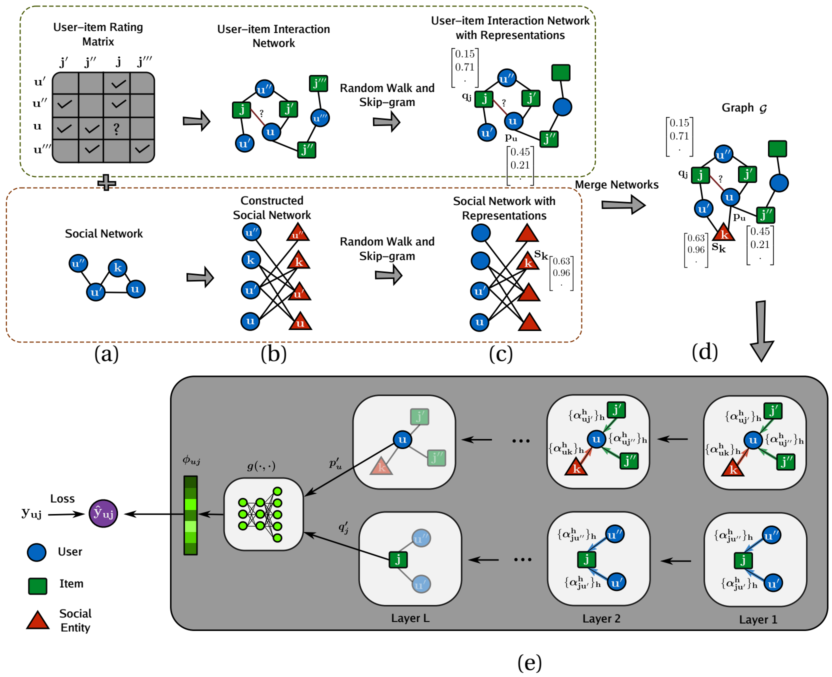

Fig. 1. The illustration of SoRecGAT recommendation setting. The transformations from a user-item rating matrix and a social network (a) to a graph with learned node representations (d) are illustrated via steps (b) and (c). The graph in (d) serves as an input to SoRecGAT (e) which employs a multi-layer and multi-head attention mechanism, and gives final user(u) and item(j) representations p'_u and q'_j and user-item pair representation ϕ_{uj} for (u, j). This ϕ_{uj} is used for predicting rating $\hat{y}_{uj}$. (Best viewed in colour.)

users and items, and edges representing interactions among them. We assume that no attribute information is available for the nodes, and initial representations (or embeddings) are learned using random walk and skip-gram techniques. We propose a simple approach by which a social network associated with the users or items can seamlessly be incorporated into this graph. The novelty of our approach lies in handling heterogeneous networks (for example, a social network with a user-item interaction network) for a personalised recommendation. Specifically, we propose to obtain the heterogeneous graph node representations in a unified space, which is essential in assigning weights to neighbouring nodes. These node representations are learned using multi-head and multi-layer attention mechanism. The attention mechanism helps in capturing complex relationships among entities in both user-item interaction network and social networks, collectively. Further, the final node representations are used for predicting the ratings. We conduct extensive experiments on eight real-world datasets – four from Amazon and four from Yelp. Experimental results demonstrate the effectiveness of SoRecGAT over state-of-the-art models.

2 The Proposed Model

Problem formulation. We represent a user-item interaction network and a social network combinedly as a graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$ where $\mathcal{V}$ represents the set of users, items and social entities,[5] and $\mathcal{E}$ represents the set of edges present in the graph. We consider the implicit rating setting where the rating between user u and item j is given as

$$y_{uj} = \begin{cases} 1, \text{ if } (u, j) \in \Omega \\ 0, \text{ otherwise} \end{cases}$$

where $\Omega = \{(u, j) : \text{user } u \text{ interacts with item } j\}$. Given $\mathcal{G}(\mathcal{V}, \mathcal{E})$, our goal in this work is to design a model which gives a top-N ranked list of items for each user.

2.1 SoRecGAT

In this section, we explain the proposed model – SoRecGAT, illustrated in Fig. 1. As shown in the figure, a user-item rating matrix can be converted to a graph whose node features (representations) (Fig. 1(c)) can be found (Section 6). A social network (e.g. a friendship network) is first converted to a bi-partite graph (Fig. 1(b)) by connecting users to "social entities", where each social entity corresponds to a user. Thus, if user $u1$ is connected to user $u2$, then, this would correspond to two edges $(u1, e2)$ and $(u2, e1)$ in Fig. 1(b), where $e1$ and $e2$ are the social entities associated with users $u1$ and $u2$, respectively. The introduction of social entities helps fuse the user-item interaction network and social network to get a combined graph (Fig. 1(d)) with node representations. This proposed idea also helps in combining multiple networks which share entities. In addition, network-specific features (side information) for the nodes can be seamlessly incorporated. A multi-head attention mechanism is then applied layerwise on this graph to predict the rating of a user-item pair. This is explained below.

Let $p_u \in \mathbb{R}^{d_p}$, $q_j \in \mathbb{R}^{d_q}$ and $s_k \in \mathbb{R}^{d_s}$ denote the features of user u, item j and social entity k respectively. Note that the feature dimensions of different entities in a given heterogeneous network can be different. Let N_p, N_q and N_s denote the number of users, items and social entities respectively. We denote the user, item and social entity features compactly as $\mathbf{p}, \mathbf{q}$ and $\mathbf{s}$ respectively, where $\mathbf{p} = (p_1, p_2, ..., p_{N_p})$, $\mathbf{q} = (q_1, q_2, ..., q_{N_q})$, and $\mathbf{s} = (s_1, s_2, ..., s_{N_s})$. The sets of neighbours of user u in the user-item interaction network and the social network are denoted by $\mathcal{N}_u^I$ and $\mathcal{N}_u^S$ respectively.

SoRecGAT contains multiple layers, and at every layer, a new set of hidden representations for the nodes $\mathbf{p'} = (p'_1, p'_2, ..., p'_{N_p})$, $p_u \in \mathbb{R}^{d'}$, $\mathbf{q'} = (q'_1, q'_2, ..., q'_{N_q})$, $q_j \in \mathbb{R}^{d'}$, and $\mathbf{s'} = (s'_1, s'_2, ..., s'_{N_s})$, $s_k \in \mathbb{R}^{d'}$ are obtained from the output of previous layers. It is essential to learn multiple levels of representations due to

[5] users/items present in a social network.

the complex nature of the relationship that exists among entities. Further, the influence of different neighbours on a given node need not be equal. Accounting for these, we explain how hidden representations are obtained in one layer. The same procedure is repeated in the other layers.

If $f(\cdot,\cdot)$ denotes an attention function, then the importance of item j's features to user u can be calculated as

$$\bar{\alpha}_{uj} = f(W_p p_u, W_q q_j), \tag{1}$$

and that of social entity k's (in a social network) features to the same user u is given by

$$\bar{\alpha}_{uk} = f(W_p p_u, W_s s_k), \tag{2}$$

where $W_p \in \mathbb{R}^{d' \times d_p}, W_q \in \mathbb{R}^{d' \times d_q}$ and $W_s \in \mathbb{R}^{d' \times d_s}$ are the weight matrices respectively for users, items and social entities. Due to different types of entities present in a network, it is important to have different weight matrices. These matrices also act as projection matrices for entities with different types and they project the representations of users, items and social entities into a unified space. The function $f(\cdot,\cdot)$ can be a feedforward neural network. In this work, we use a single layer feedforward neural network, parametrized by trainable parameter c. That is,

$$\begin{aligned} f(W_p p_u, W_q q_j) &= a(c^T[W_p p_u \| W_q q_j]), \\ f(W_p p_u, W_q s_k) &= a(c^T[W_p p_u \| W_s s_k]), \end{aligned} \tag{3}$$

where $a(\cdot)$ denotes an activation function and $\|$ denotes concatenation operation. Normalized positive attention weights of item j on user u can be calculated as

$$\begin{aligned} \alpha_{uj} &= \text{softmax}\,(\bar{\alpha}_{uj}) \\ &= \frac{\exp(\bar{\alpha}_{uj})}{\sum_{j' \in \mathcal{N}_u^I} \exp(\bar{\alpha}_{uj'}) + \sum_{k' \in \mathcal{N}_u^S} \exp(\bar{\alpha}_{uk'}) + \exp(\bar{\alpha}_{uu})} \\ \text{and} \quad & \bar{\alpha}_{uj} = f(W_p p_u, W_q q_j), \bar{\alpha}_{uk} = f(W_p p_u, W_s s_k), \bar{\alpha}_{uu} = f(W_p p_u, W_p p_u), \end{aligned} \tag{4}$$

where $\bar{\alpha}_{uj}$ and α_{uj} represent unnormalized and normalized attention weights of item j on user u. The normalized attention coefficients are then used to compute a linear combination of the features of neighbouring nodes to get a new representation of a given node. For example, a representation of user u at the current layer is calculated as

$$p'_u = a\Big(\sum_{j \in \mathcal{N}_u^I} \alpha^h_{uj} W^h_q q_j + \sum_{k \in \mathcal{N}_u^S} \alpha^h_{uk} W^h_s s_k + \alpha^h_{uu} W^h_p p_u\Big). \tag{5}$$

To exploit complex relationships that exist among entities, we employ multi-head attention mechanism. In particular, using H independent attention heads, the representation for user u can be obtained as

$$p'_u = \|_{h=1}^{H} a(\sum_{j \in \mathcal{N}_u^I} \alpha_{uj}^h W_q^h q_j + \sum_{k \in \mathcal{N}_u^S} \alpha_{uk}^h W_s^h s_k + \alpha_{uu}^h W_p^h p_u). \quad (6)$$

Similarly, one can obtain features representations of items, $\mathbf{q}'$. The final rating of user u on item j can be obtained as

$$\hat{y}_{uj} = \sigma(w \cdot \phi_{uj}), \quad \text{where} \quad \phi_{uj} = g(p'_u, q'_j), \quad (7)$$

where $\sigma(\cdot)$ is the sigmoid function defined as $\sigma(z) = \frac{1}{1+e^{-z}}$ and w denotes weight vector. Here, $g(\cdot,\cdot)$ is a function which constructs the representation for user-item interaction ϕ_{uj} for (u,j) from p'_u and q'_j. One can use a feedforward neural network for $g(\cdot,\cdot)$. In our experiments, we use $g(p'_u, q'_j) = p'_u \odot q'_j$, where $\odot$ denotes element-wise multiplication.

Note that, as mentioned earlier, it is easy to incorporate the side information of the nodes (for example, gender, age and country for users; and keywords and category for items) in the proposed model. Let user u (with the associated social entity e) be involved in a user-item interaction network and a social network. Let x_u^p and x_e^s denote the side information associated with these nodes. This information may be directly available in the dataset. Then the new representations of the user and social entity nodes can be $p_u \| x_u^p$ and $s_e \| x_e^s$ respectively. Thus, side information, if available, can be easily used in the proposed approach.

2.2 Loss Function

Some commonly used loss functions for the implicit rating setting are cross-entropy (l_{ce}) [9] and pairwise loss (l_{pair}) [22] functions, which can be defined for a user-item pair (u,j) as

$$\begin{aligned} l_{ce}(y_{uj}, \hat{y}_{uj}) &= -y_{uj} \ln(\hat{y}_{uj}) - (1 - y_{uj}) \ln(1 - \hat{y}_{uj}), \\ l_{pair}(\hat{y}_{ujj'}) &= -\ln(\sigma(\hat{y}_{uj} - \hat{y}_{uj'})), \text{ where } (u,j) \in \Omega \text{ and } (u,j') \notin \Omega. \end{aligned} \quad (8)$$

In this work, we use cross-entropy loss with negative sampling strategy [16] for training the model. For all the training interactions, the loss function is defined as follows:

$$\min_{\mathcal{W}} \quad \mathcal{L}(\mathcal{W}) = -\sum_{(u,j) \in \mathcal{D}} y_{uj} \ln \hat{y}_{uj} + (1 - y_{uj}) \ln(1 - \hat{y}_{uj}) + \lambda \, \mathcal{R}(\mathcal{W}), \quad (9)$$

where $\mathcal{R}(\cdot)$ is a regularizer, λ is a non-negative hyperparameter, and $\mathcal{W}$ denotes all the model parameters. Here, $\mathcal{D} = \mathcal{D}^+ \cup \mathcal{D}^-_{samp}$ where $\mathcal{D}^+ := \{(u,j) \in \Omega\}$ and $\mathcal{D}^-_{samp} \subset \{(u,j') \notin \Omega\}$, obtained using negative sampling.

2.3 Node Features

Initial embeddings of graph nodes, before using multi-head attention layers, are obtained using skip-gram technique [16]. Node sequences for a given graph $\mathcal{G}$ are

Algorithm 1: SoRecGAT

Input: graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, epochs T, number of layers L, minibatch size m

```
Initialize 𝒲
obtain p, q, s from user-item interaction network and social network based on
  equations (10)-(12) while t < T and not converged do
    𝒪 ← Shuffle(p, q, s)
    for each minibatch of (p̄, q̄, s̄) = (p_i, q_i, s_i)_{i=1}^m ⊆ 𝒪 do
        𝒲 ← GATEmbedding(p̄, q̄, s̄, 𝒲, L)
return
```

Algorithm 2: GATEmbedding learns weights for attention layers

Input: $\mathbf{p}, \mathbf{q}, \mathbf{s}$, network weights $\mathcal{W}$, number of layers L

Output: $\mathcal{W}^{new}$

```
for l = 1 → L − 1 do
    compute p′, q′, s′ from p, q, s and 𝒲 based on equations (3)-(6)
    (p, q, s) ← (p′, q′, s′)
compute ŷ based on equation (7)
𝒲^new ← 𝒲^old − η ∂ℒ(𝒲)/∂𝒲          //η is a learning rate
return 𝒲^new
```

first generated by random walks [18, 4]. Treating these sequences as sentences, the skip-gram technique is used to construct graph node embeddings. For a given graph $G(\mathcal{V}, \mathcal{E})$ with entities belonging to the same type, the objective of the skip-gram technique is to maximize the probability of predicting the context node c of a given node v as

$$\max_{\mathbf{x}} \prod_{v \in \mathcal{V}} \prod_{c \in \mathcal{C}(v)} Pr(c|v), \quad \text{where} \quad Pr(c|v) = \frac{e^{x_c \cdot x_v}}{\sum_{c' \in \mathcal{V}} e^{x_{c'} \cdot x_v}}. \tag{10}$$

Here, $\mathcal{C}(v)$ denotes the context of node v, $\mathbf{x} = (x_1, x_2, \ldots, x_{|\mathcal{V}|})$, and x_v represents the embedding of node v. These embeddings are learnt by solving the above optimization problem. In our setting, we have two networks: a user-item interaction network and a user-social_entity social network. The node embeddings for these networks are constructed separately. Note that, this procedure reflects meta-path based node embedding construction for *user-item* and *user-social_entity* meta-paths. Considering heterogeneity among entities has been shown to improve performance over ignoring the types and taking them as homogeneous entities [4].

To reduce the computational cost involved in computing $Pr(c|v)$ (equation (10)), we adopt the negative sampling strategy [16] as follows:

$$\ln Pr(c|v) = \ln \sigma(x_c \cdot x_v) + \sum_{m=1}^{M} \mathbb{E}_{c' \sim P_n(c')}[\ln \sigma(x_v \cdot x_{c'})]. \tag{11}$$

where M denotes number of negative samples. Hence, the loss function corresponding to skip-gram model is defined as follows:

$$\min_{\mathbf{x}} \quad \mathcal{L}_{rw}(\mathbf{x}) = -\sum_{v \in \mathcal{V}} \sum_{c \in \mathcal{C}(v)} \ln \sigma(x_c \cdot x_v) \; - \sum_{m=1}^{M} \mathbb{E}_{c' \sim P_n(c')}[\ln \sigma(x_v \cdot x_{c'})]. \quad (12)$$

The complete training procedure for learning the parameters $\mathcal{W}$ for SoRecGAT is given in Algorithms 1 and 2. During training, we maintain the whole graph structure in a sparse adjacency matrix. In our model, attention weight parameters are shared across all the edges in the graph. Unlike other graph neural network approaches, due to the shared attention weight parameters, we do not operate on embeddings of all the nodes at every mini-batch iteration. Instead, we operate only on the embeddings of the corresponding mini-batch nodes and their neighbours. We randomly select a mini-batch of user-item interactions, that is, the corresponding users and items based on their interactions in the training set. During mini-batch training the gradient propagation happens only to the respective nodes and their neighbours.

In the next section, we will discuss our experimental results.

3 Experiments

To demonstrate the effectiveness of the proposed model, in view of the following research questions, we conduct several experiments:

RQ1 Does our proposed model – SoRecGAT perform better than state-of-the-art social recommendation models? Does influence learning provide an advantage when only the user-item rating matrix is available?
RQ2 What is the effect of various sparsity levels of the training set on the performance of the proposed model?
RQ3 Employing the multi-head attention mechanism helpful for improving the performance of SoRecGAT?

We address these questions after discussing experimental settings.

3.1 Experimental Settings

Datasets. We conduct experiments on eight datasets: four from **Amazon**[6] – an e-commerce recommendation system for products ranging from books, movie DVDs to cloth items, and **Yelp**[7] – a user review platform on local businesses ranging from restaurants, hotels to real estates. Amazon dataset contains co-purchased information for the items which we use as the item-social network. Similarly, Yelp dataset contains friendship connections which we use as the user-social network. Datasets contain ratings on the scale, [1-5]. We do the following preprocessing as done in [19, 9]: (1) Ratings having value more than 3 are retained

[6] http://jmcauley.ucsd.edu/data/amazon [7] https://www.yelp.com/dataset/challenge

and treated as positive interactions (rating value 1 is assigned to them); (2) Those users and items who have at least five ratings associated with them are retained; and (3) Social connections between entities $e1 \rightarrow e2$ for which either $e1$ or $e2$ is a part of user-item interaction network are retained. The details of the datasets are given in Table 1.

Dataset		# users	# items	# ratings	# social entities	# social connections
Amazon	Music	2412	1923	33237	6769	129848
	Movie	9498	4786	156633	5835	85495
	CD	7878	7247	137610	18687	485526
	Book	10041	6477	143805	16711	264283
Yelp	Art	3071	1122	31438	7203	458322
	Food	12615	4222	151394	13053	819044
	Hotel	11040	3925	128130	14432	893278
	Restaurant	13877	2233	158384	16702	1076506

Table 1. Dataset statistics.

Evaluation procedure. For evaluating the performance of the models, we closely follow [9] and adopt the well-known *leave-one-out* procedure. That is, one item for each user from the dataset is held-out for validation and test purpose respectively and the remaining items are used for training the model. Since it is too time-consuming to rank all the items for each user during the evaluation time, following [9], we randomly sample 50 non-interacted items for each user along with the held-out item to construct validation and test set. Likewise, we randomly extract five such sets. Mean and standard deviation of the models on the test set with respect to best validation set performance is reported as the final result.

Metric. We use two widely adopted ranking metrics – HitRatio@N (HR@N) and normalized discounted cumulative gain (NDCG@N) for comparing the performance of different models [19, 9]. While HR@N measures the existence of the items a user has interacted with, NDCG@N emphasizes the position of the same item from the predicted top-N ranked list.

Comparison with different models. To evaluate the performance in rating-only and social recommendation setting, we compare SoRecGAT with the following four groups of models. They are: rating-only models based on (i) matrix factorization and (ii) neural networks; and social recommendation models based on (iii) matrix factorization and (iv) neural networks. We select representatives for each group and detail them below:

- **SAMN** [3] is a state-of-the-art model for top-N social recommendation setting. It contains two components. The first component – attention-based memory module learns aspect-level differences among friends, whereas the

second component – friend-level attention module learns influence strength of his friends.
- **DeepSoR** [5] follows the two-stage procedure. In the first stage, it obtains user representations from social networks by leveraging random walks. It extends PMF (Probabilistic Matrix Factorization) [17] for social recommendation in the second stage. The representations obtained from the first stage are used as regularizers for users.
- **SBPR** [31] is a state-of-the-model for the top-N recommendation setting. It extends BPR for the social recommendation.
- **TrustSVD** [8] extends the MF based model [11] to social recommendation. It jointly factorizes both social network and user-item rating matrices to learn richer representations.
- **NeuMF** [9] is a recently proposed state-of-the-art model for rating-only setting. It fuses multi-layer perceptron with matrix factorization model in order to exploit both deep and wide representations.
- **GMF** is a generalization of matrix factorization and proposed as a part of NeuMF [9].
- **BPR** [22] is a standard baseline for top-N ranking setting. It optimizes the pairwise loss function during training.
- **MF** [11] is a standard and widely adopted baseline for collaborative filtering.
- **RecGAT** is a special case of our model which uses only user-item interaction network.

Note that, SAMN and DeepSoR are neural network models, and SBPR and TrustSVD are matrix factorization models for social recommendation. In addition, NeuMF is based on a neural network model, and MF, GMF and BPR are matrix factorization models for the rating-only setting.

Parameter setting and reproducibility. We use Python, Tensorflow 1.12 for our implementation. Our implementation is available at `https://github.com/mvijaikumar/SoRecGAT`

We use the dropout regularizer and adopt RMSProp [7] with mini-batch for optimization. The number of layers, number of heads per layer and number of activation functions per head are sensitive hyperparameters for RecGAT and SoRecGAT. Hyperparameters are tuned using the validation set. From the validation set performance, the number of layers are set to two for RecGAT, SoRecGAT, DeepSoR and NeuMF. Further, for SoRecGAT, the batch size is set to 1024, the number of heads for layers are set to [8,6] for Food dataset and [12,6] for other datasets the number of activation functions per head is set to 32 in the first layer and 96 for Movie, Book and CD, 48 for Hotel and 64 for other datasets in the second layer, the dropout ratio is set to 0.2 for Art and Book datasets and 0.5 for other datasets, learning rate is set to 0.0004 for Music, 0.0001 for Art and 0.00008 for other datasets. We use LeakyRELU as the activation function in equation (3) and exponential linear unit (ELU) as the activation function in other places. Further, we tune l_2-regularization values for SBPR, TrustSVD, DeepSoR, SAMN from {0.005, 0.01, 0.05, 0.1, 0.5, 1, 1.5} and the number of factors for MF, GMF, BPR, SBPR, TrustSVD, DeepSoR and SAMN from {16, 32, 64, 80, 128}, respectively. We use early stopping criterion with the maximum number of epochs for training set to 60.

Model	Music		CD	
	HR@5	NDCG@5	HR@5	NDCG@5
MF	0.6482 ± 0.0158	0.4844 ± 0.0107	0.6779 ± 0.0039	0.5198 ± 0.0032
BPR	0.6555 ± 0.0102	0.4855 ± 0.0082	0.6901 ± 0.0052	0.5340 ± 0.0054
GMF	0.6835 ± 0.0106	0.5163 ± 0.0109	0.7163 ± 0.0061	0.5609 ± 0.0056
NeuMF	0.6854 ± 0.0084	0.5182 ± 0.0095	0.7251 ± 0.0030	0.5776 ± 0.0036
RecGAT (ours)	0.7104 ± 0.0116*	0.5416 ± 0.0098*	0.7504 ± 0.0065*	0.6019 ± 0.0047*
SBPR	0.6646 ± 0.0122	0.4914 ± 0.0092	0.6985 ± 0.0047	0.5485 ± 0.0062
TrustSVD	0.6712 ± 0.0113	0.5015 ± 0.0087	0.7043 ± 0.0072	0.5713 ± 0.0049
DeepSoR	0.6759 ± 0.0082	0.5130 ± 0.0084	0.7373 ± 0.0026	0.5841 ± 0.0036
SAMN	0.6795 ± 0.0080	0.5008 ± 0.0046	0.7245 ± 0.0061	0.5695 ± 0.0042
SoRecGAT (ours)	**0.7333 ± 0.0029**	**0.5582 ± 0.0129**	**0.7796 ± 0.0023**	**0.6225 ± 0.0033**
	Movie		Book	
MF	0.5370 ± 0.0021	0.3799 ± 0.0027	0.7193 ± 0.0008	0.5614 ± 0.0014
BPR	0.5401 ± 0.0047	0.3843 ± 0.0042	0.7144 ± 0.0042	0.5626 ± 0.0021
GMF	0.5590 ± 0.0023	0.4006 ± 0.0012	0.7397 ± 0.0038	0.5931 ± 0.0027
NeuMF	0.5607 ± 0.0053	0.4022 ± 0.0037	0.7457 ± 0.0035	0.5965 ± 0.0033
RecGAT (ours)	0.5815 ± 0.0018*	0.4243 ± 0.0015*	0.7734 ± 0.0012*	0.6241 ± 0.0017*
SBPR	0.5493 ± 0.0034	0.3918 ± 0.0021	0.7217 ± 0.0029	0.5998 ± 0.0018
TrustSVD	0.5531 ± 0.0065	0.3973 ± 0.0032	0.7265 ±0.0032	0.5910 ± 0.0024
DeepSoR	0.5610 ± 0.0042	0.4079 ± 0.0035	0.7478 ±0.0009	0.5964 ± 0.0024
SAMN	0.5621 ± 0.0065	0.4107 ± 0.0033	0.7405 ± 0.0041	0.5937 ± 0.0021
SoRecGAT (ours)	**0.5888 ± 0.0043**	**0.4306 ± 0.0019**	**0.7805 ± 0.0014**	**0.6297 ± 0.0011**

Table 2. Performance of different models on four real-world datasets – Music, CD, Movie, Book from **Amazon**. Social recommendation models are separated from rating-only models. The best overall scores are indicated in boldface, while the best scores among rating-only models are highlighted by asterisk (*). We conduct paired t-test and the improvements using SoRecGAT are statistically significant with $p < 0.01$.

3.2 Results and Discussion

***Overall performance* (RQ1).** Tables 2 and 3 detail the performance of our models and the other comparison models on eight datasets from Amazon and Yelp. Learning influence strength among entities in both user-item interaction network and social network is crucial. To understand this phenomenon, we study two cases here – without social network (RecGAT) and with social network (SoRecGAT). RecGAT achieves better performance consistently across the datasets as compared to the rating-only alternatives – MF, BPR, GMF and NeuMF. From this, we observe that when only implicit ratings are available, understanding the influence

Model	Art		Hotel	
	HR@5	NDCG@5	HR@5	NDCG@5
MF	0.7111 ± 0.0063	0.5124 ± 0.0091	0.8147 ± 0.0006	0.6127 ± 0.0015
BPR	0.7051 ± 0.0057	0.5123 ± 0.0027	0.7994 ± 0.0028	0.6009 ± 0.0025
GMF	0.7235 ± 0.0065	0.5319 ± 0.0068	0.8350 ± 0.0024	0.6359 ± 0.0018
NeuMF	0.7204 ± 0.0083	0.5314 ± 0.0059	0.8313 ± 0.0022	0.6364 ± 0.0017
RecGAT (ours)	0.7371 ± 0.0048*	0.5370 ± 0.0036*	0.8462 ± 0.0044*	0.6454 ± 0.0032*
SBPR	0.7284 ± 0.0062	0.5334 ± 0.0046	0.8332 ± 0.0037	0.6318 ± 0.0026
TrustSVD	0.7310 ± 0.0056	0.5391 ± 0.0032	0.8382 ± 0.0027	0.6353 ± 0.0019
DeepSoR	0.7322 ± 0.0065	0.5363 ± 0.0047	0.8357 ± 0.0040	0.6364 ± 0.0023
SAMN	0.7345 ± 0.0104	0.5374 ± 0.0067	0.8292 ± 0.0025	0.6215 ± 0.0033
SoRecGAT (ours)	**0.7460 ± 0.0051**	**0.5407 ± 0.0038**	**0.8506 ± 0.0039**	**0.6546 ± 0.0035**
	Food		Restaurant	
MF	0.8087 ± 0.0022	0.6086 ± 0.0025	0.7744 ± 0.0017	0.5649 ± 0.0025
BPR	0.7862 ± 0.0027	0.5895 ± 0.0025	0.7536 ± 0.0034	0.5499 ± 0.0023
GMF	0.8285 ± 0.0024	0.6314 ± 0.0015	0.7925 ± 0.0037	0.5881 ± 0.0022
NeuMF	0.8387 ± 0.0038	0.6403 ± 0.0032	0.7945 ± 0.0044	0.5896 ± 0.0034*
RecGAT (ours)	0.8420 ± 0.0016*	0.6442 ± 0.0012*	0.7961 ± 0.0031*	0.5860 ± 0.0029
SBPR	0.8295 ± 0.0028	0.6277 ± 0.0019	0.7904 ± 0.0041	0.5811 ± 0.0028
TrustSVD	0.8380 ± 0.0034	0.6390 ± 0.0024	0.7946 ± 0.0038	0.5874 ± 0.0027
DeepSoR	0.8294 ± 0.0022	0.6333 ± 0.0023	0.7963 ± 0.0025	0.5937 ± 0.0031
SAMN	0.8218 ± 0.0032	0.6119 ± 0.0016	0.7777 ± 0.0034	0.5658 ± 0.0029
SoRecGAT (ours)	**0.8471 ± 0.0074**	**0.6515 ± 0.0017**	**0.8038 ± 0.0042**	**0.5972 ± 0.0033**

Table 3. Performance of different models on four real-world datasets – Art, Hotel, Food and Restaurant from **Yelp**. Social recommendation models are separated from rating-only models. The best overall scores are indicated in boldface, while the best scores among rating-only models are highlighted by asterisk (*). We conduct paired t-test and the improvements using SoRecGAT are statistically significant with $p < 0.01$.

of users and items on each other is essential. RecGAT achieves this by utilizing the multi-head attention mechanism layerwise.

SoRecGAT performs better than both rating-only and other social recommendation models. Note that DeepSoR and SAMN are neural network models. Further, SAMN leverages attention-based memory network and friend-level attention mechanism to learn the influence strength of users from the social network. However, the above procedure is insufficient when we are given access to only implicit ratings. This is because the users may not have an equal opinion on all the items they interact within a system. In contrast, SoRecGAT accounts for this by integrating both user-item interaction network and social network together, and captures the influence strength in an end-to-end fashion using graph attention mechanism. Also note that, in SoRecGAT, the representations of any entity in the graph is obtained from all its neighbours irrespective of its entity type. This provides a more unified framework than DeepSoR and SAMN.

Performance of models with respect to different sparsity levels **(RQ2).**

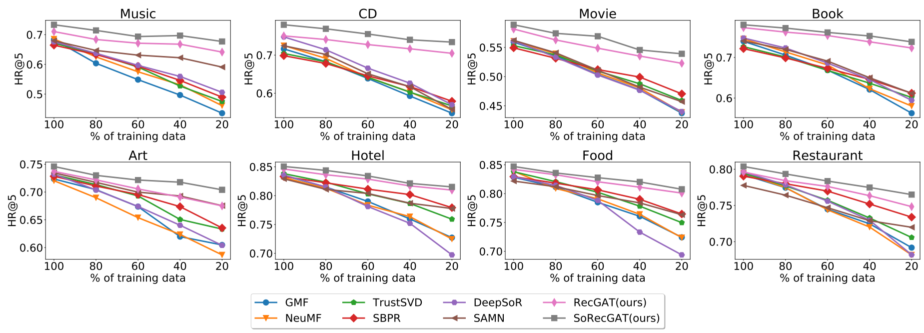

Fig. 2. Performance (**HR@5**) comparison of different models with respect to different sparsity levels on the datasets: Music, CD, Movie, Book, Art, Hotel, Food and Restaurant. Here, we report the mean value obtained from five different experiments for each sparsity level.

To investigate the effectiveness of our models under various sparsity levels, we do the following. We start from the full training set and randomly remove 20% ratings at each step. We continue this until only 20% of the ratings are left in the training set. We repeat this for five different experiments for each sparsity level, and report the mean value. Figs. 2 and 3 show the detailed comparison using the metrics HR@5 and NDCG@5, respectively.

As can be seen from Figs. 2 and 3, RecGAT and SoRecGAT consistently perform better than the other models across different datasets, and their performance does not deteriorate drastically as the sparsity level increases. This is particularly evident for Amazon datasets ((a), (b), (c) and (d) in Figs. 2 and 3). This shows that RecGAT and SoRecGAT are more robust to the situations where data are extremely sparse. From this, we can conclude that learning influence strength among entities in the user-item interaction network and social network by our approach helps in alleviating data sparsity issues.

***Effect of multi-head attention for obtaining influence* (RQ3).** Here, we study the advantage of employing multiple attention heads in layers. We keep two layers, and vary the number of attention heads from [2,1] to [20,10] in the respective layers. The performance of SoRecGAT, in terms of HR@5 and NDCG@5, is depicted in Fig. 4 for Music and Art datasets. From this figure, it is clear that the performance improves, as we increase the number of attention heads. However, in our experiments, we notice that the performance starts deteriorating once the number of attention heads exceeds [12,6] as this results in overfitting. We thus observe that each attention head provides different complementary knowledge about the relationship that exists among entities, which boosts the overall performance of SoRecGAT.

Effect of attention mechanism. Here, we study the effect of attention mechanism in our graph networks. We use the same architecture (two layers with the number of heads set to [12,6], the number of activation functions set to 32

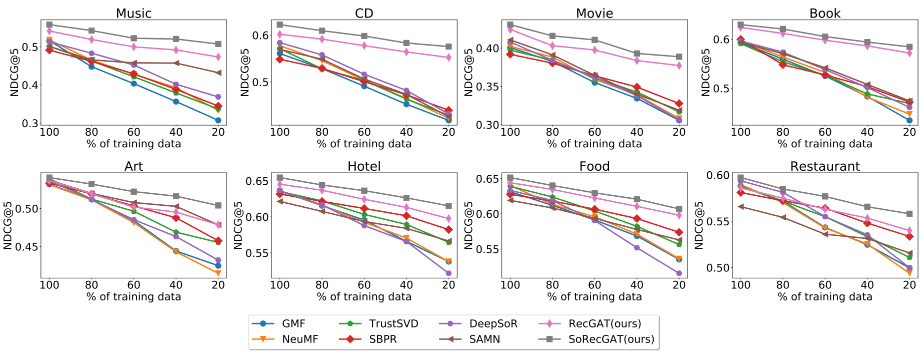

Fig. 3. Performance (**NDCG@5**) comparison of different models with respect to different sparsity levels on the datasets: Music, CD, Movie, Book, Art, Hotel, Food and Restaurant. Here, we report the mean value obtained from five different experiments for each sparsity level.

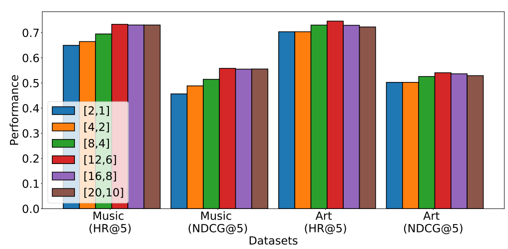

Fig. 4. Performance of SoRecGAT with respect to different number of attention heads in the layers on Music and Art datasets.

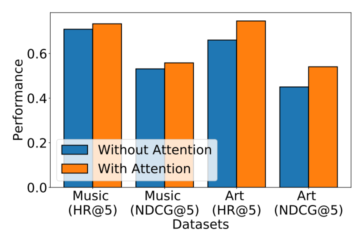

Fig. 5. Performance of the proposed architecture without and with attention mechanism on Music and Art datasets.

and 64 respectively in the first and second layers, and the dropout set to 0.5 for Music and 0.2 for Art) without and with attention mechanism on Music and Art datasets. The performance is shown in Fig. 5 for the two datasets. From this figure, we can observe that attention mechanism in the proposed approach improve the performance.

4 Related Work

In the literature of recommendation systems, early successful models are mostly based on matrix factorization techniques [11, 17, 22]. In particular, [2, 22] are proposed for top-N recommendation framework where only implicit ratings are available. Despite being simple, MF models act as strong baselines among collaborative filtering techniques. Owing to its rich representation capability [7, 12], a surge of neural networks and deep learning models have been proposed for recommendation systems recently [9, 13, 28–30]. In contrast to MF, these models replace the simple dot product between latent representation of users and items with neural networks. Further, He *et al.* [9] proposed NeuMF that marries multi-layer perceptron with generalized matrix factorization model to get the best of both MF and neural network world. Nevertheless, these aforementioned models suffer from data sparsity issues.

Exploiting social connections along with the user-item ratings have been shown to greatly improve the performance of recommendation systems over traditional models that use only ratings [8, 10, 14, 15, 1, 25]. Most existing works on social recommendation extend matrix factorization techniques to incorporate social network information into the recommendation system framework. For instance, SocialMF [10] considers social influence by trust propagation mechanism; SoReg [15] incorporates social connections as regularizers to user representations learned from user-item ratings; and TrustSVD [8] extends SVD++ model [11] to trust and social recommendation. Further, [19, 20, 31] have been proposed specifically for top-N social recommendation tasks. Neural network models [5, 21] also have been proposed for social recommendation framework. However, the above models assume that there exists equal influence across users in the social network, which is not true in practice.

Our work is related to [6, 23, 27, 29] in terms of using graph framework, and [3, 24, 32] in terms of using attention mechanism for the top-N recommendation setting. However, inspired by GAT [26], we employ multiple levels of attention mechanism to account for complex relationships that exist among entities. Further, in contrast to GAT which is proposed for node classifications in graphs, our model is proposed for top-N recommendation setting and the objective function is designed to predict future links between the users and items. Thus, here, the social network helps in fine-tuning the user and item representations.

Furthermore, the models [23, 24, 27] are proposed for session-based social recommendations which require temporal information and [32] requires context information in addition to user-item interaction network and social network. In particular, Wu *et al.* [27] proposed SR-GNN that models session sequences as graph structured data. Further, they employ graph neural networks to capture complex transitions of items. Fan *et al.* [6] proposed GraphRec for social recommendation to jointly model interactions and opinions in the user-item graph. In [29], a graph neural network algorithm called PinSage was proposed. PinSage employs low latency random walks and localized graph convolution operations to learn rich representations for nodes. The model [23] uses graph attention mechanism for learning the influence of users in a social network. In contrast, our model is more general and unified than [23], and the former learns influence from both the social network and user-item interaction network, collectively.

5 Conclusion

In this paper, we presented a novel graph attention-based model, SoRecGAT, for top-N social recommendation. More importantly, our model integrates social network with user-item interaction network and learns the complex relationships among entities by multi-head and multi-layer attention mechanism. We conducted extensive experiments on eight real-world datasets, and demonstrated the effectiveness of the proposed model over state-of-the-art models under various settings. Further, the proposed model has an advantage of using network-specific side information, if available of nodes. Our model is more general and it can be used for recommendations with any number of external networks. In future, we plan to extend these ideas to a multimedia recommendation system where data come from different modalities such as audios, images and videos.

References

1. Abbasi, M.A., Tang, J., Liu, H.: Trust-aware recommender systems. Machine Learning book on computational trust, Chapman & Hall/CRC Press (2014)
2. Balakrishnan, S., Chopra, S.: Collaborative ranking. In: WSDM. pp. 143–152. ACM (2012)
3. Chen, C., Zhang, M., Liu, Y., Ma, S.: Social attentional memory network: Modeling aspect-and friend-level differences in recommendation. In: WSDM. pp. 177–185. ACM (2019)
4. Dong, Y., Chawla, N.V., Swami, A.: metapath2vec: Scalable representation learning for heterogeneous networks. In: SIGKDD. pp. 135–144. ACM (2017)
5. Fan, W., Li, Q., Cheng, M.: Deep modeling of social relations for recommendation. In: AAAI (2018)
6. Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., Yin, D.: Graph neural networks for social recommendation. In: WWW. pp. 417–426 (2019)
7. Goodfellow, I., Bengio, Y., Courville, A., Bengio, Y.: Deep learning, vol. 1. MIT press Cambridge (2016)
8. Guo, G., Zhang, J., Yorke-Smith, N.: Trustsvd: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In: AAAI. vol. 15, pp. 123–125 (2015)
9. He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S.: Neural collaborative filtering. In: WWW. pp. 173–182 (2017)
10. Jamali, M., Ester, M.: A matrix factorization technique with trust propagation for recommendation in social networks. In: RecSys. pp. 135–142. ACM (2010)
11. Koren, Y., Bell, R., Volinsky, C.: Matrix factorization techniques for recommender systems. Computer (8), 30–37 (2009)
12. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. nature **521**(7553), 436 (2015)
13. Li, X., She, J.: Collaborative variational autoencoder for recommender systems. In: SIGKDD. pp. 305–314. ACM (2017)
14. Ma, H., Yang, H., Lyu, M.R., King, I.: Sorec: social recommendation using probabilistic matrix factorization. In: CIKM. pp. 931–940. ACM (2008)
15. Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I.: Recommender systems with social regularization. In: WSDM. pp. 287–296. ACM (2011)
16. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013)
17. Mnih, A., Salakhutdinov, R.R.: Probabilistic matrix factorization. In: Advances in neural information processing systems. pp. 1257–1264 (2008)
18. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: SIGKDD. pp. 701–710. ACM (2014)

19. Rafailidis, D., Crestani, F.: Joint collaborative ranking with social relationships in top-n recommendation. In: CIKM. pp. 1393–1402. ACM (2016)
20. Rafailidis, D., Crestani, F.: Learning to rank with trust and distrust in recommender systems. In: RecSys. pp. 5–13. ACM (2017)
21. Rafailidis, D., Crestani, F.: Recommendation with social relationships via deep learning. In: SIGIR. pp. 151–158. ACM (2017)
22. Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L.: Bpr: Bayesian personalized ranking from implicit feedback. In: Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence. pp. 452–461. AUAI Press (2009)
23. Song, W., Xiao, Z., Wang, Y., Charlin, L., Zhang, M., Tang, J.: Session-based social recommendation via dynamic graph attention networks. In: WSDM. pp. 555–563. ACM (2019)
24. Sun, P., Wu, L., Wang, M.: Attentive recurrent social recommendation. In: SIGIR. pp. 185–194. ACM (2018)
25. Tang, J., Hu, X., Liu, H.: Social recommendation: a review. Social Network Analysis and Mining **3**(4), 1113–1133 (2013)
26. Veličković, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. International Conference on Learning Representations (2018)
27. Wu, S., Tang, Y., Zhu, Y., Wang, L., Xie, X., Tan, T.: Session-based recommendation with graph neural networks. In: AAAI (2019)
28. Xue, H.J., Dai, X., Zhang, J., Huang, S., Chen, J.: Deep matrix factorization models for recommender systems. In: IJCAI. pp. 3203–3209 (2017)
29. Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W.L., Leskovec, J.: Graph convolutional neural networks for web-scale recommender systems. In: SIGKDD. pp. 974–983. ACM (2018)
30. Zhang, S., Yao, L., Sun, A., Tay, Y.: Deep learning based recommender system: A survey and new perspectives. ACM Computing Surveys (CSUR) **52**(1), 5 (2019)
31. Zhao, T., McAuley, J., King, I.: Leveraging social connections to improve personalized ranking for collaborative filtering. In: CIKM. pp. 261–270. ACM (2014)
32. Zhou, C., Bai, J., Song, J., Liu, X., Zhao, Z., Chen, X., Gao, J.: Atrank: An attention-based user behavior modeling framework for recommendation. In: AAAI (2018)