Heterogeneous Graph Augmented Multi-Scenario Sharing Recommendation with Tree-Guided Expert Networks

Xichuan Niu[1*], Bofang Li[2*], Chenliang Li[3†], Jun Tan[2], Rong Xiao[2], Hongbo Deng[2]
[1]School of Remote Sensing and Information Engineering, Wuhan University, China
[2]Alibaba Group, Hangzhou, China
[3]Key Laboratory of Aerospace Information Security and Trusted Computing, Ministry of Education, School of Cyber Science and Engineering, Wuhan University, China
[1]{niuxichuan}@whu.edu.cn, [2]{bofang.lbf, tanjun.tj, xiaorong.xr, dhb167148}@alibaba-inc.com
[3]cllee@whu.edu.cn

ABSTRACT

Sharing recommendation is becoming ubiquitous at almost every e-commerce website, where a user will be recommended a list of users when he wants to share something with others. With the tremendous growth of online shopping users, sharing recommendation confronts several distinct difficulties: 1) how to establish a unified recommender model for large numbers of sharing scenarios; 2) how to handle with long-tail even cold start scenarios with limited training data; 3) how to incorporate social influence in order to make more accurate recommendations.

To tackle with the above challenges, we firstly build multiple expert networks to integrate different scenarios. During model training one specific scenario can learn to differentiate importance of each expert network automatically based on corresponding context information. With respect to the long-tail issue, we propose to maintain a complete scenario tree such that each scenario can utilize context knowledge from root node to leaf node to select the expert networks. At the same time, making use of the tree-based full path message contributes to alleviating training data sparsity problem. Moreover, we construct a large-scale heterogeneous user-to-user graph which is derived from various social behaviors at e-commerce websites. Then a novel scenario-aware multi-view graph attention network is leveraged to augment user representations socially. In addition, an auxiliary inconsistency loss is applied to balance the load of expert networks, along with main click-through rate (CTR) prediction loss, the whole framework is trained in an end-to-end fashion. Both offline experiments and online A/B test results demonstrate the superiority of proposed approach over a bunch of state-of-the-art models.

∗ Euqal contribution.
† Corresponding author.

WSDM '21, March 8–12, 2021, Virtual Event, Israel

ACM ISBN 978-1-4503-8297-7/21/03…$15.00
https://doi.org/10.1145/3437963.3441729

CCS CONCEPTS

• **Information systems** → **Retrieval models and ranking**; • **Computing methodologies** → **Neural networks**.

KEYWORDS

Heterogeneous Graph, E-Commerce, Sharing Recommendation

ACM Reference Format:
Xichuan Niu, Bofang Li, Chenliang Li, Jun Tan, Rong Xiao, Hongbo Deng. 2021. Heterogeneous Graph Augmented Multi-Scenario Sharing Recommendation with Tree-Guided Expert Networks. In *Proceedings of the Fourteenth ACM International Conference on Web Search and Data Mining (WSDM '21), March 8–12, 2021, Virtual Event, Israel.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3437963.3441729

1 INTRODUCTION

Social influence has been shown to be the essential factor that affects user's activities at e-commerce websites [9, 41]. Understanding and leveraging social relations effectively ensures more accurate personalized recommendations and better user experience. Consequently, great efforts have been made to direct users to establish more relationships, such as combining e-commerce with social platforms (*e.g.,* F-commerce of Facebook) and promoting group buying (*e.g.,* Groupon). Particularly, to cater to users' desire to share with others, most e-commerce websites provide the functionality of sharing, which also plays an important role in building and capturing new social relations. Sharing recommendation, as shown in Figure 1, would recommend a source user a list of potential destination users that he wants to share with. A high-quality recommended user list helps bring about fresh social relations smoothly and greatly improves user experience.

Despite the importance of sharing recommendation, it is not easy to develop an efficient sharing recommender system. The challenges of designing algorithms for sharing recommendation are three-fold. The first is *how to establish a unified recommender model that can be applied accross different scenarios.* Every scenario possesses its own unique features, for instance, a user at the item sharing scenario might want to share with someone totally different with live sharing scenario. So the recommender model should be capable of grasping the characteristics of diverse scenarios to generate more precise recommendations. Second, *how to handle with considerable long-tail even cold-start scenarios where training data*

could be very limited. Many times, we do not have adequate context information of the scenario (*e.g.,* some temporary promotion), how to adapt the recommender model to work as normal as possible at these scenarios presents a huge challenge. The third one is *how to incorporate social influence to characterize a user more comprehensively*. Usually a user has numerous social behaviors at e-commerce websites (*e.g.,* being friend of someone, paying on behalf of someone), how to organize and utilize these multi-view side information to build a more precise user portrait remains an open problem.

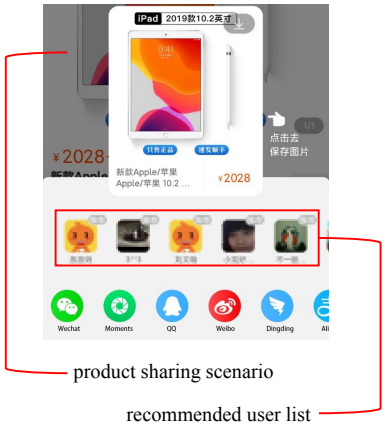


Figure 1: Sharing recommendation example of Taobao mobile application.

Recently, several prior works [5, 23, 27] investigate the mixture of experts architecture to cope with task differences in multi-task learning setting. The basis is that each task can automatically select a particularly useful subnets (*i.e.,* experts) in terms of the characteristics of the current task at both training and serving time. The ensemble of subnetworks and parameter modulation strategy have been proven to be able to enhance the model performance [13]. Therefore, motivated by these advances, we take advantage of mixture of expert networks to unify sharing recommender model of all scenarios. Each scenario leverages its specific context information to determine which experts to use for recommendation. However, considering the large number of long-tail scenarios, directly adopting vanilla expert networks most likely gives rise to suboptimal results.

Therefore, taking into account the traits of e-commerce scenarios, we propose a novel framework named tree-guided expert networks. First of all, we maintain a scenario tree where the following rules must be satisfied: the closer to the root node, the more general description of scenario; the closer to the leaf node, the more concrete characterization of scenario. Then if we are dealing with a specific sharing scenario *Makeups*, we will exploit the corresponding full path starting from root node to leaf node, which is *C2C→Sharing→Entity→Product→Makeups*. This tree-based hierarchical context information will be leveraged to discriminate usefulness of parallel experts and help relieve sparsity issue of training data in long-tail scenarios. Besides, an innovative auxiliary inconsistency loss is involved for the purpose of load balancing of expert networks.

Also, graph based models have achieved remarkable success in search and recommendation tasks [6, 7, 20, 24, 31, 36, 37]. The existing methods can be divided into two categories. The first category takes advantage of graph embedding techniques (*e.g.,* DeepWalk [25], Node2Vec [10]) and then employs the pretrained node embeddings for specific downstream task (*e.g.,* CTR prediction). This two-stage manner may restrict the representation capacity of resultant embeddings so that the model cannot obtain optimal results. The second class resorts to graph neural networks [12, 19, 30] and combines neighbor aggregation and main prediction task in an end-to-end fashion. The various social behaviors of users at e-commerce sites can be modelled as multi-view heterogeneous graph naturally, where each view represents one specific relationship respectively. Hence, we can adapt single-view graph neural networks to multi-view setting with relation-aware aggregation and view-level combination. End-to-end training allows efficient message passing within the graphs and generates more robust user representations.

To this end, in this paper, we propose a heterogeneous graph augmented multi-scenario sharing recommendation framework with tree-guided expert networks, named TreeMS. To the best of our knowledge, this is the first work to focus on the task of sharing recommendation in e-commerce, especially for large numbers of long-tail scenarios. In detail, on the basis of the forementioned traits of e-commerce sharing scenarios, we firstly build and maintain a scenario tree where each single scenario corresponds to a specific full path starting from root node to leaf node semantically. The full path context knowledge is utilized to selectively combine outputs of each layer of expert networks for scenario-aware model learning. As a side effect, this semantic correspondence plays a critical role in relieving training data sparsity issue for long-tail scenarios. Besides, we construct a large-scale multi-view heterogeneous graph that is mined from a variety of users social behaviors at e-commerce websites. Scenario-aware neighbor aggregation and multi-view combination are applied on the heterogeneous graph to augment user representation learning. For model training, besides primary CTR prediction loss, an auxiliary inconsistency loss is proposed for load balance of the experts and the whole architecture is trained in an end-to-end manner.

The main contributions of our work can be summarized as follows:

(1) We propose a novel tree-guided mixture of expert networks to accommodate all sharing scenarios under a single unified recommendation model. Leveraging scenario tree based full path knowledge contributes to mitigating data sparsity issue of long-tail scenarios.

(2) We construct a large-scale user-to-user heterogeneous graph based on user social behaviors at e-commerce websites. And a multi-view graph attention network is applied to enhance user representation learning.

(3) To balance the load of experts, an auxiliary inconsistency loss is proposed to assist the main CTR prediction loss. Both offline and online evaluation results demonstrate the effectiveness of our proposed TreeMS.

2 RELATED WORK

In this section, we review existing literatures which are highly related to our work respectively: graph representation learning for recommendation, social recommendation, and mixture of expert networks.

Graph Representation Learning for Recommendation. In recent years, graph representation learning (GRL) has been a promising topic in both research and industrial community. The methods of GRL can be categorized into two broad classes: network embedding (NE) methods and graph neural network (GNN) algorithms. NE methods [10, 25, 29] mostly tackle the embedding problem by resorting to matrix factorization framework [26] either explicitly or implicitly. GNN methods [12, 19, 30] aim to generalize traditional neural network algorithms (*e.g.,* convolution, attention) to non-Euclidean graph structure data. GRL has also achieved magnificent success in various search and recommendation tasks [8, 31, 33, 36, 37]. In these tasks, the previous works leverage graph representation learning techniques either in two-stage manner or in an end-to-end fashion. Two-stage based methods [2, 31, 36] usually pretrain node embeddings by NE models and then inject the pretrained embeddings into specific downstream tasks (*e.g.,* CTR/CVR prediction). Note that this node embedding learning process is not optimized directly for the final performance, resulting in suboptimal performance. To repair this imperfection, many end-to-end learning solutions [6, 20, 37] are proposed to directly couple heterogeneous or homogeneous graph neural network and the downstream task together, which is able to unleash potential of graph structure as side information to its maximum. In this work, we incorporate multi-view heterogeneous graph into sharing recommendation task in the second (*i.e.,* end-to-end) manner.

Social Recommendation. Social recommendation utilizes user social connections as side information to enhance performance of recommender systems [11, 17, 35]. The basic social influence theory is about the phenomenon that similar preferences are observed among social neighbors [1]. Earlier work on social recommendation attempts to incorporate social network with a social regularization term [21]. Later, more social network aware models are proposed to improve the corresponding traditional recommendation models, for instance, ContextMF [18] for traditional MF, SBPR [38] for traditional pair-wise BPR and TrustSVD [11] for traditional SVD++. Recently, some uptodate efforts [7, 28, 32] leverage graph neural network to model user social network for more accurate social recommendation. In comparison, Our task presented here is similar to social recommendation and the utilization of multi-view heterogeneous graph is derived from various user social behaviors at e-commerce websites.

Mixture of Expert Networks. In deep learning, ensemble of subnetworks or submodules have been proven to be useful for enhancing model performance [13]. The original mixture of experts (MoE) model is proposed in [16] and the recent MoE layer is described in [5, 27]. MMoE [23] is designed to improve MoE layer with multiple gates to model task differences. Furthermore, an advanced subnetwork routing strategy [22] is proposed for flexible parameter sharing in multi-task setting. Our work makes use of the mixture of expert networks architecture and maintain a scenario tree to guide the selection of experts for that scenario.

3 THE PROPOSED APPROACH

In this section, we firstly give a formal problem formulation of sharing recommendation. And then introduce some basics of mixture of experts briefly and show the details of proposed tree-guided expert networks. Next, we describe the components of multi-view heterogeneous graph attention network to enhance user representation learning. At last, the model optimization procedures are described.

3.1 Problem Definition

A scenario-aware sharing recommender system can be formally defined as $\{\mathcal{U}_s, \mathcal{U}_d, \mathcal{S}\}$, where $\mathcal{U}_s$ and $\mathcal{U}_d$ represent the set of source users and destination users respectively and $\mathcal{S}$ is the scenario set. $\mathcal{U}_s$ and $\mathcal{U}_d$ jointly form the complete user set $\mathcal{U} = \mathcal{U}_s \cup \mathcal{U}_d$. A scenario $s \in \mathcal{S}$ is defined as a context environment in which user can share specific content with others. $(u_s, u_d) \in \mathbf{Y}_s$ denotes user u_s has shared with user u_d in scenario s. For each user $u \in \mathcal{U}$, he is associated with exhaustive demographic features and interaction history. Besides, we have a large-scale multi-view heterogeneous graph derived from copious user social behaviors at e-commerce websites, which is formulated as $G_r = (\mathcal{V}, \mathcal{E})$. $\mathcal{V}$ is the user nodes set, $\mathcal{E}$ is the edges set and $r \in \mathcal{R}$ denotes specific view (relation) type. Given extensive features of users and scenarios as well as the multi-view heterogeneous graph, the goal of our model is to learn a rating function $f_s(u_s, u_d) : \mathcal{U} \times \mathcal{U} \rightarrow \mathbb{R}$. Accordingly, a recommended destination users list will be ordered and displayed for source user u_s under sharing scenario s.

3.2 Basics of Mixture of Experts

We provide a brief review of mixture of expert networks commonly used in recent works [23, 39], as illustrated in Figure 2. Usually expert networks are implemented as parallel multi-layer perceptrons which are added on top of a shared bottom layer. Let $\mathbf{x}$ denote the input vector of expert networks, assuming we have n experts and the output vector of kth expert can be expressed as $f^k(\mathbf{x})$. Every scenario (task) s selectively combines outputs of expert networks to reach to the final prediction layer $p^s(\cdot)$, as shown in the following equation:

$$\hat{y}^s = p^s(\sum_{k=1}^{n} g_k^s(\mathbf{x}) f^k(\mathbf{x})) \tag{1}$$

where $\hat{y}^s$ represents prediction of scenario s, $g_k^s(\mathbf{x})$ is the gating scalar value for kth expert, which is calculated as follows:

$$g^s(\mathbf{x}) = softmax(\mathcal{W}_{g^s}\mathbf{x}) \tag{2}$$

where $g^s(\mathbf{x}) \in \mathbb{R}^n$, $\mathcal{W}_{g^s}$ is the linear transformation matrix of gating network for scenario s.

Such design of expert networks guarantees soft-parameter sharing to model scenario (task) relations and conflicts. Nevertheless, in real-world e-commerce websites, we could have hundreds of sharing scenarios, of which most are long-tail even cold-start ones. As a result, we have to allow more parameter sharing explicitly and make full use of context information to solve the severe long-tail issue.

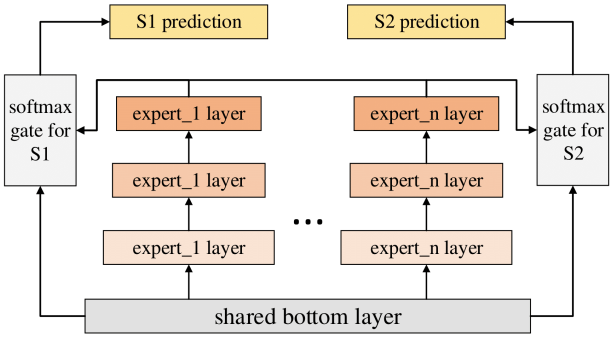


Figure 2: Basics of mixture of experts, where S1 and S2 stand for two different scenarios.

3.3 Tree-guided Expert Networks

The overall model architecture of tree-guided expert networks is illustrated in Figure 3. It is observed that concepts existing in e-commerce sharing scenarios can be formulated as a tree-based hierarchical structure. Within this scenario tree, the closer to the root node, the more generic description of scenario, on the other hand, the closer to the leaf node, the more exact characterization of scenario. For example, when we refer to *Makeups* scenario, the relevant tree hierarchy is *C2C→Sharing→Entity→Product→Makeups*. Obviously *C2C* is a more generalized concept while *Makeups* refers to a specific scenario. More importantly, we can tell that there exist many similar nodes (*e.g., Women Clothes, Women Shoes*) under level *C2C→Sharing→Entity→Product* and even more under level *C2C→Sharing→Entity*. The rich information contained in the scenario tree provides valuable knowledge especially for sharing recommendation in long-tail scenarios. Furthermore, we also exploit this hierarchical structure to selectively combine the outputs of expert networks.

Still we use $\mathbf{x}$ to denote the input of expert networks, which is a wide-concatenated vector in our setting:

$$\mathbf{x} = concat(\mathbf{u}, \mathbf{v}, \mathbf{t}, \mathbf{c}) \tag{3}$$

where $\mathbf{u}$ and $\mathbf{v}$ denote representations of source user u and destination user v respectively. How to augment them with multi-view heterogeneous graph learning will be described later. $\mathbf{t}$ refers to interaction history of u and v and $\mathbf{c}$ is the context information of current scenario.

Different from traditional mixture of experts, we propose to reserve output vectors of each layer of expert networks. Specifically, we have n experts and every one has H layers, so $f_i^k(\mathbf{x})(1 \le i \le H, 1 \le k \le n)$ represents the ith layer output of kth expert. The number of layers of expert networks should be equivalent to the depth of the scenario tree so as to associate them semantically. Along a full path of the scenario tree (*i.e.,* starting with root node and ending with leaf node), each node o belonging to scenario s will be embedded into a unique vector $\mathbf{o}_i^s(1 \le i \le H)$ which is shared across all the scenarios that cover node o in their full paths. Then to retain the sequential nature, we input the node embeddings into an LSTM [14]:

$$\mathbf{h}_i^s = LSTM(\mathbf{o}_i^s), 1 \le i \le H \tag{4}$$

where $\mathbf{h}_i^s$ is the output hidden state vector of $\mathbf{o}_i^s$.

Accordingly, $\mathbf{h}_i^s$ will be used to selectively integrate ith layer outputs of expert networks, which is presented in the following equation:

$$\mathbf{z}_i^s = \sum_{k=1}^{n} g_i^k(\mathbf{h}_i^s) f_i^k(\mathbf{x}) \tag{5}$$

$$g_i(\mathbf{h}_i^s) = softmax(\mathcal{W}_i \mathbf{h}_i^s) \tag{6}$$

where $g_i(\mathbf{h}_i^s) \in \mathbb{R}^n$ is the gating network of ith layer and $g_i^k(\mathbf{h}_i^s)$ is the kth entry. Note that here we make transformation matrix of gating network (*i.e.,* $\mathcal{W}_i$) only depend on layer i, since the input of gating network is already composed of full path information offered by the scenario tree. To some extent, this parameter sharing strategy reduces the computational burden caused by enormous number of sharing scenarios.

As mentioned above, $\mathbf{z}_i^s(1 \le i \le H)$ embodies selective combination of each layer of expert networks in terms of scenario s, to make the most of full path context knowledge, we concatenate $\mathbf{z}_i^s$ before prediction layer $p(\cdot)$:

$$\hat{y}_{uv}^s = p(MLPs(concat(\mathbf{z}_i^s,\ 1 \le i \le H))) \tag{7}$$

where $\hat{y}_{uv}^s$ is the final predicted probability of user u choosing user v in sharing scenario of s. We add several multi-layer perceptrons after concatenation operation to facilitate more feature fusions. The prediction layer is shared accross all scenarios to lessen memory cost (*i.e.,* the number of parameters).

It is worth pointing out that expert networks are devised as tower-like structure (see Figure 3), leading to that the dimension of $\mathbf{z}_i^s$ gradually decreases as the layer i increases. This is reasonable because we would like to utilize more shared knowledge and less particular information about long-tail scenarios considering the training data sparsity issue.

3.4 Multi-view Heterogeneous Graph Augmentation

As stated in last subsection, we leverage multi-view heterogeneous graph that is derived from various social behaviors at e-commerce websites to augment user representation learning, as depicted in Figure 4.

For simplicity, we only describe the procedures for source user and the same operations are carried out for destination user as well. Let $\mathcal{R}$ denote the set of all views, where each view r refers to a specific relation type (*e.g.,* share, friend, pay). We propose to split views and treat each relationship sperately for a certain user u_i (see Figure 4). The lth level embedding of node u_i on relation type r is $\mathbf{u}_{i,r}^l$, which is aggregated from neighbors' embeddings of same view, followed by a update function:

$$\widetilde{\mathbf{u}}_{i,r}^l = aggregate(\mathbf{u}_{j,r}^{l-1}, \forall u_j \in \mathcal{N}_r(i)) \tag{8}$$

$$\mathbf{u}_{i,r}^l = update(\mathbf{u}_{i,r}^{l-1}, \widetilde{\mathbf{u}}_{i,r}^l) \tag{9}$$

where $\mathcal{N}_r(i)$ represents neighbors of u_i on view r.

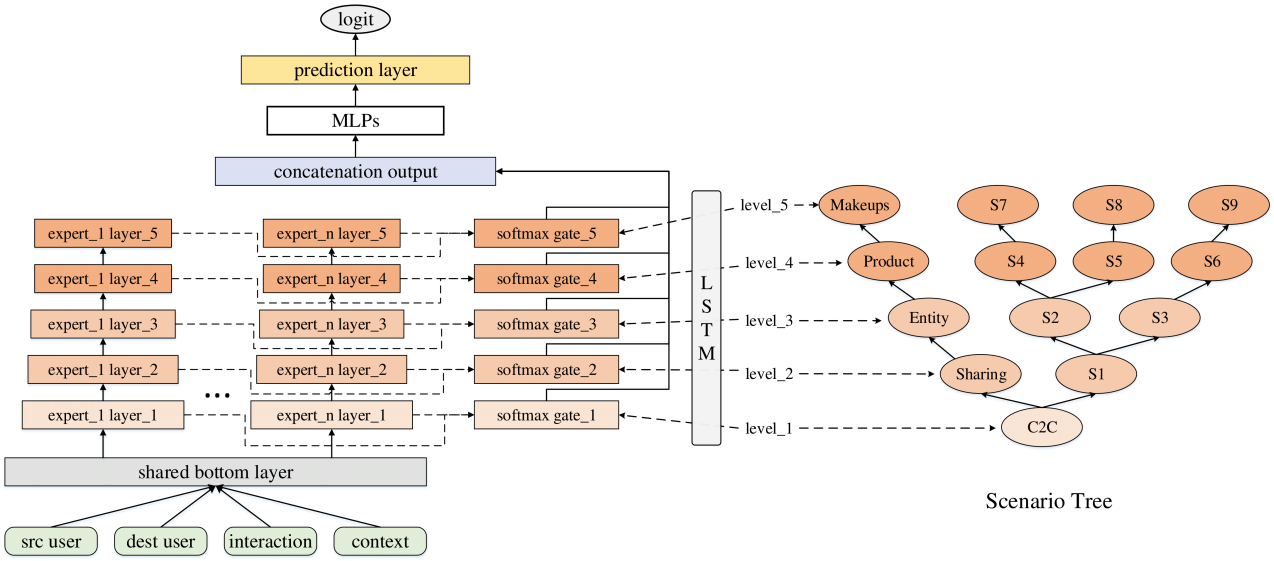


Figure 3: Illustration of tree-guided expert networks, where S1 - S9 stand for different scenarios within the tree and number of expert layers H is set as 5.

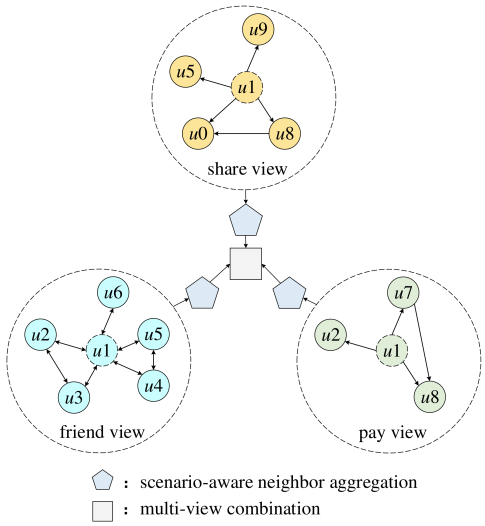


Figure 4: Leveraging multi-view heterogeneous graph to augment user representation learning. The node $u1$ with dashed circle is the target user and the other users are his neighbors under different views.

One intuitive idea is that under different scenarios, contributions of neighbors for the target user are definitely not the same. Hence, we propose to inject context information $\mathbf{c}$ (see Eq. (3)) when implementing the aggregate function. Specifically, we choose the attention mechanism as the backbone of aggregation:

$$\alpha^l_{i,r,j} = \frac{exp(\sigma(\mathbf{a}^\top_{r,l}[\mathbf{u}^{l-1}_{i,r} \| \mathbf{u}^{l-1}_{j,r} \| \mathbf{c}]))}{\sum_{u_m \in \mathcal{N}_r(i)} exp(\sigma(\mathbf{a}^\top_{r,l}[\mathbf{u}^{l-1}_{i,r} \| \mathbf{u}^{l-1}_{m,r} \| \mathbf{c}]))} \tag{10}$$

$$\widetilde{\mathbf{u}}^l_{i,r} = \sum_{u_j \in \mathcal{N}_r(i)} \alpha^l_{i,r,j} \mathbf{u}^{l-1}_{j,r} \tag{11}$$

where $\mathbf{a}_{r,l}$ is the attention parameter of view r at lth level and $\|$ denotes concatenation operation. In practice, we employ $LeakyReLU$ as activation function (*i.e.,* σ). The update function of Eq. (9) is implemented as follows:

$$\mathbf{u}^l_{i,r} = \sigma(\mathcal{W}^l_r \cdot [\mathbf{u}^{l-1}_{i,r} \| \widetilde{\mathbf{u}}^l_{i,r}]) \tag{12}$$

where $\mathcal{W}^l_r$ represents update parameter matrix of view r at lth level and dimensions of $\mathbf{u}^l_{i,r}$, $\mathbf{u}^{l-1}_{i,r}$ and $\widetilde{\mathbf{u}}^l_{i,r}$ should be the same.

Supposing that we stack L graph attention layers in total, then after scenario-aware neighbor aggregation, we get representations of user u_i with regard to all views ($\mathbf{u}^L_{i,r}, \forall r \in \mathcal{R}$). A natural idea is to do multi-view combination to get the final user representation:

$$\mathbf{u}_i = combine(\mathbf{u}^L_{i,r},\ \forall r \in \mathcal{R}) \tag{13}$$

where we adopt simple but effective concatenation as combine function practically. Up to now, we have augmented user representations through scenario-aware neighbor aggregation and multi-view combination, which will be fed into shared bottom layer together with other features as shown in Figure 3.

3.5 Model Optimization

For model training, we adopt the commonly used binary cross entropy as main CTR prediction loss function:

$$\mathcal{L}_{main} = -\sum_s \sum_u \sum_v y^s_{uv} log(\hat{y}^s_{uv}) + (1 - y^s_{uv}) log(1 - \hat{y}^s_{uv}) \tag{14}$$

where y^s_{uv} is the ground truth of user u choosing user v under scenario s and $\hat{y}^s_{uv}$ is the corresponding model prediction.

Table 1: Statistics of dataset and graph relations.

data	# interactions / relations	# covered users
train set	$2.4 * 10^8$	$1.2 * 10^7$
test set	$1.0 * 10^6$	$3.7 * 10^4$
friend view	$1.3 * 10^9$	$3.2 * 10^8$
pay view	$4.7 * 10^7$	$3.6 * 10^7$
share view	$7.4 * 10^8$	$1.3 * 10^8$

Note that we do not want some single expert network to dominate while others remain nearly useless. However, in our preliminary evaluation, the training with the loss defined in Eq. (14) could easily meet this adverse situation. Inspired by the related work in [15], to balance the load of each parallel expert network, we propose the following auxiliary inconsistency loss:

$$\mathcal{L}_{aux} = -\sum_{i=1}^{H}\sum_{k=1}^{n}\sum_{k'=k+1}^{n} g_i^k(\mathbf{h}) \times g_i^{k'}(\mathbf{h}) \tag{15}$$

where the model is optimized towards guiding each expert network to be evenly utilized to its maximum.

Finally, combining main CTR prediction loss and auxiliary inconsistency loss, we get the overall loss function for our model:

$$\mathcal{L} = L_{main} + \gamma L_{aux} + \lambda \|\Theta\|_2^2 \tag{16}$$

where γ and λ are predefined hyper-parameters and $\|\Theta\|_2^2$ represents $L2$ regularization over model parameters. We take AdaGrad [4] as the optimizer to perform the gradient backpropagation and optimize the model.

4 EXPERIMENTS

In this section, we conduct extensive experiments on a real-world e-commerce sharing dataset by evaluating the proposed TreeMS against a series of state-of-the-art baseline algorithms. Performances of offline and online are reported for both head and long-tail scenarios. We also give some in-depth analysis about model components and experimental results.

4.1 Experimental Setup

4.1.1 Datasets. We collect the dataset from user daily sharing logs of Taobao mobile application platform during July 2020, which covers more than two hundred scenarios. Within the sharing recommender system, the clicked recommended destination users are held as positive samples while the impressions but not clicked are treated as negative ones. The user logs of first day are taken for model training and the next day is uniformly sampled for testing. For the purpose of leveraging heterogeneous graph, we construct multi-view relationships (*e.g.,* friend, pay, share) from various user social behaviors accross one month before the training day. Besides, we also select representative head scenarios and long-tail scenarios according to the page views to verify model's advantage under different settings. The detailed statistics of dataset and graph relations are summerized in Table 1.

4.1.2 Baseline Methods. We choose three distinct categories of baseline methods for comparison: (1) deep social recommendation models (DNN, CUNE [35], DiffNet [32]), (2) graph based representation learning algorithms (PinSage [34], GAT [30]), (3) multi-task and meta learning methods (Multi-Scenario, MMoE [23], ScenarioMeta [3]).

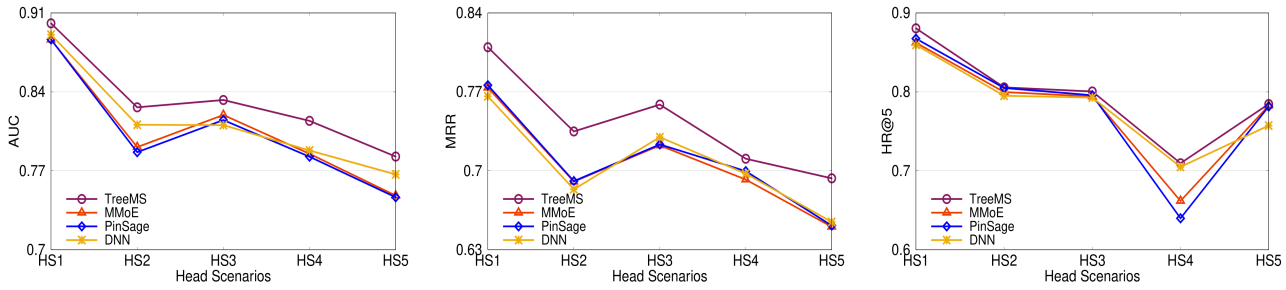
- DNN: This is the fundamental deep neural network model, where we simply replace expert networks of our model with several multi-layer perceptrons.
- CUNE [35]: A collaborative user network embedding model for social recommender systems that incorporates top-k semantic friends information into MF and BPR frameworks.
- DiffNet [32]: The state-of-the-art deep influence propagation model to simulate how users are affected by the recursive diffusion for social recommendation.
- PinSage [34]: The latest web-scale recommender system of Pinterest that leverages GraphSAGE [12] as the backbone model. We report empirical results using three different neighbor aggregation strategies respectively: Average (PinSage_A), Max-pooling (PinSage_M), LSTM (PinSage_L).
- GAT [30]: The self-attention based graph neural network model that aggregates homogeneous neighbors as well as target node attentively.
- Multi-Scenario: This model is adapted from basic DNN model to fit for multiple scenarios, where we add a fully connected layer for scenario dependent transformation before prediction layer.
- MMoE [23]: The multi-gate mixture of experts model from Google, which explicitly learns to model task relationships from data in multi-task learning setting.
- ScenarioMeta [3]: The state-of-the-art sequential scenario specific meta learning based model for online recommendation.

4.1.3 Evaluation Protocol. Our primary goal is to predict the click probability of a candidate user for a target user under specific sharing scenario, according to which a destination user list will be ordered and displayed. Hence we adopt four widely used evaluation metrics, which reflect model performance from different prospectives:

- **AUC** (Area Under the receiver operating characteristic Curve). AUC is a commonly used metric that actually reflects the ranking quality of the model.
- **GAUC** (Group AUC). This metric is firstly proposed in [40], which calculates AUC with regard to each user and sums over all users using frequency as weight.
- **MRR** (Mean Reciprocal Rank). This is a standard metric for evaluating performance of recommender systems. MRR computes average reciprocal rank for the destination users that users actually shared with.
- **HR** (Hit Ratio). HR@k measures whether ground truth items will appear in the predicted top-k list. In our experiments, we report results when $k = 5, 10$ respectively.

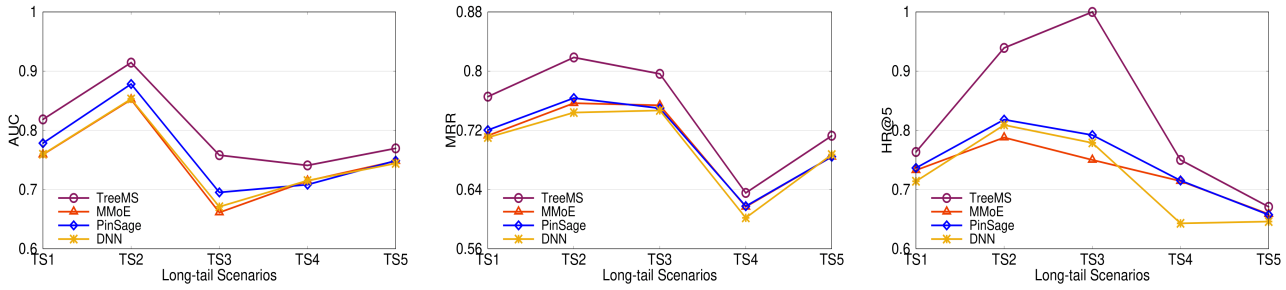
4.2 Overall Performance

We report the model performances over all scenarios in Table 2, from which we can make the following observations:



(a) AUC (b) MRR (c) HR@5

Figure 5: *AUC*, *MRR* and *HR@5* in five representative head scenarios (HS1: product sharing, HS2: Tmall farm, HS3: shop sharing, HS4: live, HS5: fission coupon).



(a) AUC (b) MRR (c) HR@5

Figure 6: *AUC*, *MRR* and *HR@5* in five representative long-tail scenarios (TS1: qingwa2, TS2: yangtaodashang, TS3: fenxiangyouli, TS4: fenxiangyouhaohuo, TS5: animals3).

- Our proposed model consistently performs best among all methods over all scenarios on all evaluation metrics. Specifically, our model outperforms best baseline by 2.08%, 1.14%, 1.42%, 0.94%, 1.32% on AUC, GAUC, MRR, HR@5, HR@10 respectively. These relative performance gains are attributed to the elaborated design of tree-guided expert networks and multi-view heterogeneous graph augmented user representation learning.
- Traditional social recommendation algorithms perform unsatisfactorily because they cannot distinguish the characteristics of each scenario and neglect serious long-tail issue. Graph based methods achieve better performance since they take advantage of neighbor relationships to obtain more robust user representations. Nevertheless, PinSage and GAT cannot take into consideration the nuances of multi-view user social behaviors. Multi-tasking related models can learn the similarities and differences among scenarios automatically but we should employ sophisticated mixture of experts architecture instead of simple scenario-aware transformation.
- Meanwhile, the proposed TreeMS attains significant performance gains on five distinct evaluation metrics, which reflects different aspects of a model. Thus we can conclude that our model is effective and scalable in both point-wise prediction and list-wise ordering.

4.3 Scenario Specific Performance

Besides the overall results over all scenarios, we pay more attention to performance under specific scenarios. Therefore, we pick out five representative head scenarios and five long-tail ones according to their page views. The experimental results of our model, MMoE, PinSage_A and DNN in terms of AUC, MRR and HR@5 are drawn in Figure 5 and Figure 6.

Figure 5 is about five head scenarios of *product sharing, Tmall farm, shop sharing, live, fission coupon*. On the other hand, Figure 6 is concerning five obscure long-tail scenarios of *qingwa2, yangtaodashang, fenxiangyouli, fenxiangyouhaohuo, animals3*. As presented in the two figures, we can tell that our model significantly outperforms all the three other methods by a large margin, especially under long-tail scenarios. It proves that the establishment of scenario tree and utilization of full path context knowledge indeed benefit a great quantity of long-tail scenarios.

4.4 Visualization

4.4.1 Analysis of Expert Utilization. Recall that we propose Eq. (15) in order to balance the load of each expert network. To validate the

Table 2: Overall performance of our proposed model and baseline methods. The * represents the best performance of the baselines. Best results of all methods are highlighted in boldface. Improvement over the best baseline are shown in the last row.

Model	AUC	GAUC	MRR	HR@5	HR@10
DNN	0.7347	0.6946	0.6741	0.6396	0.8325
CUNE	0.7429	0.6963	0.6669	0.6401	0.8232
DiffNet	0.7530	0.6974	0.6683	0.6416	0.8256
PinSage_A	0.7548	0.7001	0.6679	0.6450	0.8350*
PinSage_M	0.7536	0.6982	0.6645	0.6427	0.8314
PinSage_L	0.7552	0.7017*	0.6691	0.6483*	0.8342
GAT	0.7513	0.6978	0.6628	0.6413	0.8306
Multi-Scenario	0.7385	0.6949	0.6752*	0.6411	0.8323
MMoE	0.7579*	0.7009	0.6657	0.6467	0.8349
ScenarioMeta	0.7557	0.6948	0.6637	0.6417	0.8278
TreeMS	**0.7737**	**0.7097**	**0.6848**	**0.6544**	**0.8460**
Impv.	2.08%	1.14%	1.42%	0.94%	1.32%

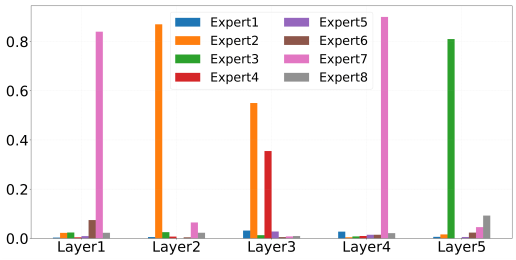


Figure 7: Visualization of expert utilization of each layer without inconsistency loss.

efficacy of additional inconsistency loss, we randomly select a scenario and visualize the expert utilization of each layer without and with Eq. (15), as illustrated in Figure 7 and Figure 8. Note that in our experiments the number of experts is set to 8.

As can be seen from these two figures, without the inconsistency loss, it is usual that some expert dominate the prediction decision making in each layer, leading to a large waste of computing resources. On the contrary, when we combine the main CTR prediction loss with auxiliary inconsistency loss, the domination of certain experts disappears and we acquire a relatively balanced distribution of expert utilization. This verifies the necessity of adding the auxiliary loss for load balance of the expert networks.

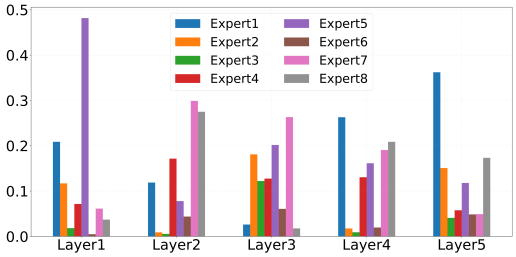
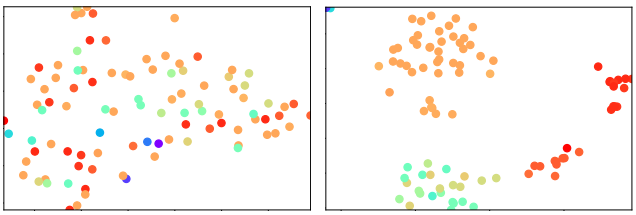


Figure 8: Visualization of expert utilization of each layer with inconsistency loss.



(a) MMoE (b) TreeMS

Figure 9: Visualization of learned scenario-aware representations of MMoE and Our model.

4.4.2 Exploration of Learned Scenario-aware Representations. We also examine the semantic expression ability of learned scenario-aware representations of different models. Figure 9(a) is a visualization of scenario-aware context embeddings that MMoE obtained after training. Figure 9(b) gives another display of tree-based scenario-aware representations of our model. Note that we paint similar scenarios with unanimous colors.

From Figure 9, we observe that the embeddings of MMoE follow a practically irregular distribution, which indicates that MMoE model cannot generate semantically meaningful scenario-aware representations. While the tree-based scenario-aware representations of our model present several obvious clusterings, further suggesting correctness and effectiveness of maintaining the scenario tree.

4.5 Online A/B Test

We deploy our model on sharing recommendation platform of Taobao community relation engine and conduct online A/B test (*i.e.*, bucket test) with comparison to MMoE, since the latter offers strong offline performance among baseline methods. Another deployed industrial algorithm is a variant of our model without incorporation of graphs. The multi-view heterogeneous graph data covers the period of last one month and can be updated incrementally everyday. We choose online CTR as the metric of evaluating bucket test results. The average relative improvements of seven days over MMoE and the variant model are **1.80%** and **1.47%**, respectively.

5 CONCLUSION AND FUTURE WORK

In this paper, we present a heterogeneous graph augmented multi-scenario sharing recommendation framework with tree-guided expert networks. Specifically, we take the first step to construct a scenario tree where the hierarchical structure contains more plentiful information of each scenario. Then we integrate it with expert networks to assign all scenarios under one unified model. The utilization of full path context starting from root node to leaf node also benefits a multitude of long-tail scenarios where training data can be very sparse. After that we introduce a large-scale multi-view user-to-user heterogeneous graph, which strengthens representations of users by fusing a variety of social influence. We conduct comprehensive experiments on real-world datasets and both offline evaluation and online A/B tests prove the superiority of proposed model over strong baseline methods, particularly under long-tail scenarios.

For future work, we plan to investigate user interaction history on other platforms (*e.g.,* search, guess you like) to help improve performance of recommender system of sharing scenarios. And other advanced heterogeneous graph representation learning methods are also considered to better characterize users.

ACKNOWLEDGMENTS

This work was supported by Alibaba Group through Alibaba Innovative Research Program and National Natural Science Foundation of China (No. 61872278).

REFERENCES

[1] Aris Anagnostopoulos, Ravi Kumar, and Mohammad Mahdian. 2008. Influence and correlation in social networks. In *KDD*. 7–15.
[2] Yukuo Cen, Xu Zou, Jianwei Zhang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Representation learning for attributed multiplex heterogeneous network. In *KDD*. 1358–1368.
[3] Zhengxiao Du, Xiaowei Wang, Hongxia Yang, Jingren Zhou, and Jie Tang. 2019. Sequential Scenario-Specific Meta Learner for Online Recommendation. In *KDD*. 2895–2904.
[4] John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research* 12, Jul (2011), 2121–2159.
[5] David Eigen, Marc'Aurelio Ranzato, and Ilya Sutskever. 2013. Learning factored representations in a deep mixture of experts. *arXiv preprint arXiv:1312.4314* (2013).
[6] Shaohua Fan, Junxiong Zhu, Xiaotian Han, Chuan Shi, Linmei Hu, Biyu Ma, and Yongliang Li. 2019. Metapath-guided heterogeneous graph neural network for intent recommendation. In *KDD*. 2478–2486.
[7] Wenqi Fan, Yao Ma, Qing Li, Yuan He, Eric Zhao, Jiliang Tang, and Dawei Yin. 2019. Graph neural networks for social recommendation. In *WWW*. 417–426.
[8] Jingyue Gao, Yang Lin, Yasha Wang, Xiting Wang, Zhao Yang, Yuanduo He, and Xu Chu. 2020. Set-Sequence-Graph: A Multi-View Approach Towards Exploiting Reviews for Recommendation. In *CIKM*. 395–404.
[9] David Gefen. 2000. E-commerce: the role of familiarity and trust. *Omega* 28, 6 (2000), 725–737.
[10] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *KDD*. 855–864.
[11] Guibing Guo, Jie Zhang, and Neil Yorke-Smith. 2015. TrustSVD: collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In *AAAI*. 123–129.
[12] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *NeurIPS*. 1024–1034.
[13] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531* (2015).
[14] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997), 1735–1780.
[15] Wan-Ting Hsu, Chieh-Kai Lin, Ming-Ying Lee, Kerui Min, Jing Tang, and Min Sun. 2018. A Unified Model for Extractive and Abstractive Summarization using Inconsistency Loss. In *ACL*. 132–141.
[16] Robert A Jacobs, Michael I Jordan, Steven J Nowlan, and Geoffrey E Hinton. 1991. Adaptive mixtures of local experts. *Neural computation* 3, 1 (1991), 79–87.
[17] Mohsen Jamali and Martin Ester. 2010. A matrix factorization technique with trust propagation for recommendation in social networks. In *RecSys*. 135–142.
[18] Meng Jiang, Peng Cui, Fei Wang, Wenwu Zhu, and Shiqiang Yang. 2014. Scalable recommendation with social contextual information. *IEEE Transactions on Knowledge and Data Engineering* 26, 11 (2014), 2789–2802.
[19] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).
[20] Feng Li, Zhenrui Chen, Pengjie Wang, Yi Ren, Di Zhang, and Xiaoyu Zhu. 2019. Graph Intention Network for Click-through Rate Prediction in Sponsored Search. In *SIGIR*. 961–964.
[21] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *WSDM*. 287–296.
[22] Jiaqi Ma, Zhe Zhao, Jilin Chen, Ang Li, Lichan Hong, and Ed H Chi. 2019. Snr: Sub-network routing for flexible parameter sharing in multi-task learning. In *AAAI*. 216–223.
[23] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *KDD*. 1930–1939.
[24] Xichuan Niu, Bofang Li, Chenliang Li, Rong Xiao, Haochuan Sun, Hongbo Deng, and Zhenzhong Chen. 2020. A Dual Heterogeneous Graph Attention Network to Improve Long-Tail Performance for Shop Search in E-Commerce. In *KDD*. 3405–3415.
[25] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *KDD*. 701–710.
[26] Jiezhong Qiu, Yuxiao Dong, Hao Ma, Jian Li, Kuansan Wang, and Jie Tang. 2018. Network embedding as matrix factorization: Unifying deepwalk, line, pte, and node2vec. In *WSDM*. 459–467.
[27] Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538* (2017).
[28] Weiping Song, Zhiping Xiao, Yifan Wang, Laurent Charlin, Ming Zhang, and Jian Tang. 2019. Session-based social recommendation via dynamic graph attention networks. In *WSDM*. 555–563.
[29] Jian Tang, Meng Qu, Mingzhe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. In *WWW*. 1067–1077.
[30] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).
[31] Jizhe Wang, Pipei Huang, Huan Zhao, Zhibo Zhang, Binqiang Zhao, and Dik Lun Lee. 2018. Billion-scale commodity embedding for e-commerce recommendation in alibaba. In *KDD*. 839–848.
[32] Le Wu, Peijie Sun, Yanjie Fu, Richang Hong, Xiting Wang, and Meng Wang. 2019. A neural influence diffusion model for social recommendation. In *SIGIR*. 235–244.
[33] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. 2019. Session-based recommendation with graph neural networks. In *AAAI*. 346–353.
[34] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *KDD*. 974–983.
[35] Chuxu Zhang, Lu Yu, Yan Wang, Chirag Shah, and Xiangliang Zhang. 2017. Collaborative User Network Embedding for Social Recommender Systems. In *Proceedings of the 2017 SIAM International Conference on Data Mining, Houston, Texas, USA, April 27-29, 2017*. 381–389.
[36] Yuan Zhang, Dong Wang, and Yan Zhang. 2019. Neural IR Meets Graph Embedding: A Ranking Model for Product Search. In *WWW*. 2390–2400.
[37] Jun Zhao, Zhou Zhou, Ziyu Guan, Wei Zhao, Wei Ning, Guang Qiu, and Xiaofei He. 2019. IntentGC: a scalable graph convolution framework fusing heterogeneous information for recommendation. In *KDD*. 2347–2357.
[38] Tong Zhao, Julian McAuley, and Irwin King. 2014. Leveraging social connections to improve personalized ranking for collaborative filtering. In *CIKM*. 261–270.
[39] Zhe Zhao, Lichan Hong, Li Wei, Jilin Chen, Aniruddh Nath, Shawn Andrews, Aditee Kumthekar, Maheswaran Sathiamoorthy, Xinyang Yi, and Ed Chi. 2019. Recommending what video to watch next: a multitask ranking system. In *RecSys*. 43–51.
[40] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *KDD*. 1059–1068.
[41] Lina Zhou, Ping Zhang, and Hans-Dieter Zimmermann. 2013. Social commerce research: An integrated view. *Electronic commerce research and applications* 12, 2 (2013), 61–68.