Attribute Graph Neural Networks for Strict Cold Start Recommendation

Tieyun Qian, *Member, IEEE*, Yile Liang, Qing Li, *Senior Member, IEEE*, and Hui Xiong, *Fellow, IEEE*

Abstract—Rating prediction is a classic problem underlying recommender systems. It is traditionally tackled with matrix factorization. Recently, deep learning based methods, especially graph neural networks, have made impressive progress on this problem. Despite their effectiveness, existing methods focus on modeling the user-item interaction graph. The inherent drawback of such methods is that their performance is bound to the density of the interactions, which is however usually of high sparsity. More importantly, for a strict cold start user/item that neither appears in the training data nor has any interactions in the test stage, such methods are unable to learn the preference embedding of the user/item since there is no link to this user/item in the graph. In this work, we develop a novel framework *Attribute Graph Neural Networks* (AGNN) by exploiting the attribute graph rather than the commonly used interaction graph. This leads to the capability of learning embeddings for the strict cold start users/items. Our AGNN can produce the preference embedding for a strict cold user/item by learning on the distribution of attributes with an extended variational auto-encoder (eVAE) structure. Moreover, we propose a new graph neural network variant, i.e., gated-GNN, to effectively aggregate various attributes of different modalities in a neighborhood. Empirical results on three real-world datasets demonstrate that our model yields significant improvements for strict cold start recommendations and outperforms or matches the state-of-the-art performance in the warm start scenario.

Index Terms—Recommender systems, rating prediction, graph neural networks, strict cold start recommendation

1 Introduction

Rating prediction is a well-known recommendation task aiming at predicting a user's ratings for those items which are not rated yet by the user. Collaborative filtering (CF) [1], which mainly makes use of historical ratings, has been successfully used to build recommender systems in various domains. Matrix factorization (MF) [2] is one of the most prevalent method in CF due to its high predicting performance and scalability. Given a $M \times N$ user-item rating matrix, MF first performs a low rank approximation to learn the user's and item's latent representation, also known as *preference embedding* of a user or an item, and then uses a score function over the learnt preference embeddings to generate ratings for the missing entries in the matrix. Sparsity and its special case of *cold start, where a new user/item that does not appear in the training data*, are the severe problem in recommender systems. The performance of MF methods will drop quickly in the sparsity or the cold start settings. Conventional approaches to this issue are to generate *feature embedding* using side or external information [3], [4], [5], [6], [7], [8], [9]. Such methods often introduce additional objective terms which make the learning and inference process very complicated.

The cold start users/items refer to those not appearing in the training data. We call them as *normal cold start users/items*. In this paper, we are interested in an extreme scenario, i.e., *the strict cold start users/items that neither appear in the training data nor have any interactions (user-item links) at the test stage*. Conventional approaches like inductive learning [10], [11], optimization-based meta learning [12], [13], and HIN based methods [13], [14], [15], [16], are inappropriate for the strict cold start problem since they all require the users/items have interactions in test.

Recent advances in deep learning, especially graph neural networks (GNNs), shed new light on the classic rating prediction problem. The main advantage of GNN is that it can represent information from its neighborhood [17], [18], [19], [20], [21], [22]. GNN allows learning high-quality user and item representations, and consequently achieves the state-of-the-art performance. However, almost all existing GNN based methods are built upon the user-item bipartite graph, where the node denotes a user or an item, and the edge is the interaction between the user and the item. Hence such methods cannot be used for strict cold start recommendation if no side information is involved.

Indeed, little attention has been paid on using graph neural network architectures to address the (strict) cold start issues. We are aware of a few GNN based methods towards this problem, i.e., STAR-GCN [23] and HERS [24]. Despite its effectiveness, STAR-GCN has an inherent limitation, i.e., its performance is bound to the number of interactions. The reason is that STAR-GCN relies on the interaction graph. It requires an ask-to-rate technique to rate the new item during the test phase. This means that STAR-GCN can be applied to the normal cold start only. Moreover, though HERS utilizes user-user and item-item relations to address the strict cold start problem by referring to the influential

- *Tieyun Qian and Yile Liang are with the Wuhan University, Wuhan 430072, China. E-mail: {qty, liangyile}@whu.edu.cn.*
- *Qing Li is with the Hong Kong Polytechnic University, Hong Kong. E-mail: qing-prof.li@polyu.edu.hk.*
- *Hui Xiong is with the Rutgers, the State University of New Jersey, New Brunswick, NJ 08854 USA. E-mail: hxiong@rutgers.edu.*

Manuscript received 11 May 2020; revised 23 Sept. 2020; accepted 12 Nov. 2020. Date of publication 16 Nov. 2020; date of current version 7 July 2022. (Corresponding author: Tieyun Qian.) Recommended for acceptance by M. Wang. Digital Object Identifier no. 10.1109/TKDE.2020.3038234

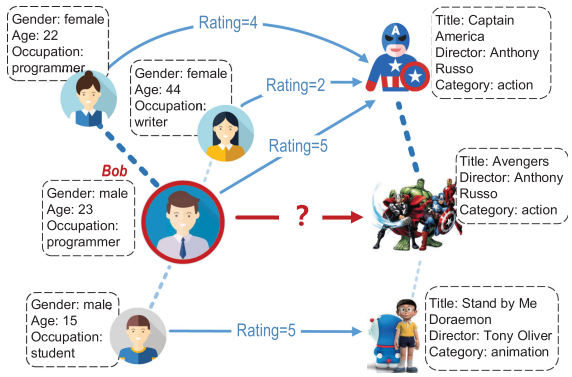


Fig. 1. Motivation example. Solid line denotes a user's rating on the item. Dotted line denotes the user-user or item-item link constructed from their proximity.

nodes in contexts, the drawback is that it might recommend the popular item to the new user, or vice versa, as it represents the new node by neighbor aggregation without considering the node's own attributes.

In order to address the above limitations, we propose a novel framework *Attribute Graph Neural Networks* (AGNN) by exploiting the attribute graph instead of the widely used user-item graph. Unlike the ratings, the attributes are available even for strict cold start users/items. For example, when a merchant starts to sale its products online, there is no interaction for the product. However, it is necessary to provide the product attributes such as the category, description, and image. Similarly, many web-sites ask users to fill their profile like gender and location at the time of registration. In case that the user's profile information is unavailable due to the privacy concerns, the user's links, which are ubiquitous in almost all social networks or online sites, can be treated as his/her attributes (as we will later show in our experiments).

Motivation Example. The difficulty in strict cold start recommendation lies in the lack of preference information (i.e., historical interactions) of users and items. Fig. 1 presents the users' historical rating behaviors on various movies. When a new movie "Avengers" is released, it is a strict cold start item since it is not included in the training data and it does not have any interactions, and it is hard to predict the user Bob's rating on this movie. Fortunately, the attribute information such as the movie's director and its category can be exploited to represent the movie. Furthermore, the movies having the same attributes can form a graph which will be used to the pass preference information from the neighbor movie like "Captain America" to "Avengers".

While it seems to be ready to exploit the attribute information for strict cold start recommendations, there are two key challenges that hinder its potential. One is how to transform the attribute representation into the preference representation. The other is how to effectively aggregate attributes of different modalities, e.g., textual description and image, of the nodes in a neighborhood. In this work, we first develop *an extended variational auto-encoder (eVAE) structure* to generate preference embedding from the reconstructed attribute distribution, with the perception that users' or items' preference can be inferred from their attributes. For example, a female user may prefer the romantic movie to the horror one. We further design *a gated-GNN structure* to aggregate the complicated node embeddings in the same neighborhood, which enables a leap in model capacity since it can assign different importance to each dimension of the node embeddings.

We conduct extensive experiments on three real-world datasets. Results demonstrate that our proposed AGNN model yields significant improvements over the state-of-the-art baselines for strict cold start recommendations, and it also outperforms or matches the performance of these baselines in the warm start scenario. In summary, the contributions of this work are three-fold.

- We highlight the importance of exploiting the attribute graph rather than the interaction graph in addressing strict cold start problem in neural graph recommender systems.
- We design a novel eVAE structure to effectively infer the users'/items' preference embeddings from their attribute distributions with the empowered approximation ability.
- We address the key challenges in aggregating various attributes in a neighborhood by developing a gated-GNN structure which greatly improves the model capacity.

2 Related Work

In this section, we first review the literature in rating prediction and then focus on the highly relevant work on GNN based recommendation and cold start recommendation.

2.1 Collaborative Filtering

Collaborative filtering is commonly used to leverage the user-item interaction data for recommendation. It mainly consists of neighbor-based methods [25], [26] and matrix factorization methods [2], [27]. Recently, the CF approaches are extended with various types of deep learning techniques [19], [28], [29], [30]. Although these methods have improved the performance of CF, none of them utilizes the side information for enhancing the performance of recommender systems.

2.2 Graph Neural Network Based Recommendation

The first GNN architecture employed for recommendation is graph convolution network (GCN). After that, a good number of GNN methods have been proposed for recommendation. We discuss these methods by the graph types and the approaches in utilizing the GNN.

Graph types. Most of previous methods employ the existing graph as their input of GNN, such as the user-item interaction graph [17], [19], [23], [31], [32], [33], users' social graph [20], [21], [24], and knowledge graph [9], [34]. A few studies present graph construction methods to fit the user-user or item-item graph in GNN for recommendation. For example, RMGCNN [31] constructs the user-user or item-item graphs as un-weighted k-nearest neighbor graphs in the space of user and item features. DANSER [21] calculates the relevance between two items by the common users who click or rate them to construct item-item graph. HERS [24] builds the item-item links based on the number of common tags between items.

Approaches in utilizing GNN. RMGCNN [31] adopts GCN framework to aggregate information from user-user and

item-item graphs. PinSage [18] combines random walks and graph convolutions to incorporate graph structure and node feature. STAR-GCN [23] designs a stacked and reconstructed GCN to improve the prediction performance. The GNN architecture is also used for the recursive diffusion in social recommendation [20], [35]. For example, NGCF [19] encodes the high-order connectivity by performing the embedding propagation. DANSER [21] deploys graph attention networks for modeling social effects for recommendation tasks.

A brief summary. Our proposed model differs from prior GNN based methods in both the graph type and the GNN method. First, our model is based on the attribute graph while most of existing methods are based on the interaction graph. Though RMGCNN [31] also employs the attribute graph, it directly applies GCN to the attribute graph without a preference reconstruction process. Second, we present a novel GNN architecture with an injected eVAE and gated-GNN structure. To be specific, our eVAE structure can generate preference embedding from attribute embedding of different modality. To our knowledge, this is the first time that VAE [36] is used for this purpose, while previous researches in recommendation adopt VAE to reconstruct the latent representations with the same modality [37], [38], [39]. Moreover, our gated-GNN structure can differentiate the importance of each dimension of node embeddings. Though gated mechanism has been introduced in the broader research area [40], [41], it is developed for information passing between edges with different types and directions [40] or between nodes at different layers [41], and the detailed design is totally different from that in our model.

2.3 Dealing With Cold Start Issues

Recommender systems often suffer from sparsity and cold start problem. A promising approach to this problem is to leverage side information such as contextual information [4], [42], [43], the user and item relations [5], [44]. Conventional methods mainly exploit side information as regularization in MF objective function [3], [6]. More recently, a number of studies focused on developing various types of neural networks to incorporate side information [7], [8], [9], [14], [15], [16], [21], [34], [44], [45]. A related line of research is at the model level, such as developing inductive methods that can be generalized to users/items unseen during the training, and active learning methods that ask a number of users to interact with items, as well as the meta learning methods that can be adapted to a new task by deriving general knowledge across different learning tasks. We categorize these methods by their ability in dealing with normal and strict cold start issues.

Addressing normal cold start issues. The inductive learning methods [10], [11] can alleviate normal cold start problem. Specifically, IGMC [11] trains inductive matrix completion models without using side information. GraphSage [10] leverages node feature information to generate node embeddings for previously unseen data. Both these methods require that cold start nodes have some interactions or links at the test phase. The optimization based meta learning methods MeLU [12] and MetaHIN [13] utilize a support set such that the meta-learner adapts the global prior to task-specific parameters with respect to the loss on this set in testing stage. The active learning scheme [46], [47], [48], [49], where a number of users are selected for rating or commenting on a new item, can also tackle this problem but with the extra costs or budgets. STAR-GCN [23] actually falls into this active learning category. Finally, by utilizing the heterogeneous side information [13], [14], [15], [16], the meta path based and the multi-view based methods can boost the recommendation performance of their counterpart without deploying meta path or only using single view. However, the JRL [14] and HIRE [15] methods are applicable to sparsity problem only because the users' or items' side information serves as one type of view or as the loss to guide the learning process which should be included in the training set, and the performance of meta path based methods [13], [16] will drop since some of meta paths do not exist any more for cold start users/items.

Addressing strict cold start issues. Both the model based meta-learning methods LWA and NLBA [50] and the gradient based meta-learning method MetaEmb [51] can be applied to the strict cold start scenario. LWA and NLBA focus on the specific twitter cold start where a user has many interactions with previous twitters that can be used to train the meta-learner. The drawback of MetaEmb lies in that it does not explore the information in the neighborhood. Among the methods that integrate auxiliary data, DropoutNet [52] trains DNNs by incorporating content and preference information such that the model can be generalized to missing input. However, the training of the DropoutNet model is still dependent on the existing interactions since its objective is to reconstruct the rating of the user-item pairs. HERS [24] aggregates user and item relations without fully exploiting users' or items' attributes. We notice a recent study LLAE [53] also employs the user's attribute for strict cold start recommendation. However, LLAE does not make use of the graph network, and thus it loses the ability to borrow information from neighbors with the same attributes.

A brief summary. Though several models have been proposed for dealing with the strict cold start issues, almost all these models [50], [51], [52], [53] disregard the potential of GNN, and only a few models [24], [53] utilize the attribute information. Our proposed framework is distinct in that we exploit the users'/items' inherent attribute information via a specially designed GNN architecture to better absorb the information from the neighbors.

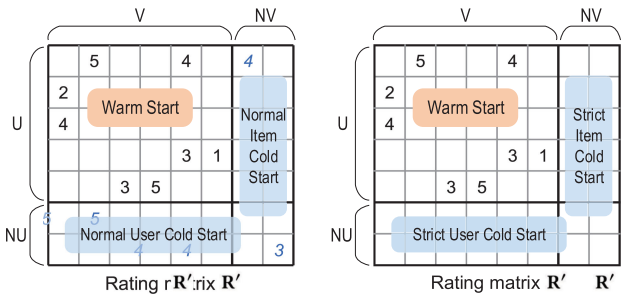
3 Proposed Model

3.1 Problem Definition

Let $U = \{u_1, u_2, \ldots, u_M\}$ be a set of users and $V = \{v_1, v_2, \ldots, v_N\}$ be a set of items, where M and N denote the corresponding cardinalities. In addition, each user or item is associated with a set of attributes from different fields. Each attribute value has a separated encoding, and all attributes are concatenated into a multi-hot attribute encoding $\mathbf{a} \in \mathbb{R}^K$. Below is an example of user attribute encoding $\mathbf{a}_u$.

$$\mathbf{a}_u = \underbrace{[0,1]}_{gender}\ \underbrace{[1,0,0,\ldots,0]}_{age}\ \underbrace{[0,1,0,\ldots,0]}_{occupation},$$

Let $\mathbf{R} \in \mathbb{R}^{M \times N}$ be the user-item interaction matrix, which consists of real-valued ratings for explicit interactions, or binary entries for implicit feedbacks such as click or not. We



(a) warm start and normal cold start scenarios (b) warm start and strict cold start scenarios

Fig. 2. Warm start, normal cold start, and strict cold start scenarios in rating prediction task. ($\mathbf{R}' \in \mathbb{R}^{(M+\Delta M)\times(N+\Delta N)}$).

tackle the recommendation task with explicit interactions, where each $r_{ij} \in \mathbf{R}$ is either a rating score denoting u_i gives a rating to v_j, or 0 denoting the unknown ratings of items that the users have not interacted yet. The goal of classic *warm start rating prediction problem* is to predict the unknown ratings for the users/items that already exist in the original interaction matrix $\mathbf{R}$ and also have interaction history, while *the normal cold start rating prediction problem* refers to predicting the ratings for the users/items unseen during training but having interactions at the test stage, as shown in Fig. 2a.

In this paper, we are interested in *the strict cold start rating prediction problem*. Specifically, let $NU = \{u_{|M|+1}, \ldots, u_{|M|+\Delta|M|}\}$ be a set of new users and $NV = \{v_{|N|+1}, \ldots, v_{|N|+\Delta|N|}\}$ be a set of new items. Note that these new users/items are neither included in the original matrix $\mathbf{R}$ nor they have interactions in the test stage, but they should have attribute information. Fig. 2b shows warm start and the strict cold start scenarios in rating prediction. The goal of strict cold start user/item recommendation is to predict new users' ratings on items or to predict users' ratings on new items.

3.2 An Overview

The architecture of the proposed AGNN model is shown in Fig. 3a. It consists of an input layer, an interaction layer, a gated-GNN layer, and a prediction layer.

We first present an input layer to construct the user (item) attribute graph. We then design an interaction layer to integrate one node's different information into a unified embedding. We also develop an injected eVAE structure to generate the preference embedding for strict cold start nodes in this layer. Next, we propose a gated-GNN layer to aggregate the complicated node embeddings in a neighborhood in the attribute graph. Finally, we add a prediction layer to let the aggregated representations of user and item interact with each other to calculate the rating score.

In general, our model adopts GNN as the main framework to deal with the strict cold start problem using items' and users' attributes, where the graph construction component is used to prepare the input of the GNN since in the strict cold start scenario we cannot use user-item interaction graph, and the eVAE structure is used to approximate the users' preference from their attribute and gated-GNN is for information filtering and aggregating during the process of propagation in GNN. The entire framework and the injected components are all designated for addressing the strict cold start problem.

3.3 Model Architecture

3.3.1 Input Layer

Our model differs from exiting ones in that it is upon the homogeneous attribute graph rather than the bipartite user-item graph. This enables our model to free from sparse interactions and to deal with the strict cold start issues.

We construct the attribute graph using attribute information in this layer. The quality of attribute graph plays a key role in our task. In this paper, we resort to a natural proximity-based way to construct the attribute graph, and we will investigate the impacts of different graph construction methods in our experiments. We first define two types of proximities, i.e., *preference proximity* and *attribute proximity*.

1) The preference proximity measures the historical preference similarity between two nodes. If two users have similar rating record list (or two items have similar rated record list), they will have a high preference proximity. Note we cannot calculate the preference proximity for strict cold start nodes as they do not have historical ratings.

2) The attribute proximity measures the similarity between the attributes of two nodes. If two users have similar user profiles, e.g., gender, occupation (or two items have similar properties, e.g., category), they will have a high attribute proximity.

Both types of proximity can be measured by cosine distance. It is calculated as

$$proximity(\mathbf{w}, \mathbf{v}) = 1 - \frac{\mathbf{w} \cdot \mathbf{v}^T}{\|\mathbf{w}\|\|\mathbf{v}\|}, \tag{1}$$

where $\mathbf{w}$ and $\mathbf{v}$ are two nodes' preference representations or their multi-hot attribute encodings. Two types of proximity are summed after the min-max normalization to get an overall proximity.

After calculating the overall proximity between two nodes, it becomes a natural choice to build a k-NN graph as adopted in [31]. Such a method will keep a fixed number of neighbors once the graph is constructed. It may work well when the graph is constructed on the single type of node attribute like a social graph. However, since our similarity is defined on multiple types of attributes, it is necessary to maintain a diversity of neighborhood to some extent. The rationale is that we wish the age is the dominant factor in determining the neighborhood in some cases while the occupation holds the lead in other cases. To this end, we propose *a dynamic graph construction strategy*. To be specific, for a node u, we add all the nodes which have a top $p\%$ proximity with node u to the candidate pool N_u^C. During each round of the training process, the neighbors of node u are sampled according to the proximity from the candidate pool.

3.3.2 Attribute Interaction Layer

In the constructed attribute graph $\mathcal{A}_u$ and $\mathcal{A}_i$, each node has an attached multi-hot attribute encoding and a unique one-hot representation denoting its identity. Due to the huge number of users and items in the web-scale recommender systems, the dimensionality of nodes' one-hot representation is extremely high. Moreover, the multi-hot attribute representation simply combines multiple types of attributes into one long vector without considering their interactive relations.

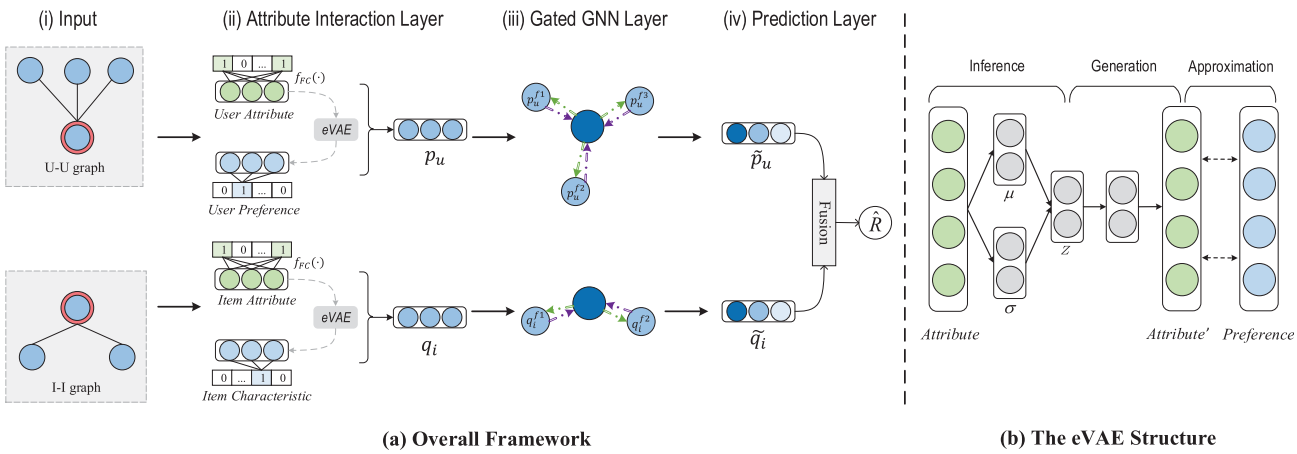


Fig. 3. The left is the framework of our model; and the right is an illustration the eVAE structure.

The goal of interaction layer is to reduce the dimensionality for one-hot identity representation and learn the high-order attribute interactions for multi-hot attribute representation.

To this end, we first set up a lookup table to transform a node's one-hot representation into the low-dimensional dense vector. The lookup layers correspond to two parameter matrices $\mathbf{M} \in \mathbb{R}^{M \times D}$ and $\mathbf{N} \in \mathbb{R}^{N \times D}$. Each entry $\mathbf{m}_u \in \mathbb{R}^D$ and $\mathbf{n}_i \in \mathbb{R}^D$ encodes the user u's preference and the item i's property, respectively. Note that $\mathbf{m}_u$ and $\mathbf{n}_i$ for strict cold start nodes are meaningless, since no interaction is observed to train their preference embedding. We will discuss the solution to this problem later.

Inspired by [43], we capture the high-order attribute interactions with a Bi-Interactive pooling operation, in addition to the linear combination operation. To be specific, let $\mathbf{v}_i$ and $\mathbf{v}_j$ be the embedding vector for the ith and jth type of attribute in the multi-hot attribute encoding $\mathbf{a} \in \mathbb{R}^K$, respectively, the Bi-Interactive and linear combination operation are defined as

$$f_{BI}(\mathbf{a}) = \sum_{i=1}^{K} \sum_{j=i+1}^{K} a_i \mathbf{v}_i \odot a_j \mathbf{v}_j, \quad f_L(\mathbf{a}) = \sum_{i=1}^{K} a_i \mathbf{v}_i, \tag{2}$$

where $\odot$ denotes the element-wise product.

Finally, a fully connected layer is added on both the second-order interaction and linear combination to learn the high-order feature interactions

$$f_{FC}(\mathbf{a}) = LeakyReLU(\mathbf{W}_{fc}^{(1)} f_{BI}(\mathbf{a}) + \mathbf{W}_{fc}^{(0)} f_L(\mathbf{a}) + \mathbf{b}_{fc}), \tag{3}$$

where $\mathbf{W}_{fc}, \mathbf{b}_{fc}, LeakyReLU$ are weight matrix, bias vector, and activation function, respectively.

We can then get the attribute embedding $\mathbf{x}_u$ and $\mathbf{y}_i$ for a user u and for an item i by feeding their respective attribute encoding $\mathbf{a}_u$ and $\mathbf{a}_i$ into the f_{FC} function, i.e.,

$$\mathbf{x}_u = f_{FC}(\mathbf{a}_u), \quad \mathbf{y}_i = f_{FC}(\mathbf{a}_i). \tag{4}$$

Next, we fuse the preference embedding and attribute embedding into the node embedding such that each node contains both historical preferences and its own attributes.

$$\mathbf{p}_u = \mathbf{W}_u[\mathbf{m}_u;\ \mathbf{x}_u] + \mathbf{b}_u, \quad \mathbf{q}_i = \mathbf{W}_i[\mathbf{n}_i;\ \mathbf{y}_i] + \mathbf{b}_i, \tag{5}$$

where $[;]$ denotes vector concatenation operation, $\mathbf{W}_{u(i)}$, $\mathbf{b}_{u(i)}$ are weight matrix and bias vector. For strict cold start nodes without any interactions, we will generate preference embeddings for them in this layer. We will detail our solution to this strict cold start problem in a separate subsection later.

3.3.3 Injected eVAE Structure in Attribute Interaction Layer

The strict cold start problem is caused by the lack of any interactions/links for the new nodes in the test phase. We view this as a missing preference problem. Unlike the methods in [23], [52] which reconstruct the same node embedding with the dropout or mask technique, we aim to reconstruct the node's missing preference embedding from its attribute embedding.

Basically, one specific type of users might be interested in the similar items, and vice versa. For example, animation is the mainstream entertainment among teenage children (the users have the similar age attribute). This indicates that the attribute and preference embeddings are not only close to each other in the latent space but also have the similar distribution. Hence we tackle the missing preference problem by employing the variational auto-encoder structure to reconstruct the preference from the attribute distribution.

Our proposed extended VAE structure is shown in Fig. 3b, which contains three parts: inference, generation, and approximation. The first two parts are the standard VAE and the third one is our extension. Take the strict cold start user node u as an example. In the generation part, u is given a latent variables $\mathbf{z}_u$. The reconstructed embedding $\mathbf{x}'_u$ is generated from its latent variables $\mathbf{z}_u$ through generation network as MLP parameterized by θ [36]

$$\mathbf{x}'_u \sim p_\theta(\mathbf{x}'_u | \mathbf{z}_u). \tag{6}$$

In the inference part, variational inference approximates the true intractable posterior of the latent variable $\mathbf{z}_u$ by introducing an inference network parameterized by ϕ [36]

$$q_\phi(\mathbf{z}_u) = \mathcal{N}(\boldsymbol{\mu}_u,\ diag(\boldsymbol{\sigma}_u^2)), \tag{7}$$

The objective of variational inference is to optimize the free variational parameters so that the KL-divergence $KL(q(\mathbf{z}_u) \parallel p(\mathbf{z}_u|\mathbf{x}_u))$ is minimized. With the reparameterization trick [36], we sample $\boldsymbol{\epsilon} \sim \mathcal{N}(0, \mathbf{I})$ and reparameterize $\mathbf{z}_u =$

$\mu_\phi(\mathbf{x}_u) + \epsilon \odot \sigma_\phi(\mathbf{x}_u)$. In this case, the gradient towards ϕ can be back-propagated through the sampled $\mathbf{z}_u$.

In the approximation part, we constrain the reconstructed embedding $\mathbf{x}'_u$ to be close to the preference embedding $\mathbf{m}_u$. This is practical because the system should have collected a certain amount of interactions in reality. During the training phase, the nodes with historical ratings actually have the preference embeddings. Such information can be explored to improve the VAE. Hence we require the reconstructed embedding $\mathbf{x}'_u$ to be similar with both the preference embedding (by the constraint) and the original attribute distribution (by the standard VAE). To summarize, the reconstruction loss function in our proposed eVAE is defined as follows.

$$\begin{aligned}\mathcal{L}_{recon} = &- KL(q_\phi(\mathbf{z}_u|\mathbf{x}_u)\|p(\mathbf{z}_u)) \\ &+ \mathbb{E}_{q_\phi(\mathbf{z}_u|\mathbf{x}_u)}[\log p_\theta(\mathbf{x}'_u|\mathbf{z}_u)] + \|\mathbf{x}'_u - \mathbf{m}_u\|_2,\end{aligned} \tag{8}$$

where the first two terms are same as those in standard VAE, and the last one is our extension for the approximation part. The strict cold start item i's preference embedding $\mathbf{n}_i$ can be generated similarly from its attribute embedding $\mathbf{y}'_i$, and thus we have $\mathbf{m}_u \sim \mathbf{x}'_u$ and $\mathbf{n}_i \sim \mathbf{y}'_i$.

3.3.4 Gated-GNN Layer

Intuitively, different neighbors have different relations to a node. Furthermore, one neighbor usually has multiple attributes. For example, in a social network, a user's neighborhood may consist of classmates, family members, colleagues, and so on, and each neighbor may have several attributes such as age, gender, and occupation. Since all these attributes (along with the preferences) are now encoded in the node's embedding, it is necessary to pay different attentions to different dimensions of the neighbor node's embedding. However, existing GCN [54] or GAT [55] structures cannot do this because they are at the coarse granularity. GCN treats all neighbors equally and GAT differentiates the importance of neighbors at the node level. To solve this problem, we design a gated-GNN structure to aggregate the fine-grained neighbor information.

Our proposed gated-GNN structure is shown in Fig. 4. It contains an *aggregate gate* (denoted as $\mathbf{a}_{gate}$) and a *filter gate* (denoted as $\mathbf{f}_{gate}$). In order to better capture the homophily phenomenon in networks, the $\mathbf{a}_{gate}$ controls what information should be aggregated from neighbors to the target node, while $\mathbf{f}_{gate}$ controls what information in the target node should be filtered out if it is not consistent with that in the neighbors. These two gates work as follows.

Given a user node u, its node embedding $\mathbf{p}_u$, its neighbor set N_u, and the node embedding $\mathbf{p}_u^{f_i}$ for the ith neighbor f_i in N_u, we first apply $\mathbf{a}_{gate}$ to the neighbors to obtain the aggregated representation $\mathbf{p}_u^{u\leftarrow N_u}$ by selectively passing the neighbor embeddings to the target node u.

$$\mathbf{a}_{gate}^{f_i} = \sigma(\mathbf{W}_a[\mathbf{p}_u;\ \mathbf{p}_u^{f_i}] + \mathbf{b}_a), \tag{9}$$

$$\mathbf{p}_u^{u\leftarrow N_u} = \frac{1}{|N_u|}\sum_{i=1}^{|N_u|}(\mathbf{p}_u^{f_i} \odot \mathbf{a}_{gate}^{f_i}), \tag{10}$$

where $\mathbf{W}_a, \mathbf{b}_a, \sigma$ are weight matrix, bias vector, and the sigmoid activation function.

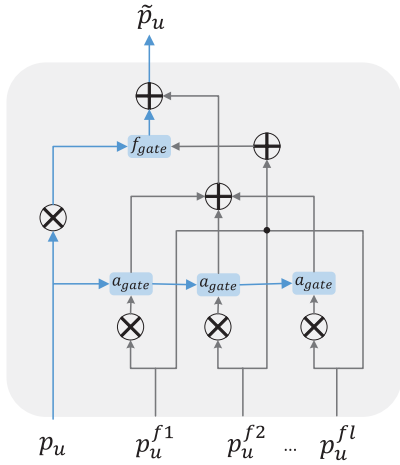


Fig. 4. The gated-GNN structure.

Homophily is a fundamental characteristic in social networks. Known as "birds of a feather flock together" [56], homophily has been observed in many online social networks [57], [58], [59]. In this section, we propose to reflect the homophily phenomenon by applying the filter gate $\mathbf{f}_{gate}$ to the target node u, such that the node u's information which is inconsistent with the averaged representations of the neighbors will be filtered out. More formally,

$$\mathbf{f}_{gate} = \sigma\left(\mathbf{W}_f[\mathbf{p}_u;\ \frac{1}{|N_u|}\sum_{i=1}^{|N_u|}\mathbf{p}_u^{f_i}] + \mathbf{b}_f\right), \tag{11}$$

$$\mathbf{p}_u^- = \mathbf{p}_u \odot (\mathbf{1} - \mathbf{f}_{gate}), \tag{12}$$

where $\mathbf{p}_u^-$ is the node u's remaining representation after the filtering operation.

Combining the aggregated representation $\mathbf{p}_u^{u\leftarrow N_u}$ and the remaining representation $\mathbf{p}_u^-$ together, we can get the user node u's final embedding $\tilde{\mathbf{p}}_u$ as follows:

$$\tilde{\mathbf{p}}_u = LeakyReLU(\mathbf{p}_u^- + \mathbf{p}_u^{u\leftarrow N_u}). \tag{13}$$

The item i's final embedding, denoted as $\tilde{\mathbf{q}}_i$, can be obtained from the item attribute graph in a similar way.

3.3.5 Prediction Layer

Given a user u's final representation $\tilde{\mathbf{p}}_u$ and an item i's final representation $\tilde{\mathbf{q}}_i$ after the gated-GNN layer, we model the predicted rating of the user u to the item i as

$$\hat{R}_{u,i} = MLP([\tilde{\mathbf{p}}_u;\ \tilde{\mathbf{q}}_i]) + \tilde{\mathbf{p}}_u\tilde{\mathbf{q}}_i^T + b_u + b_i + \mu, \tag{14}$$

where the MLP function is the multi-layer perception implemented with one hidden layer, and b_u, b_i, and μ denotes user bias, item bias, and global bias, respectively. In Eq. (14), the second term is inner product interaction function [2], and we add the first term to capture the complicated nonlinear interaction between the user and the item.

3.4 Loss

The overall loss function for training is defined as

$$\mathcal{L} = \mathcal{L}_{pred} + \lambda\ \mathcal{L}_{recon}, \tag{15}$$

where $\mathcal{L}_{pred}$ is the task-specific rating prediction loss, $\mathcal{L}_{recon}$ is the reconstruction loss defined in Eq. (8), and λ is a constant weighting factor.

For the rating prediction loss, we employ the square loss as the objective function

$$\mathcal{L}_{pred} = \sum\nolimits_{u,i \in \mathcal{T}} (\hat{R}_{u,i} - R_{u,i})^2, \tag{16}$$

where $\mathcal{T}$ denotes the set of instances for training, i.e., $\mathcal{T} = \{(u, i, r_{u,i}, \mathbf{a}_u, \mathbf{a}_i)\}$, $R_{u,i}$ is ground truth rating in the training set $\mathcal{T}$, and $\hat{R}_{u,i}$ is the predicted rating.

4 Experiments

In this section, we evaluate our model by comparing it with the state-of-the-art baselines. We begin with the experimental setup, and then analyze the experimental results.

4.1 Experimental Setup

4.1.1 Datasets

We use three real-world datasets to evaluate our proposed AGNN model in this section.

MovieLens[1] is a widely used movie rating dataset. We employ the ML-100K and ML-1M versions. We take categories, stars, directors, writers, and countries as movie features, and gender, age, and occupations as user features. Since the original datasets do not provide sufficient item attributes, we extend them by crawling stars, directors, writers, and countries from IMDb[2] according to the movie title and release year. We will release these two extended datasets to the public.[3]

Yelp[4] is adopted from the 2017 edition of the Yelp challenge. Wherein, the local businesses like restaurants and hotels are viewed as the items. Since the raw Yelp data is extremely large and sparse, we preprocess this dataset by removing nodes with less than 20 ratings following the settings in [60], [61]. Note that such a pre-processing does not contradict to the target of strict cold start evaluation, as we will remove the interactions of strict cold start nodes from training set. That is to say, these nodes have never been seen before the test phase, just like the newly-released movies, nor they have any interactions.

We take categories, located states, and located cities as item features. Due to the lack of user profile information on Yelp, we use social links as user-user graph and also as attributes for users. Specifically, we take each row of the social matrix as the user's attribute encoding, where each dimension of the social vector represents an attribute value. The statistics of the datasets are shown in Table 1.

TABLE 1
Statistics of the Datasets

Datasets	#Users	#Items	#Ratings	Sparsity
ML-100K	943	1,682	100,000	93.70%
ML-1M	6,040	3,883	1,000,209	95.74%
Yelp	23,549	17,139	941,742	99.77%

4.1.2 Baselines

We choose the following twelve comparative methods as our baselines. All these methods leverage side information or attributes. Moreover, many of them are specially designed for addressing sparsity or normal/strict cold start problem. Finally, these methods cover the state-of-the-art models with conventional techniques and those with graph neural networks.

- *NFM* [43] combines factorization machines with neural networks for rating prediction under sparse settings. It projects the user and item features into a dense vector, and then uses a multi-layered neural network to model feature interactions.
- *DiffNet* [20] fuses user and item features with the free node embedding. It contains a GCN-alike layer-wise diffusion procedure to model dynamic diffusion on user-user social graph for recommendation.
- *DANSER* [21] is a GAT-based method to incorporate multifaceted social effects. It employs users' social relations as user-user graph and builds item-item graphs using a similarity coefficient between items clicked by a same user.
- *sRMGCNN* [31] employs multiple GCNs on user-user, item-item, and user-item graphs for matrix completion. It builds user-user and item-item graphs using a k-nearest neighbor strategy on the attributes.
- *GC-MC* [17] includes side information in the form of user and item feature vectors which are later combined into the node representation. It applies a GCN framework to user-item graph for matrix completion.
- *STAR-GCN* [23] adopts the stacked GCN on user-item interaction graph. It concatenates the feature embedding and free embedding to represent a node. It alleviates the normal cold start problem by applying mask technique to node embeddings.
- *MetaHIN* [13] combines meta-learning and meta-path methods for normal cold start recommendation on heterogeneous information networks.
- *IGMC* [11] constructs enclosing user-item subgraphs and learns local graph patterns via relational graph convolutional operator for inductive matrix completion. It is developed to address normal cold start issues.
- *DropoutNet* [52] incorporates content and preference information with a DNN model. It presents a dropout technique to tackle the strict cold start issues.
- *LLAE* [53] develops a linear low-rank denoising auto-encoder to reconstruct a user's behaviors from this user's attributes for strict cold start recommendation.
- *HERS* [24] employs users' social relations as user-user graph and builds item-item graph based on the common tags between two items. It aggregates user and item relations for addressing strict cold start issues.
- *MetaEmb* [51] is a meta-learning approach to address strict cold start problem. It trains an embedding generator for new IDs through gradient-based meta-learning.

We partition the baselines into three parts. The first part (from NFM to GC-MC) includes the methods designed for

1. https://grouplens.org/datasets/movielens/
2. https://www.imdb.com
3. Both the code and the data are available at: https://github.com/NLPWM-WHU/AGNN
4. https://www.yelp.com/dataset_challenge

warm start. The second part (from STAR-GCN to IGMC) includes the methods developed for normal cold start. The third part (from DropoutNet to MetaEmb) includes the methods that can deal with the strict cold start problem.

4.1.3 Evaluation Metrics

We use Rooted Mean Square Error (RMSE) and Mean Absolute Error (MAE) as the evaluation metrics, which are both widely used for rating prediction tasks [21], [35], [43], [62], [63]. The definition of RMSE and MAE are as follows:

$$RMSE = \sqrt{\frac{1}{N}\sum_{u,i}(\hat{R}_{u,i} - R_{u,i})^2}, \tag{17}$$

$$MAE = \frac{1}{N}\sum_{u,i}|\hat{R}_{u,i} - R_{u,i}|, \tag{18}$$

where N is the total number of data points being tested, $\hat{R}_{u,i}$ is the predicted rating and $R_{u,i}$ is the ground truth.

4.1.4 Settings

For strict cold start, we randomly choose 20 percent items (or users) along with their interactions as test set, and the remaining interactions as training set. For warm start, we randomly choose 20 percent user-item interactions as test set and the remaining 80 percent as training set. The difference is that for strict cold start nodes, their interactions are totally removed from testing.

For our AGNN, we set the batch size = 128, slop of LeakyReLU = 0.01, the initial learning rate = 0.0005, and we use Adam [64] as optimizer to self-adapt the learning rate. We also set the hyper-parameters as follows: the embedding dimension D = 40, the threshold p = 5 for the candidate set in graph construction, and the weighting factor λ = 1. We will investigate the impacts of these hyper-parameters later.

For the baselines, we follow the same hyperparameter settings if they are reported by the authors and we fine tune the hyper-parameters if they are not reported. The baselines designed for top-N recommendation are revised to optimize RMSE scores. We implement sRMGCNN with its public source code, but it cannot scale to large dataset like Yelp.

Please note that all baselines use the same attribute information as our model. Consequently, many baselines without specific design to tackle the strict cold start problem can still be applied to the strict scenario. The attributes are used to be integrated into the node embeddings, to model the feature interactions, or to construct graphs. Since there is no social relationship on ML-100K and ML-1M, we use the attribute features to initialize the user embeddings for DANSER and HERS. Such initialization is also done for item embeddings of DANSER as it is not designed for incorporating attributes. Besides, for HERS, we use the common attributes instead of tags to build item-item graph due to the lack of tags in our datasets.

As for graph construction for baselines, there is no such step for NFM, DropoutNet, LLAE, and MetaEmb since they are not graph based method, and we use the users' and items' attribute for these three baselines as their original paper does. Meanwhile, DiffNet utilizes user-user graph, while GC-MC, IGMC, and STAR-GCN employ user-item graph, and MetaHIN utilizes heterogeneous information network. Moreover, DANSER, HERS, and sRMGCNN utilize both user-user graph and item-item graph.

When constructing user-user graph, we follow the settings in their original paper. On Yelp dataset, DiffNet, DANSER, and HERS use the social links to construct user-user graph. sRMGCNN uses the users' attribute to construct user-user graph, and such a construction method is applied to DiffNet, DANSER, and HERS on Movielens, since there is no social links on this dataset.

We also follow the original methods in the corresponding paper to construct item-item graph. Specifically, DANSER constructs item-item relation according to the similarity coefficient of the number of co-clicked users. HERS and sRMGCNN construct item-item graph by connecting an item with its K-nearest (K=10) neighbors in the space of item attributes or tags.

The user-item graph can be directly constructed from the historical interactions. Note that we do not add newly rated edges to the strict cold start nodes in the testing phase of STAR-GCN, for a fair comparison with all other methods and also for simulating the real world strict cold start scenario.

4.2 Comparison With Baselines

Our model is mainly developed for strict cold start recommendation. However, for a fair comparison with the baselines, we examine the model performance in both the strict cold and warm start scenario in this subsection. The results of AGNN and the baselines on three datasets are reported in Table 2.

First, it is clear that our AGNN outperforms all baselines in the strict cold start scenarios. Its improvements over the strongest baselines are statistically significant at the 0.01 level. The results verify the superiority of our proposed architecture by exploring attribute graph for strict cold start recommendation. Moreover, in the warm start scenario, our AGNN also yields the best results on Yelp and ML-1M, and is the second best on ML-100K with a performance slightly inferior to STAR-GCN (the difference is not statistically significant).

Second, among five baselines that are mainly designed for warm start, NFM performs well in some cases due to its ability to learn high-order feature interactions. DiffNet and DANSER performs better in many cases of strict cold start users since they both utilize social graph for recommendation. Further more, DiffNet combines user embedding with preference and attribute information. DANSER constructs item-item graph according to the number of co-purchased items. This results in its poor performance in strict item cold start. sRMGCNN is poor since it uses attributes to construct user-user or item-item graph without including them into the convolution operation. Moreover, it cannot handle large dataset like Yelp as its convolution is defined on Chebyshev expansion. The performance of GC-MC is also limited as it incorporates content information after the convolution layer.

Third, among three baselines for normal cold start, STAR-GCN utilizes graph convolutional network on the user-item graph. It can get the second best performance in

TABLE 2
Performance Comparison on Three Datasets for RMSE and MAE, Respectively

RMSE	ML-100K			ML-1M			Yelp		
	ICS	UCS	WS	ICS	UCS	WS	ICS	UCS	WS
NFM	1.0416	1.0399	0.9533	1.0403	0.9885	0.9130	1.1231	1.1045	1.0620
DiffNet	1.0418	1.0379	0.9221	1.0363	0.9809	0.8622	1.1072	1.1267	1.0444
DANSER	1.1190	1.0490	0.9823	1.1246	0.9808	0.9797	1.1302	1.0927	1.0525
sRMGCNN	1.1532	1.0479	0.9376	1.2978	1.2118	1.1770	-	-	-
GC-MC	1.0392	1.0444	0.9106	1.0526	0.9922	0.8656	1.1229	1.1020	1.0254
STAR-GCN	1.0376	1.0428	**0.9049**	1.0456	0.9878	0.8573	1.1173	1.0988	1.0232
MetaHIN	1.0712	1.1328	0.9955	1.1162	1.0036	0.9870	1.1184	1.1031	1.0252
IGMC	1.1053	1.0589	0.9318	1.1353	1.0453	0.8883	1.0965	1.0994	1.0512
DropoutNet	1.0844	1.0654	0.9428	1.1008	1.0396	0.9254	1.1891	1.1724	1.1524
LLAE	3.3700	3.2653	3.1786	3.3169	3.3223	3.3384	3.8057	3.8416	3.8008
HERS	1.1027	1.0493	0.9344	1.1219	0.9823	0.9137	1.1977	1.1596	1.0240
MetaEmb	1.0432	1.0408	0.9427	1.0290	0.9863	0.8648	1.0869	1.0928	1.0265
AGNN	**1.0187***	**1.0208***	0.9078	**1.0091***	**0.9743***	**0.8533**†	**1.0749***	**1.0657***	**1.0106***
Improvement	1.82%	1.65%	-0.32%	2.62%	0.67%	0.47%	1.10%	2.47%	1.23%

MAE	ML-100K			ML-1M			Yelp		
	ICS	UCS	WS	ICS	UCS	WS	ICS	UCS	WS
NFM	0.8525	0.8404	0.7565	0.8478	0.7934	0.7221	0.9077	0.8832	0.8372
DiffNet	0.8476	0.8380	0.7250	0.8349	0.7884	0.6760	0.9012	0.9144	0.8241
DANSER	0.9414	0.8542	0.7830	0.9434	0.7863	0.7847	0.9095	0.8818	0.8319
sRMGCNN	0.9434	0.8411	0.7458	1.0685	1.0012	0.9790	-	-	-
GC-MC	0.8470	0.8647	0.7150	0.8615	0.8030	0.6847	0.9111	0.9235	0.8205
STAR-GCN	0.8440	0.8596	**0.7116**	0.8494	0.7975	0.6705	0.9088	0.9162	0.8201
MetaHIN	0.8946	0.9309	0.8321	0.9266	0.8348	0.8218	0.9150	0.9196	0.8222
IGMC	0.9299	0.8495	0.7298	0.9256	0.8615	0.7036	0.8983	0.8844	0.8403
DropoutNet	0.8722	0.8571	0.7399	0.8866	0.8398	0.7296	0.9628	0.9624	0.9254
LLAE	3.1749	3.0701	2.9797	3.1047	3.1453	3.1280	3.6300	3.6702	3.6237
HERS	0.8745	0.8572	0.7360	0.8923	0.7878	0.7236	0.9691	0.9289	0.8056
MetaEmb	0.8457	0.8504	0.7495	0.8330	0.7971	0.6842	0.8929	0.8823	0.8102
AGNN	**0.8171***	**0.8198***	0.7138	**0.8093***	**0.7794***	**0.6677**†	**0.8715***	**0.8586***	**0.7945***
Improvement	3.19%	2.17%	-0.31%	2.85%	0.89%	0.42%	2.40%	2.63%	3.12%

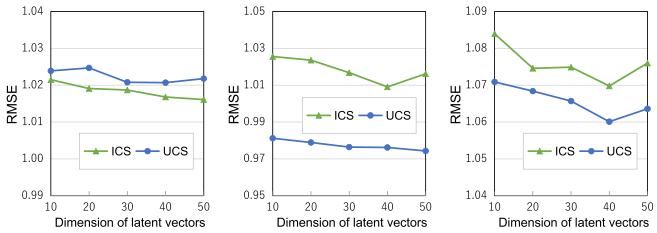
*The best performance among all is in bold while the second best one is marked with an underline. ICS, UCS, and WS are the abbreviations for the strict cold start item, strict cold start user, and warm start, respectively. The last row indicates the percentage of improvements gained by our proposed AGNN model compared with the best baseline. * and † denote the statistical significance between each pair of our AGNN and the best baseline at $p < 0.01$ and $p < 0.05$ level, respectively.*

some cases as it integrates the content information into node embedding and also because it avoids the leakage issue when convoluting on user-item graph. The performance of MetaHIN is ordinary since it requires a support set for adapting the prior knowledge. In this study, we remove the interactions from support set to query set during testing to ensure that the links of new nodes is unseen, and thus its performance drops. This confirms that the optimization based meta learning is not suitable for the strict cold start scenario. IGMC is not good due to the fact that it still requires some interactions to construct subgraph for the target user-item pairs, whereas the strict cold start user/item does not have any links at all.

Finally, among four baselines for strict cold start, LLAE performs extremely poorly because it aims to fit a user's entire rating vector over all items with the vector reconstructed from the user's attributes, whereas the target of rating prediction is to optimize the rating score for each user-item pair. Note that LLAE only optimizes the reconstruction function while both the rating prediction loss and reconstruction loss are included in our objective function. To exclude the impacts caused by the loss function, we will further conduct the component replacement study for LLAE to examine its ability in addressing strict cold start issues in Section 5.1. DropoutNet is not good since it requires the content information to approximate the results of matrix factorization, and its performance is dependent on the pre-trained preference embeddings. HERS is the best baseline on Yelp in warm-start scenario since this dataset is very sparse and HERS uses both item and user relations to alleviate the sparsity problem. MetaEmb performs relatively well in some cases. It learns the ID embedding generator by meta-learning which can be adapted to different cold start scenarios. However, it is still inferior to our AGNN model because AGNN can exploit the attribute information with the powerful graph neural networks with injected eVAE and gated-GNN structures.

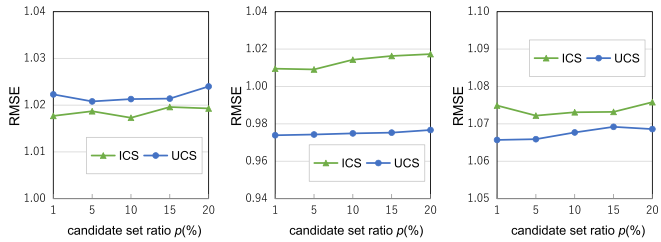
4.3 Parameter Study

In this subsection, we analyze the impacts of three hyper-parameters in AGNN, including: the latent vector dimension D, the weighting factor λ, and the threshold p for the candidate set in graph construction. Figs. 5, 6, and 7 show the results by varying the dimensionality in the set of $\{10, 20, 30, 40, 50\}$, tuning the weighting factor of reconstruction amongst $\{0, 0.01, 0.1, 1, 10\}$, and varying the threshold in the candidate



(a) ML-100K (b) ML-1M (c) Yelp

Fig. 5. Impacts of latent vector dimension D.



(a) ML-100K (b) ML-1M (c) Yelp

Fig. 7. Impacts of neighbor candidate set threshold p.

set of $\{1, 5, 10, 15, 20\}$ on three datasets, respectively. Due to the space limitation, we only report the RMSE results. Note that the smallest score is the best for RMSE.

We first fix λ to 1 and p to 5 and vary D. We can see from Figs. 5a and 5b that, with the increase of the dimension of latent vector, the performance shows a general upward trend on two MovieLens datasets. This indicates that a larger dimension could capture more latent factors of users, items and their respective attributes, which may bring better representation capability. However, as is shown in Fig. 5c on Yelp, when the dimension of the latent vector exceeds a certain value, e.g., 40, it will cause overfitting and brings about the decrease of performance.

We then fix D to 40 and p to 5 and vary λ. It can be seen that the weighting factor λ is critical in maintaining a balance between the rating prediction loss and the reconstruction loss of the eVAE component. As shown in Fig. 6, we find that the optimal value of λ is around 1 on different datasets. When λ is too small, the reconstruction loss becomes trivial in the model. That is to say, AGNN is unable to capture the relationship between the preference and attribute information, which leads to the poor performance. On the other hand, when λ is too large, e.g., 10, the model has a bias target towards the reconstruction loss and thus destroys the learning process of the rating prediction task.

We finally fix D to 40 and λ to 1 and vary p. We observe from Fig. 7 that the candidate set threshold p does not have big impacts on the performance. Indeed, the curves are rather steady with the changing values of p. The reason might be that when sampling from the candidate set, the top-ranked samples always have higher probability to be selected though the larger candidate set can provide more samples. In most cases, $p = 5$ can generate good enough results. Hence we choose this as the setting for p on all datasets.

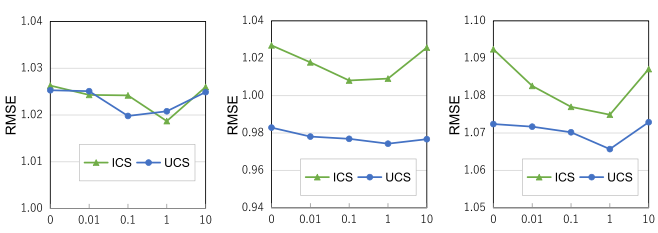
4.4 Performance w.r.t. Strict Cold Start Ratio

In the strict cold start scenario, a higher ratio of strict cold start nodes indicates that fewer nodes and user-item interactions can be utilized for training. This subsection compares our model with three strongest baselines, i.e., DiffNet, STAR-GCN, and MetaEmb, to examine the impacts of the ratio of strict cold start nodes.

We randomly choose 10, 30 and 50 percent nodes along with their interactions as test set, and the remaining nodes and their interactions as training set. Figs. 8a, 8b and 8c show the results in strict item and user cold start scenario on ML-100K, ML-1M, and Yelp, respectively. From the results, we have the following findings.

AGNN consistently outperforms DiffNet, STAR-GCN, and MetaEmb in different proportions of strict cold start nodes. This proves that the performance of our AGNN is stable among various strict cold start settings. When increasing the ratio of strict cold start nodes, the performance of DiffNet and STAR-GCN degrades more quickly than that of AGNN. The main reason is that DiffNet and STAR-GCN are dependent on the user-item interactions, and thus are sensitive to the number of strict cold start nodes which is proportional to the number of edges in the user-item graph.

On the other hand, MetaEmb is better than DiffNet and STAR-GCN with larger ratio of strict cold start nodes yet still worse than our AGNN. The embedding generator in MetaEmb relies on the previously seen users/items, and the embedding of new users/items in MetaEmb is produced by feeding their contents and attributes without exploiting the relations with neighbors in the attribute graph. Consequently, the high ratio of the strict cold start users/items severely harms the performance of MetaEmb. In contrast, AGNN focuses on modeling the attribute graph and is less affected by the limited number of training nodes and their interactions.



(a) ML-100K (b) ML-1M (c) Yelp

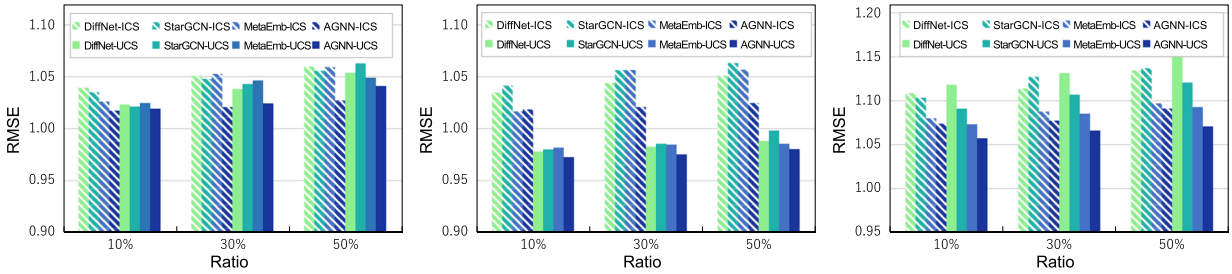
Fig. 6. Impacts of weighting factor λ.

5 Analysis

This section illustrates the detailed analysis on the components and computational cost of our model.

5.1 Ablation and Replacement Study

In order to verify the effectiveness of our model, we perform two types of studies. One is the ablation study which removes one component from our AGNN. This is used to show the effects of different components like eVAE and gated-GNN in our model. The other is the replacement study which adapts the key technique in baselines such as mask or dropout to replace the one in our AGNN. This is to examine the impacts of the specific techniques in baselines while keeping all other conditions exactly same as ours. The results for ablation and replacement study are shown in Tables 3 and 4, respectively.



(a) ML-100K (b) ML-1M (c) Yelp

Fig. 8. The performance comparison with varying percentage of testing data in strict cold start scenario on different datasets.

5.1.1 Analysis on Ablation Study

In ablation study, we first investigate the impacts of preference proximity (PP) and attribute proximity (AP) in AGNN_{PP} and AGNN_{AP} by using only one proximity. It can be seen from Table 3 that AP is more effective in measuring the closeness between two nodes. This is natural since attribute is the inherent property of a user/item, while preference is reflected by the user's behavior which might be indirect and incomplete.

We then examine the impacts of gated-GNN structures. Specifically, $\text{AGNN}_{-\text{gGNN}}$ removes the entire gated-GNN, while $\text{AGNN}_{-\text{agate}}$ and $\text{AGNN}_{-\text{fgate}}$ removes aggregate gate and filter gate from the original gated-GNN structure, respectively. The removal of gated-GNN incurs the worst results since it contains the complete functionality of the gate mechanism. Furthermore, the removal of two fine-grained gates results in the decrease of performance on different datasets. This verifies the significance of key design in our gated-GNN, where $\mathbf{a}_{gate}$ regulates how the neighbor information spreads out and $\mathbf{f}_{gate}$ controls how the homogeneous information is reserved. Finally, the downward trend of $\text{AGNN}_{-\text{agate}}$ is more obvious than $\text{AGNN}_{-\text{fgate}}$. This is consistent with our assumption that $agate$ enhances the ability of information propagation and thus it well leverages the power of GNN architectures.

We finally show the impacts of eVAE structure in the last two rows in Table 3. $\text{AGNN}_{-\text{eVAE}}$ removes the entire eVAE structure from AGNN while AGNN_{VAE} removes the approximation part in eVAE and degrades to a standard VAE. It can be seen that the removal of eVAE greatly harms the performance. Especially on the sparse Yelp dataset, $\text{AGNN}_{-\text{eVAE}}$ is the worst in the strict item cold start scenario. Moreover, the performance of AGNN_{VAE} also decreases a lot compared to standard AGNN. This reveals that the approximation part is essential to our eVAE structure. Without the the approximation, the normal VAE is unable to generate the mapping from attribute to preference and thus loses the ability to well utilize the attribute information.

Overall, all the above variants perform worse than the complete AGNN model, showing that each component is essential to the proposed framework.

5.1.2 Analysis on Replacement Study

In replacement study, we first show the influence of the graph construction methods in RMGCNN [31] and DANSER [21]. Specifically, AGNN_{knn} constructs user-user and item-item graph by choosing 10-nearest neighbors in the user and item attribute space [31]. AGNN_{cop} constructs item-item graph according to the number of co-purchased items [21]. The user-user graph is constructed in a similar way if social links are not available. Since there is no common tags between items in our datasets, the graph construction method in HERS [24] which is now based on the common attributes between items, becomes identical to that in RMGCNN. Hence we omit the replacement experiment for HERS.

As shown in Table 4, the performance of AGNN_{cop} declines dramatically on Movielens since there will be no neighbor for strict cold start items using only the co-purchase information. Its performance for strict cold start users does not change much on Yelp because social links already form the user-user graph and serve as the users' attribute. Furthermore, the superior performance of AGNN over AGNN_{knn} demonstrates that both attribute and preference information are useful for graph construction. Finally,

TABLE 3
Results for Ablation Study

	ML-100K				ML-1M				Yelp			
	ICS		UCS		ICS		UCS		ICS		UCS	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
AGNN	**1.0187**	**0.8171**	**1.0208**	**0.8198**	**1.0091**	**0.8093**	**0.9743**	**0.7794**	**1.0749**	**0.8715**	**1.0657**	**0.8586**
AGNN_{PP}	1.0667	0.8722	1.0322	0.8406	1.0310	0.8391	0.9877	0.8014	1.0842	0.8878	1.0770	0.8702
AGNN_{AP}	1.0271	0.8288	1.0250	0.8229	1.0156	0.8200	0.9770	0.7849	1.0768	0.8729	1.0695	0.8628
$\text{AGNN}_{-\text{gGNN}}$	1.0357	0.8410	1.0328	0.8305	1.0193	0.8259	0.9868	0.8013	1.0785	0.8803	1.0869	0.8866
$\text{AGNN}_{-\text{agate}}$	1.0284	0.8275	1.0284	0.8298	1.0182	0.8187	0.9788	0.7884	1.0766	0.8668	1.0702	0.8551
$\text{AGNN}_{-\text{fgate}}$	1.0230	0.8262	1.0264	0.8196	1.0175	0.8220	0.9760	0.7819	1.0754	0.8696	1.0680	0.8520
$\text{AGNN}_{-\text{eVAE}}$	1.0263	0.8302	1.0253	0.8306	1.0269	0.8320	0.9829	0.7924	1.0924	0.8894	1.0724	0.8707
AGNN_{VAE}	1.0252	0.8241	1.0240	0.8250	1.0238	0.8310	0.9839	0.7972	1.0936	0.8873	1.0729	0.8621

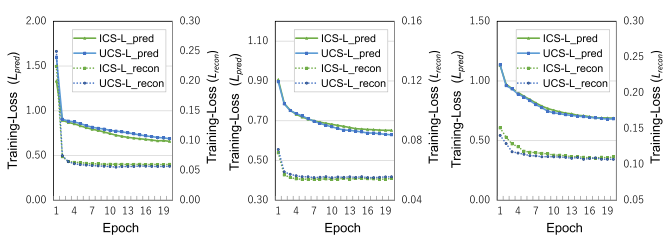
TABLE 4
Results for Replacement Study

	ML-100K				ML-1M				Yelp			
	ICS		UCS		ICS		UCS		ICS		UCS	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
AGNN	**1.0187**	**0.8171**	**1.0208**	**0.8198**	**1.0091**	**0.8093**	**0.9743**	**0.7794**	**1.0749**	**0.8715**	**1.0657**	**0.8586**
AGNN_{knn}	1.0298	0.8357	1.0282	0.8343	1.0149	0.8177	0.9797	0.7901	1.0805	0.8824	1.0762	0.8708
AGNN_{cop}	1.0717	0.8779	1.0310	0.8341	1.0314	0.8381	0.9858	0.7994	1.0788	0.8787	1.0734	0.8658
AGNN_{GCN}	1.0308	0.8343	1.0280	0.8295	1.0165	0.8224	0.9818	0.7919	1.0772	0.8749	1.0766	0.8711
AGNN_{GAT}	1.0262	0.8280	1.0274	0.8289	1.0152	0.8227	0.9785	0.7825	1.0768	0.8752	1.0811	0.8746
$\text{AGNN}_{\text{mask}}$	1.0230	0.8298	1.0250	0.8243	1.0176	0.8228	0.9770	0.7831	1.0847	0.8861	1.0687	0.8611
$\text{AGNN}_{\text{drop}}$	1.0256	0.8262	1.0246	0.8247	1.0163	0.8179	0.9816	0.7808	1.0885	0.8831	1.0719	0.8693
$\text{AGNN}_{\text{LLAE}}$	1.0399	0.8428	1.0325	0.8344	1.0364	0.8424	0.9872	0.7970	1.1104	0.9193	1.0823	1.0887
$\text{AGNN}_{\text{LLAE+}}$	1.0259	0.8215	1.0259	0.8276	1.0210	0.8240	0.9793	0.7847	1.1033	0.9090	1.0686	0.8688

AGNN benefits from its dynamic construction strategy as it allows to access diversified neighbors, and thus yields better performance than two variants with the fixed neighbors.

We then examine how the performance changes if the gated-GNN is replaced by the conventional GNN structures. To this end, AGNN_{GCN} employs an ordinary GCN used in GC-MC [17], i.e., aggregating all neighbors' representations with a summation operation. AGNN_{GAT} adopts the graph attention technique in DANSER [21], i.e., using an attention layer to learn the weight of each neighbor before aggregation. As can be seen, AGNN_{GCN} is inferior to AGNN_{GAT} in many cases, indicating that attention mechanism can improve the performance. Moreover, they are both worse than our AGNN. This verifies that differentiating the importance of each dimension of the node can further enhance the performance since it greatly enlarges the model capacity.

We finally investigate the impacts of other techniques for addressing the (strict) cold start issues by replacing eVAE with mask [23], dropout [52], and denoising autoencoder [53]. Specifically, $\text{AGNN}_{\text{mask}}$ randomly masks 20 percent of the input nodes and adds a decoder after gated-GNN layer to reconstruct the initial input node embedding. $\text{AGNN}_{\text{drop}}$ has the similar adaption except that it randomly sets 20 percent preference embedding of the input nodes to 0. $\text{AGNN}_{\text{LLAE}}$ reconstructs the preference embedding from the attribute embedding using a denoising auto-encoder and removes the gated-GNN structure since it is not included in the original LLAE. $\text{AGNN}_{\text{LLAE+}}$ is an enhanced version which contains gated-GNN to aggregate neighbor information. Note that both these two variants of LLAE adopt our optimization objective to exclude the influence caused by the loss function in the original LLAE [53].



(a) ML-100K (b) ML-1M (c) Yelp

Fig. 9. Training curves of prediction loss and reconstruction loss on different datasets.

As can be seen, $\text{AGNN}_{\text{mask}}$ is in general better than $\text{AGNN}_{\text{drop}}$ since it can be viewed as a fine-grained dropout at the dimension level rather than the node level. Two variants of LLAE are worse than $\text{AGNN}_{\text{mask}}$ and $\text{AGNN}_{\text{drop}}$ in most cases, showing that the reconstruction of attribute via auto-encoder does not work well. Moreover, $\text{AGNN}_{\text{LLAE}}$ is the worst because it does not make use of neighbor information. Finally, our AGNN outperforms all four variants. The results clearly demonstrate that our proposed architecture is more effective than the mask [23], dropout [52], and the reconstruction [53] techniques in addressing the (strict) cold start problem.

5.2 Complexity Analysis

In this subsection, we investigate the complexity of proposed AGNN. It is obvious that the aggregation and propagation in Gated-GNN is the main operation. During training process, we need to update dynamic neighbors of users and items for each user-item pair, which requires time complexity $O(|\mathcal{R}^+||N_u||N_i|D)$, where $|\mathcal{R}^+|$ denotes the number of none-zero entries in user-item interaction matrix, $|N_u|$ and $|N_i|$ is a small number (=10 in this study) of dynamic neighbors of target user u and item i, respectively, and D is the embedding dimension. Therefore, AGNN can scale linearly w.r.t the number of user-item interactions.

We plot the curves of training loss in Fig. 9 to reveal the training process. We show the prediction loss and reconstruction loss separately in terms of strict item cold start and strict user cold start, respectively. At the early training stages, both these losses decrease rapidly. With the training process forward, the prediction loss declines smoothly and the reconstruction loss converge gradually at about the 4th epoch. This proves that our model is stable and easy to train.

6 Conclusion

In this paper, we propose a novel model, namely AGNN for the strict cold start rating prediction tasks. We first highlight the importance of exploiting the attribute graph rather than the interaction graph in addressing strict cold start problem in neural graph recommender systems. We then present an eVAE structure to infer preference embedding from attribute distribution. Moreover, we address the key challenges in aggregating various information in a neighborhood by developing a gated-GNN structure which greatly improves the model capacity.

We conduct extensive experiments on three real-world datasets. Results prove that our AGNN model significantly outperforms the state-of-the-art baselines in strict cold start recommendation. It also achieves the best or the second-best performance in warm start scenario.

ACKNOWLEDGMENTS

The work described in this paper was supported in part by the NSFC project (61572376, 91646206).

REFERENCES

[1] D. Goldberg, D. Nichols, B. M. Oki, and D. Terry, "Using collaborative filtering to weave an information tapestry," *Commun. ACM*, vol. 35, no. 12, pp. 61–70, 1992.
[2] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, 2009.
[3] H. Ma, D. Zhou, C. Liu, M. R. Lyu, and I. King, "Recommender systems with social regularization," in *Proc. 4th ACM Int. Conf. Web Search Data Mining*, 2011, pp. 287–296.
[4] S. Rendle, "Factorization machines," in *Proc. IEEE Int. Conf. Data Mining*, 2010, pp. 995–1000.
[5] S. Kabbur, X. Ning, and G. Karypis, "FISM: Factored item similarity models for top-n recommender systems," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2013, pp. 659–667.
[6] G. Guo, J. Zhang, and N. Yorke-Smith, "TrustSVD: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings," in *Proc. 29th AAAI Conf. Artif. Intell.*, 2015, pp. 123–125.
[7] L. Zheng, V. Noroozi, and P. S. Yu, "Joint deep modeling of users and items using reviews for recommendation," in *Proc. 10th ACM Int. Conf. Web Search Data Mining*, 2017, pp. 425–434.
[8] C. Chen, M. Zhang, Y. Liu, and S. Ma, "Neural attentional rating regression with review-level explanations," in *Proc. World Wide Web Conf.*, 2018, pp. 1583–1592.
[9] X. Wang, X. He, Y. Cao, M. Liu, and T. S. Chua, "KGAT: Knowledge graph attention network for recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 950–958.
[10] W. L. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1024–1034.
[11] M. Zhang and Y. Chen, "Inductive matrix completion based on graph neural networks," in *Proc. Int. Conf. Learn. Representations*, 2020.
[12] H. Lee, J. Im, S. Jang, H. Cho, and S. Chung, "MeLU: Meta-learned user preference estimator for cold-start recommendation," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2019, pp. 1073–1082.
[13] Y. Lu, Y. Fang, and C. Shi, "Meta-learning on heterogeneous information networks for cold-start recommendation," in *Proc. 26th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2020, pp. 1563–1573.
[14] Y. Zhang, Q. Ai, X. Chen, and W. B. Croft, "Joint representation learning for top-n recommendation with heterogeneous information sources," in *Proc. ACM Conf. Inf. Knowl. Manage.*, 2017, pp. 1449–1458.
[15] T. Liu, Z. Wang, J. Tang, S. Yang, G. Y. Huang, and Z. Liu, "Recommender systems with heterogeneous side information," in *Proc. World Wide Web Conf.*, 2019, pp. 3027–3033.
[16] C. Shi, B. Hu, W. X. Zhao, and P. S. Yu, "Heterogeneous information network embedding for recommendation," *IEEE Trans. Knowl. Data Eng.*, vol. 31, no. 2, pp. 357–370, Feb. 2019.
[17] R. V. D. Berg, T. N. Kipf, and M. Welling, "Graph convolutional matrix completion," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018.
[18] R. Ying, R. He, K. Chen, P. Eksombatchai, W. L. Hamilton, and J. Leskovec, "Graph convolutional neural networks for web-scale recommender systems," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 974–983.
[19] X. Wang, X. He, M. Wang, F. Feng, and T.-S. Chua, "Neural graph collaborative filtering," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 165–174.
[20] L. Wu, P. Sun, Y. Fu, R. Hong, X. Wang, and M. Wang, "A neural influence diffusion model for social recommendation," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 235–244.
[21] Q. Wu *et al.*, "Dual graph attention networks for deep latent representation of multifaceted social effects in recommender systems," in *Proc. World Wide Web Conf.*, 2019, pp. 2091–2102.
[22] X. He, K. Deng, X. Wang, Y. Li, Y. Zhang, and M. Wang, "LightGCN: Simplifying and powering graph convolution network for recommendation," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 639–648.
[23] J. Zhang, X. Shi, S. Zhao, and I. King, "STAR-GCN: Stacked and reconstructed graph convolutional networks for recommender systems," in *Proc. 28th Int. Joint Conf. Artif. Intell.*, 2019, pp. 4264–4270.
[24] L. Hu, S. Jian, L. Cao, Z. Gu, Q. Chen, and A. Amirbekyan, "HERS: Modeling influential contexts with heterogeneous relations for sparse and cold-start recommendation," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 3830–3837.
[25] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, 2001, pp. 285–295.
[26] Y. Koren, "Factorization meets the neighborhood: A multifaceted collaborative filtering model," in *Proc. 14th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2008, pp. 426–434.
[27] A. Mnih and R. R. Salakhutdinov, "Probabilistic matrix factorization," in *Proc. 20th Int. Conf. Neural Inf. Process. Syst.*, 2007, pp. 1257–1264.
[28] H. Wang, N. Wang, and D. Y. Yeung, "Collaborative deep learning for recommender systems," in *Proc. 21th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2015, pp. 1235–1244.
[29] G. K. Dziugaite and D. M. Roy, "Neural network matrix factorization," 2015, *arXiv:1511.06443*.
[30] H. Zhang, L. Nie, X. Hu, X. He, L. Liao, and T.-S. Chua, "Neural collaborative filtering," in *Proc. 26th Int. Conf. World Wide Web*, 2017, pp. 173–182.
[31] F. Monti, M. M. Bronstein, and X. Bresson, "Geometric matrix completion with recurrent multi-graph neural networks," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 3700–3710.
[32] L. Chen, L. Wu, R. Hong, K. Zhang, and M. Wang, "Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach," in *Proc. 34th AAAI Conf. Artif. Intell.*, 2020, pp. 27–34.
[33] L. Wu, Y. Yang, K. Zhang, R. Hong, Y. Fu, and M. Wang, "Joint item recommendation and attribute inference: An adaptive graph convolutional network approach," in *Proc. 43rd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2020, pp. 679–688.
[34] H. Wang, M. Zhao, X. Xie, W. Li, and M. Guo, "Knowledge graph convolutional networks for recommender systems," in *Proc. World Wide Web Conf.*, 2019, pp. 3307–3313.
[35] W. Fan *et al.*, "Graph neural networks for social recommendation," in *Proc. World Wide Web Conf.*, 2019, pp. 417–426.
[36] D. P. Kingma and M. Welling, "Auto-encoding variational bayes," in *Proc. Int. Conf. Learn. Representations*, 2014.
[37] X. Li and J. She, "Collaborative variational autoencoder for recommender systems," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2017, pp. 305–314.
[38] D. Liang, R. G. Krishnan, M. D. Hoffman, and T. Jebara, "Variational autoencoders for collaborative filtering," in *Proc. World Wide Web Conf.*, 2018, pp. 689–698.
[39] N. Sachdeva, G. Manco, E. Ritacco, and V. Pudi, "Sequential variational autoencoders for collaborative filtering," in *Proc. 12th ACM Int. Conf. Web Search Data Mining*, 2019, pp. 600–608.
[40] Y. Li, R. Zemel, M. Brockschmidt, and D. Tarlow, "Gated graph sequence neural networks," in *Proc. Int. Conf. Learn. Representations*, 2016.
[41] A. Rahimi, T. Cohn, and T. Baldwin, "Semi-supervised user geolocation via graph convolutional networks," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, 2018, pp. 2009–2019.
[42] T. Chen, W. Zhang, Q. Lu, K. Chen, and Y. Yong, "Svdfeature: A toolkit for feature-based collaborative filtering," *J. Mach. Learn. Res.*, vol. 13, no. 1, pp. 3619–3622, 2012.
[43] X. He and T.-S. Chua, "Neural factorization machines for sparse predictive analytics," in *Proc. 40th Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2017, pp. 355–364.
[44] X. Xin, X. He, Y. Zhang, Y. Zhang, and J. Jose, "Relational collaborative filtering: Modeling multiple item relations for recommendation," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 125–134.

[45] W. Fu, Z. Peng, S. Wang, Y. Xu, and J. Li, "Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 94–101.
[46] Y. Zhu *et al.*, "Addressing the item cold-start problem by attribute-driven active learning," *IEEE Trans. Knowl. Data Eng.*, vol. 32, no. 4, pp. 631–644, Apr. 2020.
[47] M. Aharon, O. Anava, N. Avigdor-Elgrabli, D. Drachsler-Cohen, S. Golan, and O. Somekh, "ExcUseMe: Asking users to help in item cold-start recommendations," in *Proc. 9th ACM Conf. Recommender Syst.*, 2015, pp. 83–90.
[48] O. Anava *et al.*, "Budget-constrained item cold-start handling in collaborative filtering recommenders via optimal design," in *Proc. 24th Int. Conf. World Wide Web*, 2015, pp. 45–54.
[49] M. Nasery, M. Braunhofer, and F. Ricci, "Recommendations with optimal combination of feature-based and item-based preferences," in *Proc. Conf. User Model. Adaptation Personalization*, 2016, pp. 269–273.
[50] M. Vartak, A. Thiagarajan, C. Miranda, J. Bratman, and H. Larochelle, "A meta-learning perspective on cold-start recommendations for items," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 6904–6914.
[51] F. Pan, S. Li, X. Ao, P. Tang, and Q. He, "Warm up cold-start advertisements: Improving CTR predictions via learning to learn ID embeddings," in *Proc. 42nd Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2019, pp. 695–704.
[52] M. Volkovs, G. Yu, and T. Poutanen, "DropoutNet: Addressing cold start in recommender systems," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 4957–4966.
[53] J. Li, M. Jing, K. Lu, L. Zhu, Y. Yang, and Z. Huang, "From zero-shot learning to cold-start recommendation," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 4189–4196.
[54] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017.
[55] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Lio, and Y. Bengio, "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018.
[56] M. McPherson, L. S. Lovin, and J. M. Cook, "Birds of a feather: Homophily in social networks," *Annu. Rev. Sociol.*, vol. 27, no. 1, pp. 415–444, 2001.
[57] J. Leskovec, L. A. Adamic, and B. A. Huberman, "The dynamics of viral marketing," *ACM Trans. Web*, vol. 1, no. 1, pp. 5–es, 2007.
[58] P. Singla and M. Richardson, "Yes, there is a correlation - from social networks to personal behavior on the web," in *Proc. 17th Int. Conf. World Wide Web*, 2008, pp. 655–664.
[59] Y. Zhao, G. Wang, P. S. Yu, S. Liu, and S. Zhang, "Inferring social roles and statuses in social networks," in *Proc. 19th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2013, pp. 695–703.
[60] Y. Lu, R. Dong, and B. Smyth, "Coevolutionary recommendation model: Mutual learning between ratings and reviews," in *Proc. World Wide Web Conf.*, 2018, pp. 773–782.
[61] Y. Tay, A. T. Luu, and S. C. Hui, "Multi-pointer co-attention networks for recommendation," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2018, pp. 2309–2318.
[62] Z. Cheng, Y. Ding, X. He, L. Zhu, X. Song, and M. S. Kankanhalli, "A^3NCF: An adaptive aspect attention model for rating prediction," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, 2018, pp. 3748–3754.
[63] W. Fu, Z. Peng, S. Wang, Y. Xu, and J. Li, "Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems," in *Proc. 33rd AAAI Conf. Artif. Intell.*, 2019, pp. 94–101.
[64] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. Learn. Representations*, 2015.

Tieyun Qian (Member, IEEE) received the BS degree in computer science from the Wuhan University of Technology, China, in 1991, and the PhD degree in computer science from the Huazhong University of Science and Technology, China, in 2006. She is a professor at the School of Computer Science, Wuhan University, China. Her current research interests include text mining, Web mining, and natural language processing. She has published more than 60 papers in leading conferences and top journals including ACL, AAAI, CIKM, TOIS, and TKDD. She is a member of ACM and CCF. She has served as program committee member of many premium conferences like WWW, AAAI, IJCAI, and DASFAA.

Yile Liang received the BS degree in computer science from Hunan University, China, in 2018. He is currently working toward the master's degree in the School of Computer Science, Wuhan University, China. His current research interests are in the recommender systems and web mining. He has authored/coauthored several papers in leading conferences and journals such as DASFAA, AAAI, and Neurocomputing.

Qing Li (Senior Member, IEEE) received the BEng degree from Hunan University, Changsha, China, and the MSc and PhD degrees from the University of Southern California, Los Angeles, California, all in computer science. He is a chair professor at the Department of Computing, the Hong Kong Polytechnic University, Hong Kong. His research interests include multi-modal data management, machine learning, social media, Web services, and e-learning systems. He has authored/coauthored more than 400 publications in these areas. He has actively involved in the research community and served as an associate editor of a number of major technical journals including the *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, *ACM Transactions on Internet Technology (TOIT)*, *Data Science and Engineering (DSE)*, *World Wide Web (WWW)*, and *Journal of Web Engineering*, in addition to being a Conference and program chair/co-chair of numerous major international conferences. He also sat in the Steering Committees of DASFAA, ER, ACM RecSys, IEEE U-MEDIA, and ICWL. He is a fellow of IEE/IET, and a distinguished member of CCF (China).

Hui Xiong (Fellow, IEEE) received the PhD degree from the University of Minnesota (UMN), Minneapolis, Minnesota. He is currently a full professor at the Rutgers, the State University of New Jersey, Brunswick, New Jersey, where he received the 2018 Ram Charan Management Practice Award as the Grand Prix winner from the Harvard Business Review, RBS Dean's Research Professorship (2016), the Rutgers University Board of Trustees Research Fellowship for Scholarly Excellence (2009), the ICDM Best Research Paper Award (2011) and the IEEE ICDM Outstanding Service Award (2017). He is a co-editor-in-chief of Encyclopedia of GIS, an associate editor of the *IEEE Transactions on Big Data (TBD)*, *ACM Transactions on Knowledge Discovery from Data (TKDD)*, and *ACM Transactions on Management Information Systems (TMIS)*. He has served regularly on the organization and program committees of numerous conferences, including as a program co-chair of the Industrial and Government Track for the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), a program co-chair for the IEEE 2013 International Conference on Data Mining (ICDM), a general co-chair for the IEEE 2015 International Conference on Data Mining (ICDM), and a program co-chair of the Research Track for the 2018 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. He is an ACM distinguished scientist.