Semi-supervised User Profiling with Heterogeneous Graph Attention Networks

Weijian Chen[1], **Yulong Gu**[2], **Zhaochun Ren**[3*], **Xiangnan He**[1*], **Hongtao Xie**[1], **Tong Guo**[1], **Dawei Yin**[2] and **Yongdong Zhang**[1]
[1] University of Science and Technology of China, Hefei, China
[2] JD.com, China
[3] Shandong University, China
{naure,gt1996}@mail.ustc.edu.cn, {htxie,zhyd73}@ustc.edu.cn,
{guyulongcs,xiangnanhe}@gmail.com, zhaochun.ren@sdu.edu.cn, yindawei@acm.org

Abstract

Aiming to represent user characteristics and personal interests, the task of user profiling is playing an increasingly important role for many real-world applications, e.g., e-commerce and social networks platforms. By exploiting the data like texts and user behaviors, most existing solutions address user profiling as a classification task, where each user is formulated as an individual data instance. Nevertheless, a user's profile is not only reflected from her/his affiliated data, but also can be inferred from other users, e.g., the users that have similar co-purchase behaviors in e-commerce, the friends in social networks, etc. In this paper, we approach user profiling in a semi-supervised manner, developing a generic solution based on heterogeneous graph learning. On the graph, nodes represent the entities of interest (e.g., users, items, attributes of items, etc.), and edges represent the interactions between entities. Our *heterogeneous graph attention networks* (HGAT) method learns the representation for each entity by accounting for the graph structure, and exploits the attention mechanism to discriminate the importance of each neighbor entity. Through such a learning scheme, HGAT can leverage both unsupervised information and limited labels of users to build the predictor. Extensive experiments on a real-world e-commerce dataset verify the effectiveness and rationality of our HGAT for user profiling.

1 Introduction

By inferring user personality traits from user-generated data, the task of user profiling is playing an important role in providing personalized services in real-world applications, e.g., e-commerce and social networks, etc. [Farnadi *et al.*, 2018; Lu *et al.*, 2016; Chen *et al.*, 2019; Wu *et al.*, 2019a; Liao *et al.*, 2018]. Existing approaches consider user profiling as a classification task to classify a user's personal profile (e.g., gender and age) with either textual or behavior information, where each user is set as an individual data instance [Zamal *et al.*, 2012; Dong *et al.*, 2014; Miura *et al.*, 2017]. Nevertheless, we argue that existing user profiling methods suffer from two common issues:

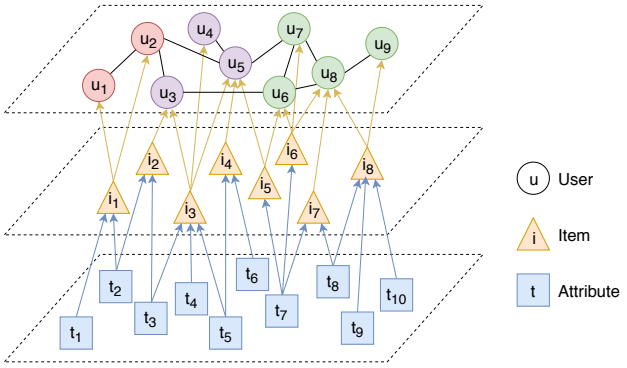

Figure 1: User Profiling in the heterogeneous graph

1. Only one type of information is used to infer user profiles [Dong *et al.*, 2014], whereas other types of data cannot be naturally integrated. Very few related works [Miura *et al.*, 2017; Farnadi *et al.*, 2018] touch the multi-type user profiling problem. However, they need to carefully design hand-crafted features or the fusion method.
2. Only self-generated data is exploited to learn the user profiling representation, whereas the rich interactions among data instances are neglected. Such interactions, e.g., the co-clicks and co-purchases in e-commerce, the friendship relations among users in social network, can be considered as semi-supervised signals which are valuable to improve the inference of user profiles [Rahimi *et al.*, 2018].

Recent studies have demonstrated that graph is capable to provide a general representation to integrate multiple types of data [Kipf and Welling, 2017; Ma *et al.*, 2018; Cao *et al.*, 2019]. Figure 1 shows an example of a graph with heterogeneous information, where three kinds of nodes are applied to represent three types of data, i.e., users, items, and attributes, respectively. Inspired by the recent success of graph neural network approaches in several tasks [Hamilton *et al.*, 2017; Wang *et al.*, 2019b], we expect that they work well to learn user profiles from multiple types of data. One rationality is

*Corresponding Authors

that the neighborhood features could provide valuable semi-supervised signals that are beneficial to infer user profiles. For example, users that have similar co-purchase behaviors in e-commerce are likely to be in the same age range. Although a recent work [Rahimi *et al.*, 2018] has explored graph neural network for user profiling, it only deals with one type of data, and needs non-trivial efforts (e.g., fusion strategy) to migrate it to heterogeneous graph with multiple types of data. Towards this research gap, we consider developing a heterogeneous graph neural network method for user profiling. In this paper, we propose a new framework, named *heterogeneous graph attention networks* (HGAT), to infer user profiles within a multi-type data environment. HGAT is capable to model the rich unsupervised information in heterogeneous graph by encoding both the graph structure and node features. Specifically, HGAT first learns user representation by propagating information in the heterogeneous graph using attention operations. Secondly, HGAT trains an end-to-end semi-supervised user profiling predictor using limited labels of users. To evaluate the performance of HGAT, we collect a large-scale real-world dataset from an e-commerce portal. Extensive experiments conducted on this dataset verify the effectiveness of HGAT. Further studies verify the rationality of each module designed. To sum up, the contributions of this work are as follows:

- We propose to approach user profiling as a semi-supervised classification task in the heterogeneous graph, opening up an opportunity of developing generic solutions to incorporate multiple types of data.
- We develop a heterogeneous graph attention networks framework HGAT, sufficiently leveraging the graph structure and node features to learn user profiles from limited labeled data.
- We conduct extensive experiments on a large-scale real-world dataset and verify the effectiveness of our method.

2 Related Work

2.1 User Profiling

Existing methods for user profiling usually firstly extract features from texts, relations, behaviors and so on, and then exploit machine learning techniques to infer users' profiles [Rosenthal and McKeown, 2011; Zamal *et al.*, 2012; Dong *et al.*, 2014; Liang *et al.*, 2018; Li *et al.*, 2012; Wu *et al.*, 2019b]. For example, [Rosenthal and McKeown, 2011] exploited logistic regression to predict users' age based on blog texts and online behavior of blog authors. [Zamal *et al.*, 2012] extracted designed features from twitter users and their neighbors and then used SVM and GBDT models to infer attributes of users. [Dong *et al.*, 2014] proposed a graphical model to infer users' gender and age based on their communication records in mobile networks.

Recently, modeling heterogeneous information in multiple sources of user data for user profiling using deep learning has gained significant interest. Specifically, [Miura *et al.*, 2017] used a complex neural network to unify text, metadata, and user network representations with attention mechanism and predict users' geolocations. [Farnadi *et al.*, 2018] proposed a hybrid user profiling framework which uses separate deep neural networks to extract information from different sources and then integrates the decisions of these networks.

However, these approaches need to design either handcrafted features or fusion methods. What's more, they need large scale labels of users for supervised learning, whereas many semi-supervised signals have been neglected. These problems may limit the universality or the performance of these approaches in various applications.

2.2 Graph Attention Networks

Graph Convolutional Network (GCN) [Kipf and Welling, 2017], which performs convolutional operations on graph-structured data, has recently achieved appealing performance in a variety of tasks, such as node classification [Kipf and Welling, 2017], recommendation [Wang *et al.*, 2019b] and stock prediction [Feng *et al.*, 2019]. They can encode both graph structure and features of nodes without the need for designing features or fusion methods. Graph Attention Network (GAT) [Velickovic *et al.*, 2018; Wang *et al.*, 2019a] extends the graph convolutional operations in GCN with masked self-attentional layers, which enable attending different weights to different neighborhoods. [Hamilton *et al.*, 2017; Chen *et al.*, 2018; Gao *et al.*, 2018] proposed sampling strategies and subgraph training methods and enabled the efficient application of GCN and GAT in large-scale graphs.

Recently, there are some preliminary works of applying GCN for user profiling. [Rahimi *et al.*, 2018] proposed a multiview model based on GCN to infer users' geolocations in social media based on text and network information. They also encounter the problem of needing the design of fusion architectures. In this paper, we propose a general framework that can directly model the information in heterogeneous networks and build user profiles.

3 Problem Formulation

In this section, we introduce Heterogeneous Graph and formulate the Semi-supervised User Profiling problem.

3.1 Heterogeneous Graph

In this paper, we represent information networks (e.g., e-commerce, social networks and so on) as heterogeneous graphs. The structure of a heterogeneous graph is shown in Figure 1. For a heterogeneous graph $G = (V, E)$, V and E denote the nodes and edges in the graph. The nodes V are consisted of the set of users U, items I and attributes T. The edges E have three types: User-User edges E_{uu} that reflect the relationships between users, Item-User edges E_{ui} that express the interactions between users and items, and Attribute-Item edges E_{it} that describe the attribute information of items.

For example, in e-commerce, items are products and attributes can be the words in the titles of products. Each item (i.e., product) has some attributes (e.g., words), and each user (i.e., consumer) may purchase some items. An Attribute-Item edge describes a word exists in the title of a product, an Item-User edge means that a user has purchased (or clicked) a product, and a User-User edge represents that the two users have co-purchased (or co-clicked) some same products.

Subgraphs in Heterogeneous Graph

In this paper, according to the types of edges, the heterogeneous graph can be divided into three subgraphs: User-User subgraph, Item-User subgraph and Attribute-Item subgraph.

User-User subgraph. User-User subgraph is consisted of users and edges between them in the heterogeneous graph.

Item-User subgraph. The nodes in Item-User subgraph are items and users in the heterogeneous graph. The edges in the subgraph are the interactions between items and users.

Attribute-Item subgraph. The nodes in Attribute-Item subgraph are attributes and items in the heterogeneous graph, while the edges are the attribute information of items.

3.2 Semi-supervised User Profiling

In information networks, user profiles, which represent the labels (e.g., demographic characteristics, interests, etc.) or interests of users, are significant for personalized search, recommendation, advertisements and so on. However, in the real world, user profiles are usually unknown due to privacy concerns and other reasons. Consequently, user profiling, which aims to infer user profiles, is significant for real applications. One important user profiling problem is to infer the gender and age labels of users [Zamal *et al.*, 2012; Dong *et al.*, 2014; Farnadi *et al.*, 2018].

Existed solutions usually follow the supervised paradigm [Rosenthal and McKeown, 2011; Zamal *et al.*, 2012; Dong *et al.*, 2014; Li *et al.*, 2012]. In this paper, we solve the user profiling problem under the semi-supervised learning paradigm [Kipf and Welling, 2017]. Specifically, our goal is to use both the labels of some users and a large amount of unsupervised information in the heterogeneous graph, such as the interactions between users and items and the attribute information of items.

Definition 3.1 (Semi-supervised User Profiling Problem) *Semi-supervised User Profiling aims to infer the labels of users based on supervised labels of some users and large scale unsupervised information in the heterogeneous graph.*

4 Heterogeneous Graph Attention Networks

In this paper, we propose the *heterogeneous graph attention networks* (HGAT) framework to solve the Semi-supervised User Profiling problem in the heterogeneous graph.

As shown in Figure 2, HGAT is consisted of three parts: the Input and Embedding Layer, the Heterogeneous Graph Attention Layers and the Output Layer.

4.1 Input and Embedding Layer

In our framework, the input is the information of nodes and edges derived from the heterogeneous graph.

For the embeddings of attributes (i.e., words), we use *FastText* [Bojanowski *et al.*, 2017] to learn the embeddings of the words using the whole item titles corpus. These embeddings are regarded as low-dimension representations of attributes.

4.2 Heterogeneous Graph Attention Layers

For the representation learning in a heterogeneous graph, there are two critical problems: (1) How embeddings of nodes are updated? (2) How the information is propagated across heterogeneous graph? For the first problem, we use three Heterogeneous Graph Attention Operations for embedding updating. For the second problem, we propose Meta-path aware Graph Propagation to define the information propagation method in the heterogeneous graph.

Heterogeneous Graph Attention Operations

Heterogeneous Graph Attention Operations update the embedding of a node based on information in its neighbors. They transform the embeddings of nodes in graph G from $\mathbf{H} \in \mathbb{R}^{|V|\times F}$ into new embedding matrix $\mathbf{H}' \in \mathbb{R}^{|V|\times F'}$. The difference between these operations is the calculation method. Given a node i whose neighbors are $\mathcal{N}_i$, Heterogeneous Graph Attention Operations will transform its embedding from $\mathbf{h}_i$ to $\mathbf{h}'_i$.

We exploit three kinds of Heterogeneous Graph Attention Operations : Vanilla Attention Operation, Graph Convolutional Operation and Graph Attention Operation.

Vanilla Attention Operation. This operation uses the vanilla attention mechanism [He *et al.*, 2018]. Given a node i and its neighbors $\mathcal{N}_i$, its new embedding $\mathbf{h}'_i$ is calculated as follows:

$$\begin{aligned} e_{ij} &= \mathbf{c}^T tanh(\mathbf{W}\mathbf{h}_j + \mathbf{b}) \\ \alpha_{ij} &= \frac{\exp(e_{ij})}{\sum_{k\in\mathcal{N}_i}\exp(e_{ik})} \\ \mathbf{h}'_i &= \sum_{j\in\mathcal{N}_i}\alpha_{ij}\mathbf{h}_j \end{aligned} \tag{1}$$

where the context vector $\mathbf{c} \in \mathbb{R}^{F'}$, the weight matrix $\mathbf{W} \in \mathbb{R}^{F'\times F}$ and the bias vector $\mathbf{b} \in \mathbb{R}^{F'}$ are parameters. These parameters are used to calculate the attention score α_{ij}, which measures the importance of a neighborhood node j to the node i.

Graph Convolutional Operation. The Graph Convolutional Operation uses the graph convolutional operation proposed in GCN [Kipf and Welling, 2017]. The operation is defined by the following formula:

$$\begin{aligned} \hat{\mathbf{A}} &= \widetilde{\mathbf{D}}^{-1/2}\tilde{\mathbf{A}}\widetilde{\mathbf{D}}^{-1/2} \\ \mathbf{H}' &= \sigma(\hat{\mathbf{A}}\mathbf{H}\mathbf{W}^T) \end{aligned} \tag{2}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}_{|V|}$ is the adjacency matrix with added self-loops (corresponding to the identity matrix $\mathbf{I}_{|V|}$), $\widetilde{\mathbf{D}} \in \mathbb{R}^{|V|\times|V|}$ is the degree matrix of $\tilde{\mathbf{A}}$, $\mathbf{W} \in \mathbb{R}^{F'\times F}$ is the weight matrix, and σ denotes an activation function.

Graph Attention Operation. The Graph Attention Operation exploits the multi-head graph attention operation used in Graph Attention Network [Velickovic *et al.*, 2018]. Firstly, it computes the attention scores between nodes based on a shared attentional mechanism $att : \mathbb{R}^{F'} \times \mathbb{R}^{F'} \to \mathbb{R}$. The neighborhood of node i is denoted as $\mathcal{N}_i$. For each node

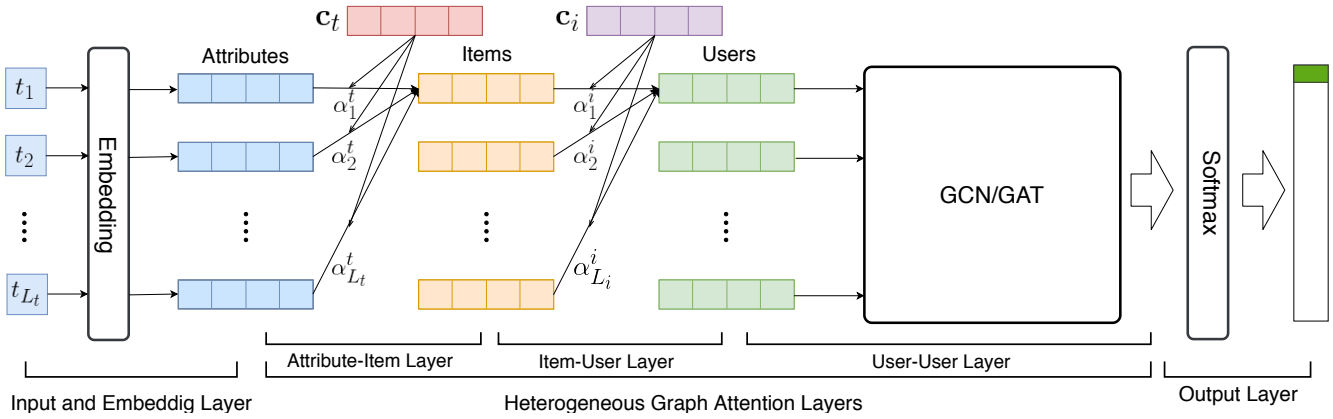

Figure 2: The architecture of our heterogeneous graph attention networks

$j \in \mathcal{N}_i$, the attention coefficient between node i and j is e_{ij}. To make coefficients easily comparable across different nodes, the $softmax$ function is used to normalize the attention coefficient e_{ij} into the attention score α_{ij}.

$$e_{ij} = att(\mathbf{W}\mathbf{h}_i, \mathbf{W}\mathbf{h}_j)$$
$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k \in \mathcal{N}_i} exp(e_{ik})} \quad (3)$$

where the attention mechanism att is a single-layer feedforward neural network, parameterized by $\mathbf{W} \in \mathbb{R}^{F' \times F}$ and $\mathbf{a} \in \mathbb{R}^{2F'}$, and applying the LeakyReLU nonlinearity (with negative slope 0.2):

$$att(\mathbf{W}\mathbf{h}_i, \mathbf{W}\mathbf{h}_j) = LeakyReLU(\mathbf{a}^T[\mathbf{W}\mathbf{h}_i || \mathbf{W}\mathbf{h}_j]) \quad (4)$$

where $||$ represents the concatenation between two vectors. Secondly, the node's new embedding $\mathbf{h}'_i$ is computed by using the sum on its neighbors' features, weighted by the attention scores.

$$\mathbf{h}'_i = \sigma(\sum_{j \in \mathcal{N}_i} \alpha_{ij} \mathbf{W}\mathbf{h}_j) \quad (5)$$

Multi-head attention is used to increase the representation power. Specifically, K independent attention mechanisms are performed using the transformation of Equation 5, and then the results $\mathbf{h}_i^{'(1)}, \mathbf{h}_i^{'(2)}, ..., \mathbf{h}_i^{'(K)}$ are aggregated together:

$$\mathbf{h}'_i = \sigma(agg(\mathbf{h}_i^{'(1)}, \mathbf{h}_i^{'(2)}, ..., \mathbf{h}_i^{'(K)})) \quad (6)$$

The aggregation function agg can be concatenation or averaging. Specially, if we perform multi-head attention on the final layer, we employ averaging. Otherwise, we apply the concatenation operation.

Meta-path aware Graph Attention Propagation

In this paper, we propose Meta-path aware Graph Attention Propagation to define the method of information propagation.

Attribute-Item, Item-User and User-User edges are key pieces of information in the heterogeneous graph. Heterogeneous Graph Attention Layers exploit three layers, i.e., the Attribute-Item Layer, Item-User Layer and User-User Layer, to model these edges. These layers use the Attribute-Item, Item-User, and User-User meta-paths, which correspond to information propagation in the respective subgraph.

Attribute-Item Layer. The Attribute-Item Layer propagates information from attributes to items. In this layer, we use Vanilla Attention Operation to propagate information in the Attribute-Item subgraph and learn embeddings of items leveraging the embeddings of attributes and the edges between items and attributes. For example, in e-commerce, an item's embedding can be learned by using the embeddings of words in the titles of this item (i.e., product).

Item-User Layer. The Item-User Layer propagates information from items to users. In this layer, we use Vanilla Attention Operation to propagate information in the Item-User subgraph and learn embeddings of users leveraging the embeddings of items and the edges between users and items. For example, in e-commerce, a user's embedding can be learned by using the embeddings of items she has bought.

User-User Layer. The User-User Layer propagates information from users to users. In this layer, we apply Graph Convolutional Operation or Graph Attention Operation to propagate information in the User-User subgraph and learn embeddings of users. Finally, our model will embed all relevant users into a matrix $\mathbf{H}_u \in \mathbb{R}^{|U| \times F_Y}$ where F_Y is the dimension of the users' embedding vectors, which is set to be equal to the number of labels in the user profiling task.

4.3 Output Layer

We view user profiling as a semi-supervised and multi-class classification task. The Output Layer predicts the labels of users based on the learned embedding vectors of users. For the user profiling task where F_Y is the number of categories (i.e., labels of users), we apply row-wise $softmax$ function on the users' embedding matrix $\mathbf{H}_u$ and obtain $\mathbf{Z} \in \mathbb{R}^{|U| \times F_Y}$, the predicted distribution of users' labels.

4.4 Model Learning

For the learning of our method, cross-entropy is adopted as the loss function to carry out end-to-end training for the model. Specifically, the loss function is defined as cross-entropy error over all labeled users:

$$\mathcal{L} = -\sum_{u \in U_L} \sum_{f=1}^{F_Y} Y_{uf} \log Z_{uf} \quad (7)$$

Nodes			Edges		
Users	Items	Attributes	User-User	Item-User	Attribute-Item
54,161	203,712	10,218	36,043,982	817,136	1,580,202

Table 1: Statistics of nodes and edges in the dataset

where U_L is the set of users that have labels, $\mathbf{Y}$ and $\mathbf{Z}$ are the ground-truth and the predicted probabilistic distribution of users' labels respectively.

4.5 Mini Heterogeneous Graph Sampling

In real-world information networks, such as e-commerce and social networks, the number of nodes and edges can be millions or billions. Traditional graph convolution networks [Kipf and Welling, 2017] need all the nodes in the graph are present simultaneously during the training procedure, which is not appropriate to be applied in real applications. [Hamilton *et al.*, 2017; Qiu *et al.*, 2018] propose some sampling methods to perform operations on large graphs. However, they are designed for the homogeneous graph.

In this paper, we extend these methods and propose a new sampling method for heterogeneous graph. To be specific, in the training procedure, for each user in a batch, we sample some nodes and edges corresponding to the meta-paths from the heterogeneous graph and build User-User, Item-User and Attribute-Item mini graphs. We apply the same weight for all the edges in the sampling as previous work did [Hamilton *et al.*, 2017].

User-User mini graph. For each user, we first sample L_{u_1} users from the user's neighbors and denote them as u_{s_1}. Then for each user in u_{s_1}, we sample L_{u_2} users from the user's neighbors. We iteratively perform these operations k times to obtain k-hop neighborhood information. The k-hop user-centered mini graph is called the User-User mini graph.

Item-User mini graph. For each user in the k-hop mini graph, we sample L_i items that the user has interacted with. The resulted graph is denoted as the Item-User mini graph.

Attribute-Item mini graph. For each item in the Item-User mini graph, we sample L_t attributes to describe the item, which leads to the Attribute-Item mini graph.

The representations of users are learned by applying Meta-path aware Graph Attention Propagation operations on these mini graphs, instead of the original graph.

5 Experiments

5.1 Datasets

To evaluate our proposed method in user profiling, we collect a large scale real-world dataset from JD.com*, one of the most popular e-commerce portals in China. In this dataset, users, items and attributes are consumers, products and words in the titles of products respectively. The statistics of nodes and edges in the dataset is shown in Table 1.

The profiles of users are the gender and age labels. The statistics of each label is demonstrated in Table 2.

*https://www.jd.com

Gender		Age			
Male	Female	< 26	$26 - 35$	$36 - 55$	> 55
31,717	22,444	3,403	29,322	12,888	8,548

Table 2: Statistics of each label in the dataset

5.2 Experimental Methods

In the experiments, we implement two instantiated models of our framework: HGAT and HGCN.

- **HGAT**: a instantiated model of our framework. It uses Vanilla Attention Operation in the Attribute-Item Layer and Item-User Layer and the Graph Attention Operation in the User-User Layer.
- **HGCN**: another instantiated model of our framework. The only difference is that HGCN uses the Graph Convolutional Operation in the User-User Layer.

We compare our methods HGAT and HGCN with several baseline methods for user profiling task.

- **Logistic Regression (LR)** is widely used in user profiling due to its advantages of efficiency and good interpretation ability [Rosenthal and McKeown, 2011].
- **Support Vector Machine (SVM)** is widely used to solve classification and user analysis problems [Zamal *et al.*, 2012].
- **Graph Convolutional Network (GCN)** is a semi-supervised learning algorithm on graph structured data. It is widely used for node classification [Kipf and Welling, 2017; Rahimi *et al.*, 2018].
- **Graph Attention Network (GAT)** is a state-of-the-art graph neural network model. By learning different attention coefficients to neighbors, it can acquire a better representation of nodes [Velickovic *et al.*, 2018].

5.3 Evaluation Metrics

In the experiments, there are two classification tasks: gender prediction and age prediction. We choose two metrics *Accuracy* and *Macro-F_1* [Wu *et al.*, 2019a], which are widely used in classification and user profiling problems, to evaluate the performance of our model.

5.4 Implementation Details

In the experiment, we randomly split labeled users into training set, validation set and test set with the ratio 75:12.5:12.5 following previous works [Qiu *et al.*, 2018].

During the training stage, we use the embeddings of all relevant users (i.e., the user and her k-hop neighbors), but only the labels assigned to the users in the training set. In the validation and testing stages, we use the labels of users in the validation and test set to evaluate our model respectively.

In the Mini Heterogeneous Graph Sampling procedure, the number of neighborhood samples is set as follows: $k = 2, L_{u_1} = 10, L_{u_2} = 4$ for User-User mini graph, $L_i = 10$ for Item-User mini graph, $L_t = 10$ for Attribute-Item mini graph. When we use *FastText* [Bojanowski *et al.*, 2017] to process the texts, we set the output dimension of word embedding as 200. We adopt Adam as the optimizer and set weight decay to $5e^{-4}$. The value of multi-head $K = 8$. The

Methods	Gender Prediction		Age Prediction	
	Accuracy	*Macro-F_1*	*Accuracy*	*Macro-F_1*
LR	0.505	0.499	0.220	0.203
SVM	0.501	0.492	0.189	0.183
GCN	0.453	0.411	0.370	0.231
GAT	0.463	0.433	0.397	0.206
HGCN	0.508	0.509	0.415	**0.246**
HGAT	**0.570**	**0.561**	**0.440**	0.232

Table 3: Performance of methods for User Profiling

learning rate, dropout rate, mini-batch size, are set to 0.005, 0.6, 64 for gender prediction and 0.1, 0.2, 32 for age prediction, respectively.

6 Results and Analysis

6.1 Overall Comparison

The performance of HGAT, HGCN and baseline methods is presented in Table 3.

From this table, we can find that:

(1) For gender prediction, HGAT achieves the best results in both *Accuracy* and *Macro-F_1*.

(2) For age prediction, HGAT achieves the best results in the metric of *Accuracy* and HGCN achieves the best results in the metric of *Macro-F_1*.

(3) HGAT and HGCN both achieve impressive improvements than state-of-the-art methods like GCN and GAT, which proves the superiority of our framework.

6.2 Ablation Study

Both HGCN and HGAT use three layers in the Heterogeneous Graph Attention Layers: Attribute-Item Layer, Item-User Layer, and User-User Layer. To investigate the effectiveness of the first two layers in our framework, we conduct experiments using variants of HGCN and HGAT.

- **HGCN$_1$**: It is a variant of HGCN, but it only uses the User-User Layer in the Heterogeneous Graph Attention Layers (i.e., no Attribute-Item Layer and Item-User Layer), which is the way we implement GCN on such large graph. The feature of a user is the average of embeddings of words in the items which the user has interacted with.
- **HGCN$_2$**:It is a variant of HGCN, but it only uses the Item-User Layer and User-User Layer in the Heterogeneous Graph Attention Layers (i.e., no Attribute-Item Layer). The feature of an item is the average of embedding of words in the title of the item.
- **HGAT$_1$**: It is a variant of HGAT, but it only uses the User-User Layer in the Heterogeneous Graph Attention Layers, which is the way we implement GAT on such large graph. The features of users are the same as HGCN$_1$.
- **HGAT$_2$**: It is a variant of HGAT, but it only uses the Item-User Layer and User-User Layer. The features of items are the same as HGCN$_2$.

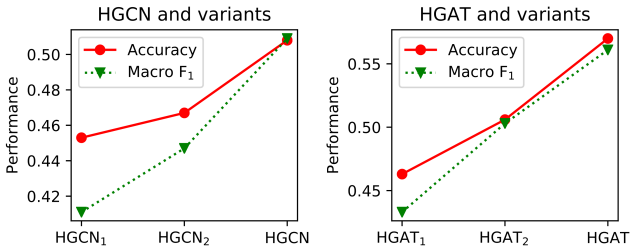

Figure 3: Performance of HGCN, HGAT and their variants

We evaluate HGCN, HGAT and these variants on the gender prediction task and demonstrate the performance in Figure 3. The experiment results on the age prediction task are similar. From the experiment results in this figure, we can find that:

(1) The prediction performance is improved when we add the Attribute-Item Layer and Item-User Layer, which exploit attention operations to automatically learn embeddings of items and users based on information in their neighborhoods.

(2) The information propagation operations between heterogeneous nodes are effective for user profiling in the heterogeneous graph.

(3) Our framework can effectively integrate the heterogeneous data in the network and achieves appealing performance for user profiling.

7 Conclusion and Future Work

In this paper, we have addressed the task of user profiling in a semi-supervised manner, which aims to solve two challenges in user profiling: single type of input data and negligence of semi-supervised signals. Unlike previous work that considered the user profiling as a classification task with self-generated user data, we have proposed *heterogeneous graph attention networks* (HGAT) to learn the representation for each entity by accounting for the graph structure, and present attention mechanisms to examine the importance of neighborhood entities. Thus, HGAT is capable to leverage both unsupervised information and limited labels of users to construct the predictor. Experiments on a large-scale real-world dataset have shown that HGAT outperforms state-of-the-art baselines for user profiling. We have also verified the effectiveness of components in HGAT.

To our best knowledge, our framework is the first method that can automatically model multi-relation graph structure and node features in heterogeneous networks for user profiling, without the need of designing hand-crafted features or fusion method. This opens up a new opportunity of solving various problems based on heterogeneous networks.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (2017YFC0820600), National Defense Science and Technology Fund for Distinguished Young Scholars (2017-JCJQ-ZQ-022), the National Nature Science Foundation of China (61525206, 61771468), the Youth Innovation Promotion Association Chinese Academy of Sciences (2017209).

References

[Bojanowski *et al.*, 2017] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146, 2017.

[Cao *et al.*, 2019] Yixin Cao, Xiang Wang, Xiangnan He, Zikun Hu, and Tat-Seng Chua. Unifying knowledge graph learning and recommendation: Towards a better understanding of user preferences. In *WWW*, pages 151–161, 2019.

[Chen *et al.*, 2018] Jie Chen, Tengfei Ma, and Cao Xiao. Fastgcn: Fast learning with graph convolutional networks via importance sampling. In *ICLR*, 2018.

[Chen *et al.*, 2019] Shiqian Chen, Chenliang Li, Feng Ji, Wei Zhou, and Haiqing Chen. Review-driven answer generation for product-related questions in e-commerce. In *WSDM*, pages 411–419, 2019.

[Dong *et al.*, 2014] Yuxiao Dong, Yang Yang, Jie Tang, Yang Yang, and Nitesh V. Chawla. Inferring user demographics and social strategies in mobile social networks. In *KDD*, pages 15–24, 2014.

[Farnadi *et al.*, 2018] Golnoosh Farnadi, Jie Tang, Martine De Cock, and Marie-Francine Moens. User profiling through deep multimodal fusion. In *WSDM*, pages 171–179, 2018.

[Feng *et al.*, 2019] Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. Temporal relational ranking for stock prediction. *ACM Transactions on Information Systems*, 37(2):27:1–27:30, 2019.

[Gao *et al.*, 2018] Hongyang Gao, Zhengyang Wang, and Shuiwang Ji. Large-scale learnable graph convolutional networks. In *KDD*, pages 1416–1424, 2018.

[Hamilton *et al.*, 2017] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, pages 1025–1035, 2017.

[He *et al.*, 2018] Xiangnan He, Zhankui He, Jingkuan Song, Zhenguang Liu, Yu-Gang Jiang, and Tat-Seng Chua. NAIS: Neural attentive item similarity model for recommendation. *IEEE Transactions on Knowledge and Data Engineering*, 30(12):2354–2366, 2018.

[Kipf and Welling, 2017] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[Li *et al.*, 2012] Rui Li, Shengjie Wang, Hongbo Deng, Rui Wang, and Kevin Chen-Chuan Chang. Towards social user profiling: unified and discriminative influence model for inferring home locations. In *KDD*, pages 1023–1031, 2012.

[Liang *et al.*, 2018] Shangsong Liang, Xiangliang Zhang, Zhaochun Ren, and Evangelos Kanoulas. Dynamic embeddings for user profiling in twitter. In *KDD*, pages 1764–1773, 2018.

[Liao *et al.*, 2018] Lizi Liao, Xiangnan He, Hanwang Zhang, and Tat-Seng Chua. Attributed social network embedding. *IEEE Transactions on Knowledge and Data Engineering*, 30(12):2257–2270, 2018.

[Lu *et al.*, 2016] Zhongqi Lu, Sinno Jialin Pan, Yong Li, Jie Jiang, and Qiang Yang. Collaborative evolution for user profiling in recommender systems. In *IJCAI*, pages 3804–3810, 2016.

[Ma *et al.*, 2018] Yao Ma, Zhaochun Ren, Ziheng Jiang, Jiliang Tang, and Dawei Yin. Multi-dimensional network embedding with hierarchical structure. In *WSDM*, pages 387–395, 2018.

[Miura *et al.*, 2017] Yasuhide Miura, Motoki Taniguchi, Tomoki Taniguchi, and Tomoko Ohkuma. Unifying text, metadata, and user network representations with a neural network for geolocation prediction. In *ACL*, pages 1260–1272, 2017.

[Qiu *et al.*, 2018] Jiezhong Qiu, Jian Tang, Hao Ma, Yuxiao Dong, Kuansan Wang, and Jie Tang. Deepinf: Social influence prediction with deep learning. In *KDD*, pages 2110–2119, 2018.

[Rahimi *et al.*, 2018] Afshin Rahimi, Trevor Cohn, and Timothy Baldwin. Semi-supervised user geolocation via graph convolutional networks. In *ACL*, pages 2009–2019, 2018.

[Rosenthal and McKeown, 2011] Sara Rosenthal and Kathleen R. McKeown. Age prediction in blogs: A study of style, content, and online behavior in pre- and post-social media generations. In *ACL*, pages 763–772, 2011.

[Velickovic *et al.*, 2018] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[Wang *et al.*, 2019a] Xiang Wang, Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua. KGAT: Knowledge graph attention network for recommendation. In *KDD*, 2019.

[Wang *et al.*, 2019b] Xiang Wang, Xiangnan He, Meng Wang, Fuli Feng, and Tat-Seng Chua. Neural graph collaborative filtering. In *SIGIR*, 2019.

[Wu *et al.*, 2019a] Chuhan Wu, Fangzhao Wu, Junxin Liu, Shaojian He, Yongfeng Huang, and Xing Xie. Neural demographic prediction using search query. In *WSDM*, pages 654–662, 2019.

[Wu *et al.*, 2019b] Libing Wu, Cong Quan, Chenliang Li, Qian Wang, Bolong Zheng, and Xiangyang Luo. A context-aware user-item representation learning for item recommendation. *ACM Transactions on Information Systems*, 37(2):22:1–22:29, 2019.

[Zamal *et al.*, 2012] Faiyaz Al Zamal, Wendy Liu, and Derek Ruths. Homophily and latent attribute inference: Inferring latent attributes of twitter users from neighbors. In *ICWSM*, 2012.