Community Value Prediction in Social E-commerce

Guozhen Zhang[1], Yong Li[1*], Yuan Yuan[1], Fengli Xu[1], Hancheng Cao[2], Yujian Xu[3], Depeng Jin[1]
[1]Beijing National Research Center for Information Science and Technology,
Department of Electronic Engineering, Tsinghua University, Beijing 100084, China
[2]Department of Computer Science, Stanford University, [3]Beibei Group
*Corresponding author: liyong07@tsinghua.edu.cn

ABSTRACT

The phenomenal success of the newly-emerging social e-commerce has demonstrated that utilizing social relations is becoming a promising approach to promote e-commerce platforms. In this new scenario, one of the most important problems is to predict the value of a community formed by closely connected users in social networks due to its tremendous business value. However, few works have addressed this problem because of 1) its novel setting and 2) its challenging nature that the structure of a community has complex effects on its value. To bridge this gap, we develop a Multi-scale Structure-aware Community value prediction network (MSC) that jointly models the structural information of different scales, including peer relations, community structure, and inter-community connections, to predict the value of given communities. Specifically, we first proposed a Masked Edge Learning Graph Convolutional Network (MEL-GCN) based on a novel masked propagation mechanism to model peer influence. Then, we design a Pair-wise Community Pooling (PCPool) module to capture critical community structures. Finally, we model the inter-community connections by distinguishing intra-community edges from inter-community edges and employing a Multi-aggregator Framework (MAF). Extensive experiments on a large-scale real-world social e-commerce dataset demonstrate our method's superior performance over state-of-the-art baselines, with a relative performance gain of 11.40%, 10.01%, and 10.97% in MAE, RMSE, and NRMSE, respectively. Further ablation study shows the effectiveness of our designed components. Our code and dataset are available[1].

CCS CONCEPTS

• **Information systems** → **Online shopping**; • **Applied computing** → *Online shopping*; • **Computing methodologies** → *Modeling methodologies.*

KEYWORDS

Community value prediction, graph neural networks, pooling for graph neural networks

[1]https://github.com/tsinghua-fib-lab/MSC

WWW '21, April 19–23, 2021, Ljubljana, Slovenia

ACM ISBN 978-1-4503-8312-7/21/04.
https://doi.org/10.1145/3442381.3449793

ACM Reference Format:
Guozhen Zhang, Yong Li, Yuan Yuan, Fengli Xu, Hancheng Cao, Yujian Xu, Depeng Jin. 2021. Community Value Prediction in Social E-commerce. In *Proceedings of the Web Conference 2021 (WWW '21), April 19–23, 2021, Ljubljana, Slovenia.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3442381.3449793

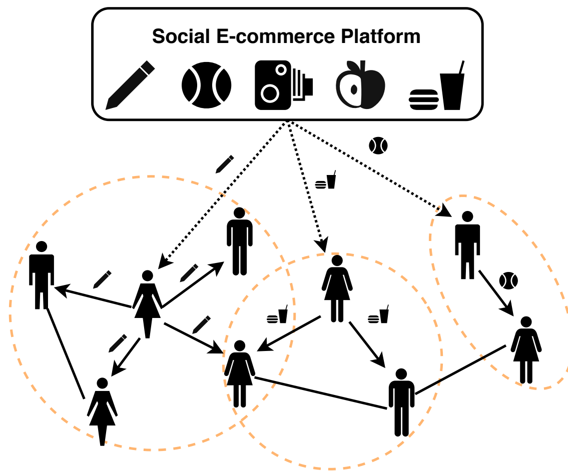

Figure 1: Illustration of how communities form in social e-commerce. The solid lines represent social networks, the arrows represent actions such as recommending an item or issuing a group buying invitation, and the orange circles are the boundaries of the formed communities.

1 INTRODUCTION

Social e-commerce is a newly-emerging form of e-commerce that utilizes social networks to promote online transactions [18]. Despite the early less successful attempts such as developing storefront sites inside social media (e.g., F-commerce, T-commerce), the immense success of several recently developed social e-commerce platforms that facilitate group buying (e.g., Pinduoduo[2], Groupon[3]) or promotes customer referrals (e.g., Beidian[4], Yunji[5]) have shown a promising future of social e-commerce and attracted extensive attention from both academia and industry [3, 4, 6, 34].

[2]https://www.pinduoduo.com
[3]https://www.groupon.com/
[4]https://www.beidian.com
[5]https://www.yunji.com

One common feature among these new social e-commerce platforms is that customers are often encouraged by rewards to express their item preferences on social media and recommend items to their relatives and friends [22]. As a result, customers' purchase decisions are easily affected by one another [20], and highly engaged communities around selling and buying are formed based on social networks, as illustrated in Figure 1. Take Beidian, a leading social e-commerce platform in China that promotes customer referrals as an example. Motivated by rewards, some users have developed the habit of sharing their favorite items with their social connections and some further become opinion leaders that strongly affect their friends' purchase decisions. Over time, people connected to these users form into communities that have a common purchase trend.

Assessing the economic value of such communities, defined as the total transaction amount inside the community, is of great importance to the prosperity of social e-commerce platforms because when people form into communities, their value is generally boosted, and thereby these platforms typically make their marketing campaigns towards communities and manage their resources around communities. This problem is also receiving increasing attention in research communities owing to the prevalence of online group activities and related business, which is not limited to social e-commerce [2]. For example, nowadays, a group of travelers can make their trip plan online together, and thus the key concern of related companies is shifting from an individual perspective to a group perspective. However, to the best of our knowledge, few works have systematically addressed this problem.

To fill this gap, we seek to develop an effective predictive model to forecast the value of given communities in social e-commerce. Compared to predicting each user's value, predicting community value is much more challenging since different levels of the underlying social networks can influence the value of communities. Specifically, we identify three levels of structure important. The first level is peer relationships. Existing theories have indicated a prominent effect of social homophily on customers' decision making processes [23]. For example, it is likely for one to purchase on the best friend's recommendations. Thus, the more tightly connected pairs in the community, the higher the community value. The second level is the community structure beyond the pair-wise relations. Take the edge density of a community as an example: the denser the network, the easier for social influence to diffuse across the community [21], and thus the higher the community value. Third, in addition to the community structure, inter-community connections can also be a crucial factor that influences community value. For example, high-value communities' purchase trends may pass to low-value communities through inter-community connections, thereby increasing their value. This phenomenon is also referred to as the strength of weak ties in social network literature [10].

In this work, we address the above challenges by developing a Multi-scale Structure-aware Community value prediction network, MSC for short, to jointly model the multi-scale structural information at the same time. It is built based on graph convolution networks (GCN) [13], a recent proposed state-of-the-art graph representation model. Specifically, we first develop a Masked Edge Learning Graph Convolution Network (MEL-GCN) based on our proposed masked propagation mechanism, which can efficiently learn the effects of peer relations on community value. In addition to the power of GCN to model the structural information itself, we propose a Pair-wise Community Pooling module (PCPool) to capture the critical community structure for prediction. Finally, to model the inter-community connections, we distinguish intra-community edges from inter-community edges by a Multi-aggregator Framework (MAF).

We highlight our contributions as follows:

- To the best of our knowledge, this paper formally proposed the community value prediction problem for the first time, and we provide in-depth analyses and empirical evidence demonstrating that modeling the multi-scale structure of communities is keenly important for this problem.
- To predict community value, we develop a deep learning framework, MSC, with three novel components, including MEL-GCN, PCPool, and MAF, which can jointly model peer relations, community structure, and inter-community connections at the same time.
- We conduct extensive experiments on a large-scale real-world dataset. The results show the superior performance of MSC compared with various types of state-of-the-art methods, with relative performance gains of at least 11.40%, 10.01%, and 10.97% in MAE, RMSE, and NRMSE, respectively. Further ablation study verifies the effectiveness of each designed components, and the case study and sensitivity study show that MSC provides robust and interpretable predictions.

2 OBSERVATION AND PROBLEM FORMULATION

2.1 Observation and Motivation

As we discussed before, community value is influenced by multi-scale structural information, including peer relations, community structure, and inter-community connections. As we seek empirical evidence supporting the theoretical analyses, we find that prior literature only provides empirical evidence of the effects of peer relations, while the impacts of community structure and inter-community connections are under-explored. Thus, we conduct data analysis on a large-scale dataset collected from Beidian (See Section 4.1.1 for details). The results are discussed as follows:

Peer Relations. Prior work on social e-commerce has already provided abundant empirical evidence on the non-negligible effects of peer relations on community value. For example, Xu et al. [34] demonstrate that when recommended by friends, the purchase conversion rate in social e-commerce is three to ten times higher than that of the traditional e-commerce scenarios. In other words, such peer influence can significantly increase community value.

Community Structure. We use the purchase amount of one month as a proxy of community value and visualize the correlations between the number of edges inside a community and the community value in Figure 2, controlling the size of the community. The results reveal a positive correlation between the number of edges inside the community and the community value. Note that the number of edges inside a community reflects the community's density when the community size is controlled. Therefore, the above results further validated the theory that the denser the network, the easier for social influence to diffuse, thereby the higher the

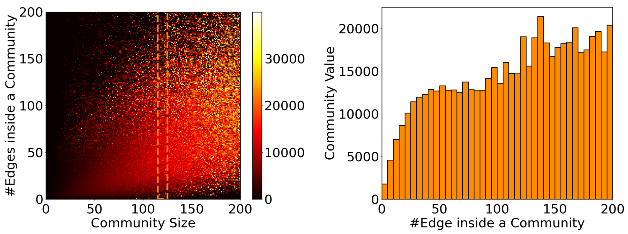

Figure 2: The correlation between community value and the number of edges inside a community. The right figure shows the details of the orange box area of the left figure. Note that the absence of data points causes the black part in the upper left of the left figure.

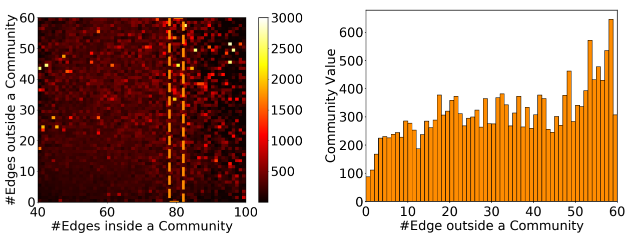

Figure 3: The correlation between community value and the number of edges outside a community. The right figure shows the details of the orange box area of the left figure.

community value. In conclusion, community structure is indeed vital for community value prediction.

Inter-community Connections. We also visualize the correlations between the number of edges outside a community and community value in Figure 3, controlling the number of edges inside a community. Here, we also controlled the community size by selecting only the communities with a size of 50. The results show a positive correlation between the number of edges outside the community and community value, which suggests inter-community connections affect community value.

The above observations further motivate us to model the multi-scale structural information.

2.2 Problem Statement

The goal of community value prediction is to predict the total purchase amount of given sets of users in a future time period. Specifically, it can be formally defined as follows: given a social network G, a user feature matrix $\boldsymbol{X_v}$, a user interaction feature matrix $\boldsymbol{X_e}$, and a set of communities C, the objective is to learn a mapping function to predict the value of each given community $\boldsymbol{y}$, which can be formulated as follows,

$$\boldsymbol{y} = F(G, \boldsymbol{X_v}, \boldsymbol{X_e}, C), \tag{1}$$

where $\boldsymbol{X_v} \in \mathbb{R}^{d_{v0}\times N}$, $\boldsymbol{X_e} \in \mathbb{R}^{d_{e0}\times K}$, $\boldsymbol{y} \in \mathbb{R}^M$, and $C = \{C_1, C_2, ..., C_M\}$ where C_i denotes subset of users, with d_{v0} and d_{e0} as the dimension of node features and edge features, respectively. Here, N, K, M is the number of users, user relations and communities, respectively.

In this paper, we model the social network G as a graph, with users modeled as nodes and user relations modeled as edges. Thus, we have $G = (\mathcal{V}, \mathcal{E})$ with $|\mathcal{V}| = N$, $|\mathcal{E}| = K$, and its adjacency matrix $\boldsymbol{A} \in \mathbb{R}^{N\times N}$. We use the sum of purchase of all the members in each community in a future period as a proxy of community value.

3 MODEL FRAMEWORK

As shown in Figure 4, our proposed MSC model consists of four layers with three key components: (1) the Masked Edge Learning Graph Convolutional Network (MEL-GCN), which models peer relations by a novel graph convolutional network that leveraging both the node features and edge features to learn a mask vector that controls the propagation step of GCN, (2) the Pair-wise Community Pooling Module (PCPool), which facilitates the GCN-based model to capture the key community structure by a novel pair-wise pooling mechanism to map the node embeddings learned by MEL-GCN to community embeddings for prediction, (3) the Multi-aggregator Framework (MAF), which models the inter-community connections by distinguishing intra-community edges from inter-community edges. In the following sections, we elaborate on the details of the above key components of our designed framework.

3.1 Masked Edge Learning GCN

To capture the community value that mainly resides in one's social network in social e-commerce, our first step is to model peer relations. It is a challenging task due to the complex nature of social influence. For instance, for close friends who did not get in touch for long, their relationship may still contain great value. Thus, we need to take both the node features and edge features into consideration and jointly learn the relation between each pair of users.

Graph Convolutional Network (GCN) is a natural choice for modeling the above problem. However, the original GCN model cannot effectively utilize the edge features. To tackle this problem, most prior work either treats edges as new nodes [1] or learns a single edge weight as the adjacency matrix [13]. Both approaches lack a fine-grained granularity in leveraging the edge features. Motivated by the shortcomings of existing work, we proposed MEL-GCN, which modifies the propagation step of GCN to model the peer relations with a fine-grained granularity. We illustrate the details of this mechanism in Figure 5. Specifically, MEL-GCN contains two modules, namely the Edge Learning module and the Masked Edge Learning Convolutional (MELConv) module. The Edge Learning module takes the node embeddings $\boldsymbol{H_v} = [\boldsymbol{h_{v_i}}, \boldsymbol{h_{v_2}}, ..., \boldsymbol{h_{v_N}}]$, edge embeddings $\boldsymbol{H_e} = [..., \boldsymbol{h_{e_{ij}}}, ...]$, $((i,j) \in \mathcal{E})$ from the embedding layer, along with the adjacency matrix $\boldsymbol{A}$ as inputs. For each pair of connected nodes, it learns a mask vector leveraging both the node embeddings $\boldsymbol{h_{v_i}}$ and $\boldsymbol{h_{v_j}}$, and the corresponding edge embeddings $\boldsymbol{h_{e_{ij}}}$, which can be formulated as follows,

$$\boldsymbol{\epsilon_{ij}} = \boldsymbol{\Theta_\epsilon^{(1)}} \sigma\left(\boldsymbol{\Theta_\epsilon^{(0)}}\left(\boldsymbol{h_{v_i}} || \boldsymbol{h_{v_j}} || \boldsymbol{h_{e_{ij}}}\right) + \boldsymbol{b_\epsilon^{(0)}}\right) + \boldsymbol{b_\epsilon^{(1)}}, \tag{2}$$

where $(\cdot||\cdot)$ denotes concatenation, $\sigma(\cdot)$ is a non-linear activation function, and we adopt ReLU [24] in our implementation. $\boldsymbol{\Theta_\epsilon^{(0)}} \in$

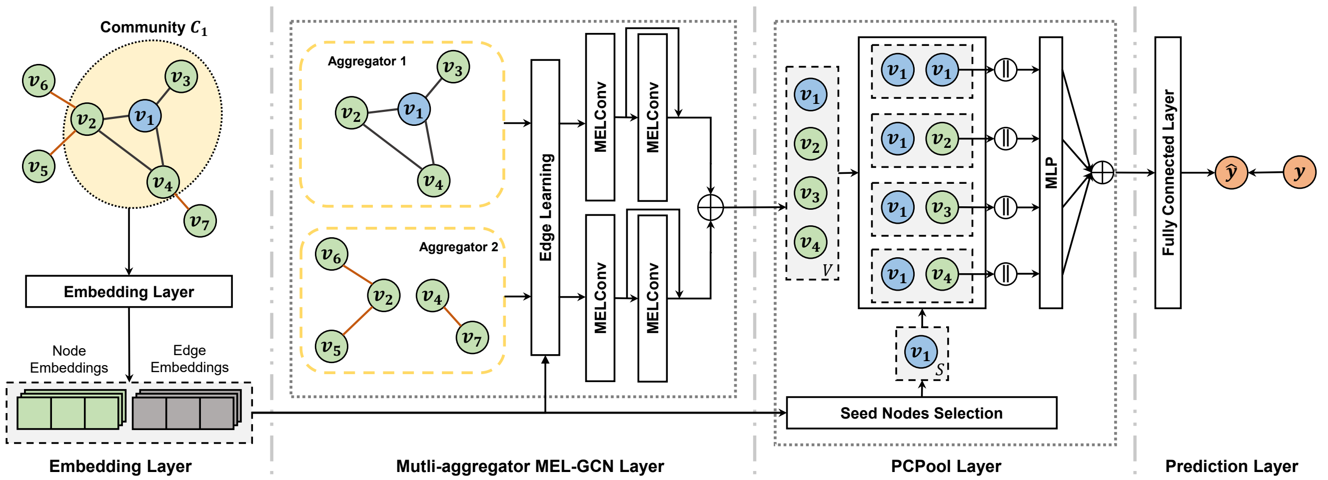

Figure 4: The architecture of our proposed Multi-scale Structure-aware Community Value Prediction Network (MSC). Here, we take the prediction of a demo community as an example. The multi-aggregator MEL-GCN layer takes the node embeddings, edge embeddings, and the adjacency matrix of community C_1 as inputs, and learns a node representation for each node by a masked edge learning graph convolutional network with different aggregators for intra-community edges and inter-community edges. The PCPool layer transforms the learned node representations to community representations for prediction through a pair-wise pooling mechanism controlled by the selected seed nodes that introduce prior knowledge.

$\mathbb{R}^{d_\epsilon \times (2d_v + d_e)}$, $\Theta_\epsilon^{(1)} \in \mathbb{R}^{d_\epsilon \times d_\epsilon}$, $b_\epsilon^{(0)}, b_\epsilon^{(1)} \in \mathbb{R}^{d_\epsilon}$ are model parameters. d_v, d_e, d_ϵ represent the dimension of node embeddings, edge emgeddings and MELConv output, respectively. $\epsilon_{ij} \in \mathbb{R}^{d_\epsilon}$ is the mask vector learned for the propagation step of GCN, and it is worth noting that $\epsilon_{ij} \neq \epsilon_{ji}$. Intuitively, ϵ_{ij} characterizes how a user is influenced by its neighbors. This influence can either be positive or negative, thereby the value of the mask vector is not restricted in $[0, 1]$. After learning such a mask vector for each edge in the graph, we feed them into the MELConv module. For each node, MELConv updates the hidden states of layer l $h_{v_i}^l$ by aggregating of all its neighbors controlling by the learned masked vector, which can be formulated as follows,

$$h_{v_i}^{l+1} = \sigma\left(\Theta_h^l h_{v_i}^l + \sum_{j \in \mathcal{N}(i)} (\Theta_h^l h_{v_j}^l) \odot (P_\epsilon^l \epsilon_{ij})\right), \tag{3}$$

where $\odot$ refers to element-wise multiplication. $\mathcal{N}(i)$ is the set that contains the neighbors of user node i. $\Theta_h^l \in \mathbb{R}^{d_{l+1} \times d_l}$ is the model parameter, where d_l is the hidden dimension of l layer of MELConv. $P_\epsilon^l \in \mathbb{R}^{d_{l+1} \times d_\epsilon}$ is the dimension transformation matrix designed to adapt the dimension of the mask vector to the output dimension of layer l, which is also a learnable parameter. In our framework, we stack two MELConv layers and add an readout operation after each MELConv to enhance the representation power of our model.

To sum up, MEL-GCN models peer relations as a mask vector that controls the propagation step of GCN and outputs the learned representations of each node.

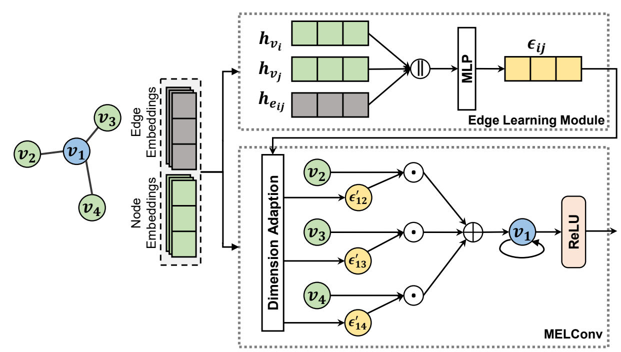

Figure 5: An illustration of the framework of MEL-GCN.

3.2 Pair-wise Community Pooling

The community value prediction task is essentially a graph prediction task. Thus, after we obtain node-level representations from MEL-GCN, we still need to learn a pooling function to map the node-level representations to graph-level representations of each community for prediction.

As illustrated in Section 2, the community structure is keenly important for predicting community value in social e-commerce. However, it is difficult for GCN with the state-of-the-art pooling methods to accurately capture the high-order structure information of a graph [17]. To tackle this challenge, we design a novel pairwise community pooling method, PCPool for short, to facilitate our model to capture the critical community structure for prediction. The main idea is to integrate the prior knowledge into the pooling process as a supervised signal. Specifically, our method consists of two steps. The first step is to select a set of most important nodes $\mathcal{S}$ utilizing the prior knowledge. In social e-commerce, we can reasonably assume that users with more connections are more important for community. Thus, we select the top $r\%$ of nodes with the highest degrees as seed nodes. Here, r is a hyper-parameter of the model. Note that this selection algorithm can be easily extended to other complicated ones. This step can be formally formulated as follows,

$$\mathcal{S} = \{i | i \in \text{rank}(\text{degree}(V, E), \lceil rN \rceil)\}, \tag{4}$$

where $rank(\cdot, k)$ is the function that returns the indices of the top k value, $\lceil\cdot\rceil$ is the ceiling function, and $degree(V, E)$ is the function that returns the degree of each nodes in a given graph G. Given the set of seed nodes $\mathcal{S}$ and the output node representations $\boldsymbol{h}^{g}_{v_i}$ from MEL-GCN, the seccond step of PCPool is to conduct a pair-wise concatenation operation between the seed nodes and all the nodes in the graph. Then, we feed the results to a linear layer followed by a mean pooling to learn the representations of community $\boldsymbol{h}^{c}_{m}$, $(m \in C)$, formulated as follows,

$$\boldsymbol{h}^{c}_{m} = \frac{1}{N_m \times S} \sum_{(i,j) \in \mathcal{V}_m \times \mathcal{S}} \sigma\left(\boldsymbol{\Theta_c}\left(\boldsymbol{h}^{g}_{v_i} || \boldsymbol{h}^{g}_{v_j}\right) + \boldsymbol{b_c}\right), \tag{5}$$

where N_m, S are the number of nodes in community m and the number of seed nodes, respectively; $\boldsymbol{\Theta_c} \in \mathbb{R}^{d_c \times 2d_g}$ and $\boldsymbol{b_c} \in \mathbb{R}^{d_c}$ are the model parameters with d_g, d_c as the dimension of the output of MEL-GCN and the output of PCPool, respectively. Intuitively, PCPool can capture the important community structure by leveraging the prior knowledge to identify important nodes and learning the important relationships between nodes.

3.3 Multi-Aggregator Framework

As we illustrated in Section 2, inter-community connections are also a crucial factor that affects the value of a community. Even though the users between community could be weakly connected, this impact of such connection is non-negligible for its ability of diffusing the purchase trend from one community to another.

To effectively model the inter-community connections, we have two main designs. First, we take the inter-community edges into consideration and formulated the problem as a subgraph prediction task. Second, we noticed the fact that a normal GCN cannot differentiate the edges inside community with the edges outside the community, yet there are fundamental differences between the effects of inter-community edges and intra-community edges on community value. As such, we propose to use different GCN aggregators for neighbor nodes inside community and outside community, and thus Equation (3) can be rewritten as follows,

$$\begin{aligned}
\boldsymbol{h}^{l+1}_{va_i} &= \sigma\left(\Theta^{l}_{ha}\boldsymbol{h}^{l}_{va_i} + \sum_{j \in \mathcal{N}_a(i)} (\Theta^{l}_{ha}\boldsymbol{h}^{l}_{va_j}) \odot (P^{l}_{\epsilon a}\epsilon_{ij})\right), \\
\boldsymbol{h}^{l+1}_{vo_i} &= \sigma\left(\Theta^{l}_{ho}\boldsymbol{h}^{l}_{vo_i} + \sum_{j \in \mathcal{N}_o(i)} (\Theta^{l}_{ho}\boldsymbol{h}^{l}_{vo_j}) \odot (P^{l}_{\epsilon o}\epsilon_{ij})\right), \\
\boldsymbol{h}^{l+1}_{v_i} &= \boldsymbol{h}^{l+1}_{va_i} + \boldsymbol{h}^{l+1}_{vo_i},
\end{aligned} \tag{6}$$

where $\mathcal{N}_a(i)$ and $\mathcal{N}_o(i)$ are the sets that contain the intra-community neighbors and inter-community neighbors of node i, respectively; $\boldsymbol{h}^{l+1}_{va_i}$ and $\boldsymbol{h}^{l+1}_{vo_i}$ are the output of the GCN aggregators for intra-community neighbor nodes and inter-community neighbor nodes, respectively. In this way, we can effectively capture the different effects of inter-community connections on community value.

3.4 Inputs, Outputs and Training

Inputs and the Embedding Layer. The embedding layer takes the raw features, including node features $\boldsymbol{X_v}$ that contains user demographics and users' history purchase amount and edge features $\boldsymbol{X_e}$ that contains the history interaction times of each pair of users as inputs and transforms them into nodes embeddings $\boldsymbol{H_v}$ and edge embeddings $\boldsymbol{H_e}$ with two fully connected layers, respectively, which can be formulated as follows,

$$\begin{aligned}
\boldsymbol{H_v} &= \sigma\left(\boldsymbol{\Theta^{(1)}_v}\sigma\left(\boldsymbol{\Theta^{(0)}_v}\boldsymbol{X_v} + \boldsymbol{b^{(0)}_v}\right) + \boldsymbol{b^{(1)}_v}\right), \\
\boldsymbol{H_e} &= \sigma\left(\boldsymbol{\Theta^{(1)}_e}\sigma\left(\boldsymbol{\Theta^{(0)}_e}\boldsymbol{X_e} + \boldsymbol{b^{(0)}_e}\right) + \boldsymbol{b^{(1)}_e}\right),
\end{aligned} \tag{7}$$

where $\boldsymbol{\Theta^{(0)}_v} \in \mathbb{R}^{d_v \times d_{v0}}$, $\boldsymbol{\Theta^{(0)}_e} \in \mathbb{R}^{d_e \times d_{e0}}$, $\boldsymbol{\Theta^{(1)}_v} \in \mathbb{R}^{d_v \times d_v}$, $\boldsymbol{\Theta^{(1)}_e} \in \mathbb{R}^{d_e \times d_e}$, $\boldsymbol{b^{(0)}_v}$, $\boldsymbol{b^{(1)}_v} \in \mathbb{R}^{d_v}$, $\boldsymbol{b^{(0)}_e}$, $\boldsymbol{b^{(1)}_e} \in \mathbb{R}^{d_e}$ are model parameters. The outputs of embedding layer are feed into the Mutli-aggregator MEL-GCN Layer and PCPool Layer for follow-up operations.
Outputs. The prediction layer takes community embeddings $\boldsymbol{H_p} \in \mathbb{R}^{d_p \times M}$ from the PCPool layer as inputs to predict community value. Here, we use two fully connected layers to predict labels, which is a widely-adopted framework[28]. It can be formulated as follows,

$$\hat{\boldsymbol{y}} = \sigma\left(\boldsymbol{\Theta_{fc}}\boldsymbol{H_p} + \boldsymbol{b_{fc}}\right)\boldsymbol{p}, \tag{8}$$

where $\boldsymbol{\Theta_{fc}} \in \mathbb{R}^{d_p/2 \times d_p}$, $\boldsymbol{b_{fc}} \in \mathbb{R}^{d_p/2}$, $\boldsymbol{p} \in \mathbb{R}^{d_p/2}$ are model parameters. Note that this prediction function can be extended to other complicated ones.
Training. After we obtain the predictions, we use the widely-adopted mean absolute error loss function[9] with l_2 regularization on the parameters in MSC to prevent over-fitting, which can be formulated as follows:

$$\mathcal{L} = \frac{1}{M}\sum |\hat{\boldsymbol{y}} - \boldsymbol{y}| + \lambda \sum_{\Theta \in \mathcal{P}} \|\Theta\|_2, \tag{9}$$

where $\boldsymbol{y}$ and $\hat{\boldsymbol{y}}$ are ground truths and model predictions, respectively; $\mathcal{P}$ denotes the set that contains all model parameters. We use MAE loss rather than MSE loss because the standard deviation of community value in our dataset is large, and MAE is more stable to outliers.

4 EXPERIMENTS

To comprehensively evaluate our proposed model, we conduct extensive experiments on a large-scale real-world dataset to answer the following research questions:

- **Q1**: How is the overall prediction performance of MSC compared with various state-of-the-art methods?
- **Q2**: How do different components of MSC, including MEL-GCN, PCPool, and MAF, contribute to the performance?

Statistics	Value
The number of communities	1500
The average number of nodes inside community	56.4
The number of nodes	76649
The number of edges	153823
The number of edges inside community	119302
The number of edges outside community	34521
Average node degree	4.013
Average degree centrality of communities	0.968
Average clustering coefficient of communities	0.267

Table 1: The basic statistics of the dataset.

- **Q3**: Is MSC robust to different community sizes and different community overlap ratios?
- **Q4**: Can MSC provides a certain level of interpretability and capture the key structural information that affects the community value?
- **Q5**: How do different hyper-parameter settings affect the performance of MSC?

4.1 Experiment Setup

4.1.1 ***Datasets.*** We evaluate our proposed MSC based on a large-scale real-world dataset collected from a leading social e-commerce platform in China, Beidian. To the best of our knowledge, we are the first to study the community value prediction problem in social e-commerce, and thereby no public datasets are available. We make our dataset public to motivate future researches[1].

Our dataset covers all the user relations with interaction data, user demographic data, purchase data on the platform from 11/2018 to 12/2019, and the community partition of Beidian. Specifically, the interaction data records the number of times that users share items with others. The user demographic data includes each user's age, gender, status, and registration time. The community partition is directly adapted from Beidian's official version, which is naturally defined by Beidian's social e-commerce business model that focuses on active sharing users or key opinion leaders. Those that usually interact with these users are defined as communities in Beidian. Note that in practice, we can not only obtain communities by naturally defined rules in application scenarios but also by existing community detection algorithms based on social network structures and user interaction histories [29].

The statistics of this dataset are reported in Table 1. In our experiments, we use data from 11/2018 to 11/2019 to predict the community value in 12/2019. We use the user purchase data to compute the label of each community and get a result with a mean of 808.2 and a standard deviation of 1372.9.

4.1.2 ***Baseline Methods.*** To the best of our knowledge, we are the first to leverage the network structure information to model the community value, and thereby no previous methods can be directly applied to the problem. Thus, we compared the performance of MSC with the state-of-the-art methods from four research lines with minimum modification to adapt them to the problem.

Traditional Machine Learning Methods: Prior work on predicting customer value usually designs various hand-crafted rules to extract user-specific features and feed these features into regression models for prediction. This method is also widely used in the industrial environment. We adopt three baselines, including:

- Random Forest (RF) [30]: An ensemble learning method based on decision trees.
- Support Vector Regression (SVR) [33]: A classical supervised learning model.
- Light Gradient Boosting Machine (LGBT) [14]: A recently proposed state-of-the-art tree-based machine learning model.

For a fair comparison, besides user-specific features used in prior work [30], we also extract various network-specific features, including degree, the number of edges and triangles inside each community, network average clustering coefficient [32] and network average degree centrality [25] as the inputs of these baselines.

Network Embedding Methods: We also compare our model with the state-of-the-art network embedding models, including DeepWalk [26] and Node2Vec [11], which are briefly introduced as follows:

- DeepWalk [26]: An unsupervised node embedding model based on the random walk and skip-gram algorithms. Nodes sharing lots of links have similar embeddings.
- Node2Vec [11]: A generalization of DeepWalk that balances between network homophily and structural equivalence.

To adapt network embedding models to our problem, we take three steps. First, we pre-train the network embedding model to learn node representations of each node in a community. Second, we concatenate the learned node embeddings with feature embeddings learned from the embedding layer of MSC, and obtain community embeddings via a mean pooling operation. Finally, we feed the community embeddings into two fully connected layers for prediction, which is the same as MSC.

GCN-based Methods: We also compare our model with state-of-the-art GCN models with different pooling methods. For this type of methods, the input and prediction layers are the same as our MSC model for a fair comparison. We introduce methods for comparison as follows,

- GCN [13]: A current state-of-the-art variant of GCN that can efficiently generate node embeddings. A mean pooling method is used to obtain community embeddings for prediction.
- GCN + gPool [8]: gPool is one of the state-of-the-art pooling methods for GCN. We use it to obtain community embeddings in this method.
- GCN + Self-Attention Graph Pooling (SAGPool) [19]: SAGPool is an attention-based pooling algorithm for GCN, which can learn a soft assignment of node embeddings to graph embeddings. We use SAGPool to obtain community embedding in this method.

Customer Value Prediction Methods: In this work, we take the sum of the purchase amount as a proxy of community value. Thus, we can compare our model with its variants that first predict each customer's purchase amount and then sum up the results of

Group	Model	MAE	Gain	RMSE	Gain	NRMSE	Gain
Traditional ML	RF [30]	482.97 ± 21.02	−15.09%	911.14 ± 43.38	−10.01%	0.099 ± 0.009	−19.27%
	SVR [33]	483.19 ± 28.85	−15.15%	939.13 ± 42.57	−13.39%	0.106 ± 0.011	−27.71%
	LGBM [14]	475.12 ± 21.45	−13.22%	918.01 ± 38.26	−10.84%	0.101 ± 0.010	−21.68%
Network Embedding	DeepWalk [26]	510.55 ± 22.17	−21.67%	1077.63 ± 62.81	−30.11%	0.151 ± 0.013	−81.92%
	Node2Vec [11]	499.05 ± 23.10	−18.93%	991.43 ± 51.34	−19.70%	0.098 ± 0.009	−18.07%
GCN-based Models	GCN [13]	469.42 ± 27.54	−11.87%	936.59 ± 49.95	−13.08%	0.091 ± 0.006	−10.97%
	GCN + gPool [8]	467.45 ± 22.38	−11.40%	946.22 ± 54.89	−14.24%	0.092 ± 0.006	−12.19%
	GCN + SAGPool [19]	474.89 ± 21.44	−13.17%	941.27 ± 53.92	−13.64%	0.091 ± 0.006	−10.97%
Customer Value	CVP	516.25 ± 21.81	−23.03%	962.76 ± 59.79	−16.24%	0.121 ± 0.008	−45.78%
Ours	MSC	**419.61 ± 15.30**	-	**838.23 ± 34.24**	-	**0.082 ± 0.003**	-

Table 2: Performance comparison with different categories of baseline models in terms of MAE, RMSE, and NRMSE.

users inside each community to get community value. We implement this method by deleting our PCPool layer in MSC and directly send the node embeddings that output from the MEL-GCN layer to a fully connected layer for customer value prediction without pooling. We denote this model by CVP.

4.1.3 ***Evaluation Protocols.*** In our experiments, we perform 10-fold cross-validation on our dataset following prior work [5]. We evaluate the prediction results by widely used metrics in regression tasks, including Mean Absolute Error (MAE), Root Mean Square Error (RMSE), and Normalized Root Mean Square Error (NRMSE). Specifically, we include MAE due to its robustness to outliers, and NRMSE to show the relative performance of the models.

4.1.4 ***Parameter Settings and Reproducibility.*** In our experiments, we use a node embedding size of 40 and an edge embedding size of 20. For all GCN-based layers, including normal GCN and our proposed MELConv, we stacked two layers with the size of hidden nodes set to 32. For each GCN-based layer's output, we perform a readout operation that saves each node's hidden state, and we concatenate the saved results of every GCN-based layer as the input of the prediction layer. To prevent overfitting, we add a dropout operation before the prediction layer. For MSC, we set the selection ratio of pair-wise community pooling as 0.2. During the training process, we used the Adam optimizer [15] for gradient-based model optimization in a mini-batch mode, and we perform a grid search on learning rate, batch size, l_2 regularization coefficient, and dropout rate to find the best parameters for each model including all the baselines. Specifically, we search the learning rates $\in \{0.025, 0.02, 0.015, 0.01, 0.005, 0.001\}$, batch sizes $\in \{45, 54, 75, 90, 135\}$, l_2 regularization coefficients $\in \{1e-3, 5e-4, 1e-4, 5e-5, 1e-5, 5e-6\}$, and dropout rates $\in \{0.4, 0.5, 0.6\}$. We make our implementation code of the MSC public[1].

4.2 Overall Performance Comparison (Q1)

Table 2 shows the prediction performance of all categories of methods on our dataset with relative performance gains. We summarize key observations and insights as follows:

- MSC achieves significantly better performance over different types of state-of-the-art methods in terms of all three evaluation metrics. Specifically, it provides a relative performance gain of 11.40%, 10.01%, and 10.97% in MAE, RMSE, and NRMSE, respectively, comparing to the best baselines, which demonstrates the effectiveness of our proposed methods.
- Among all the baselines, we find that all GCN-based methods, including MSC, GCN, GCN+gPool, and GCN+SAGPool, perform consistently better than others. On the one hand, this observation indicates the promising power of GCN in capturing the hidden patterns of non-Euclidean data. On the other hand, it also suggests predicting the value of community needs to model the community's network structure and individual attributes jointly.
- One interesting observation is that among GCN-based methods, prior state-of-the-art pooling methods, including gPool and SAGPool, perform worse than mean pooling. The reason lies in two aspects. First, as demonstrated in [17], these pooling methods are not suitable for capturing the high-order structural information of a community, which is the key to predicting community value. Second, these pooling methods are selection-based pooling methods. In other words, these methods only select a part of the nodes to compute the community embeddings and thereby may result in information loss.
- Although MSC is also a GCN-based model, it achieves consistent better performance than GCN with mean pooling, which suggests that our novel designs, including MEL-GCN, PCPool, and MAF, indeed facilitate the original GCN model to better characterize the key structure for community value prediction.
- MSC performs significantly better than MVP, which suggests that summing up the prediction of customer value to get community value results in an accumulation of errors, and the community value prediction problem is essentially different from the customer value prediction problem.

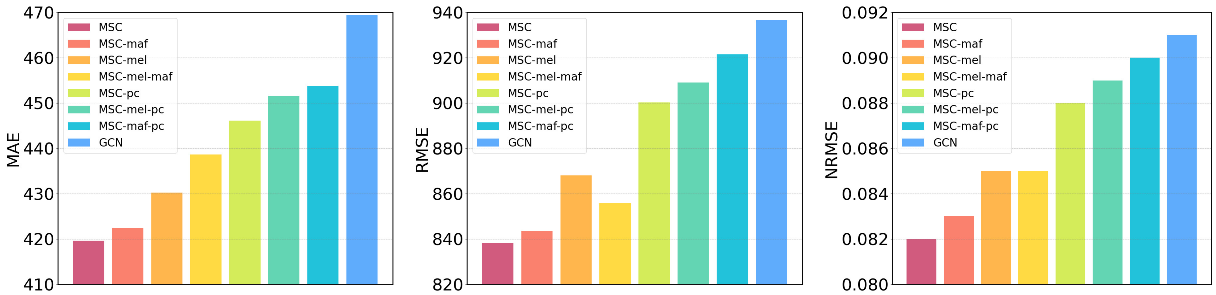

Figure 6: Ablation study with six variants of MSC across different metrics.

4.3 Ablation Study (Q2)

Our proposed model MSC consists of three key components, namely MEL-GCN, PCPool, and MAF. To better understand how each component contributes to the model performance, we conduct an ablation study. Specifically, we consider six variants of our proposed method, including *MSC-mel, MSC-pc, MSC-maf, MSC-mel-maf, MSC-mel-pc, MSC-maf-pc*. Here, *mel* stands for the MEL-GCN, *pc* stands for PCPool, and *maf* stands for the MAF. For the six models mentioned above, having a specific suffix name means substituting the corresponding MSC module with a simplified one. In particular, the simplified version of MEL-GCN, PCPool, and MAF are normal GCN propagation, mean pooling, and one aggregator for all kinds of nodes, respectively. The evaluation results are reported in Figure 6, and we have the following three key observations:

- The full version of MSC achieves the best performance across all evaluation metrics. Removing any of the components results in a certain level of decrease of the performance, which suggests that all the components are effective in characterizing the structure of different scales.
- Each combination of the three components has a performance gain, which suggests the three components capture the structural information that affects community value from different perspectives.
- Among the three key components, PCPool is the most effective one for predicting the community value. Specifically, removing PCPool from MSC (i.e., *MSC-pc*) results in a performance degradation of 6.32%, 8.69%, and 7.31% in MAE, RMSE, and NRMSE, respectively. It is even worse than removing both MEL-GCN and multi-aggregator from MSC. This observation indicates that pair-wise community pooling can effectively capture the useful community structure that informs community value, and community structure is more informative for community value prediction, comparing to peer relations and inter-community connections.

4.4 Robustness Analysis (Q3)

In practice, the model's robustness determines its applicability. Thus, evaluating whether the performance of our model is consistently better than baseline methods is also valuable. In the community value prediction problem, we identify two factors keenly

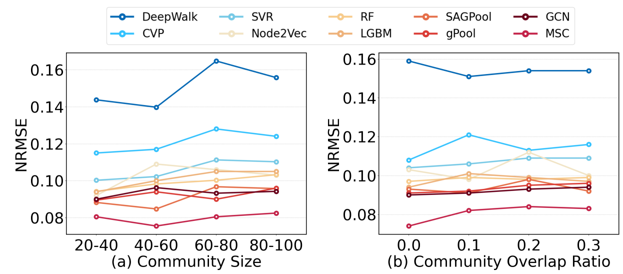

Figure 7: Performance comparison on datasets with different community sizes and different community overlap ratios.

important for robustness, including the community size and community overlap ratio.

Communities with different sizes differ dramatically in their value and structure, and we can plausibly assume that how different structural information contributes to community value is different for communities with very different sizes. To examine the robustness of our model's performance to community size, we divide the model outputs on the test set into different groups according to the community size and calculated NRMSE for different groups. As shown in Figure 7(a), our model performs consistently better than all the baseline methods for communities of different sizes, and the performance is stable across different community sizes, which demonstrates the robustness of our model.

In practice, communities defined by rules or detected by algorithms may differ significantly in their overlap ratios, defined as the ratio of overlap nodes of communities in all nodes. Thus, whether the model is robust to different community overlap ratios greatly affects its applicability. To examine this problem, we first construct four different datasets with different community overlap ratios based on the original dataset. Specifically, the community overlap ratio for our original dataset is 11%. By assigning the nodes in multi communities to the community they interact with the most, we get two new dataset with lower community overlap ratios. By iteratively assigning the node-pairs connected by the most interacted inter-community edges to an additional community, we get two new datasets with higher community overlap ratios. We test all the models on the four constructed datasets and show the results

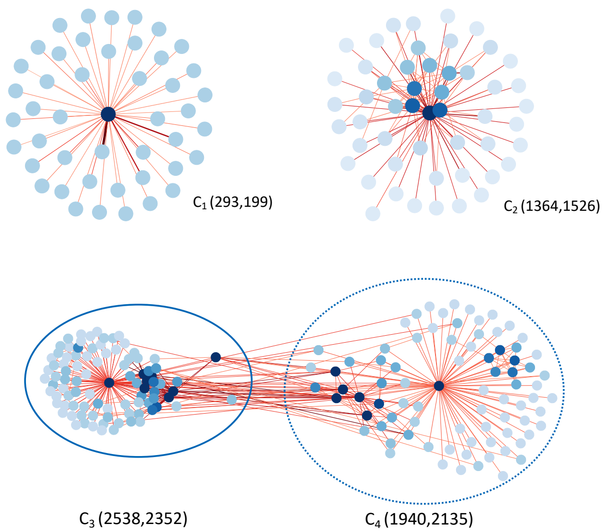

Figure 8: Case study of the edge learning weights of MEL-GCN, where the numbers in parentheses represent the prediction value and true value of the community.

in Figure 7(b). We find that MSC performs consistently better in datasets with different community overlap ratios, which indicates that our proposed method is robust in vast practical scenarios.

4.5 Case Study (Q4)

To validate whether our proposed model can provide a certain degree of interpretability, we conduct a case study on the learned weights of the mask vectors in MEL-GCN, which characterizes the influence of a user on its neighbors. We calculate the mean value of the mask vectors as the weight of the edges and visualized the community structure of four communities, as shown in Figure 8, where the color of the edge represents the strength of the influence, and the color of the node reflects its degree. The darker the color, the stronger the influence, the higher the degree.

The results show that the predictions can be well explained. Specifically, the value of community C_1 is low because most of its users are loosely connected, and it lacks users with strong influence. In contrast, for the densely connected community C_2, MSC identifies many users with strong influence. In terms of C_3 and C_4, we find that although C_4 is loosely connected, there are many strong connections between it and a high-value community C_3, which suggests the high value of C_3 may result from the strong influence of C_4. These results show the ability of MSC to provide interpretations for the prediction results.

5 HYPER-PARAMETER SENSITIVITY ANALYSIS (Q5)

To explore how MSC's performance is affected by different hyper-parameters, we further study the sensitivities of several key hyper-parameters by varying them in different scales. Specifically, we investigate how the node embedding size d_v and the seed ratio r of

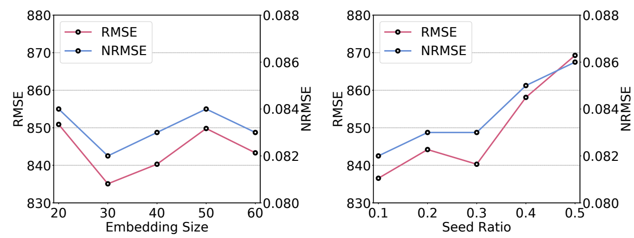

Figure 9: Hyper-parameter sensitivity analysis on the node embedding size and the seed ratio of PCPool.

the PCPool module affect the prediction performance. As shown in Figure 9, among all the settings, $d_v = 30, r = 0.1$ achieves the best performance. Moreover, we find that there is a trend that the performance decreases as the seed ratio increases. The reason is that in our model, the seed ratio controls the prior information that we introduce to the model as a supervised signal. When it is large, this signal becomes noisy and meaningless.

6 RELATED WORK

Customer Value Prediction. Community value prediction is closely related to the long-standing problem of customer value prediction [5, 12], yet exhibits significant differences. These two problems are both concerned with the financial benefits an entity can bring to a company. However, Customers' value mainly refers to their own purchase and can be inferred from their past behaviors. In contrast, the value of a community is more about the marginal gain that results from users' different organizational forms, and it is closely related to the community's underlying social network structure. This work focuses on designing a deep learning framework that can effectively model different structure information for community value prediction.

Graph Convolutional Networks. The recent proposed graph convolutional network (GCN) has successfully adapted the representation power of deep neural networks to graph-structured data [16] and achieved state-of-the-art performance in various network learning tasks [27, 31, 36]. Its core idea is to iteratively update the state of each node according to the states of their neighbors. However, common GCNs cannot effectively model edge features. To tackle this problem, previous work either uses the edge attributes to learn a weight as the adjacency matrix [13], which lacks fine-grained granularity, or treats edges as new nodes [1], which lacks interpretability. How to better utilize the edge features for community value prediction is one of the key concerns of this work, and we developed a Masked Edge Learning Graph Convolutional Network based on a novel masked propagation mechanism that efficiently utilized the edge features to model peer influence.

Pooling Techniques for GCNs. To transfer the node embeddings to community embeddings, we need pooling operations. Existing pooling methods for GCNs can be categorized into clustering-based pooling methods and selection-based pooling methods. The core idea of clustering-based pooling methods, including DiffPool [36] and EdgePool [7], is to assign nodes to a set of clusters, which is often computationally expensive. Selection-based pooling methods, including gPool [8] and SAGPool [19], use a top-K node selection procedure to form an induced subgraph, which may lose the original graph structure information. Moreover, it is difficult for existing pooling methods to effectively capture the high-order structure information critical for community value. This work develops a novel pair-wise pooling method to better capture the high-order graph structure information for community value prediction.

7 CONCLUSION

In this work, we propose the community value prediction problem for the first time, and we present a GCN-based framework MSC to effectively capture the multi-scale structural information to address this problem in social e-commerce. Specifically, MSC addresses the challenges with three novel components: (1) a masked edge learning graph convolutional network that explicitly characterizes peer relations; (2) a pair-wise community pooling module that effectively captures the critical high-order community-level structural information; (3) a multi-aggregator framework that enables us to model the inter-community connections. Extensive experiments on a large-scale real-world social e-commerce dataset show the superior performance of our proposed MSC.

We open our dataset and call for more attention to the community value prediction problem. In future, it is valuable to explore the community value prediction problem on industry scale datasets. Note that while in this work, we adopted GCN-based architecture, it is possible for MSC to extend to industry scale dataset with neighbor sampling technique [35]. Further, it is also valuable to explore how to extend MSC to dynamic communities that change over time.

ACKNOWLEDGMENTS

This work was supported in part by the National Natural Science Foundation of China under U1936217, 61971267, 61972223, 61941117, 61861136003, and Beijing Natural Science Foundation under L182038.

REFERENCES

[1] James Atwood and Don Towsley. 2016. Diffusion-convolutional neural networks. In *Advances in neural information processing systems*. 1993–2001.

[2] Da Cao, Xiangnan He, Lianhai Miao, Yahui An, Chao Yang, and Richang Hong. 2018. Attentive group recommendation. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 645–654.

[3] Hancheng Cao, Zhilong Chen, Mengjie Cheng, Shuling Zhao, Tao Wang, and Yong Li. 2020. You Recommend, I Buy: How and Why People Engage in Instant Messaging Based Social Commerce. *arXiv preprint arXiv:2011.00191* (2020).

[4] Hancheng Cao, Zhilong Chen, Fengli Xu, Tao Wang, Yujian Xu, Lianglun Zhang, and Yong Li. 2020. When Your Friends Become Sellers: An Empirical Study of Social Commerce Site Beidian. In *Proceedings of the International AAAI Conference on Web and Social Media*, Vol. 14. 83–94.

[5] Benjamin Paul Chamberlain, Angelo Cardoso, CH Bryan Liu, Roberto Pagliari, and Marc Peter Deisenroth. 2017. Customer lifetime value prediction using embeddings. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 1753–1762.

[6] Zhilong Chen, Hancheng Cao, Fengli Xu, Mengjie Cheng, Tao Wang, and Yong Li. 2020. Understanding the Role of Intermediaries in Online Social E-commerce: An Exploratory Study of Beidian. *Proceedings of the ACM on Human-Computer Interaction* 4, CSCW2 (2020), 1–24.

[7] Frederik Diehl, Thomas Brunner, Michael Truong Le, and Alois Knoll. 2019. Towards graph pooling by edge contraction. In *ICML 2019 Workshop on Learning and Reasoning with Graph-Structured Data*.

[8] Hongyang Gao and Shuiwang Ji. 2019. Graph u-nets. *arXiv preprint arXiv:1905.05178* (2019).

[9] Aritra Ghosh, Himanshu Kumar, and PS Sastry. 2017. Robust loss functions under label noise for deep neural networks. In *Thirty-First AAAI Conference on Artificial Intelligence*.

[10] Mark Granovetter. 1983. The strength of weak ties: A network theory revisited. *Sociological theory* (1983), 201–233.

[11] Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*. 855–864.

[12] Sunil Gupta, Dominique Hanssens, Bruce Hardie, Wiliam Kahn, V Kumar, Nathaniel Lin, Nalini Ravishanker, and S Sriram. 2006. Modeling customer lifetime value. *Journal of service research* 9, 2 (2006), 139–155.

[13] Will Hamilton, Zhitao Ying, and Jure Leskovec. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*. 1024–1034.

[14] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in neural information processing systems*. 3146–3154.

[15] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[16] Thomas N Kipf and Max Welling. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907* (2016).

[17] Boris Knyazev, Graham W Taylor, and Mohamed Amer. 2019. Understanding Attention and Generalization in Graph Neural Networks. In *Advances in Neural Information Processing Systems*. 4204–4214.

[18] Hugo KS Lam, Andy CL Yeung, Chris KY Lo, and TCE Cheng. 2019. Should firms invest in social commerce? An integrative perspective. *Information & Management* 56, 8 (2019), 103164.

[19] Junhyun Lee, Inyeop Lee, and Jaewoo Kang. 2019. Self-attention graph pooling. *arXiv preprint arXiv:1904.08082* (2019).

[20] Matthew KO Lee, Na Shi, Christy MK Cheung, Kai H Lim, and Choon Ling Sia. 2011. Consumer's decision to shop online: The moderating role of positive informational social influence. *Information & management* 48, 6 (2011), 185–191.

[21] Kristina Lerman and Rumi Ghosh. 2010. Information contagion: An empirical study of the spread of news on digg and twitter social networks. In *Fourth International AAAI Conference on Weblogs and Social Media*.

[22] Ting-Peng Liang, Yi-Ting Ho, Yu-Wen Li, and Efraim Turban. 2011. What drives social commerce: The role of social support and relationship quality. *International journal of electronic commerce* 16, 2 (2011), 69–90.

[23] Miller McPherson, Lynn Smith-Lovin, and James M Cook. 2001. Birds of a feather: Homophily in social networks. *Annual review of sociology* 27, 1 (2001), 415–444.

[24] Vinod Nair and Geoffrey E Hinton. 2010. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the 27th international conference on machine learning (ICML-10)*. 807–814.

[25] Tore Opsahl, Filip Agneessens, and John Skvoretz. 2010. Node centrality in weighted networks: Generalizing degree and shortest paths. *Social networks* 32, 3 (2010), 245–251.

[26] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. 701–710.

[27] Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European Semantic Web Conference*. Springer, 593–607.

[28] Vivienne Sze, Yu-Hsin Chen, Tien-Ju Yang, and Joel S Emer. 2017. Efficient processing of deep neural networks: A tutorial and survey. *Proc. IEEE* 105, 12 (2017), 2295–2329.

[29] Lei Tang, Xufei Wang, and Huan Liu. 2012. Community detection via heterogeneous interaction analysis. *Data mining and knowledge discovery* 25, 1 (2012), 1–33.

[30] Ali Vanderveld, Addhyan Pandey, Angela Han, and Rajesh Parekh. 2016. An engagement-based customer lifetime value system for e-commerce. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. 293–302.

[31] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* (2017).

[32] Duncan J Watts and Steven H Strogatz. 1998. Collective dynamics of 'small-world'networks. *nature* 393, 6684 (1998), 440.

[33] Ian H Witten and Eibe Frank. 2002. Data mining: practical machine learning tools and techniques with Java implementations. *Acm Sigmod Record* 31, 1 (2002), 76–77.

[34] Fengli Xu, Jianxun Lian, Zhenyu Han, Yong Li, Yujian Xu, and Xing Xie. 2019. Relation-Aware Graph Convolutional Networks for Agent-Initiated Social E-Commerce Recommendation. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*. 529–538.

[35] Rex Ying, Ruining He, Kaifeng Chen, Pong Eksombatchai, William L Hamilton, and Jure Leskovec. 2018. Graph convolutional neural networks for web-scale recommender systems. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 974–983.

[36] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, Will Hamilton, and Jure Leskovec. 2018. Hierarchical graph representation learning with differentiable pooling. In *Advances in neural information processing systems*. 4800–4810.