Towards Consumer Loan Fraud Detection: Graph Neural Networks with Role-Constrained Conditional Random Field

Bingbing Xu[1,2*], Huawei Shen[1,2†], Bingjie Sun[3], Rong An[3†], Qi Cao[1,2], Xueqi Cheng[1,2]

[1]CAS Key Laboratory of Network Data Science and Technology,
Institute of Computing Technology, Chinese Academy of Sciences
[2]University of Chinese Academy of Sciences
[3]Ant Financial Services Group
{xubingbing, shenhuawei, caoqi, cxq}@ict.ac.cn, bingjie-nash@163.com,rongan0915@qq.com

Abstract

Consumer loans, i.e., loans to finance consumers to buy certain types of expenditures, is increasingly popular in e-commerce platform. Different from traditional loans with mortgage, online consumer loans only take personal credit as collateral for loans. Consequently, loan fraud detection is particularly critical for lenders to avoid economic loss. Previous methods mainly leverage applicant's attributes and historical behavior for loan fraud detection. Although these methods gain success at detecting potential charge-offs, yet they perform worse when multiple persons with various roles (e.g., sellers, intermediaries) collude to apply fraudulent loan. To combat this challenge, we consider the problem of loan fraud detection via exploiting roles of users and multi-type social relationships among users. We propose a novel Graph neural network with a Role-constrained Conditional random field, namely GRC, to learn the representation of applicants and detect loan fraud based on the learned representation. The proposed model characterizes the multiple types of relationships via self-attention mechanism and employs conditional random field to constrain users with the same role to have similar representation. We validate the proposed model through experiments in large-scale auto-loan scenario. Extensive experiments demonstrate that our model achieves state-of-the-art results in loan fraud detection on Alipay, one online credit payment service serving more than 450 million users in China.

Introduction

With the rapid development of e-commerce, consumer loans have become a popular form of credit activity, which can support the applicants to purchase valuable commodities on the online trading platform. Different from traditional loans, consumer loans only take personal credit as collateral for loans. Thus, fraud happens frequently, where applicants maliciously default on the loan although they have the ability to repay. To avoid huge economic losses, it is important to accurately assess loan risk and find out possible fraudulent applications.

Generally, on a consumer loan platform, there exist three typical roles, i.e., applicants, sellers, and intermediaries. An applicant applies for loan on the platform to purchase goods, then the platform examines the application and pays to the seller. After receiving the goods from the seller, the applicant needs to repay the loan in time. Here, the platform can be thought as a third-party financial institution. However, when faced with thousands of goods and loan products, choosing a proper one by applicants themselves is like finding a needle in a haystack, due to information asymmetry. Thus, intermediaries become another necessary role during the dealing, which bridge the gap between sellers and applicants, and help applicants solve difficulties during loan application. However, some intermediaries also play important roles in loan fraud. Specifically, they connect with sellers and applicants, and let applicants apply for the loan from the platform to purchase goods from sellers. Upon getting paid from the platform, sellers return most of the money to the applicants. The intermediaries also get a share of the money. Finally, the three roles all benefit from this dealing, while the loan will often be in arrears. Figure. 1 illustrates two common patterns of normal loans and fraudulent loans.

Our goal is to predict whether an applicant will maliciously default on the loan. Previous methods mainly leverage applicant's attributes and historical behavior for loan fraud detection. Based on these features, different classifiers are employed, e.g., tree-based models and neural networks. Although these methods gain success at detecting potential charge-offs, yet they perform worse when multiple persons with various roles (e.g., sellers, intermediaries) collude to apply fraudulent loan. The data used for loan fraud detection is a network where four roles of nodes, i.e., applicants, sellers, intermediaries, and other users (i.e., the users who do not belong to any of the above roles)[1], are connected by multiple relationships, including social connections, capital transactions, and device dependence, and each link stores the starting and ending time, as well as other attributes of the relationship. With the network representation, loan fraud detection is formulated as a node classification problem, i.e., predict whether an applicant node is fraudulent given node attribute and social relationships among nodes.

*Work done while Bingbing Xu was an intern at Ant Financial Services Group.

†Corresponding authors

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

[1]Note that one node may have multiple roles, e.g., one node can be both an intermediary and an applicant.

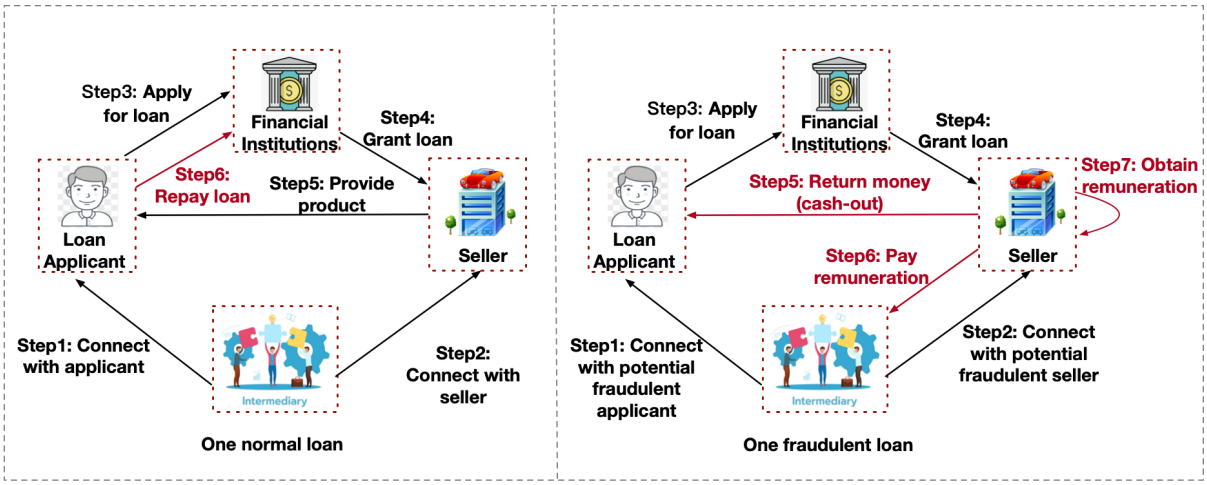

Figure 1: The different patterns between one normal loan and one fraudulent loan.

Recently, graph neural networks (GNN) (Defferrard, Bresson, and Vandergheynst 2016; Xu et al. 2019b,a, 2020) have gained remarkable success in node classification, leveraging the feature information from neighboring nodes to improve the representation learning of the target node. Based on our analysis, role information is rather important among all the features of a node, since the applicant who connects with many sellers and intermediaries are more likely to be a fraud (Sec. Data Analysis). However, GNN treats different features equally. All the features are mapped and propagated together to get the representations of nodes. Considering that the role of each node is just a one-dimensional feature in the high dimensional feature space, there exist more features which are not related to the role, e.g., age, gender, education. As illustrated in Figure. 2 (a), under the joint effect of high-dimensional features, the representation of each node may not distinguish among roles in feature space. As a result, the representation of applicants with neighbors of different roles have no distinction in representation space after neighbor aggregation, then the traditional GNNs fails. Besides, due to complex motives, fraudulent applicants often build new relations with other users before the application, thus the newly connected relations are more likely to be related to fraud. It is significant to model the temporal and type information of relations, while traditional GNNs cannot handle it.

To address the above challenges, we propose Graph neural networks with Role-constrained Conditional random field for loan fraud detection, namely **GRC**, which aims to make full use of the graph structure and the role of each node. Since the relationships around the applicant reflect the loan risk, we build sub-networks for an applicant based on different relationships. To handle the problem in Fig. 2 (a), we leverage conditional random field (CRF) to constrain the nodes of the same type to have similar representations. Under this constraint, Fig. 2 (b) demonstrates that GRC can generate better representations after information aggregation to distinguish fraudulent applicants from normal ones. Furthermore, we use the attention mechanism to learn the impact of different relationships on the loan risk. Considering that new relations established before the application may be useful for our task rather than the long-existed relations, we split the relationship network into two parts for each user based on the time interval between the relationship establishment time and loan application time, and different weights are learned for each part. Extensive experiments on Alipay show that our model achieves state-of-the-art results in loan fraud detection. Also, we provide ablation studies to evaluate the importance of each part.

In summary, our work has the following contributions:

- We investigate consumer loan fraud detection in e-commerce, one important and realistic AI application problem. We formalize it as node classification task and find the role information can help to detect fraud.
- We propose a novel graph neural network with a role-constrained conditional random field, leveraging the role information to detect collusive fraud, overcoming the shortage of previous methods that only use user profile.
- Our model achieves state-of-the-art results in realistic large-scale auto-loan scenario on Alipay, one online credit payment service serving more than 450 million users in China.

Preliminary

In this section, we firstly give the definition of loan fraud detection, and then we analyze the patterns of neighbors with different roles for fraudulent applicants and normal applicants respectively on real data.

Problem Definition

In the loan fraud detection problem, we need to classify one loan into two classes: normal loan or fraudulent loan. We give the brief definition of the task. Suppose that we have R relations in our network, e.g., fund based relation, social based relation, device based relation and so on. Each sample can be defined as $\{x = \{G_1, G_2, \ldots, G_R, T\}, y =$

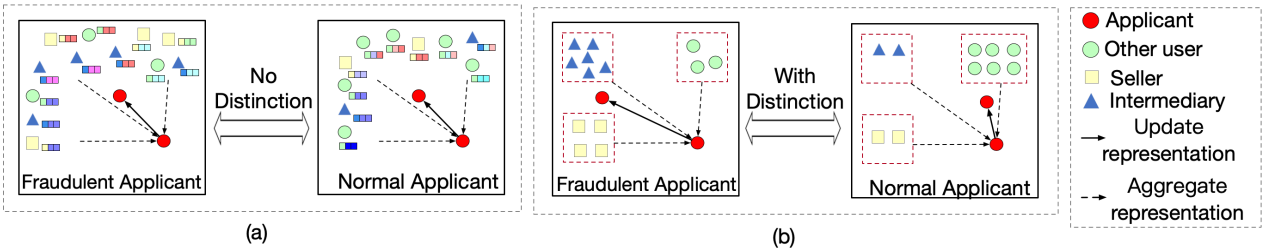

Figure 2: Node representations in the common embedding space. The color block next to each node represents its features, and the first dimension in the features denotes the role of the node. (a): The representations of neighbors with different types are mapped to the same representation position, thus it is hard to distinguish the fraudulent applicants from normal applicants (the fraudulent applicant and the normal applicant, e.g., two red nodes are mapped into the same position after neighbors aggregation). (b): Leveraging conditional random field to constrain the nodes of the same type to have similar representations can help to distinguish the fraudulent applicants from normal applicants, thus enhancing the classification.

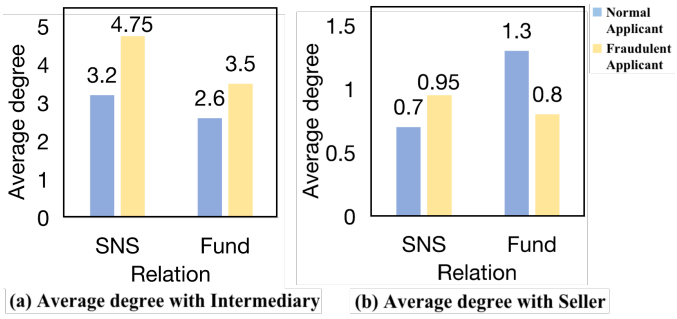

Figure 3: The fraudulent applicants tend to have more intermediary-neighbors and seller-neighbors, which suggests that leveraging the relations with intermediaries and sellers can enhance the classification.

$\{0,1\}\}$. $G_i = \{V^i, E^i, X^i_{node}, X^i_{edge}\}$ denotes the i-th relation between the applicant and other users, consisting of the node set V^i (including the central applicant and it's neighbors), the edge set E^i, the attribute information matrix of nodes $X^i_{node} \in R^{|V|\times k_1}$ and an information matrix of edges $X^i_{edge} \in R^{|E|\times k^i_2}$. Some columns in X^i_{node} indicate the role of the corresponding node, e.g, seller, and others describe its profile. T denotes the application time. The label $y \in 0, 1$ for each loan indicate whether this loan is fraudulent or not. Given some historical loan records $\{x = \{G_1, G_2, \ldots, G_R, T\}, y = \{0, 1\}\}$ in the training set, the goal is to classify one loan in the test set into two classes: normal loan or fraudulent loan.

Data Analysis

In this section, we analyzed the real cases of fraudulent loans, and find that the fraudulent applicants tend to connect with intermediaries and sellers via different relations based on the real data of Alipay.

Intermediary-neighbors aggregation. The basic idea is that fraudulent applicants tend to connect with more intermediary-neighbors and sellers. We first collect the neighbors of each applicant in each relation. For each applicant, we count the number of neighbors who are intermediary (called intermediary-neighbor). For comparison, we calculate the average degree of intermediary-neighbors for the fraudulent applicants and normal applicants respectively based on each relation. Fig. 3 (a) illustrates that the fraudulent applicants tend to have more intermediary-neighbors. This observation implies that modeling the connection with their intermediary-neighbors can help us to distinguish normal applicants from fraudulent applicants. Similarly, we also count the mean number of neighbors who are sellers (called seller-neighbor) for each type of applicant. Fig. 3 (b) illustrates that the fraudulent applicants are more likely to connect with seller-neighbors, which suggests that leveraging the relations with sellers also enhance the classification.

Methods

In this section, we present the architecture of Graph neural networks with Role-constrained Conditional random field for loan fraud detection, called **GRC** shortly.

Architecture

In this section, we give the motivations and overall architecture based on our above observations in real data. Since the connections with intermediaries and sellers can enhance the classification, we first involve the graph neural networks to model the corresponding relationship networks to learn the representation of applicant. However, previous GNNs only regard the role as one-dimensional feature and treat it equally as other features via feature engineering, leading to the failure of GNNs. To solve the challenges, we resort to conditional random field to constrain the nodes of the same type to have similar representations, thus distinguish normal applicants from fraudulent applicants easier. Furthermore, we use the attention mechanism to capture the different impacts of relations and time slots. Fig. 4 illustrates the overall architecture of our GRC, then we will introduce each part respectively in the following section.

CBlock: Neighbors Aggregation under CRF Constraint

To aggregate information of neighbors for the target node, we first need to do feature transformation for each node to

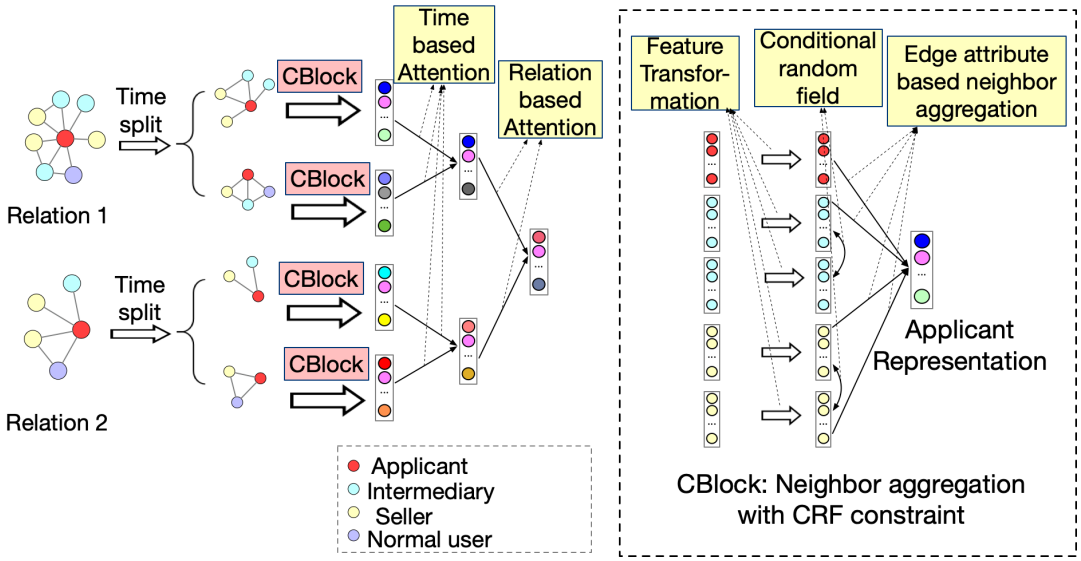

Figure 4: The overall architecture of GRC. Suppose there are two relations. We extract the relationship network for each applicant (red node) and split it into two parts based on the time interval between the relationship establishment time and loan application time. For each part, we obtain the applicant's representation via CBlock, which aggregates the neighbor's representation under the CRF constraint. We leverage the attention mechanism to learn the impacts of relations and time slots.

obtain its representation, this feature transformation is under the CRF constraint which enforce the representations of nodes with the same type to be similar. Then, we model the impact of each neighbor for the central applicant and aggregate neighbors under the learned impact.

Feature Transformation under the CRF Constraint

We first conduct feature transformation for each node as follows:

$$X^{'} = XW, \tag{1}$$

where $X \in \mathbb{R}^{n \times k_1}$ denotes the input representation and k_1 is the dimension of node attributes, $W \in \mathbb{R}^{k_1 \times q}$ is the learned parameters to transform the input dimension k_1 into the output dimension q.

As Fig.2 illustrated, the distinguishable representation for nodes with different roles can help us to distinguish the normal applicants from fraudulent applicants, thus we enforce the representations of nodes with the same role to be similar, and then propagate the node information through the new representations. Specifically, $X_u^{'} \in \mathbb{R}^{n \times q}$ in Eq. 1 denotes the preliminary representation of the node u, and H_u denotes the adjusted new representation based on $X_u^{'}$, which needs to satisfy the following two constraints,

$$\psi_u(H_u, X_u^{'}) = ||H_u - X_u^{'}||_2^2 \to 0, \tag{2}$$

$$\psi_p(H_u, H_v) = f_{uv}||H_u - H_v||_2^2 \to 0, \tag{3}$$

where f_{uv} equals to 1 if nodes u and v have the same role, or 0 otherwise. Here, we minimize the measurement Eq. 2 to enforce H_u to be close to $X_u^{'}$. Through this constraint, the new node representation can maintain the original information of the preliminary representation. Meanwhile, we minimize the measurement Eq. 3 to enforce the representations of nodes with the same type to be similar.

Based on the two constraints, an energy function for node u can be defined as:

$$E(H_u|X_u^{'}) = \alpha||H_u - X_u^{'}||_2^2 + \beta \sum_{v \in R(u)} f_{uv}||H_u - H_v||_2^2 \tag{4}$$

where $R(u)$ represent the nodes that have the same role with node u. Hyper-parameters α and β are used to define the importance of these two measurement functions. Considering that the function may be not exactly consistent with the loan fraud detection, treating this function as a regularization loss will result in optimizing not toward the global optimum. To avoid this, We resort to Conditional Random Field (CRF) to minimize it. CRF is a probabilistic graphical model first proposed in (Lafferty, McCallum, and Pereira 2001) and can capture the desired properties via defining energy function (Gao, Pei, and Huang 2019).

Specifically, we can view $X_u^{'}$ and H_u as random variables, and H_u relies on $X_u^{'}$ and H_v ($v \in R(u)$). In this formulation, the X' is the observation, and H is the label. Then, we have the following CRF model,

$$P(H|X^{'}) = \frac{1}{Z(X^{'})} \exp(-E(H|X^{'})), \tag{5}$$

where $Z(\cdot)$ serves as the normalization factor.

To maximize the probability, we leverage the mean-field approximation method. Specifically, we use the simple distribution $Q(H)$ to approximate $P(H|X^{'})$ and minimize the KL divergence between these two distributions:

$$\text{minKL}(Q(H)||P(H|X^{'})). \tag{6}$$

Although the variables in H are not independent from each other, we can leverage variational inference to represent the distribution $Q(H)$ by the product of independent

marginal distributions as $Q(H) = \prod_{u=1}^{n} Q_u(H_u)$. According to Eq. 4 and Eq. 5, we can get

$$Q_u^*(H_u) \sim \exp(-(\alpha||H_u - X_u^{'}||_2^2 + \beta \sum_{v \in R(u)} f_{uv}||H_u - H_v||_2^2)) \tag{7}$$

This form indicates that $Q_u^*(H_u)$ is a multi-dimensional Gaussian function. We compute expectation of $Q_u^*(H_u)$ to obtain the maximum probability. Thus, the updating rule for the adjusted representation is as follows:

$$H_u^{k+1} = \frac{\alpha X_u^{'} + \beta \sum_{v \in R(u)} f_{uv} H_v^k}{\alpha + \beta \sum_{v \in R(u)} f_{uv}}, \tag{8}$$

where $H_v^0 = X_v^{'}$. By iterative adjusting the representation of node u of K iterations, the adjusted representations of the nodes with the same role are similar, and they still maintain their own feature information.

Neighbors Aggregation

To model the impact of each neighbor for the central node, previous methods often use the Laplacian matrix (Kipf and Welling 2017) or attention based on the hidden representation (Velickovic et al. 2017) of each neighbor to calculate the weight. Different from previous scenarios, we have the attributes for each edge. Intuitively, the attribute of each edge should be leveraged to model the weights of each neighbor. Let k_2^i be the dimension of edge attribute for the i-th relation, given the edge attribute $X_{uv}^i \in R^{k_2^i \times 1}$ for node pair $\{u, v\}$, the weight of neighbor u to central node v is calculated as:

$$e_{uv} = a^i X_{uv}^i, \tag{9}$$

where $a^i \in R^{1 \times k_2^i}$ is a learned parameter vector to transform the edge attribute into the weight. The weight indicates the importance of node u to node v. In this scenario, the number of neighbors is important for classify one applicant into normal applicant or fraudulent applicant, thus we do not normalize the weights across all neighbors of central node.

Once we obtained the constrained representation of each node and their corresponding weights, these weights are leveraged to compute a linear combination of the representations, to serve as the aggregated representation for the applicant node.

$$h_v = \sum_{u \in N(v)} e_{uv} H_u^K + H_v^K, \tag{10}$$

where $N(v)$ denotes the neighbors of central node v and K is the hyper-parameter used in CBlock. Overall, we obtain the representation for each applicant via neighbors aggregation under CRF constraint in each time slot of each relation.

Time Slot Aggregation based on Attention Mechanism

Intuitively, the newly connected relations with other users before the application are more likely related to fraud rather than the long-existed relations, thus we hope to capture the different effects of the relations with different start time. Considering that the relation in time is sparse, i.e., only a few applicants have the new connected relations in one day, we directly split the relationship network into two time slots for each user based on the time interval between the relationship establishment time and loan application time. In this paper, we set the threshold time interval as 30 days. Since attention mechanism is proved an effective and efficient method to capture the effects of different aspects, we leverage the attention mechanism to learn the weights for each part.

For the two time slots, we obtained the representation via the above CBlock respectively. Let $h_{v,t1}^r$ and $h_{v,t2}^r$ denote the representation of applicant in two time slots of one given relation r. We leverage the attention mechanism to learn the different weights of these two time slots:

$$w_{r,t1} = a_{r,t1} h_{v,t1}^r, w_{r,t2} = a_{r,t2} h_{v,t2}^r, \tag{11}$$

where $a_{r,t1} \in R^{1 \times q}$ and $a_{r,t2} \in R^{1 \times q}$ are the learned parameters to model the impact of corresponding time slot. Then we normalize them using softmax function:

$$w_{r,ti} = \frac{\exp(w_{r,ti})}{\sum_{i \in 1,2} \exp(w_{r,ti})} \tag{12}$$

Based on these two normalized and positive weights, we calculated the representation of applicant of the given relation r as:

$$h_v^r = w_{r,t1} h_{v,t1}^r + w_{r,t2} h_{v,t2}^r. \tag{13}$$

After this, we obtained the representation for each applicant on each relation.

Relation Aggregation based on Attention Mechanism

In the real relationship networks, we often have different relations, e.g., social relationship, fund relationship and so on. Similarly, we learn the impact of each relation via leveraging the attention mechanism. For one applicant in a given relation r with the representation h_v^r, we obtain the weight of this relation as:

$$w_r = a_r h_v^r, \tag{14}$$

where $a_r \in R^{1 \times q}$ is the learned vector to model the impact of the corresponding relation. We first calculate the weights for all relations respectively, and then we normalize them using softmax function:

$$w_r = \frac{\exp(w_r)}{\sum_{r \in R} \exp(w_r)}, \tag{15}$$

where R represents the set of all relations. Based on these normalized weights, we calculated the final representation of applicant as:

$$h_v = \sum_{r \in R} w_r h_v^r. \tag{16}$$

Model Learning

We aims to make binary classification for the given applicant, thus we leverage the sigmoid function on the obtained final representation, e.g., h_v in the output layer:

$$p_v = \text{sigmoid}(W_o h_v + b_o), \tag{17}$$

where W_o and b_o denote the learned weight matrix and bias vector of the output layer and sigmoid is the active function to project the output value into $[0, 1]$. For this classification task, we model the objective function with maximum likelihood estimation, which can be formulated as follows:

$$L = \sum_{(v,y_v)\in T} y_v log(p_v) + (1 - y_v) log(1 - p_v), \quad (18)$$

where T denotes the training set, y_v represents the ground-truth label for applicant v, and p_v denotes the predicted fraudulent probability of applicant v.

Experiments

To validate the proposed GRC, we show the experimental results of our model applied to Alipay.

Datasets

We use the real dataset in Alipay to validate our model. Alipay is online credit payment service provided by Ant Financial Services Group, serving more than 450 millions of users. We use one scenario in Alipay named auto-loan. This scenario provides service for the user who want to apply for the loan to buy car. The users in car-finance includes applicants, sellers, intermediaries and other users. Meanwhile, there exists lots of fraudulent applicants. We hope to detect these applicants when they apply the loan and reject them.

We extract the year dataset (2018/08/01-2019/08/01) to detect fraud, each loan record in the dataset can be regard as one sample. If the loan overdue happens within 3 months after the loan origination, the court will put the lender on trial for fraud. However, if it is overdue after 3 months of repayment on time, the court will treat it as the credit overdue. Malicious fraudsters usually take this way to avoid legal sanctions. Therefore, we mark the overdue lenders as fraud, i.e., malicious fraudsters who are overdue than 30 days after 3 months of repayment on time. We define the normal applicants who always repay in time within six months. Each loan record includes the attributes of applicant and the apply time. In addition, we have the four relationship networks with edge attributes and node attributes corresponding with four different relations of the applicant. The attributes for each user include user profile, transaction summarizing and so on. Considering that the relationship networks and attributes of all users are hard to load into memory, we extract the attributes and the relationship networks by only retaining the neighbors for each applicant.

Baselines

We consider several representative methods for the loan fraud detection, which can be divided into two types: (1) Attribute only. We use SVM (Suykens and Vandewalle 1999) and MLP as our baselines. (2) Leveraging Structure and Attribute. GNNs are popular, however, it is full of challenges to leverage GNN into our task directly because of the edge attributes and different relations of our task. Considering this, we take GCN (Kipf and Welling 2017) as our baseline, and we use the normalized Laplacian matrix to capture the weights between applicant and its neighbors, and give the equal weight to all relations and time slots. To be noted that since each sample have four graphs based on four relationship networks, it is unreasonable to use network embedding methods, e.g., DeepWalk (Perozzi, Al-Rfou, and Skiena 2014), LINE (Tang et al. 2015) and node2vec (Grover and Leskovec 2016), to solve this problem directly. It's the reason why we don't take the methods that using structure only as our baselines.

Furthermore, to validate the effect of the each component in our models, i.e., Neighbor aggregation (EA), Time attention(TA), Relation attention(RA) and Conditional random field (CRF), we also implement variants of our models and conduct ablation analysis.

Experimental Settings

We implement our models using Tensorflow. Considering that it is difficult for us to obtain the labeled data, we verify the impact of the training data size, we used 1,000, 2,000, 3,000 and 12,000 data as training data respectively. In addition, we leverage the additional validation set of 1,000 labeled samples to determine hyper-parameters. We use Adam optimizer (Kingma and Ba 2014) with an initial learning rate of 0.01 and a weight decay of 0.0005. The hyper-parameter K is set to be 1, and we set α and β as 0.5. We run 100 epochs and choose the model that performs the best on the validation set.

Evaluation Metric

In the loan fraud detection, we care more about the positive samples, i.e., the applicants who do not repay in time after 3 months. Identifying the fraudulent applicants can reduce loan risks and losses. We use the precision and recall to evaluate our GRC. To balance these two metrics, we also use F1 score to measure the performance of each methods.

Performance on Loan Fraud Detection Task

Size	Method	SVM	MLP	GCN	GRC
1,000	Precision	66.32%	72.80%	69.78%	72.28%
	Recall	51.00%	44.20%	61.20%	82.40%
	F1-Score	57.66%	55.00%	65.21%	**77.01%**
2,000	Precision	65.80%	71.25%	68.59%	74.38%
	Recall	51.60%	46.60%	69.00%	89.70%
	F1-Score	57.84%	56.35%	68.79%	**81.32%**
3,000	Precision	66.50%	69.25%	70.35%	77.09%
	Recall	47.10%	51.80%	71.60%	89.50%
	F1-Score	55.14%	59.27%	70.97%	**82.83%**
12,000	Precision	68.65%	66.82%	74.06%	77.50%
	Recall	54.10%	59.20%	84.80%	95.50%
	F1-Score	60.51%	62.78%	79.07%	**85.56%**

Table 1: Results of loan fraud detection

We now validate the effectiveness of GRC on loan fraud detection. Specifically, we evaluate GRC, varying the number of labeled samples on 1,000, 2,000, 3,000 and 12,000 training samples respectively. Experimental results are reported in Table 1 under different training size. Bold numbers indicate that our method improves the base model. When

the training samples is not enough, the "Recall" of SVM is better than MLP, which may due to the higher parameter complexity of MLP. Graph neural network methods including GCN and our GRC all perform much better than other methods, which indicate that the relationship networks can enhance the loan fraud detection. Specifically, the result on metric "Recall" achieved significant improvement over MLP and SVM methods, this phenomenon is consistent with our analysis. Lots of applicants will disguise themselves to be approved when applying for the loan. However, the connections with sellers and intermediaries may be necessary, so the relationship network can reflect the loan risk than only leveraging the attributes of applicants. Different from GCN methods, our GRC learns the different weights for time slots and relations, and leverages the edge attribute to do neighbor aggregate under CRF constraint, achieving much improvements over GCN.

Ablation Study

We do ablation study to validate each component of our GRC. Table 2 show the performance of GRC and GRC without other component, i.e., GRC(w/o *). GRC performs better than other base methods in "F1 score" , which indicate that each component achieves the positive effect. In addition, the "Recall" metric in GRC(w/o EA) increase much, which means the edge attributes are important to identify the important neighbors. Furthermore, model the impacts of different relations and time slots is also important.

Method	precision	Recall	F1 score
GRC(w/o CRF)	77.71%	91.00%	83.83%
GRC(w/o EA)	79.60%	88.20%	83.68%
GRC(w/o RA)	77.58%	92.40%	84.34%
GRC(w/o TA)	76.54%	91.70%	83.44%
GRC	77.50%	95.50%	**85.56%**

Table 2: Ablation study

Attention Analysis

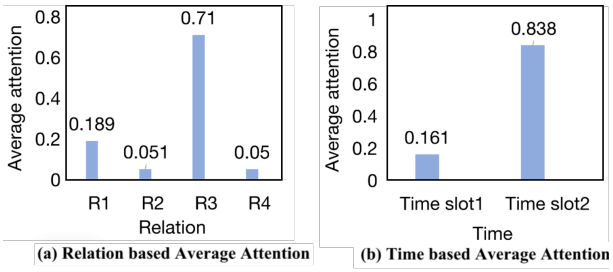

Figure 5: The average attention values for all nodes under different relations and time slots. The higher importance of "Relation3" and "Time slot2" indicates that modeling the impacts of relations and time slots is necessary.

To show the effect of attention mechanism, we show the average attention values for all nodes under different relations and time slots. Figure 5 (a) illustrates the average attention values under relations. Intuitively, the "Relation3" is more important than other relations, which suggests that modeling the impacts of relations is important. Furthermore, the importance of "Time slot2" is much higher than "Time slot1", in other words, the relationships that occurred recently before applying for a loan can be more helpful for classification, which is consistent with our expectations.

Related Work

Graph neural networks (GNNs) (Gilmer et al. 2017; Xu et al. 2018; Zhou et al. 2018; Hasanzadeh et al. 2020; Chen et al. 2020; Rong et al. 2020) leveraging the information from neighboring nodes based on weighting function to improve the representation. GraphSAGE (Bruna et al. 2014) defines the weighting function as various aggregators over neighboring nodes. MoNet (Monti et al. 2017) offers us a general framework for designing spatial methods. There also existed some research (Zhang et al. 2020; Liu et al. 2020) to applied the GNNs in fraud detection applications via involving the properties of applications. Some previous works (Mishra et al. 2019; Nguyen and Grishman 2018; Qiao et al. 2020) applied GNNs to detect the patterns in natural language and time series data. In addition, some works focused on the Internet finance. HGN (Liu et al. 2018) constructs a heterogeneous network of account devices and establishes GCN to identify fake account. Cash out detection is studied in (Hu et al. 2019). The author performs aggregation of neighbor features based on meta-path to obtain expressions and learns attention for each meta-path to obtain node expressions to classify nodes. Abnormal edge monitoring is proposed in (Zheng et al. 2019) applied in the recommendation system. HGAT (Cheng et al. 2019) finds that some nodes in the entire guarantee network are in a dominant position and leverages GAT to predict credit risk. SemiGNN (Wang et al. 2019) proposes to expand the labeled data through their social relations to get the unlabeled data, utilizing the multi-view labeled and unlabeled data for fraud detection.

Conclusion

To detect consumer loan fraud, we propose GRC to learn the representation of applicants. GRC characterizes the multiple types of relationships via self-attention mechanism and employs conditional random field to constrain users with the same role to have similar representation. We validate the proposed model through experiments in large-scale auto-loan scenario. Extensive experiments demonstrate that our model achieves state-of-the-art results in loan fraud detection on Alipay. In the future, we will apply this method to loan fraud detection beyond auto-loan scenario.

Acknowledgements

This work is funded by the National Natural Science Foundation of China under Grant Nos. 91746301 and 61802370. This work is supported by Beijing Academy of Artificial Intelligence (BAAI) under the grant number BAAI2019QN0304. Huawei Shen is also funded by K.C. Wong Education Foundation.

References

Bruna, J.; Zaremba, W.; Szlam, A.; and Lecun, Y. 2014. Spectral networks and locally connected networks on graphs. In *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*.

Chen, M.; Wei, Z.; Huang, Z.; Ding, B.; and Li, Y. 2020. Simple and Deep Graph Convolutional Networks. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 3730–3740.

Cheng, D.; Tu, Y.; Ma, Z.-W.; Niu, Z.; and Zhang, L. 2019. Risk Assessment for Networked-guarantee Loans Using High-order Graph Attention Representation. In *IJCAI*, 5822–5828.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. In *Advances in Neural Information Processing Systems*, 3844–3852.

Gao, H.; Pei, J.; and Huang, H. 2019. Conditional random field enhanced graph convolutional neural networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 276–284.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural message passing for quantum chemistry. *arXiv preprint arXiv:1704.01212* .

Grover, A.; and Leskovec, J. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, 855–864.

Hasanzadeh, A.; Hajiramezanali, E.; Boluki, S.; Duffield, N.; Zhou, M.; Narayanan, K.; and Qian, X. 2020. Bayesian Graph Neural Networks with Adaptive Connection Sampling. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 10642–10652.

Hu, B.; Zhang, Z.; Shi, C.; Zhou, J.; Li, X.; and Qi, Y. 2019. Cash-out user detection based on attributed heterogeneous information network with a hierarchical attention mechanism. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 946–953.

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *International Conference on Learning Representations (ICLR)*.

Lafferty, J.; McCallum, A.; and Pereira, F. C. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the 18th International Conference on Machine Learning (ICML)*, 282–289.

Liu, Z.; Chen, C.; Yang, X.; Zhou, J.; Li, X.; and Song, L. 2018. Heterogeneous graph neural networks for malicious account detection. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 2077–2085.

Liu, Z.; Dou, Y.; Yu, P. S.; Deng, Y.; and Peng, H. 2020. Alleviating the Inconsistency Problem of Applying Graph Neural Network to Fraud Detection. *arXiv preprint arXiv:2005.00625* .

Mishra, P.; Del Tredici, M.; Yannakoudakis, H.; and Shutova, E. 2019. Abusive language detection with graph convolutional networks. *arXiv preprint arXiv:1904.04073* .

Monti, F.; Boscaini, D.; Masci, J.; Rodola, E.; Svoboda, J.; and Bronstein, M. M. 2017. Geometric deep learning on graphs and manifolds using mixture model CNNs. In *Proc. CVPR*, volume 1, 3.

Nguyen, T. H.; and Grishman, R. 2018. Graph Convolutional Networks With Argument-Aware Pooling for Event Detection. In *AAAI*, volume 18, 5900–5907.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. Deepwalk: Online learning of social representations. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 701–710. ACM.

Qiao, Z.; Wang, P.; Fu, Y.; Du, Y.; Wang, P.; and Zhou, Y. 2020. Tree Structure-Aware Graph Representation Learning via Integrated Hierarchical Aggregation and Relational Metric Learning. In *IEEE International Conference on Data Mining (ICDM)*.

Rong, Y.; Huang, W.; Xu, T.; and Huang, J. 2020. DropEdge: Towards Deep Graph Convolutional Networks on Node Classification. In *International Conference on Learning Representations (ICLR)*.

Suykens, J. A.; and Vandewalle, J. 1999. Least squares support vector machine classifiers. *Neural processing letters* 9(3): 293–300.

Tang, J.; Qu, M.; Wang, M.; Zhang, M.; Yan, J.; and Mei, Q. 2015. Line: Large-scale information network embedding. In *Proceedings of the 24th international conference on world wide web*, 1067–1077.

Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; and Bengio, Y. 2017. Graph attention networks. *arXiv preprint arXiv:1710.10903* .

Wang, D.; Lin, J.; Cui, P.; Jia, Q.; Wang, Z.; Fang, Y.; Yu, Q.; Zhou, J.; Yang, S.; and Qi, Y. 2019. A Semi-supervised Graph Attentive Network for Financial Fraud Detection. In *2019 IEEE International Conference on Data Mining (ICDM)*, 598–607. IEEE.

Xu, B.; Huang, J.; Hou, L.; Shen, H.; Gao, J.; and Cheng, X. 2020. Label-Consistency based Graph Neural Networks for Semi-supervised Node Classification. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1897–1900.

Xu, B.; Shen, H.; Cao, Q.; Cen, K.; and Cheng, X. 2019a. Graph convolutional networks using heat kernel for semi-supervised learning. In *Proceedings of the 28th International Joint Conference on Artificial Intelligence*, 1928–1934. AAAI Press.

Xu, B.; Shen, H.; Cao, Q.; Qiu, Y.; and Cheng, X. 2019b. Graph Wavelet Neural Network. *arXiv preprint arXiv:1904.07785* .

Xu, K.; Hu, W.; Leskovec, J.; and Jegelka, S. 2018. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826* .

Zhang, S.; Yin, H.; Chen, T.; Hung, Q. V. N.; Huang, Z.; and Cui, L. 2020. GCN-Based User Representation Learning for Unifying Robust Recommendation and Fraudster Detection. *arXiv preprint arXiv:2005.10150* .

Zheng, L.; Li, Z.; Li, J.; Li, Z.; and Gao, J. 2019. AddGraph: Anomaly Detection in Dynamic Graph Using Attention-based Temporal GCN. In *IJCAI*, 4419–4425.

Zhou, J.; Cui, G.; Zhang, Z.; Yang, C.; Liu, Z.; Wang, L.; Li, C.; and Sun, M. 2018. Graph neural networks: A review of methods and applications. *arXiv preprint arXiv:1812.08434* .