ImGAGN:Imbalanced Networks Embedding via Generative Adversarial Graph Networks

Abstract

Imbalanced classification is ubiquitous yet challenging in many real-world applications, such as cancer detection in medical diagnosis and fraud detection in financial system. Recently, generative adversarial networks (GANs) based imbalanced classification methods have shown great advantages for imbalanced classification problems. However, little work has employed them to the imbalanced problem on the graph/network structural data. On the other hand, graph neural networks (GNNs) have shown promising performance on many network analysis tasks. However, most existing GNNs have almost exclusively focused on the balanced networks, and would get unappealing performance on the imbalanced networks. To bridge this gap, in this paper, we present a generative adversarial graph network model, called ImGAGN to address the imbalanced classification problem on graph. It introduces a novel generator for graph structural data, named GraphGenerator, which can simulate the distribution of the minority class nodes and generate a set of synthetic minority nodes linking to the real minority nodes to balance the network classes distribution. Then a graph convolutional network (GCN) discriminator is trained to discriminate between minority nodes and majority nodes on the synthetic balanced network classes. To validate the effectiveness of the proposed method, extensive experiments are conducted on five real-world imbalanced network datasets. Experimental results demonstrate that the proposed method ImGAGN outperforms state-of-the-art algorithms for semi-supervised imbalanced binary node classification task.

Introduction

Network data, consisting of nodes (objects) and edges (objects' relationships), is ubiquitous in many real-world problems, such as social networks, protein-protein interaction networks, citation networks and so on. Recently, network embedding (Cai, Zheng, and Chang 2017; Wu et al. 2019) techniques, which map the nodes of the original networks into the dense and low-dimensional vectors (called node embeddings) and preserve the network structure information as much as possible, have shown promising performance on many network data analysis tasks, such as node classification (Kipf and Welling 2016), link prediction (Grover and Leskovec 2016), community detection (Fortunato 2010) and so on.

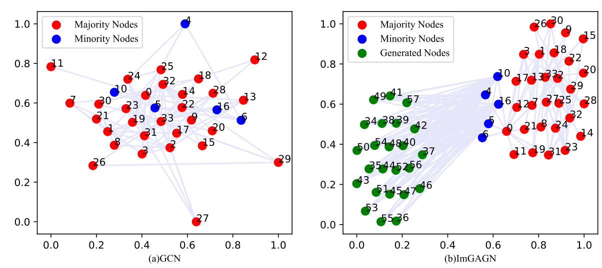

Figure 1: The 2-dimensional network embedding for the imbalanced network (Zachary's Karate Network (Zachary 1977)) using: (a) GCN (Kipf and Welling 2016) (b) proposed ImGAGN.(The red and blue circles represent the majority and minority nodes of the original network respectively, and the green circles represent the generated minority nodes by ImGAGN.)

Typical network embedding methods could be roughly divided into two categories, unsupervised network embedding methods and semi-supervised network embedding methods. The former obtains the node embeddings by preserving the network structural information. Representative method like DeepWalk (Perozzi, Al-Rfou, and Skiena 2014) utilizes the truncated random walks strategy to preserver network local information. The latter, semi-supervised network embedding methods, utilizes not only network structural information but also nodes' label information. Representative method like GCN (Kipf and Welling 2016) obtains the target node embeddings by aggregating the neighbor nodes' feature information.

However, the extensive existing network embedding methods assume that the nodes' labels are balanced, i.e., every class has roughly equal number of examples. Generally, these methods could not obtain good performance on the imbalanced networks which the number of examples of one class (minority) is far less than that of other classes (majority), and the minority usually plays an essential role in the real-world problems. For example, for the fraud detection in the online social networks, the number of fraudsters is far less than that of the normal users, and the fraudsters often try to disguise their identities as the normal users. Therefore, two key challenges of imbalanced network analysis are that: (1) The number of one class examples (minority nodes) is far less than that of other classes (majority nodes) in the network, and the labeling for minority nodes is extremely expensive. (2) The minority nodes are non-separability from the majority nodes, that is, it is difficult to find the support regions of majority and minority nodes in the networks (as shown in Figure 1(a)).

To address the above challenges, inspired by the success of GANs based methods (Shamsolmoali et al. 2020; Douzas and Bacao 2018) for imbalanced classification problems on non-graph domains, in this paper, we propose a novel semi-supervised generative adversarial graph network model, called ImGAGN. It introduces a GraphGenerator which can simulate the distribution of the minority class nodes and generate a set of minority class nodes linking to the real minority nodes to balance the original network classes distribution, then GCN discriminator is trained to discriminate between minority nodes and majority nodes on the synthetic balanced network classes. Specifically, as shown in the Figure 1(b), the generator iteratively learns to generate a set of minority nodes (green circles in Figure 1) to make the original network classes balanced. The generated nodes are linked to the original minority nodes (blue circles in Figure 1) of the network, and the features of the generated nodes are obtained by aggregating their neighbor nodes' (i.e., the real minority nodes) features. Then the discriminator (GCN) is trained to discriminate whether the node is generated by generator and whether the node is minority class. From Figure 1, we can find that ImGAGN could generate a set of appropriate minority nodes to make the original minority nodes separate from the majority nodes.

The main contributions of this paper are summarized as follows:

- In this paper, we propose an effective semi-supervised generative adversarial graph network model, called ImGAGN, which utilizes a generator to simulate the minority class node distribution and generates a set of minority nodes to make original network classes balanced. Then GCN is used to discriminate the majority and minority nodes on the synthetic balanced network classes.
- Based on ImGAGN, we propose a novel generator for graph structural data, called GraphGenerator, which can effectively learn not only the node feature distribution but also the network structural distribution.
- The proposed method is validated on five real-world imbalanced network datasets for imbalanced binary node classification and network layouts tasks. Experimental results demonstrate that the proposed method is superior to the state-of-the-art imbalanced network embedding techniques.

The rest of the paper is organized as follows. Section 2 will introduce some main related works. Section 3 will formulate the problem and provide a detailed introduction to the proposed method. In Section 4, we will introduce the experimental setups and results followed by the conclusions in Section 5.

Related Works

In this section, we introduce two main related research fields including imbalanced learning and imbalanced network embedding.

Imbalanced learning

Imbalanced learning techniques (He and Garcia 2009; Johnson and Khoshgoftaar 2019) aim at solving the problem with imbalanced data in which at least the number of one class data (minority) is far less than that of other classes (majority). Generally speaking, the minority class is often high-impact on many real-world problems, such as the cancer detection in medical diagnosis and fraud detection in financial system.

Existing methods for imbalanced learning mainly include: (1) sampling based methods, which learn the imbalanced classification by oversampling (Han, Wang, and Mao 2005) the minority class or undersampling (Liu, Wu, and Zhou 2008) the majority class. Representative method like SMOTE (Chawla et al. 2002) generates artificial data from existing minority class. (2) cost-sensitive learning based methods (Elkan 2001; Ting 2002), which utilize different cost matrices for calculating the cost of any particular data examples misclassified. (3) kernel-based methods (Akbani, Kwek, and Japkowicz 2004), which employ classifier like support vector machines (SVMs) (Suykens and Vandewalle 1999) to maximize the separation margin. and (4) GANs based methods (Shamsolmoali et al. 2020; Montahaei et al. 2018; noa 2018; Douzas and Bacao 2018), which are similar to our proposed method using the generator to create the minority class for balancing the data classes distribution. However, to our best knowledge, little work has employed these GANs based methods to the imbalanced network data.

Imbalanced network embedding

GRADE (He, Liu, and Lawrence 2008) is the classic method for imbalanced network embedding. It utilizes the global similarity matrix to obtain the compact minority class clusters, and learns the decision boundary between majority and minority classes by selecting the examples from the regions where the density changes the most. Wu et al. (Wu, He, and Liu 2018) propose a novel random walk strategy, called vertex-diminished random walk (VDRW), which discourages the random particle to the nodes visited. Based on VDRW, they introduce the semi-supervised network embedding method ImVerde which consists of the context sampling and the balanced-batch sampling strategies to improve the quality of the node-context pairs. SPARC (Zhou et al. 2018a) obtains the imbalanced node embedding in a mutually way, which can jointly predict the minority class and the neighbor context in the networks. RSDNE (Wang et al.) explores the network embedding with completely-imbalanced labels. It learns the imbalanced node embedding by allowing the intra-class nodes on the same manifold in the embedding space and removing the known connections between the inter-class nodes.

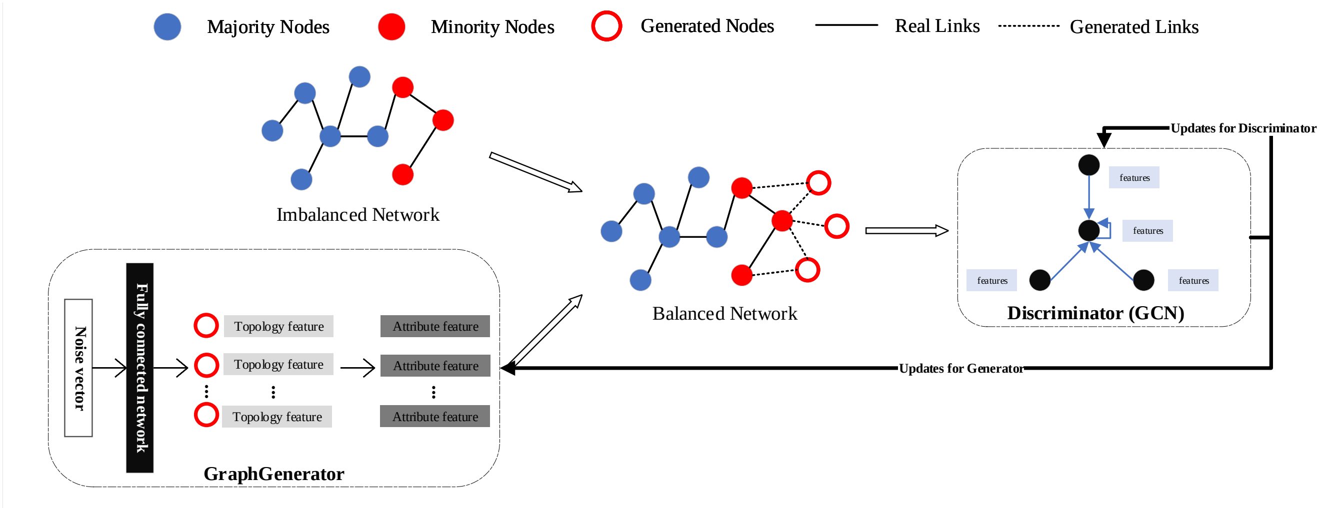

Figure 2: The architecture of ImGCN. The minority and majority nodes of original imbalanced network are represented by red and blue solid circles respectively, and the synthetic minority nodes generated by GraphGenerator are represented by red hollow circles in artificial synthetic classes balanced network. In addition, The links between real nodes are represented by solid lines, and the links between synthetic minority nodes and real minority nodes are represented by dashed lines.

Proposed Method

In this section, we first provide several needed concepts related to the proposed method. Then, we present our proposed method ImGAGN in detail. Finally, we analyze the time complexity of the proposed method.

Preliminary

Before presenting our proposed ImGAGN, we provide a brief introduction to the needed concepts for proposing our method.

- **Imbalanced network:** given an imbalanced network $\mathcal{G}_{im} = (V, E, A, X, C)$, where V is the set of n nodes, E is the set of edges, A is the adjacency matrix, $X \in R^{n \times f}$ is the node feature matrix with feature dimension f, and $C = \{c_{min}, c_{maj}\}$ is the set of node classes. $|c_{min}|$ and $|c_{maj}|$ represent the number of nodes in their classes. The network $\mathcal{G}_{im} = (V, E, A, X, C)$ is an imbalanced network if $|c_{min}|$ is far less than $|c_{maj}|$ (i.e., $|c_{min}| \ll |c_{maj}|$).
- **Imbalanced network embedding:** imbalanced network embedding aims at mapping the node $v_i \in V$ of an imbalanced network $\mathcal{G}_{im} = (V, E, A, X, C)$ into a continuous low-dimensional vector $\vec{h_i} \in R^d$ ($d \ll n$), such that the nodes with the same class label are closer than the nodes with the different class labels in the embedding space.
- **GANs:** GANs (Goodfellow et al. 2014; Gui et al. 2020) are a class of neural networks which consist of a generator and a discriminator. The key idea of generator G is that it aims at generating the fake data to simulate the real data distribution to confuse discriminator. The goal of discriminator D is to correctly classify both the real training data and fake data generated from generator G.

The GANs methods can be formulated as follows (Goodfellow et al. 2014):

$$\begin{aligned} \min_G \max_D V(D, G) = E_{x \sim p_{data}(x)}[\log D(x)] \\ + E_{z \sim p_z(z)}[\log(1 - D(G(z)))] \end{aligned} \tag{1}$$

where x is the real data obeying the distribution p_{data}, and z is the noise variable obeying the distribution p_z.

ImGAGN

To address the imbalanced classification problems on graph, we propose a novel GANs based imbalanced learning method, called ImGAGN, which incorporates GCN with a novel generator named GraphGenerator for graph structural data. GraphGenerator can effectively learn not only the node feature distribution but also the network structural distribution. The architecture of ImGAGN is shown in Figure 2.

GraphGenerator Unlike traditional GAN processing regular Euclidean data (e.g., images and text) which data is independent with each other, the generator only need to learn the data feature distribution. For graph structural data, because the data (i.e., nodes) is independent to each other, the generator need to learn not only data features distribution (e.g., the node features) but also network structure distribution (e.g., the node link relationships). In this paper, we propose a novel generator for graph data, call GraphGenerator, which can generate the node link relationships between the synthetic minority nodes and the real minority nodes, and the features of the synthetic minority nodes are obtained by aggregating the features of the real minority nodes.

The GraphGenerator $G_{graph} : Z \to F \times T$ is a fully connected network with the Softmax activation function in output layer, where Z is the noise space with d_z dimension, and F, T are network feature space and network structure space respectively. Specifically, for an imbalanced network

$\mathcal{G}_{im} = (V, E, A, X, C)$, let n_{maj} and n_{min} represent majority nodes number and minority nodes number respectively with $n = n_{maj} + n_{min}$. Let $n_g = n_{maj} - n_{min}$ represents the number of nodes needing to be generated for balancing the network classes distribution. Thus, the number of units in input layer is d_z, and the number of units in output layer is $d_o = n_g \times n_{min}$. For better understanding, we convert the output vector $\vec{o} \in R^{d_o}$ into the matrix form $O \in R^{n_g \times n_{min}}$, and each element $o_{ij} \in O$ is discretized into the $\{0, 1\}$ space by function Dis with the hyperparameter n_{min} as equation (2):

$$b_{ij} = Dis(o_{ij}) = \begin{cases} 1, & o_{ij} > \frac{1}{n_{min}} \\ 0, & o_{ij} \leq \frac{1}{n_{min}} \end{cases}, b_{ij} \in B \tag{2}$$

where $B \in \{0, 1\}^{n_g \times n_{min}}$ is the structure features of the generated minority nodes by GraphGenerator. Each row of B represents the link relationships between each generated minority node to all real minority nodes, where 1 represents link, and 0 represents unlink. X_g is the attributed features matrix of generated minority nodes by GraphGenerator, which is calculated by equation (3)

$$X_g = OX_{min} \tag{3}$$

where $X_{min} \subset X$ is the real minority node features matrix of the original imbalanced network $\mathcal{G}_{im}$.

The loss function of GraphGenerator is as equation (4).

$$\begin{aligned} \mathcal{L}_{gen} &= \mathcal{L}_{rf} + \mathcal{L}_{mi} + \mathcal{L}_{di} + \mathcal{L}_{re} \\ &+ \textstyle\sum_{i=1}^{n_g} -logPr(\widehat{y_i} = real|\vec{x_i}) \\ &+ \textstyle\sum_{i=1}^{n_g} -logPr(\widehat{y_i} = minority|\vec{x_i}) \\ &+ \frac{1}{|n_g|} \textstyle\sum_{i=1}^{n_g} \sum_{j=1}^{n_{min}} ||\vec{x_i} - \vec{x_j}||_2^2 \\ &+ \alpha||\Theta||_2^2 \end{aligned} \tag{4}$$

where this loss function consists of four terms. The first $\mathcal{L}_{rf}$ and second terms $\mathcal{L}_{mi}$ are the confusing discriminator loss over the generated minority data, in which $\widehat{y_i}$ denotes the output of the discriminator and $\vec{x_i}$ is the node feature vector. The third term $\mathcal{L}_{di}$ aims at making the generated minority nodes close to the real minority nodes. The last term $\mathcal{L}_{re}$ is regularizer, in which Θ is the set of training weights of GraphGenerator with regularization coefficient α.

Discriminator In this paper, we utilize the two-layer GCN (Kipf and Welling 2016) as our discriminator, and the input of GCN is the new network $\mathcal{G}_{bal} = (V^{'}, E^{'}, A^{'}, X^{'}, C^{'})$ with balanced classes distribution, where $V^{'}$ represents the new nodes set which consists of the nodes in $\mathcal{G}_{im}$ and the generated minority nodes by GraphGenerator, $E^{'}$ represents the new edges set which consists of the all edges in $\mathcal{G}_{im}$ and the generated edges by GraphGenerator, $A^{'}, X^{'}$ are the new adjacency matrix and feature matrix associated to $V^{'}$ respectively. $C^{'} = \{(real = 1, minority = 1), (real = 1, majority = 0), (fake = 0, minority = 1),$ $(fake = 0, majority = 0)\}$ represents the nodes label set. The goal of discriminator is to discriminate whether the nodes is generated by generator (i.e., fake) and whether the node is minority class. Therefore, we can utilize the GCN as a multi-label node classifier, and the output Y of GCN is calculated by equation (5) (Kipf and Welling 2016) as follows:

$$Y = softmax(\widehat{A^{'}}ReLU(\widehat{A^{'}}X^{'}\Omega^0)\Omega^1) \tag{5}$$

where $\widehat{A^{'}} = \widehat{D^{-\frac{1}{2}}}(\widehat{A^{'}} + I_N)\widehat{D^{-\frac{1}{2}}}$ is the pre-processing step following (Kipf and Welling 2016) with identity matrix I_N and $D_{ij} = \sum_j A_{ij}$. Ω^0 and Ω^1 are input-to-hidden and hidden-to-out weight matrices respectively. The loss function of discriminator is as equation (6):

$$\begin{aligned} \mathcal{L}_{dis} &= \mathcal{L}_{fa} + \mathcal{L}_{cl} + \mathcal{L}_{mm} + \mathcal{L}_{ree} \\ &+ \textstyle\sum_{i=1}^{n_g+n_{min}+n_{maj}} -[t_i log(Pr(\widehat{y_i} = fake|\vec{x_i})) \\ &+(1-t_i)log(1 - Pr(\widehat{y_i} = fake|\vec{x_i}))] \\ &+ \textstyle\sum_{i=1}^{n_g+n_{min}+n_{maj}} -[t_i log(Pr(\widehat{y_i} = minority|\vec{x_i})) \\ &+(1-t_i)log(1 - Pr(\widehat{y_i} = minority|\vec{x_i}))] \\ &- \textstyle\sum_{i=1}^{n_{min}} \sum_{j=1}^{n_{maj}} ||\vec{h_i} - \vec{h_j}||_2^2 \\ &+\beta||\Omega||_2^2 \end{aligned} \tag{6}$$

where this loss function consists of four terms. $t_i \in C^{'}$ and $\widehat{y_i} \in Y$ are the ground-truth label of node and output of GCN. The first term $\mathcal{L}_{fa}$ is the cross entropy loss to discriminate that the node is generated by generator or real node of the network. The second term $\mathcal{L}_{cl}$ is also the cross entropy loss to discriminate that the node is minority class or majority class. The third term $\mathcal{L}_{mm}$ aims at making the embeddings of the different class nodes are far away from each other. The last term $\mathcal{L}_{ree}$ is regularizer, in which Ω is the set of training weights of the discriminator with regularization coefficient β.

Algorithm 1 Training process of ImGAGN. λ_1 is the number of training steps to apply to the discriminator.

for number of training iterations **do**
  Generate n_g minority nodes with structure features B using equation (2)
  Generate the node attributed features X_g using equation (3)
  Synthesize the new balanced network $\mathcal{G}_{bal}$
  for λ_1 steps **do**
    Update $Discriminator$ by ascending its stochastic gradient $\nabla_\Omega \mathcal{L}_{dis}$
  end for
  Update $Generator$ by descending its stochastic gradient $\nabla_\Omega \mathcal{L}_{gen}$
end for

Time Complexity

The time complexity of Algorithm 1 is as follows. The complexity for updating generator is $O((L-1)n_g D^2 + n_g n_{min}^2)$, where L is the number of fully connected layers of generator, and D is the hidden layer dimension size of generator. The complexity for updating discriminator is $O(K|E|d + Knd^2)$, where K is the number of layers of GCN, $|E|$ is the number of edges, and d is the hidden layer dimension size of GCN. Therefore, the total time complexity of ImGAGN is $O((L-1)n_g D^2 + n_g n_{min}^2) + \lambda_1(K|E|d + Knd^2))$.

Table 1: Imbalanced binary node classification results. The best results are marked in bold.

Datasets	Metric \ Method	GCN	GraphSAGE	GCN-SMOTE	DeepWalk	Node2vec	LINE	SPARC	RECT	**ImGAGN**
Cora	Recall	0.7222	0.8611	0.8611	0.7500	0.5833	0.2222	0.6944	0.8889	**0.9722**
	Accuracy	0.9815	0.9871	0.9576	0.9539	0.9428	0.9317	0.9705	**0.9963**	**0.9963**
	F1 Score	0.8387	0.8986	0.7294	0.6835	0.5753	0.3019	0.7576	0.9714	**0.9722**
Citeseer	Recall	0.0200	0.2200	0.3600	0.1800	0	0	0.2400	0.4200	**0.9400**
	Accuracy	0.9261	0.9155	0.9140	0.9140	0.9140	0.9216	0.9306	0.9306	**0.9623**
	F1 Score	0.0392	0.2821	0.3871	0.2400	0	0	0.3429	0.4773	**0.7899**
Pubmed	Recall	0	0	0.5376	0.3006	0.3294	0.0982	0	0.4566	**0.9768**
	Accuracy	0.8657	0.9474	0.8921	0.9411	0.9496	0.9462	0.9474	0.9587	**0.9604**
	F1 Score	0	0	0.3438	0.3490	0.4071	0.1611	0	0.5320	**0.7222**
DBLP	Recall	0.0363	0	0.5273	0.3091	0	0	0	0.8182	**0.9455**
	Accuracy	0.9873	0.9834	0.9057	0.9904	0.9865	0.9869	0.9869	**0.9971**	**0.9971**
	F1 Score	0.0701	0	0.1289	0.4595	0	0	0	0.8824	**0.8966**
Wiki	Recall	0	0	0	0	0.5000	0	0	0.5000	**0.7000**
	Accuracy	0.9959	0.9959	0.9959	0.9959	0.9876	0.9959	0.9959	0.9979	**0.9988**
	F1 Score	0	0	0	0	0.2500	0	0	0.6667	**0.7000**

Experiment

In this section, we conduct the experiments on five real-world datasets to validate the effectiveness of the proposed method. Include the imbalanced binary node classification task, network layouts task and parameters sensitivity analysis task.

Experimental setup

Datasets: We conduct experiments on several node classification datasets including Cora (McCallum et al. 2000), Citeseer (Giles, Bollacker, and Lawrence 1998), Pubmed (Sen et al. 2008), DBLP (Tang et al. 2008) and Wiki (Sen et al. 2008) datasets. The statistic information of the datasets is summarized in Table 2.

Table 2: The statistic information of the network datasets

Name	Nodes	Edges	Classes	Features	Ratio of the minority class
Cora	2708	5429	7	1433	6.65%
Citeseer	3312	4715	6	3703	7.52%
Pubmed	16452	39308	3	500	5.25%
DBLP	20783	58188	10	1000	1.31%
Wiki	2405	17981	17	4973	0.37%

- Cora (McCallum et al. 2000), Citeseer (Giles, Bollacker, and Lawrence 1998), Pubmed (Sen et al. 2008), and DBLP (Tang et al. 2008) are the citation network datasets which consist of the nodes representing papers and the edges representing citation relationship between two papers. For each paper, a sparse bag-of-words vector is utilized as the feature vector. For these four original datasets, the node classes (labels) are defined according to the several research topics, and each class has the roughly equal number of node. In our experiments, for validating the effectiveness of the proposed method on the imbalanced networks, following (Zhou et al. 2018b), all these four balanced networks are reconstructed as the binary imbalanced networks by setting the smallest class as the minority class and the residual classes as the majority class. Specifically, taking Cora dataset for an example, there are seven classes[1] in total. Thus, the smallest class Rule Learning (6.65%) is used as the minority class, and the residual classes (93.35%,) are used as majority class.
- Wiki (Sen et al. 2008) is a dataset consisting of a set of Wikipedia pages. Each node represents a web page, and each edge represents a hyperlink between two pages. The node labels are defined according to the several topics, and the node features are extracted from the TF-IDF matrix. In addition, we also reconstruct this dataset in a similar way for the imbalanced network analysis.

Comparison Algorithms:

- **GCN**: Graph convolutional network (GCN) (Kipf and Welling 2016) is the most representative GNN method which obtains the node embedding by aggregating the neighbor nodes' features.
- **GraphSAGE**: GraphSAGE (Hamilton, Ying, and Leskovec 2017) is also a representative GNN method. Unlike GCN taking the full-size neighbor nodes to obtain the node embedding, GraphSAGE adopts a fixed number of neighbor nodes for each target node to save the memory.
- **GCN-SMOTE**: Synthetic minority oversampling technique (SMOTE) (Chawla et al. 2002) is the most frequently used method to address the imbalanced classification problem by generating synthetic samples from existing minority samples. In this paper, in order to fully show the performance of the GNN methods, we incorporate the SMOTE technique into GCN for improving its performance on imbalanced network embedding problem.

[1]Neural Networks: 30.21%, Rule Learning: 6.65%, Reinforcement Learning: 8.01%, Probabilistic Method: 15.73%, Theory: 12.96%, Genetic Algorithm: 15.44%, and Case Based: 11.00%.

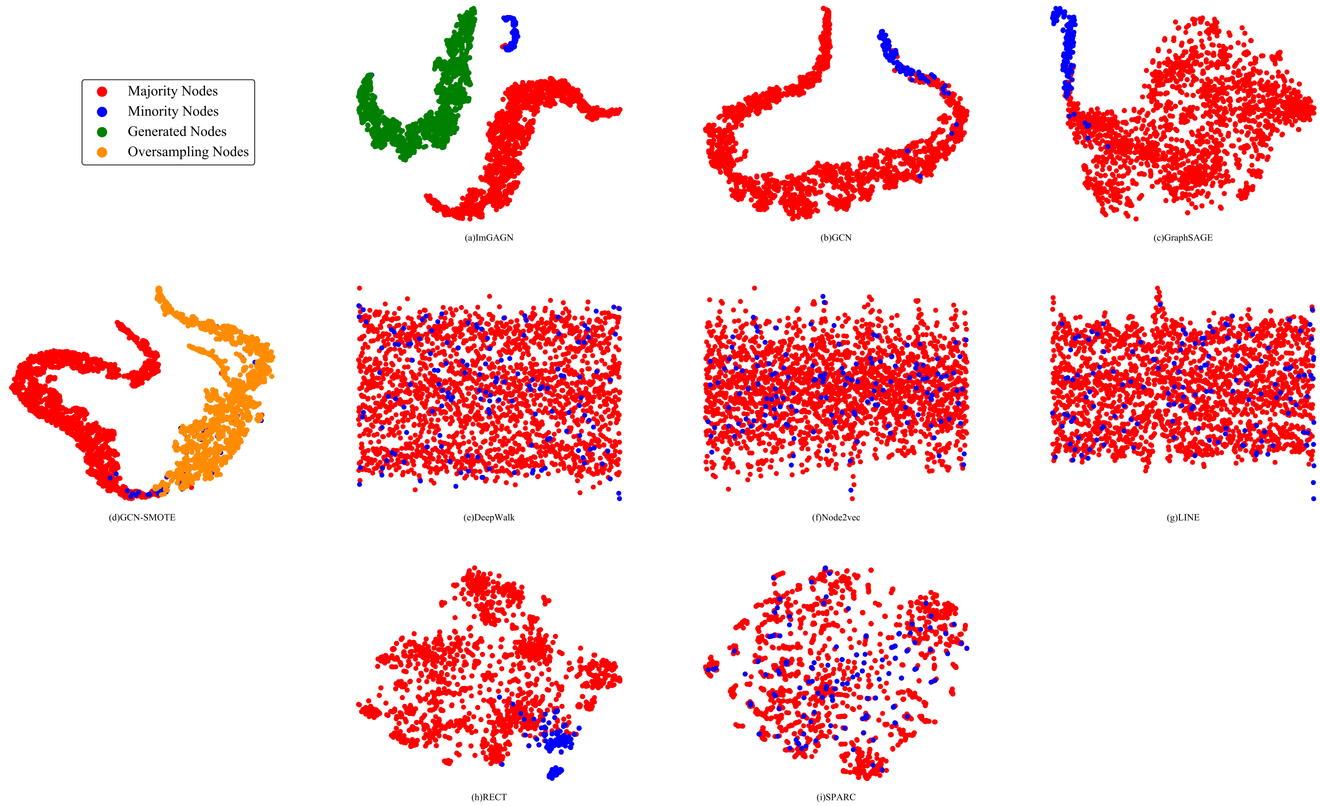

Figure 3: The 2-dimensional imbalanced network layout with t-SNE on Cora dataset. The red circles represent the majority nodes of the original networks. The blue circles represent the minority nodes of the original networks. The yellow circles represent the minority nodes generated by SMOTE. The green circles represent the minority nodes generated by the proposed ImGAGN.

- **DeepWalk**: DeepWalk (Perozzi, Al-Rfou, and Skiena 2014) is the most representative unsupervised network embedding method which adopts the random walk over the network to sample a set of network paths, and the neural language model (SkipGram) is applied to these network paths to obtain the node embedding.
- **Node2vec**: Node2vec (Grover and Leskovec 2016) is also an unsupervised network embedding method which obtains the node embedding by using a biased random walk strategy to preserve the homophily and structural equivalence relationships in the networks.
- **LINE**: LINE (Tang et al. 2015) obtains the network embedding by simultaneously optimizing the first-order and second-order proximities of the networks.
- **SPARC**: SPARC (Zhou et al. 2018a) is an imbalanced network embedding method. It obtains the imbalanced embedding in a mutually way, which can jointly predict the minority class and the neighbor context in the networks.
- **RECT**: RECT (Wang et al. 2020) is the state-of-the-art imbalanced network embedding method which is a variant of GNN. It obtains the imbalanced network embedding by learning the knowledge of class-semantic information in the networks.

Parameters: All the codes we used are provided by authors. For GCN, following (Kipf and Welling 2016), the number of layers of the networks is set $K = 2$. For GraphSAGE, we set $K = 2, S_1 = 5, S_2 = 5$ according to the author suggesting. For GCN-SMOTE, the number of generated minority samples by SMOTE is equal to the difference between the majority and minority nodes of the training set. For DeepWalk, we adopt the default hyperparameters (i.e., window size $win = 10$, walk length $len = 40$ and the number of walks $t = 90$). For Node2vec, we optimize its hyperparameters by a grid search over $p, q \in \{0.25, 0.50, 1, 2, 4\}$. For LINE, the hyperparameter negative samples $ns = 5$. For SPARC, the length of random walk sequences $\mu = 10$. Moreover, the embedding dimension of unsupervised network embedding methods (i.e., DeepWalk, Node2vec and LINE) are set as $d = 128$, and the logistic regression classifier is employed to evaluate the node embedding. For semi-supervised network embedding methods (i.e., GCN, GCN-SMOTE, GraphSAGE, SPARC and RECT), we use the outputs of their last hidden layer as the node embedding (the embedding dimension is also 128).

The hyperparameters of our proposed method ImGAGN are set as follows. For generator, it consists of 3 fully connected layers with 100 units in input layer and 200 units in hidden layer. The number of units of output layer is equal to the difference between the majority class and minority class of the training set. Tanh is utilized as the activation function. For discriminator, it consists of the two-layer GCN followed by a softmax function, and ReLU (Glorot, Bordes, and Bengio 2011) is utilized as the activation function. In addition, we perform generator and discriminator updates in $1 : 100$ ratio, and Adam SGD optimizer (Kingma and Ba 2017) is utilized as the optimizer throughout the experiments.

Repeatability: All the data and code of our algorithm are available in supplementary material, and we will release them after the paper being published. In addition, all the methods are run on a single machine with 14 CPU cores at 2.60GHZ and 2 Tesla P100 GPU with 32G memory using 1 thread.

Imbalanced binary node classification

To validate the effectiveness of the proposed method, we first conduct imbalanced binary node classification experiment on the five real-world network datasets. Three common classification metrics are used to evaluate the performance for all algorithms. Include: (1) accuracy, which measures the ratio of correctly classified nodes of all test nodes (i.e., the majority nodes and minority nodes). (2) recall, which measures the ratio of correctly classified nodes of all minority test nodes. (3) F1 score, which is widely used to balance between the precision and the recall. The train set, validation set and test set are randomly split as ratio 7:1:2. We run experiments 10 times and use average scores for each metric. The experimental results are shown in Table 1.

From experimental results, in general, we can observe that: (1) The proposed method ImGAGN can outperform all the comparison algorithms across all the datasets for all the evaluation metrics. Especially for the recall, which measures the algorithms effectiveness on minority class, our algorithm is better than the best competitor RECT. For example, ImGAGN is about 52% higher on Citeseer and Pubmed. (2) The proposed method ImGAGN can well process the extreme imbalanced network, such as Wiki (with only 0.37% ratio of the minority class). It shows that our algorithm could be well applied to the networks with very little minority nodes.

Network layout

To validate the learned node embedding whether can well discriminate the minority and majority nodes, we visualize the network layout in the embedding space, and we take Cora dataset for an example. Specifically, we firstly learn the nodes embedding in a 128-dimensional vector space for different network embedding methods, and then employ the t-SNE (Maaten and Hinton 2008) to map the 128-dimensional into the 2-dimensional space for visualization. The experimental results are shown in Figure 3.

From the experimental results, in general, we can observe that: (1) Semi-supervised network embedding methods (i.e., GCN, GCN-SMOTE, GraphSAGE, SPARC, RECT and ImGAGN) can better discriminate the majority and the minority classes than the unsupervised network embedding methods (i.e., DeepWalk, Node2vec and LINE). One explanation is that semi-supervised methods can utilize the label information. (2) The proposed ImGAGN can better discriminate the majority and minority classes than other semi-supervised methods, that is, ImGAGN can obtain a clear decision boundary between the two classes.

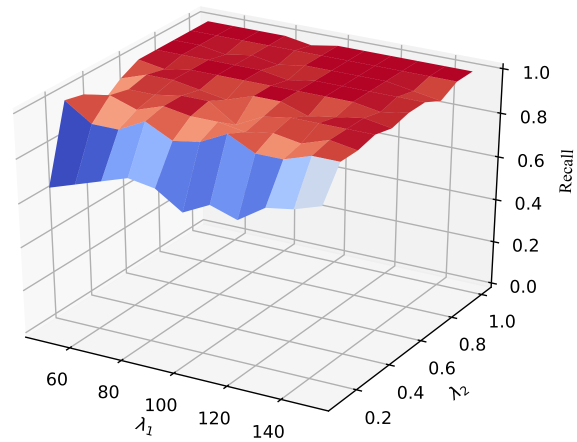

Figure 4: Parameter sensitivity analysis.

Parameters sensitivity analysis

The crucial hyperparameters of the ImGAGN are λ_1 (i.e., the number of training steps to apply to the discriminator) and λ_2 (i.e., the ratio of the number of generated minority nodes to the number of majority node is $\lambda_2 : 1$). We report the recall of ImGAGN on Citeseer dataset. The experimental results are shown in Figure 4.

From experimental results, in general, we can observe that: The recall is increasing with the values of λ_2 increasing. One explanation is that when λ_2 is small, the network is still imbalanced, which leads to bad performance. Particular speaking, we found the proposed method ImGAGN could achieve high performance with $\lambda_1 > 50$ and $\lambda_2 > 0.1$.

Conclusions

In this paper, to address the imbalanced network embedding problem, we proposed a semi-supervised network embedding method ImGAGN, which utilized a novel GraphGenerator to simulate the distribution of the minority class and generated minority class nodes to balance the network classes distribution. Then a GCN was used to train to discriminate between the minority and majority classes in the synthetic balanced networks classes. The empirical evaluation on five real-world datasets including an extreme imbalanced network demonstrated that the proposed ImGAGN can outperform the state-of-the-art imbalanced network embedding algorithms.

For future work, we plan to validate the proposed algorithm on large-scale imbalanced network.

References

2018. Effective data generation for imbalanced learning using conditional generative adversarial networks. *Expert Systems with Applications* 91: 464–471. ISSN 0957-4174. doi:10.1016/j.eswa.2017.09.030. URL https://www.sciencedirect.com/science/article/abs/pii/S0957417417306346. Publisher: Pergamon.

Akbani, R.; Kwek, S.; and Japkowicz, N. 2004. Applying support vector machines to imbalanced datasets. In *European conference on machine learning*, 39–50. Springer.

Cai, H.; Zheng, V. W.; and Chang, K. C.-C. 2017. A Comprehensive Survey of Graph Embedding: Problems, Techniques and Applications. *arXiv:1709.07604 [cs]* URL http://arxiv.org/abs/1709.07604. ArXiv: 1709.07604.

Chawla, N. V.; Bowyer, K. W.; Hall, L. O.; and Kegelmeyer, W. P. 2002. SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research* 16: 321–357. ISSN 1076-9757. doi:10.1613/jair.953. URL https://www.jair.org/index.php/jair/article/view/10302.

Douzas, G.; and Bacao, F. 2018. Effective data generation for imbalanced learning using conditional generative adversarial networks. *Expert Systems with applications* 91: 464–471.

Elkan, C. 2001. The foundations of cost-sensitive learning. In *International joint conference on artificial intelligence*, volume 17, 973–978. Lawrence Erlbaum Associates Ltd.

Fortunato, S. 2010. Community detection in graphs. *Physics reports* 486(3-5): 75–174.

Giles, C. L.; Bollacker, K. D.; and Lawrence, S. 1998. CiteSeer: An automatic citation indexing system. In *Proceedings of the third ACM conference on Digital libraries*, 89–98.

Glorot, X.; Bordes, A.; and Bengio, Y. 2011. Deep Sparse Rectifier Neural Networks 9.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

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 - KDD '16*, 855–864. San Francisco, California, USA: ACM Press. ISBN 978-1-4503-4232-2. doi:10.1145/2939672.2939754. URL http://dl.acm.org/citation.cfm?doid=2939672.2939754.

Gui, J.; Sun, Z.; Wen, Y.; Tao, D.; and Ye, J. 2020. A review on generative adversarial networks: Algorithms, theory, and applications. *arXiv preprint arXiv:2001.06937* .

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. In *Advances in neural information processing systems*, 1024–1034.

Han, H.; Wang, W.-Y.; and Mao, B.-H. 2005. Borderline-SMOTE: a new over-sampling method in imbalanced data sets learning. In *International conference on intelligent computing*, 878–887. Springer.

He, H.; and Garcia, E. A. 2009. Learning from Imbalanced Data. *IEEE Transactions on Knowledge and Data Engineering* 21(9): 1263–1284.

He, J.; Liu, Y.; and Lawrence, R. 2008. Graph-based Rare Category Detection. . . . 6.

Johnson, J. M.; and Khoshgoftaar, T. M. 2019. Survey on deep learning with class imbalance. *Journal of Big Data* 6(1): 27.

Kingma, D. P.; and Ba, J. 2017. Adam: A Method for Stochastic Optimization. *arXiv:1412.6980 [cs]* URL http://arxiv.org/abs/1412.6980. ArXiv: 1412.6980.

Kipf, T. N.; and Welling, M. 2016. Semi-Supervised Classification with Graph Convolutional Networks. *arXiv:1609.02907 [cs, stat]* URL http://arxiv.org/abs/1609.02907. ArXiv: 1609.02907.

Liu, X.-Y.; Wu, J.; and Zhou, Z.-H. 2008. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 39(2): 539–550.

Maaten, L. v. d.; and Hinton, G. 2008. Visualizing data using t-SNE. *Journal of machine learning research* 9(Nov): 2579–2605.

McCallum, A. K.; Nigam, K.; Rennie, J.; and Seymore, K. 2000. Automating the construction of internet portals with machine learning. *Information Retrieval* 3(2): 127–163.

Montahaei, E.; Ghorbani, M.; Baghshah, M. S.; and Rabiee, H. R. 2018. Adversarial Classifier for Imbalanced Problems. *arXiv:1811.08812 [cs, stat]* URL http://arxiv.org/abs/1811.08812. ArXiv: 1811.08812.

Perozzi, B.; Al-Rfou, R.; and Skiena, S. 2014. DeepWalk: Online Learning of Social Representations. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '14* 701–710. doi:10.1145/2623330.2623732. URL http://arxiv.org/abs/1403.6652. ArXiv: 1403.6652.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Galligher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI magazine* 29(3): 93–93.

Shamsolmoali, P.; Zareapoor, M.; Shen, L.; Sadka, A. H.; and Yang, J. 2020. Imbalanced Data Learning by Minority Class Augmentation using Capsule Adversarial Networks. *arXiv:2004.02182 [cs, stat]* URL http://arxiv.org/abs/2004.02182. ArXiv: 2004.02182.

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. *Proceedings of the 24th International Conference on World Wide Web - WWW '15* 1067–1077. doi:10.1145/2736277.2741093. URL http://arxiv.org/abs/1503.03578. ArXiv: 1503.03578.

Tang, J.; Zhang, J.; Yao, L.; Li, J.; Zhang, L.; and Su, Z. 2008. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD* *international conference on Knowledge discovery and data mining*, 990–998.

Ting, K. M. 2002. An instance-weighting method to induce cost-sensitive trees. *IEEE Transactions on Knowledge and Data Engineering* 14(3): 659–665.

Wang, Z.; Ye, X.; Wang, C.; Cui, J.; and Yu, P. 2020. Network Embedding with Completely-imbalanced Labels. *IEEE Transactions on Knowledge and Data Engineering* 1–1. ISSN 1558-2191. doi:10.1109/TKDE.2020.2971490. Conference Name: IEEE Transactions on Knowledge and Data Engineering.

Wang, Z.; Ye, X.; Wang, C.; Wu, Y.; Wang, C.; and Liang, K. ???? RSDNE: Exploring Relaxed Similarity and Dissimilarity from Completely-imbalanced Labels for Network Embedding 8.

Wu, J.; He, J.; and Liu, Y. 2018. ImVerde: Vertex-Diminished Random Walk for Learning Network Representation from Imbalanced Data. *arXiv:1804.09222 [cs]* URL http://arxiv.org/abs/1804.09222. ArXiv: 1804.09222.

Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; and Yu, P. S. 2019. A Comprehensive Survey on Graph Neural Networks. *arXiv:1901.00596 [cs, stat]* doi:10.1109/TNNLS.2020.2978386. URL http://arxiv.org/abs/1901.00596. ArXiv: 1901.00596.

Zachary, W. W. 1977. An Information Flow Model for Conflict and Fission in Small Groups. *Journal of Anthropological Research* 33(4): 452–473. URL http://www.jstor.org/stable/3629752.

Zhou, D.; He, J.; Yang, H.; and Fan, W. 2018a. SPARC: Self-Paced Network Representation for Few-Shot Rare Category Characterization. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining - KDD '18*, 2807–2816. London, United Kingdom: ACM Press. ISBN 978-1-4503-5552-0. doi:10.1145/3219819.3219968. URL http://dl.acm.org/citation.cfm?doid=3219819.3219968.

Zhou, D.; He, J.; Yang, H.; and Fan, W. 2018b. Sparc: Self-paced network representation for few-shot rare category characterization. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2807–2816.