GraphSR: A Data Augmentation Algorithm for Imbalanced Node Classification

Mengting Zhou[1,2], Zhiguo Gong[1,2*]

[1]State Key Laboratory of Internet of Things for Smart City, University of Macau, Macao
[2] Guangdong-Macau Joint Laboratory for Advanced and Intelligent Computing
yb97402@um.edu.mo, fstzgg@um.edu.mo

Abstract

Graph neural networks (GNNs) have achieved great success in node classification tasks. However, existing GNNs naturally bias towards the majority classes with more labelled data and ignore those minority classes with relatively few labelled ones. The traditional techniques often resort over-sampling methods, but they may cause overfitting problem. More recently, some works propose to synthesize additional nodes for minority classes from the labelled nodes, however, there is no any guarantee if those generated nodes really stand for the corresponding minority classes. In fact, improperly synthesized nodes may result in insufficient generalization of the algorithm. To resolve the problem, in this paper we seek to automatically augment the minority classes from the massive unlabelled nodes of the graph. Specifically, we propose *GraphSR*, a novel self-training strategy to augment the minority classes with significant diversity of unlabelled nodes, which is based on a Similarity-based selection module and a Reinforcement Learning(RL) selection module. The first module finds a subset of unlabelled nodes which are most similar to those labelled minority nodes, and the second one further determines the representative and reliable nodes from the subset via RL technique. Furthermore, the RL-based module can adaptively determine the sampling scale according to current training data. This strategy is general and can be easily combined with different GNNs models. Our experiments demonstrate the proposed approach outperforms the state-of-the-art baselines on various class-imbalanced datasets.

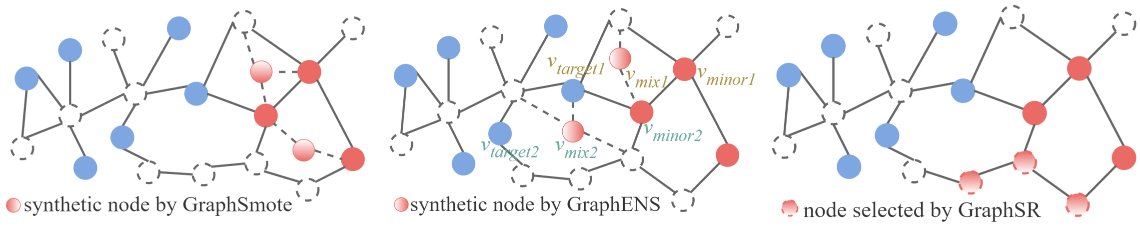
Introduction

Graph is regarded as one of the most powerful models for describing complex relationships between objects in various fields, such as natural language processing (Yao, Mao, and Luo 2019), computer vision (Chen et al. 2019), and recommendation systems (Wu et al. 2019). As the result, corresponding techniques for graph data analytics are receiving significant attention from the community. And GNNs (Graph Neural Networks) is one of the most successful techniques for node analysis. In principle, GCN (Kipf and Welling 2016) aggregates node features in the spectral space using Laplacian matrices, while GraphSAGE (Hamilton, Ying, and Leskovec 2017) aggregates features from node neighbors directly in the spatial domain. However, most existing GNNs are trained under the assumption that the node classes are balanced. Unfortunately, this assumption is not true in many real-world situations where some classes may have significantly few nodes than others in the training process. For example, the number of fraudsters in a social network is much smaller than that of benign ones in the fraud detection task (Liu et al. 2021). The class-imbalanced problem may cause the algorithm to bias towards the majority classes and ignore the minority classes in the representation learning. Therefore, it is challenging to apply GNNs directly to many real-world class-imbalanced graphs.

In fact, imbalance problem has been tackled for a long time and some significant progresses have been made in the area of machine learning. Generally speaking, the solutions can be summarized into three streams: data-level approaches, algorithm-level approaches, and hybrid approaches. Data-level approaches attempt to balance the class distribution by pre-processing training samples with over-sampling (Chawla et al. 2002) or under-sampling (Kubat, Matwin et al. 1997) techniques; algorithm-level approaches take misclassification costs into consideration (Ling and Sheng 2008) or modify the loss function (Cui et al. 2019) to alleviate the impact of class-imbalanced issue; and hybrid approaches combine above both (Batista, Prati, and Monard 2004). However, either data based or algorithm based method still solely relies on those labelled training data, therefore, inherently suffers from overfitting problem caused by the extensively reusing some samples or raising their weights.

Actually, graph data naturally present topological structures of nodes, which can be used to generate some virtual nodes as data augmentation for the training algorithm. In this respect, recently GraphSMOTE (Zhao, Zhang, and Wang 2021) extend SMOTE (Chawla et al. 2002) to perform interpolation between two minority nodes in the embedding space to synthesize new samples, and exploit an edge predictor to determine the neighborhood of the synthetic samples as shown in Fig.1a. Nevertheless, such generated nodes only rely on the minority nodes, which could not effectively extend the minority classes outwards (can still cause overfitting). To further deal with this, GraphENS (Park, Song, and Yang 2022) synthesize new minority nodes with their one-hop neighbors by mixing some existing minority node

*Corresponding Author.



(a) GraphSMOTE (b) GraphENS (c) GraphSR

Figure 1: Here is a graph with limited labelled nodes and massive unlabelled nodes, blue nodes denote majority class, red nodes denote minority class and blank nodes are unlabelled nodes. (a) synthetic nodes are generated by two minority nodes and edges are generated by an edge predictor. (b) mixed nodes and their one-hop neighbors are generated by minor node and target node. (c) displays supplemental unlabelled nodes for minority class by GraphSR.

from other classes, thus, can enrich the diversity of those minority classes as shown in Fig.1b. However, those synthetic nodes, which are generated under the subjectively designed mixing ratio between the minority nodes and other nodes, may not reveal the real situation of the underlying data nature, thus, could damage the results if the mixing ratio is not properly set. In addition, both of the above methods conduct over-sampling on minority classes heuristically with a fixed proportion, which could not be generalized across different datasets. The over-sampling ratio should be well-elaborated, otherwise the performance of majority classes will degrade when the ratio is set too large for minority classes.

As a matter of fact, the previous works fail to utilize the rich information from the massive unlabelled nodes of the graph, which are valuable resources for generating more promising data to augment the minority classes. In the computer vision area, CReST (Wei et al. 2021) experimentally finds the phenomenon that minority classes suffer from low recall but achieve surprisingly high precision, thus minority pseudo-labels are less risky to supplement the training set. However, we empirically find that there are still many misclassified nodes in minority classes in the node classification task on graph. Applying CReST directly on graph does not work well since the minority pseudo-labels are not reliable enough, and there is no mechanism to restrict these noisy unlabelled samples from being supplemented to the training set. Wrongly augmented data may damage the performance of the classification.

Motivated by above discussions, in this paper we design a novel data augmentation algorithm for minority classes, called *GraphSR*, as shown in Fig.1c, where a reinforcement learning algorithm is employed to optimize the strategy for selecting those unlabelled data to augment the minority classes.

Specifically, we firstly pre-train a baseline GNN model with labelled data, which can then generate pseudo-labels for those unlabelled data. Nevertheless, the baseline GNN model trained on imbalanced data could be biased towards majority classes, resulting in a poor prediction of unlabelled data. To tackle the problem, instead of supplementing the minority classes by randomly selecting some unlabelled nodes solely according to their pseudo-labels, GraphSR firstly utilizes a similarity-based selection module to filter out the most similar unlabelled nodes for each minority class, which aims to efficiently discover potential nodes from the numerous unlabelled nodes and maintains a pool of potential data to augment those minority classes. In the second step, in order to reduce the impact of noisy nodes, we design another module to adaptively choose the informative and reliable nodes from the candidate set through a reinforcement learning technique, named RL-based selection module. In practice, GraphSR trains a selector as an agent to decide which node in the candidate set to be preserved, and then the action is evaluated by the environment with the improved classifier trained using the augmented dataset, the reward is assigned based on the performance of a class balanced validation set. With the two-step selection, GraphSR can obtain the optimal unlabelled nodes to supplement the imbalanced training data. In this way, we can use the new training set to train an unbiased GNN classifier.

We summarize the main contributions as follows:

- We propose to study the class-imbalanced node classification problem in the semi-supervised setting, where numerous unlabelled nodes can be exploited to supplement the minority classes.
- We design a novel data augmentation strategy, GraphSR, to efficiently sample the informative and reliable unlabelled nodes to enhance the diversity of the minority classes. The proposed method can adaptively determine the sampling scale based on the current training data, making it more generalizable to different datasets.
- Experimental results on several datasets show that the proposed approach outperforms all the baselines. More importantly, the technique can be injected into any of the GNNs algorithms.

Related Work

Class Imbalanced Learning

Class imbalanced representation learning is a classical topic in machine learning domain and has been well-studied (He and Garcia 2009). The goal is to train an unbiased classifier on a labelled dataset with a class-imbalanced distribution, where majority classes have significantly more samples and minority classes have fewer samples. Prominent works include re-weighting and re-sampling approaches.

Re-weighting approaches try to modify the loss function by raising the weights of minority classes (Lin et al. 2017; Cui et al. 2019), or expanding the margins on minority classes (Cao et al. 2019; Liu et al. 2019; Menon et al. 2020). Re-sampling approaches attempt to balance the data distributions by pre-processing training samples deliberately, such as over-sampling minority classes (Chawla et al. 2002), under-sampling majority classes (Kubat, Matwin et al. 1997), and a combination of both (Batista, Prati, and Monard 2004). With the improvement of neural network, re-sampling strategies augment the minority classes through not only sampling techniques (Liu et al. 2020), but also generation idea (Kim, Jeong, and Shin 2020; Wang et al. 2021). The typical method SMOTE (Chawla et al. 2002) generates new samples by using interpolating technique over some minority samples and their nearest neighbors from the same class. While other works (Kim, Jeong, and Shin 2020; Wang et al. 2021) synthesize minority samples through transferring the common knowledge from majority classes. However, most existing methods are devoted to i.i.d. data, and can not be directly utilized to graph-based data, where the relationships among objects should be considered.

Graph Neural Networks

Graph neural networks (GNNs) are firstly proposed in 2005 (Gori, Monfardini, and Scarselli 2005). With the rapid development of deep learning, the techniques have achieved enormous success in non-Euclidean structured data. Generally speaking, GNNs follow a message-passing scheme to recursively embed a node with its neighbors into a continuous and low-dimensional space (Gilmer et al. 2017). GNN techniques can be divided into two categories: spectral-based methods and spatial-based methods. The spectral-based methods often apply the Laplacian matrix decomposition of the entire graph to collect nodes information (Defferrard, Bresson, and Vandergheynst 2016; Kipf and Welling 2016; Bianchi, Grattarola, and Alippi 2020), while the spatial-based methods employ the topological structure of the graph directly and aggregate nodes features based on the topological information of the graph (Veličković et al. 2017; Hamilton, Ying, and Leskovec 2017; You, Ying, and Leskovec 2019).

For node classification task, there are some works proposed to deal with class-imbalanced issue. DR-GCN (Shi et al. 2020) utilizes two types of regularization with class-conditioned adversarial training and latent distribution constraints on unlabelled nodes to train an unbiased classifier. GraphSMOTE (Zhao, Zhang, and Wang 2021) extends SMOTE to the embedding space and combines it with edge generation to synthesize minority nodes. ImGAGN (Qu et al. 2021) generates a set of synthetic minority nodes by modelling a generator to simulate both the attribute and topological distributions of the whole minority class. PC-GNN (Liu et al. 2021) devises a label-balanced sampler to construct the sub-graphs, and chooses neighbors for each node in the sub-graphs by a neighborhood sampler for training. GraphENS (Park, Song, and Yang 2022) synthesize the features and neighbors for minority nodes by mixing minority nodes and target nodes from other classes to avoid overfitting. For the over-sampling works, GraphSMOTE and ImGAGN only adopt the identical minority nodes to synthesize new samples, it is prone to be overfitting, and the diversity is limited. Although GraphENS utilizes the nodes from both minority classes and other classes when generating new samples, it changes graph structure and is hard to find out the optimal mixing ratio and neighbors, which may induce some noise and impact the performance instead. Moreover, these methods over-sample the minority classes with a fixed ratio and fail to take full advantage of the abundant information available from unlabelled nodes. In our work, GraphSR supplements the minority classes with unlabelled nodes to enrich the diversity and determines the sampling scale adaptively.

Problem Definition

In this work, we target at semi-supervised class-imbalanced node classification on graphs, which is with a small ratio of labelled nodes and large amount of unlabelled ones. We are going to use the limited number of labelled nodes for training a classifier, which is tested on the nodes from the same graph. In our setting, each node belongs to only one class, and the distribution of classes is imbalanced, that is the majority classes have significantly more samples than those minority classes in the training set.

Formally, an attributed graph is defined as $\mathcal{G} = (V, E, X)$, where V is the nodes set, E is the set of edges, and $X \in \mathbb{R}^{|V| \times d}$ denotes the attribute matrix where each row represents a d-dimensional attribute of the corresponding node. $\mathcal{N}(v) = \{v' \in V | \{v', v\} \in E\}$ is the set of the neighboring nodes that directly connect to v. Each node only belongs to one class y, and there are totally m classes in the graph. During training, only a subset of nodes V_L with their corresponding labels Y_L are available, and the unlabelled set of nodes is denoted as U. For the labelled nodes, we denote their class distribution as $\{C_1, ..., C_m\}$, where C_i is the node set of i-th class. Besides, we introduce an imbalance ratio ρ to measure the degree of imbalance, $\rho = \frac{min_i(|C_i|)}{max_i(|C_i|)}$.

Given $\mathcal{G}$ with a labelled node set V_L, which is class imbalanced, we aim to train an unbiased classifier f that can work well for the entire classes, with the help of unlabelled nodes in U.

Methodology

In this section, we present the details of the proposed GraphSR, which is based on the self-training technique. In fact, self-training (Scudder 1965) is a classical method which is widely used in semi-supervised learning. In principle, the algorithm iteratively trains a model on the available labelled set and uses the trained model to generate pseudo-labels for those unlabelled data; then, it selects confident samples from the unlabelled set to combine with the training set to further retrain the model, until converging.

To accommodate the class-imbalanced issue in graphs, with the idea of self-training, we propose two kinds of components to adaptively select informative and reliable nodes from unlabelled data to supplement the minority classes, as demonstrated in Fig.2. First, GraphSR trains a GNN model

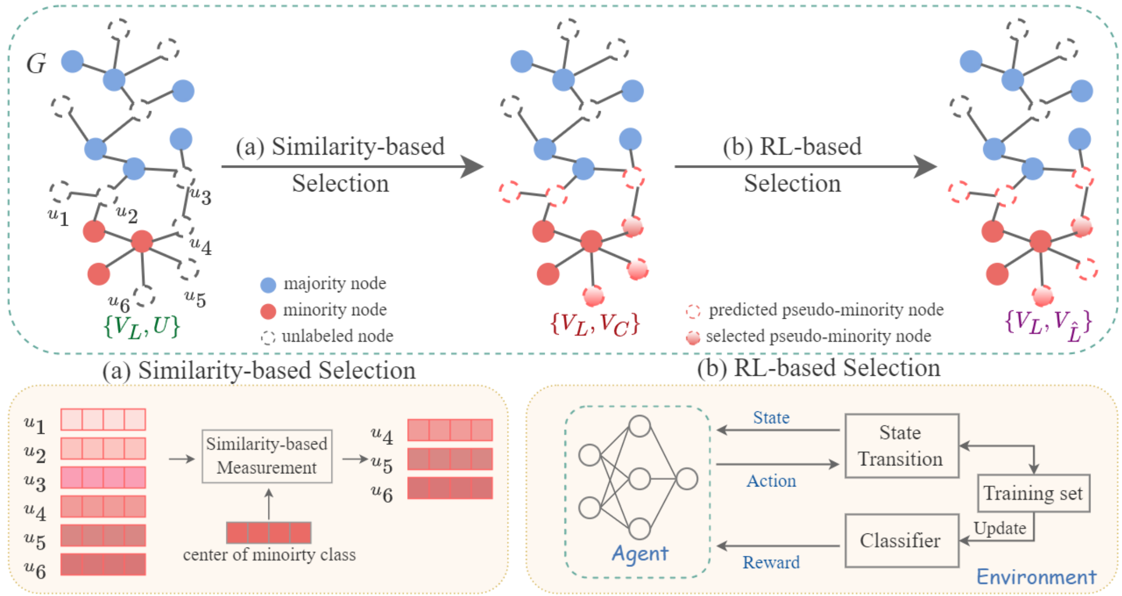


Figure 2: Overall pipeline of GraphSR

based on the labelled set V_L and generates pseudo-labels for unlabelled nodes in U. Then a similarity-based selection module is designed to identify the unlabelled nodes which are most similar to the minority nodes, to filter out candidate nodes set V_C for minority classes. Next, GraphSR utilizes a module of reinforcement learning to adaptively choose the informative and reliable nodes to get a proper supplemental set $V_{\hat{L}}$, which can optimally and effectively enrich the diversity of minority classes and finally augment the training set. With the augmented training data $\{V_L, V_{\hat{L}}\}$, we can train a class-balanced node classifier. In the following, we will show the details of each component.

Similarity-Based Selection

One simple way to augment the minority classes in the semi-supervised setting is to find the similar unlabelled nodes from the original graph. In general, node representations derived from GNNs can reflect the inter-class and intra-class relationship of nodes, that is, nodes of the same class will be closer in the embedding space, while nodes from different classes should be farther away in the latent space. Therefore, rather than comparing nodes directly using their raw attributes, we train a GNN model on the labelled set to learn the node representations, which can simultaneously capture both feature property and topological information of nodes. Specifically, we train a GNNs classifier g on imbalanced training set $\{V_L, Y_L\}$, the message passing and fusing process of which is formulated as:

$$h_v^k = \sigma\left(W^k \cdot \mathrm{CAT}(h_v^{k-1}, \mathrm{AGG}(\{h_{v'}^{k-1}, \forall v' \in \mathcal{N}(v)\}))\right) \tag{1}$$

where $\mathrm{AGG}(\cdot)$ denotes the aggregation function that aggregate the information of neighborhood $\mathcal{N}(v)$, $\mathrm{CAT}(\cdot)$ concatenates the node representations and the neighboring information, W is the learnable weight parameters, and σ refers to the nonlinear activation function, h_v^k denotes the learned representation of node v with k-hop neighbors and $h_v^0 = X[v, :]$. And we utilize z_v to indicate the embeddings of node v acquired by classifier g.

In addition, the well-trained g is able to generate pseudo-labels $\hat{y}$ for unlabelled node u, then we can filter out some unlabelled nodes that may not belong to the minority classes for efficiency. To this end, let $M_i = \{u \in U | \hat{y}_u = i\}$ denotes the set of all the unlabelled nodes which are predicted as minority class i by the classifier g. From M_i, we only select those nodes that are close enough to the center of the minority class in the embedding space by means of a similarity-based module. The center of each minority class in the latent space is computed based on the labelled nodes:

$$cen(i) = \frac{1}{|C_i|} \sum_{v \in C_i} z_v \tag{2}$$

where C_i is the labelled nodes set of class i. Then, the module only selects the top-K nearest nodes to the center $cen(i)$ from M_i as a candidate set V_C, such that

$$V_C = \{u \in M_i | \mathcal{D}(z_u, cen(i)) < \varphi \text{ and } |V_C| \leq K\} \tag{3}$$

where $\mathcal{D}(\cdot, \cdot)$ measures the similarity in the embedding space, and we adopt the Euclidean distance as the measurement $\mathcal{D}(u, v) = \| z_u - z_v \|$, φ denotes the farthest distant between node in V_C and the $cen(i)$, $\varphi = max(z_u, cen(i) | u \in V_C)$.

Through this selection, we can find the nodes which are most likely to be predicted as minority classes, however, g is not reliable because it is trained with the imbalanced data. To deal with this, GraphSR utilizes another selection module of reinforcement learning to draw out the reliable nodes that can exactly improve the performance of classifier, as well as, determine the over-sampling scale for each minority class adaptively.

RL-Based Selection

The key task of this selection module is to specify a sampling procedure that can adaptively select unlabelled nodes to supplement the minority classes. Due to the lack of supervised information on unlabelled nodes, we adopt reinforcement learning for node selection. We design an iterative sampling procedure which is formulated as a Markov Decision Process (MDP), $M = (\mathcal{S}, \mathcal{A}, \mathcal{R}, \mathcal{T})$. The procedure to generate a balanced training set can be described by a trajectory $(s_0, a_0, r_0, ..., s_T, a_T, r_T)$, where the initial state s_0 only contains the imbalanced labelled set and the last state s_T contains the final supplemented balanced nodes set. GraphSR tries to learn an optimal policy using the reinforcement learning algorithm to allow the agent to decide preserving or discarding the unlabelled nodes in a partially-observed environment. In particular, the agent (i.e. the selector) traverses the candidate minority nodes in V_C sequentially. For each node u_t, the agent takes an action through a policy network represented by π_θ based on the current state, and then the environment assigns a reward based on the action. The agent updates the policy network based on the reward. After enough interactions between the agent and the environment, the agent can learn an optimal policy to optimally select the unlabelled nodes to supplement the minority classes. With the RL-based selection, it is easier for the algorithm to be generalized to different datasets without additionally determining the over-sampling scale. In the following, we discuss the major components of RL-based selection module in detail.

State We define the state s_t of the environment to encode the information of intermediate training set V_t and the unlabelled node u_t at time step t. In order to feed the s_t into the policy network, we need to fix the dimension of s_t irrelevant to the number of nodes in V_t. Motivated by PULNS (Luo et al. 2021), we use the summation of the nodes embeddings in V_t to represent the information of V_t, i.e., $z_{V_t} = \sum_{v \in V_t} z_v$. In addition, the embedding of u_t is exploited to denote its information. For time step t, state s_t is defined as $s_t = (z_{V_t}, z_{u_t})$. At the beginning, $V_0 = V_L$ is the imbalanced labelled set, u_0 is the first node in V_C.

Action The action a_t is to decide whether the current unlabelled node u_t in V_C should be included into the current training set V_t or not at time step t. Particularly, $a_t \in \{0, 1\}$, where $a_t = 1$ means that node u_t is selected to supplement the imbalanced training set, while $a_t = 0$ indicates that u_t is not applicable. Furthermore, the action is generated by a policy function π_θ, which takes the state as input and is parameterized by θ. In this work, the policy network represents the probability distribution of action, and is specified as a Multilayer Perceptron (MLP) with nonlinear activation function, i.e.:

$$a_t = P(a_t|s_t) = \pi_\theta(s_t) = MLP_\theta(s_t) \tag{4}$$

Transition After taking action a_t, the state of the environment should change to s_{t+1}. In our work, state consists of V_t and u_t, after taking a_t,

$$V_{t+1} = \begin{cases} \{V_t \cup u_t\}, & a_t = 1 \\ V_t, & a_t = 0 \end{cases} \tag{5}$$

and then $s_{t+1} = (z_{V_{t+1}}, z_{u_{t+1}})$. The termination of transition will happen when agent has completely traversed the candidate set V_C once.

Reward The reward r_t given by environment is to evaluate the action a_t at state s_t. Without supervised information about unlabelled nodes, it is hard to exactly find the minority nodes and reward them explicitly according to the true labels. Here we train a classifier based on $\{V_t \cup u_t\}$, and evaluate its accuracy acc_t on a small balanced validation set. However, the accuracy is always non-negative, directly applying it as reward may impede the convergence of the agent. The idea of reward engineering is to assign a positive reward if adding u_t can boost the performance of classifier, or a negative reward otherwise. Hence, the reward function is designed as:

$$r_t = \begin{cases} +1, & acc_t \geq b_t \text{ and } a_t = 1 \\ -1, & acc_t < b_t \text{ and } a_t = 1 \\ +1, & acc_t < b_t \text{ and } a_t = 0 \\ -1, & acc_t \geq b_t \text{ and } a_t = 0 \end{cases} \tag{6}$$

where, b_t denotes a baseline reward which is the average of past ten accuracies, i.e., $b_t = mean\{acc_{t-11}, ..., acc_{t-1}\}$, and acc_0 means the accuracy of the initial classifier trained by the labelled nodes set V_L.

Policy Gradient Training The goal of the agent is to train an optimal policy network that can maximize the expected reward, and policy gradient based methods are broadly utilized to optimize the policy network. In this work, we use Proximal Policy Optimization (PPO) (Schulman et al. 2017) to update the parameter θ of policy network. The objective function of PPO is defined as:

$$L^{CLIP}(\theta) = \mathbb{E}_t[\min(p_t(\theta)\hat{A}_t, \text{clip}(p_t(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_t)] \tag{7}$$

where $p_t(\theta)$ is the probability ratio, $p_t(\theta) = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}$, which is clipped into the range $[1-\epsilon, 1+\epsilon]$, making a lower bound of the conservative policy iteration objective (Kakade and Langford 2002) and the agent's exploration more stable. $\hat{A}_t$ is the estimated advantage function that involves discount accumulated reward and value function V^π, and is widely used in policy gradient algorithms.

GNN-Based Classifier

As discussed before, with the similarity-based selection and RL-based selection, GraphSR can sample the most informative and reliable nodes from the unlabelled data to supplement the minority classes for training. In detail, we obtain the final training set $\{V_L, V_{\hat{L}}\}$ with labelled set $\{(v_i, y_i)\}$ and supplemented set with pseudo-label $\{(u_i, \hat{y}_i)\}$. Based on the new training data, we can train an unbiased GNN classifier f according to the message passing process as eq.1 and cross-entropy loss function.

Experiments

We conduct experiments to evaluate the effectiveness of GraphSR for class-imbalanced node classification over several datasets with different imbalanced ratios. The results are reported in this section.

	Method	Cora ACC	Cora F1	Cora AUC-ROC	CiteSeer ACC	CiteSeer F1	CiteSeer AUC-ROC	PubMed ACC	PubMed F1	PubMed AUC-ROC
GCN	Vanilla	72.25	71.72	88.36	50.30	43.86	81.79	64.20	61.14	80.66
	Re-Weighting	72.43	71.82	89.43	52.23	46.71	82.37	63.26	60.02	79.98
	EN-Weighting	73.48	72.98	88.59	51.63	46.49	82.30	62.67	58.86	79.67
	Over-Sampling	71.82	71.37	87.71	52.70	47.48	82.39	63.13	59.50	80.58
	CB-Sampling	67.72	67.07	88.49	52.26	48.45	81.82	66.33	64.99	82.37
	GraphSMOTE	68.67	67.66	90.71	46.83	44.20	76.71	66.32	64.66	82.18
	GraphENS	73.48	72.95	90.70	55.75	52.87	82.98	70.07	69.09	83.45
	RU-Selection	70.75	70.36	89.77	56.66	53.51	83.21	69.83	69.32	83.45
	SU-Selection	73.10	72.80	**90.97**	56.27	53.47	83.49	67.75	65.83	81.91
	GraphSR	**73.90**	**73.59**	90.21	**57.28**	**55.20**	**83.67**	**71.79**	**71.70**	**85.14**
GraphSAGE	Vanilla	75.67	75.26	93.71	49.99	42.75	83.79	63.67	59.89	86.88
	Re-Weighting	76.11	75.99	94.55	50.99	44.39	77.89	61.40	57.38	88.41
	EN-Weighting	76.05	75.39	94.49	50.56	43.81	77.98	61.60	57.25	88.61
	Over-Sampling	75.28	74.86	94.07	50.66	43.70	84.07	66.39	64.03	87.07
	CB-Sampling	73.53	72.97	93.64	52.73	48.67	82.45	67.13	66.24	87.14
	GraphSMOTE	75.32	75.03	94.42	43.17	38.85	76.98	67.22	65.23	85.71
	GraphENS	76.84	75.94	94.04	52.45	50.93	84.05	68.07	66.19	88.17
	RU-Selection	75.85	75.61	94.24	**56.25**	**52.74**	**85.09**	68.74	68.34	86.72
	SU-Selection	77.99	77.69	94.57	52.93	47.87	83.79	66.83	64.72	86.59
	GraphSR	**78.78**	**78.36**	**94.92**	54.30	51.15	84.21	**74.13**	**74.36**	**89.33**

Table 1: Comparisons of GraphSR with other baselines when imbalance ratio is 0.3.

	Method	$\rho = 0.1$ ACC	$\rho = 0.1$ F1	$\rho = 0.1$ AUC-ROC	$\rho = 0.2$ ACC	$\rho = 0.2$ F1	$\rho = 0.2$ AUC-ROC	$\rho = 0.4$ ACC	$\rho = 0.4$ F1	$\rho = 0.4$ AUC-ROC	$\rho = 0.5$ ACC	$\rho = 0.5$ F1	$\rho = 0.5$ AUC-ROC
GraphSAGE	Vanilla	61.53	55.40	91.52	72.67	71.42	93.49	76.04	75.64	94.69	78.49	78.28	95.09
	Re-Weighting	64.25	58.85	92.72	74.10	73.37	94.13	77.13	76.83	95.05	77.85	77.48	95.11
	EN-Weighting	64.04	58.70	93.62	74.20	73.36	94.31	76.45	76.02	94.80	77.92	78.09	95.17
	Over-Sampling	64.17	58.55	92.78	72.81	71.90	93.78	77.42	77.10	94.81	78.31	78.10	95.11
	CB-Sampling	62.28	56.07	91.50	72.49	71.40	93.80	77.10	76.80	95.08	77.03	76.66	95.04
	GraphSMOTE	65.99	63.23	92.35	71.74	70.61	92.82	75.54	74.97	94.41	77.72	77.48	94.89
	GraphENS	70.44	65.51	89.55	75.42	73.74	93.72	75.84	74.70	94.06	76.26	75.29	94.53
	RU-Selection	71.33	68.37	94.13	74.74	74.32	93.88	77.80	77.49	94.70	77.78	77.54	94.96
	SU-Selection	71.39	69.00	94.10	77.13	76.91	94.70	77.56	77.27	95.01	78.21	77.96	95.13
	GraphSR	**75.17**	**74.82**	**94.20**	**77.46**	**77.23**	**94.85**	**78.56**	**78.25**	**95.33**	**79.54**	**79.33**	**95.50**

Table 2: Comparisons of GraphSR with other baselines on Cora with different imbalance ratio ρ.

Experiment Setup

Datasets We evaluate GraphSR on several widely-used public datasets for node classification task: Cora, CiteSeer, PubMed for citation networks (Sen et al. 2008). In citation networks, we construct an imitative imbalanced setting: three classes for Cora and CiteSeer and one class for PubMed are randomly selected as minority classes. All majority classes maintain 20 nodes in the training set, and the numbers for minority classes are $20 \times \rho$, where ρ is the imbalanced ratio. When validating and testing, we sample the same numbers of nodes for all classes to make the validation and test set balanced.

Baselines We test our method over two popular architectures, GCN and GraphSAGE. And we compare GraphSR with representative approaches which handle class-imbalanced issue. Note that for all the over-sampling baselines algorithms, we oversample the minority classes until they have the same number of samples as that of the majority classes.

- Re-Weighting: A classic cost-sensitive method (Japkowicz and Stephen 2002), which modifies loss function inversely proportional to the number of each class.
- EN-Weighting: Another variant of re-weighting method, which assigns the weight for each class based on the Effective Number (Cui et al. 2019).
- Over-Sampling: A classical re-sampling method, where the minority nodes are repeatedly sampled until the number of each minority class is the same as that of the majority classes.
- CB-Sampling: A variant of re-sampling method motivated by (Butler 1956), which firstly selects a class among all classes, and then randomly samples a node from the selected class.

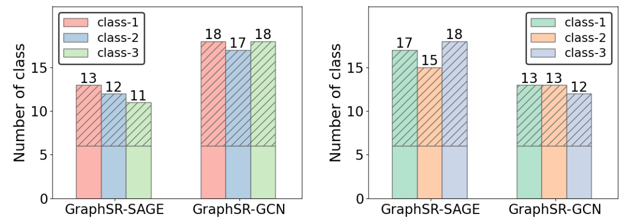
- GraphSMOTE (Zhao, Zhang, and Wang 2021): An over-sampling method for graph, which synthesizes additional minority nodes from existing nodes in the minority class.
- GraphENS (Park, Song, and Yang 2022): Another over-sampling strategy for graph, which generates minority nodes by mixing minority nodes with some nodes sampled from other classes.
- RU-Selection: A baseline model that supplements the minority class by randomly collecting the unlabelled nodes whose pseudo-labels are the minority class, until the class distribution is balanced.
- SU-Selection: An extension of RU-Selection that, rather than random collection, selects the unlabelled nodes in terms of their similarity for the minority classes.

Experiment Results

Class-Imbalanced Node Classification We compare the class-imbalanced node classification performance of GraphSR with that of other baselines on the widely-used citation datasets in semi-supervised setting. To verify the model generalization, we combine the proposed technique with two popular GNNs architectures, GCN and Gragh-SAGE. Here, we set the imbalanced ratio ρ as 0.3, and the experimental results are reported in Table 1, where we can find that GraphSR outperforms the previous baselines including re-weighting and re-sampling algorithms. Our method can effectively select proper unlabelled nodes to supplement the minority classes, which can enrich the diversity of minority classes and avoid overfitting.

Ablation Study We verify two components of GraphSR: similarity-based selection module and RL-based selection module. To this end, we introduce baselines: RU-Selection which supplements minority classes with randomly sampled unlabelled nodes (i.e., without Similarity/without RL), and SU-Selection which utilizes the similarity-based selection (i.e., with Similarity/without RL). From Table 1 and 2, we find that RU-Selection can not always work well because it is highly dependent on the predictions of GNN classifier when selecting unlabelled nodes, which is more prone to noisy nodes. On the other hand, SU-Selection can further improve the performances with the similarity-based selection module. With the further learning by RL-based module, GraphSR can identify the most informative and reliable unlabelled nodes for minority augmentation, thus,significantly boost the performances.

Over-Sampling Scale Furthermore, we investigate the over-sampling scales of minority classes which are adaptively determined by GraphSR. We visualize the number of supplemental nodes for each minority class when the imbalance ratio is fixed as 0.3 on Cora and CiteSeer datasets, based on two base architectures. The results are presented in Figure 3, where we can find that GraphSR can automatically determine different over-sampling scales for different minority classes over different datasets. Besides, even over the same dataset, GraphSR can generate different numbers of supplemental nodes when working for different base architectures, because the over-sampling scales are learned by RL-based selection based on the embeddings of nodes and current training set. For different base architectures, the embeddings of nodes are different, which make the over-sampling scale different as well. With the RL-based selection module, GraphSR can automatically decide the over-sampling scale for different imbalance ratio on different datasets, without demanding a well-elaborating over-sampling hyperparameter.



(a) Cora (b) CiteSeer

Figure 3: Over-sampling scale of minority classes with different base architectures on different datasets when imbalanced ratio $\rho = 0.3$. Note that the numbers of labelled nodes for minority classes are 6, and the shaded bar indicates the number of unlabelled nodes adaptively determined by GraphSR for each minority class.

Node Classification over Different Imbalance Ratios In this subsection, we evaluate the robustness of GraphSR on different imbalance ratio. The experiments are conducted on Cora dataset with imbalance ratio ρ ranging from 0.1 to 0.5. The results are shown in Table 2, where we can observe that GraphSR can be well adapted to different imbalance ratios. It achieves significant improvements across all different ratios. That demonstrates the validity and robustness of the proposed model. Moreover, the improvement of GraphSR is more impressive when the degree of imbalanced is extreme. On the one hand, the previous over-sampling methods can easily cause overfitting when the minority class is really limited. On the other hand, the self-training strategy of GraphSR can continuously train the GNN classifer with supplemental training set, and can generate more reliable pseudo-labels for minority classes, which can help GraphSR dig out the more valuable unlabelled nodes.

Conclusion

In this paper, we investigate the node classification with class-imbalanced problem in a semi-supervised setting. To take full advantage of rich information from the massive unlabelled nodes, we propose a novel data augmentation strategy, *GraphSR*, which can automatically supplement the minority classes from massive unlabeled nodes, with the help of a similarity-based selection module and an RL-based selection module. In addition, the RL-based module can adaptively determine the over-sampling scales for different minority classes. We verify that the proposed model can effectively enrich the diversity of minority classes and avoid overfitting to some extent. The experimental results demonstrate the effectiveness and robustness of GraphSR over various datasets with different GNN architectures.

Acknowledgments

National Key D&R Program of China (2019YFB1600704), The Science and Technology Development Fund, Macau SAR (0068/2020/AGJ, 0045/2019/A1, SKL-IOTSC(UM)-2021-2023), GDST(2020B1212030003).

References

Batista, G. E.; Prati, R. C.; and Monard, M. C. 2004. A study of the behavior of several methods for balancing machine learning training data. *ACM SIGKDD explorations newsletter*, 6(1): 20–29.

Bianchi, F. M.; Grattarola, D.; and Alippi, C. 2020. Spectral clustering with graph neural networks for graph pooling. In *International Conference on Machine Learning*, 874–883. PMLR.

Butler, J. W. 1956. Machine sampling from given probability distributions. In *Symposium on Monte Carlo Methods*, 249–264. Wiley New York.

Cao, K.; Wei, C.; Gaidon, A.; Arechiga, N.; and Ma, T. 2019. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, 1567–1578.

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.

Chen, Z.-M.; Wei, X.-S.; Wang, P.; and Guo, Y. 2019. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 5177–5186.

Cui, Y.; Jia, M.; Lin, T.-Y.; Song, Y.; and Belongie, S. 2019. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 9268–9277.

Defferrard, M.; Bresson, X.; and Vandergheynst, P. 2016. Convolutional neural networks on graphs with fast localized spectral filtering. *Advances in neural information processing systems*, 29.

Gilmer, J.; Schoenholz, S. S.; Riley, P. F.; Vinyals, O.; and Dahl, G. E. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*, 1263–1272. PMLR.

Gori, M.; Monfardini, G.; and Scarselli, F. 2005. A new model for learning in graph domains. In *Proceedings. 2005 IEEE international joint conference on neural networks*, 729–734.

Hamilton, W.; Ying, Z.; and Leskovec, J. 2017. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

He, H.; and Garcia, E. A. 2009. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9): 1263–1284.

Japkowicz, N.; and Stephen, S. 2002. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5): 429–449.

Kakade, S.; and Langford, J. 2002. Approximately optimal approximate reinforcement learning. In *In Proc. 19th International Conference on Machine Learning*. Citeseer.

Kim, J.; Jeong, J.; and Shin, J. 2020. M2m: Imbalanced classification via major-to-minor translation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 13896–13905.

Kipf, T. N.; and Welling, M. 2016. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*.

Kubat, M.; Matwin, S.; et al. 1997. Addressing the curse of imbalanced training sets: one-sided selection. In *Icml*, 179. Citeseer.

Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; and Dollár, P. 2017. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, 2980–2988.

Ling, C. X.; and Sheng, V. S. 2008. Cost-sensitive learning and the class imbalance problem. *Encyclopedia of machine learning*, 2011: 231–235.

Liu, B.; Deng, W.; Zhong, Y.; Wang, M.; Hu, J.; Tao, X.; and Huang, Y. 2019. Fair loss: Margin-aware reinforcement learning for deep face recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10052–10061.

Liu, Y.; Ao, X.; Qin, Z.; Chi, J.; Feng, J.; Yang, H.; and He, Q. 2021. Pick and choose: a GNN-based imbalanced learning approach for fraud detection. In *Proceedings of the Web Conference 2021*, 3168–3177.

Liu, Z.; Wei, P.; Jiang, J.; Cao, W.; Bian, J.; and Chang, Y. 2020. MESA: boost ensemble imbalanced learning with meta-sampler. *Advances in Neural Information Processing Systems*, 33: 14463–14474.

Luo, C.; Zhao, P.; Chen, C.; Qiao, B.; Du, C.; Zhang, H.; Wu, W.; Cai, S.; He, B.; Rajmohan, S.; et al. 2021. PULNS: Positive-Unlabeled Learning with Effective Negative Sample Selector. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 8784–8792.

Menon, A. K.; Jayasumana, S.; Rawat, A. S.; Jain, H.; Veit, A.; and Kumar, S. 2020. Long-tail learning via logit adjustment. *arXiv preprint arXiv:2007.07314*.

Park, J.; Song, J.; and Yang, E. 2022. GraphENS: Neighbor-Aware Ego Network Synthesis for Class-Imbalanced Node Classification. In *International Conference on Learning Representations*.

Qu, L.; Zhu, H.; Zheng, R.; Shi, Y.; and Yin, H. 2021. Imgagn: Imbalanced network embedding via generative adversarial graph networks. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, 1390–1398.

Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; and Klimov, O. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Scudder, H. 1965. Probability of error of some adaptive pattern-recognition machines. *IEEE Transactions on Information Theory*, 11(3): 363–371.

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.

Shi, M.; Tang, Y.; Zhu, X.; Wilson, D.; and Liu, J. 2020. Multi-class imbalanced graph convolutional network learning. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI-20)*.

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

Wang, J.; Lukasiewicz, T.; Hu, X.; Cai, J.; and Xu, Z. 2021. Rsg: A simple but effective module for learning imbalanced datasets. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3784–3793.

Wei, C.; Sohn, K.; Mellina, C.; Yuille, A.; and Yang, F. 2021. Crest: A class-rebalancing self-training framework for imbalanced semi-supervised learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 10857–10866.

Wu, S.; Tang, Y.; Zhu, Y.; Wang, L.; Xie, X.; and Tan, T. 2019. Session-based recommendation with graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, 346–353.

Yao, L.; Mao, C.; and Luo, Y. 2019. Graph convolutional networks for text classification. In *Proceedings of the AAAI conference on artificial intelligence*, 7370–7377.

You, J.; Ying, R.; and Leskovec, J. 2019. Position-aware graph neural networks. In *International conference on machine learning*, 7134–7143. PMLR.

Zhao, T.; Zhang, X.; and Wang, S. 2021. Graphsmote: Imbalanced node classification on graphs with graph neural networks. In *Proceedings of the 14th ACM international conference on web search and data mining*, 833–841.