Multi-Class Imbalanced Graph Convolutional Network Learning

Min Shi[1] , **Yufei Tang**[1] , **Xingquan Zhu**[1] , **David Wilson**[1] and **Jianxun Liu**[2]
[1]Department of Computer & Electrical Engineering and Computer Science,
Florida Atlantic University, USA
[2]School of Computer Science and Engineering, Hunan University of Science and Technology, China
{mshi2018, tangy, xzhu3, davidwilson2016}@fau.edu, ljx529@gmail.com

Abstract

Networked data often demonstrate the Pareto principle (*i.e.*, 80/20 rule) with skewed class distributions, where most vertices belong to a few majority classes and minority classes only contain a handful of instances. When presented with imbalanced class distributions, existing graph embedding learning tends to bias to nodes from majority classes, leaving nodes from minority classes under-trained. In this paper, we propose Dual-Regularized Graph Convolutional Networks (DR-GCN) to handle multi-class imbalanced graphs, where two types of regularization are imposed to tackle class imbalanced representation learning. To ensure that all classes are equally represented, we propose a class-conditioned adversarial training process to facilitate the separation of labeled nodes. Meanwhile, to maintain training equilibrium (*i.e.*, retaining quality of fit across all classes), we force unlabeled nodes to follow a similar latent distribution to the labeled nodes by minimizing their difference in the embedding space. Experiments on real-world imbalanced graphs demonstrate that DR-GCN outperforms the state-of-the-art methods in node classification, graph clustering, and visualization.

1 Introduction

Graphs are commonly used to encode both direct and implicit relationships between objects, *e.g.*, in a social network the interconnected users tend to share similar interests and represent a unique class collectively [Lee *et al.*, 2018]. Accordingly, current graph-based data mining tasks such as graph representation learning or embedding mainly focus on modeling the relative affinities between nodes from both topological and attribute content perspectives [Zhang *et al.*, 2020], such that nodes belonging to same classes (*e.g.*, “research area” in a citation network) can be clustered together in the embedding space. In the past, significant research efforts have been applied to supervised or semi-supervised graph neural models [Wu *et al.*, 2020], including the recently proposed Graph Convolutional Networks (GCN) [Kipf and Welling, 2016] and many its variants [Zhang *et al.*, 2019]. These methods typically adopt an end-to-end learning paradigm by training a node-level multi-class classifier after convolutional representation learning from the input graph [Veličković *et al.*, 2018], *i.e.*, each node forms its representation by aggregating features from all immediate neighborhoods. Despite the remarkable performance achieved in many application domains such as text classification [Yao *et al.*, 2019], image recognition [Chen *et al.*, 2019] and recommender systems [Wang *et al.*, 2019], existing methods often assume that the input class distributions are nearly or perfectly balanced, *i.e.*, balanced label samples for each class are deliberately provided to ensure representation learning equilibrium across multiple classes thereby avoiding the class imbalance problem entirely.

However, many real-world datasets naturally demonstrate highly-skewed class distributions due to the asymmetric and unrestricted evolution of different parts in these real-world graph-based systems [Huang *et al.*, 2016]. For example, in the NCI chemical compound graph [Pan and Zhu, 2013], only about 5% of molecules are active in the anti-cancer bioassay test. In the Cora citation network [Lin and Cohen, 2010], 26.8% of the papers belong to the *Neural Network* domain compared with the *Rule Learning* and *Reinforcement Learning* domains which only contain 7.9% and 4.8% respectively. When generalizing to graphs with an imbalanced class distribution, existing GCN methods have a tendency to overfit to majority classes, resulting in undesirable embedding results for the minority classes. For example, Figure 1 presents the node classification result for each class on the Cora citation network, where L1 and L6 are minority classes (*i.e.*, they contain far fewer instances than other majority classes such as L0). We observe that, in most cases, nodes from all seven classes can be correctly classified in the balanced setting, whereas for imbalanced setting, nodes from the two minority classes, L1 and L6, are frequently misclassified.

The main issue of class-imbalanced learning lies in that one or more classes may severely overrepresent others, which significantly compromises the performance of most standard learning algorithms [He and Garcia, 2009]. This issue is exacerbated in the case of graph-structured data due to the following two reasons:

- **Topological Interplay:** In addition to rich features associated with each graph node, different nodes can have

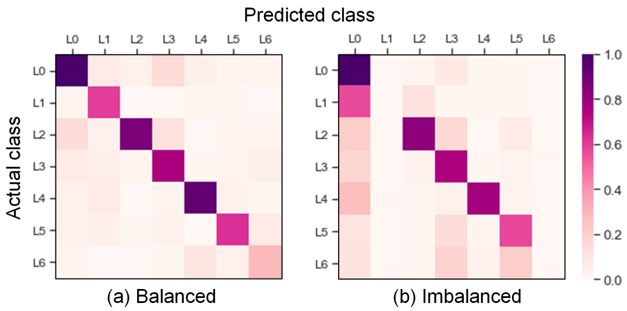


Figure 1: The confusion matrices of multi-class node classification results on Cora using a GCN with 3% of labeled nodes for semi-supervised representation learning and classification. (a) and (b) correspond to the balanced (*e.g.*, deliberately class-balanced training nodes) and imbalanced (*e.g.*, randomly sampled class-imbalanced training nodes) settings, respectively.

frequent topological connections with each other, meaning the class assignment for each node is no longer simply determined by its respective features but is also strongly impacted by its connected nodes.

- **Unclear Boundaries:** Graph data often involve multiple highly-skewed node classes, which makes it hard to balance representation learning with accurately identifying class boundaries since the learning of a particular class could be seriously impacted by other nearby class structures throughout the graph, *i.e.*, the majority classes would dominate feature propagation between nodes.

In this paper, we focus on a more general setting of multi-class imbalanced graph learning and develop a novel graph convolutional network incorporating two types of regularization. To the best of our knowledge, this is the first work that studies the node-level class-imbalanced graph embedding problem with graph neural networks. In the proposed framework, we first use a two-layer graph convolution network to derive node representations trained on class-imbalanced labels. To make representation learning for different classes of node more distinguishable (*e.g.*, clear boundaries), we incorporate a conditional adversarial training process to help separate the labeled node representations of different classes. In addition, to reduce the negative propagation influence from the convolution training of majority classes enforced on their structure-nearby minority classes, we train all unlabeled nodes to fit a similar data distribution to the well-trained labeled nodes in the learned embedding space, which promotes counterbalanced training between majority and minority classes.

In summary, our contribution is twofold: 1) we propose to study a node-level graph embedding problem that takes class distribution into account; 2) we propose DR-GCN, adopting a conditional adversarial training together with distribution alignment to learn robust node representations for both majority and minority classes.

2 Related Work

Graph neural networks. Driven by the promising learning capability of deep neural networks on grid-like data (*e.g.*, images), Graph Neural Networks (GNNs), architectures designed with non-Euclidean geometric data in mind, have seen an explosion in attention over these past five years [Wu *et al.*, 2020]. In essence, GNNs seek to exploit the characteristics of geometric data to provide a more powerful mechanism by which node representations are generated using both structural and contextual information. Graph Convolutional Networks (GCN) [Kipf and Welling, 2016] use a spectral-based convolution filter through which a node's features are aggregated from its direct neighborhood. Such convolution learning has been proven efficient and successfully applied in many problem domains [Yao *et al.*, 2019]. Graph Attention Networks (GAT) [Veličković *et al.*, 2018] are another recently proposed class of end-to-end GNNs similar to GCN, which introduce an attention mechanism that assigns larger weights to more important nodes, walks, or models. Some other useful feature aggregation methods have been proposed [Hamilton *et al.*, 2017], including the Tree-LSTM [Tai *et al.*, 2015] that learns representations for parent nodes by using child-sum tree long short-term memory networks to gather information from all child nodes.

Class imbalanced learning. Class imbalanced learning is a long-standing challenge faced by machine learning [Sun *et al.*, 2009], which aims to avoid model learning bias towards majority classes by lifting the influence of minority classes [Japkowicz and Stephen, 2002]. Conventional methods address this problem either from the data or algorithm level [He and Garcia, 2009]. The data level approach tries to rebalance the prior class distributions through a pre-processing step including over-sampling minority classes [Chawla *et al.*, 2002] or under-sampling majority classes [Drummond *et al.*, 2003]. However, these techniques could cause over-fitting or discard valuable information. In comparison, approaches at the algorithm level seek to modify existing algorithms to emphasize minority classes such as cost-sensitive learning [Dong *et al.*, 2018].

3 Problem Definition and Preliminary

3.1 Problem Definition

Given a graph with imbalanced node label distributions represented by $\mathcal{G} = (\mathbf{V}, \mathbf{E}, \mathbf{X}, \mathbf{L})$, where $\mathbf{V} = \{v_i\}_{i=1,\cdots,n}$ is a set of n unique nodes, $\mathbf{E} = \{e_{i,j}\}_{i,j=1,\cdots,n;\ i \neq j}$ is a set of edges that can be equal to a $n \times n$ adjacency matrix $\mathbf{A}$, with $\mathbf{A}_{i,j} = 1$ if $e_{i,j} \in \mathbf{E}$ or $\mathbf{A}_{i,j} = 0$ otherwise, and self-loops removed. $\mathbf{X}$ is a matrix $\mathbf{R}^{n \times m}$ containing all n nodes with their associated features, *i.e.*, $\mathbf{X}_i \in \mathbf{R}^m$ is the feature vector of node v_i, where m is the number of unique features in the graph. Graph $\mathcal{G}$ has multiple classes, denoted by $\mathbf{L} = \{\mathbf{L}_k\}_{k=1,\cdots,|\mathbf{L}|}$, that partition $\mathcal{G}$ to $|\mathbf{L}|$ clusters, where each class $\mathbf{L}_k$ categorizes a set of similar nodes. The class distribution may be highly-skewed as one or more classes contain many more nodes than others, *i.e.*, $|\mathbf{L}_1| \gg |\mathbf{L}_2|$. In such a case, $\mathbf{L}_1$ belongs to the majority classes while $\mathbf{L}_2$ belongs to minority classes.

The task in this paper is to represent graph $\mathcal{G}$ in a d-dimensional semantic space $\mathcal{H}^d$ with naturally imbalanced class labels for semi-supervised training, *i.e.*, randomly sample a few labeled instances from the whole node population.

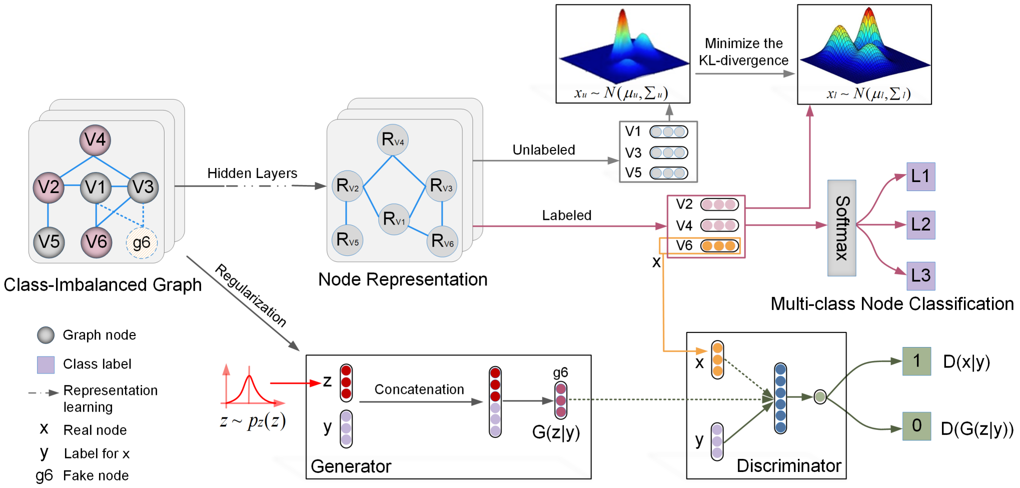


Figure 2: The proposed DR-GCN model. The node representations are obtained through a two-layer semi-supervised convolution learning with multi-class node classification (**the middle panel**). During training, a multi-class conditioned adversarial training (**the bottom panel**) ensures that embedding features can well represent node in each class to enhance the separation of different classes. Meanwhile, a distribution alignment training (**the top panel**) between labeled and unlabeled node representations strikes an influence balance between the majority and minority classes. Both the adversarial and alignment learning will help to train the convolution layers through back-propagation.

During training, the node space $\mathbf{V} = \mathbf{V}_l \cup \mathbf{V}_u$ is actually the union of labeled ($\mathbf{V}_l$) and unlabeled ($\mathbf{V}_u$) nodes both with imbalanced class distributions.

3.2 Conditional Generative Adversarial Networks

Conditional Generative Adversarial Networks (cGAN) [Mirza and Osindero, 2014] consist of two components: a generator $G(z|y)$ that maps the noise data z (*e.g.*, sampled from a prior distribution $z \sim p_z(z)$) to the real data distribution space, and a discriminator that assigns probability $D(x|y)$ to indicate whether or not x is a given real training sample or the probability $(1 - D(x|y))$ to indicate x is a fake generated sample (*e.g.*, $x = G(z|y)$). The training of cGAN tries to find the optimal discrimination between the real and fake samples, and meanwhile encourages $G(z|y)$ to approach the real data distribution. The objective of optimizing these two aspects is given as:

$$\begin{aligned}\min_G \max_D \mathcal{L}(D,G) &= \mathbb{E}_{x \sim p_{data}(x)} \log D(x|y) \\ &+ \mathbb{E}_{z \sim p_z(z)} \log(1 - D(G(z|y)))\end{aligned} \quad (1)$$

Both D and G are conditioned on some information y (*e.g.*, class labels), thus G can finally generate samples associated with y and D can well discriminate samples bound with y of varying value assignments.

4 The Proposed Method

The proposed DR-GCN model for multi-class imbalanced graph learning is shown in Figure 2, which involves three cooperative components as follows.

4.1 Class-Imbalanced Convolution Learning

In this paper, we adopt two-layer graph convolutional network [Kipf and Welling, 2016] to perform node-level representation learning on the input graph $\mathcal{G}$, where the first-order and second-order neighborhood relations can be sequentially modeled as:

$$O = \tilde{\mathbf{A}} ReLU(\tilde{\mathbf{A}}\mathbf{X}\mathbf{W}_0)\mathbf{W}_1 \quad (2)$$

Here, $\tilde{\mathbf{A}} = \tilde{\mathbf{D}}^{-\frac{1}{2}}(\mathbf{A} + \mathbf{I})\tilde{\mathbf{D}}^{-\frac{1}{2}}$ is the normalized symmetric adjacency matrix, where $\mathbf{I}$ is the identity matrix and $\tilde{\mathbf{D}}_{ii} = \sum_j (\mathbf{A} + \mathbf{I})_{ij}$. $\mathbf{W}_0 \in \mathbf{R}^{m \times r}$ and $\mathbf{W}_1 \in \mathbf{R}^{r \times d}$ are respectively the learned parameters for the first and second convolution layers, where r is the dimensionality of the first layer hidden representation. $ReLU$ is the activation function represented by $f(x) = \max(0, x)$.

The second-layer node embeddings have the same dimension as the number of classes (*i.e.*, $d = |\mathbf{L}|$) passed through a *softmax* classifier to perform multi-class node classification training by:

$$Z = softmax(O) = \frac{exp(O)}{\sum_i exp(O_i)} \quad (3)$$

$$\mathcal{L}_{gcn} = -\sum_{v_i \in \mathbf{V}_l} \sum_{j=1}^{d} Y_{ij} \ln Z_{ij} \quad (4)$$

Eq. (4) computes the cross-entropy error of classification results, where $\mathbf{V}_l$ is a set of labeled training nodes and $Y \in \mathbf{R}^{n \times |\mathbf{L}|}$ is the one-hot label indicator matrix of graph nodes.

In opposition to the deliberately balanced training samples in existing works [Kipf and Welling, 2016; Veličković *et al.*, 2018] (*i.e.*, each training class has a similar number of labeled nodes), we focus on the more practical case of naturally class-imbalanced distributions, *i.e.*, $\mathbf{V}_l$ is constructed with randomly sampled nodes from the whole population. However, as demonstrated in Figure 1 traditional GCN fails to handle the class-imbalanced graphs, we thus introduce two types of regularization training to mitigate this problem.

4.2 Class-Conditioned Adversarial Regularization

With standard convolutional learning under the imbalanced setting, the minority classes could be easily assimilated by nearby majority classes. To enhance the separation of different classes, we impose a conditional adversarial training on all labeled nodes. For each real training sample $x \in \mathbf{V}_l$ (*e.g.*, v_6) with its class indicator y (*e.g.*, one-hot vector), the generator takes a noise z generated from the prior normal distribution $z \sim p_z(z)$ as input, which then transforms to a real-like fake sample g_x (*e.g.*, g_6) after a concatenation with y and through learning with a standard multi-layer perceptron (MLP). On the other hand, the discriminator learns to classify the real and fake samples conditioned on y. To improve the learning capacity of generator, we add a regularization that forces the generated fake node could reconstruct the respective neighborhood relations (*e.g.*, g_6 has similar topology role as v_6) in the graph by:

$$\mathcal{L}_{reg} = \sum_{v_i \in \mathbf{N}(x)} \|\mathbf{h}_{g_x} - \mathbf{h}_{v_i}\|_2 \tag{5}$$

where Eq. (5) denotes the pairwise distance between g_x and x's neighbors $\mathbf{N}(x)$. Finally, the adversarial training objective is given as:

$$\begin{aligned}\min_{G,\mathcal{L}} \max_{D} \mathcal{L}(D,G) &= \mathbb{E}_{x\sim p_{data}(x)} \log D(x|y) \\ &+\mathbb{E}_{z\sim p_z(z)} \left[\log(1 - D(G(z|y))) + \mathcal{L}_{reg}\right]\end{aligned} \tag{6}$$

Eq. (6) is trained in a mini-batch fashion with balanced training classes. The main idea of the adversarial training is that while the discriminator learns to correctly classify training samples conditioned on different classes, it would in return encourage the convolution layers to learn distinguishing representations for different classes of nodes.

4.3 Latent Distribution Alignment Regularization

While the adversarial training can promote distinguishing results for labeled nodes, it could lead to overfitting within the labeled space, leaving the minority classes in the unlabeled space under-trained [Japkowicz and Stephen, 2002; He and Garcia, 2009]. We thus propose imposing a distribution alignment training between labeled and unlabeled node representations, where the assumption is that balanced convolution training across multi-imbalanced classes in the unlabeled space will be enforced in order to match the well-trained class-imbalanced nodes in the labeled space.

We assume that representations in labeled space (*e.g.*, $\mathbf{h}_{x_l} \in \mathcal{H}^d, x_l \in \mathbf{V}_l$) and unlabeled space ($\mathbf{h}_{x_u} \in \mathcal{H}^d, x_u \in \mathbf{V}_u$) follow two d-dimensional multivariate Gaussian distributions $x_l \sim \mathcal{N}(\mu_l, \Sigma_l)$ and $x_u \sim \mathcal{N}(\mu_u, \Sigma_u)$, where their probability density functions are given as:

$$p(x_l; \mu_l, \Sigma_l) = \frac{\exp\left(-\frac{1}{2}(x_l - \mu_l)^T \Sigma_l^{-1}(x_l - \mu_l)\right)}{(2\pi)^{d/2}|\Sigma_l|^{1/2}} \tag{7}$$

$$p(x_u; \mu_u, \Sigma_u) = \frac{\exp\left(-\frac{1}{2}(x_u - \mu_u)^T \Sigma_u^{-1}(x_u - \mu_u)\right)}{(2\pi)^{d/2}|\Sigma_u|^{1/2}} \tag{8}$$

where $\mu_l, \mu_u \in \mathbf{R}^d$ and $\Sigma_l, \Sigma_u \in \mathbf{R}^{d\times d}$ are the mean and covariance, respectively. For the situation in which class labels have no correlations with each other, Eqs. (7) and (8) can be respectively represented as the product of d independent Gaussian distributions with diagonal covariance matrices $\Sigma_l = \text{diag}(\sigma_{l,1}^2, \sigma_{l,2}^2, \cdots, \sigma_{l,d}^2)$ and $\Sigma_u = \text{diag}(\sigma_{u,1}^2, \sigma_{u,2}^2, \cdots, \sigma_{u,d}^2)$ [Ahrendt, 2005] by:

Algorithm 1: Training the DR-GCN model

```
Input  : An information network: $\mathcal{G} = (\mathbf{V}, \mathbf{E}, \mathbf{X}, \mathbf{L})$
Output: The node embeddings: $O \in \mathbf{R}^{n\times d}$
Initialization: $j = 1$, training epoch $I$ and $M$, batch size
  $N$, labeled set $\mathbf{V}_l$ and unlabeled set $\mathbf{V}_u$ that are both
  trainable through convolution layers by Eq. (2)

while $j \le I$ do
    $O \leftarrow$ learn convolution representations by Eq. (2)
    $\mathcal{N}(\mu_l, \Sigma_l) \leftarrow$ learn latent distribution from $\mathbf{V}_l$
    $\mathcal{N}(\mu_u, \Sigma_u) \leftarrow$ learn latent distribution from $\mathbf{V}_u$
    $\mathcal{L} \leftarrow$ compute the classification loss by Eq. (14)
    $[\mathbf{W}_0, \mathbf{W}_1] \leftarrow$ update network parameters in Eq. (2)
    for $\tau \in [1, M]$ do
        $\mathbf{x}_\Delta \leftarrow$ sample a batch of $N$ class-balanced
          training samples from $\mathbf{V}_l$, where $\mathbf{x}_\Delta$ is
          trainable;
        $\mathbf{z}_\Delta \leftarrow$ sample a batch of $N$ noise data from
          $z \sim p_z(z)$
        Update the generator with its stochastic gradient
          (where $x_k \in \mathbf{x}_\Delta$ and $z_k \in \mathbf{z}_\Delta$):
          $\bigtriangledown \frac{1}{N} \sum_{k=0}^{N} [\log D(x_k|y_k) + \log(1 - D(G(z_k|y_k)))$
              $+ \sum_{v_i \in \mathbf{N}(x_k)} \|\mathbf{h}_{g_{x_k}} - \mathbf{h}_{v_i}\|_2]$
        Update the discriminator and the convolution
          layers (e.g., $\mathbf{W}_0$ and $\mathbf{W}_1$) with their stochastic
          gradient:
          $\bigtriangledown \frac{1}{N} \sum_{k=0}^{N} [\log D(x_k|y_k) + \log(1 - D(G(z_k|y_k)))]$
    end
    $j = j + 1$
end
```

$$p(x_l; \mu_l, \Sigma_l) = \prod_{k=1}^{d} \frac{1}{\sqrt{2\pi}\sigma_{l,k}} \exp\left(-\frac{(x_{l,k} - \mu_{l,k})^2}{2\sigma_{l,k}^2}\right) \tag{9}$$

$$p(x_u; \mu_u, \Sigma_u) = \prod_{k=1}^{d} \frac{1}{\sqrt{2\pi}\sigma_{u,k}} \exp\left(-\frac{(x_{u,k} - \mu_{u,k})^2}{2\sigma_{u,k}^2}\right) \tag{10}$$

where the parameters can be approximated from the labeled and unlabeled samples as:

$$\mu_l = \frac{1}{|\mathbf{V}_l|} \sum_{v_i \in \mathbf{V}_l} \mathbf{h}_{v_i},\ \mu_u = \frac{1}{|\mathbf{V}_u|} \sum_{v_j \in \mathbf{V}_u} \mathbf{h}_{v_j} \tag{11}$$

$$\Sigma_l = \frac{1}{|\mathbf{V}_l|} \sum_{v_i \in \mathbf{V}_l} (\mathbf{h}_{v_i} - \mu_l)^2,\ \Sigma_u = \frac{1}{|\mathbf{V}_u|} \sum_{v_j \in \mathbf{V}_u} (\mathbf{h}_{v_j} - \mu_u)^2 \tag{12}$$

Items	Cora	Citeseer	PubMed	DBLP
# Nodes	2708	3327	19717	17725
# Edges	5429	4732	44338	52890
# Features	1433	3703	500	6974
# Classes	7	6	3	4

Table 1: Graph dataset characteristics.

Class Labels	$\mathbf{L}_0$	$\mathbf{L}_1$	$\mathbf{L}_2$	$\mathbf{L}_3$	$\mathbf{L}_4$	$\mathbf{L}_5$	$\mathbf{L}_6$
Cora	29	9	16	13	15	11	7
Citeseer	18	20	21	8	15	18	–
PubMed	39	21	40	–	–	–	–
DBLP	45	12	32	10	–	–	–

Table 2: Class distributions for graphs (%).

We finally minimize the difference between $\mathcal{N}(\mu_l, \Sigma_l)$ and $\mathcal{N}(\mu_u, \Sigma_u)$ based on the Kullback-Leibler divergence (both Σ_l and Σ_u are non-singular) [Joyce, 2011] as:

$$\mathcal{L}_{dist} = \frac{1}{2}\left(\log\frac{|\Sigma_u|}{|\Sigma_l|} - d + \mathrm{tr}(\Sigma_u^{-1}\Sigma_l) + (\mu_u - \mu_l)^T\Sigma_u^{-1}(\mu_u - \mu_l)\right) \tag{13}$$

4.4 Algorithm Training and Optimization

Algorithm 1 illustrates the proposed framework. DR-GCN is trained through three components: the standard convolution training in a semi-supervised manner, the conditional adversarial training to promote the distinguishing representations for various classes and the distribution alignment training that maintains the learning equilibrium between majority and minority classes. To avoid the strong constraints introduced by the distribution alignment training on the standard representation convolution learning, we combine them together by:

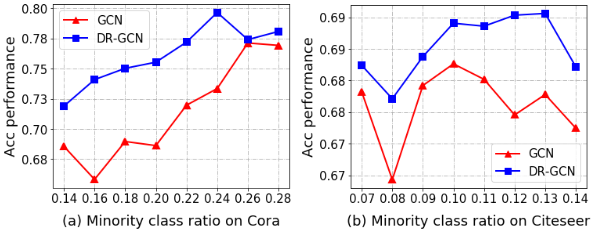
$$\mathcal{L} = (1 - \alpha)\mathcal{L}_{gcn} + \alpha\mathcal{L}_{dist} \tag{14}$$

where α is set to balance the two aspects of training.

5 Experimental Setup

Datasets. We use four widely-used benchmark graph datasets [Wu *et al.*, 2020], including Cora, Citeseer, Pubmed, and DBLP. The data statistics are summarized in Table 1. All four graphs are naturally class-imbalanced and their class distributions are shown in Table 2. We can observe that for each graph some classes contain much less number of nodes than others, *i.e.*, for Cora dataset 29% of graph nodes belong to class $\mathbf{L}_0$ while only 7% belong to class $\mathbf{L}_6$.

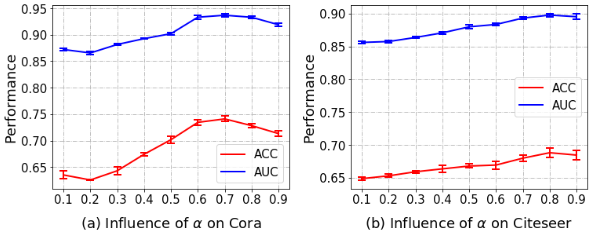
Compared methods. We compare with the following state-of-the-art embedding methods, including DeepWalk [Perozzi *et al.*, 2014] that learns node representations based on the SkipGram model [Mikolov *et al.*, 2013], Graph-LSTM [Tai *et al.*, 2015], standard GCN [Kipf and Welling, 2016], GCN combined with random under-sampling (GCN$_{RUS}$) [Liu *et al.*, 2008] and GAT [Veličković *et al.*, 2018] that all learn node representations from both graph structure and content with spectrum-based convolution filters. We also compare with two variants DR-GCN$_{gan}$ and DR-GCN$_{dist}$ that respectively incorporate the class-conditioned adversarial regularization and the latent distribution alignment regularization.



(a) Minority class ratio on Cora

(b) Minority class ratio on Citeseer

Figure 3: Influence of class imbalance ratio.



(a) Influence of α on Cora

(b) Influence of α on Citeseer

Figure 4: Influence of parameter α.

Settings. We evaluate the model by conducting node classification, graph clustering, and visualization, respectively. We randomly sample 3% of nodes from the entire graph for training as per label rates used by standard GCN methods. The remaining nodes are split into validation and testing sets where 10% are used for hyperparameter optimization, and 90% are used for testing respectively. For node classification, the performance is computed in terms of Accuracy (Acc) and AUC score [Kipf and Welling, 2016]. Each experiment is repeated 10 times and we report the mean values with standard errors. For graph clustering task, the $k-$means algorithm is used with learned node embeddings as inputs and the performance is computed w.r.t four metrics [Xia *et al.*, 2014] including accuracy (Acc), precision, F1-score (F1) and normalized mutual information (NMI). For GCN-based methods, we set the hidden embedding size r as 10, the dropout rate as 0.3, the L_2 norm regularization weight decay as 0.03 and the learning rate for the gradient decent algorithm as 0.002. We set the maximum training epoch I as 1000 with an early stopping of 200. In our approach, the default values for M, N and α are set as 1, $|\mathbf{V}_l|/2$ and 0.7, where $|\mathbf{V}_l|$ is the total number of labeled nodes.

5.1 Node Classification

Experiment results. The classification results are shown in Table 3. We can observe that both DR-GCN$_{gan}$ and DR-GCN$_{dist}$ perform better than GCN, which demonstrates the effectiveness of the introduced class-conditioned adversarial regularization and latent distribution alignment regularization for class-imbalanced node classification. In addition, on all four class-imbalanced graphs our DR-GCN model outperforms the random under-sampling method GCN$_{RUS}$ and other state-of-the-art methods such as GAT and Graph-LSTM, which verifies the superiority of our approach.

Varying imbalance ratio. We test the performance of DR-GCN w.r.t training ratio between minority class and majority class on Cora and Citeseer. For Cora dataset, we assume $\mathbf{L}_1$

Datasets	Cora		Citeseer		Pubmed		DBLP	
Metrics	Acc	AUC	Acc	AUC	Acc	AUC	Acc	AUC
DeepWalk	35.95±2.63	72.47±1.56	24.44±1.75	59.30±1.53	41.36±0.52	64.89±0.35	38.77±0.50	64.64±0.67
Graph-LSTM	70.87±0.76	87.73±1.46	66.04±0.88	85.39±0.18	81.05±0.64	92.96±1.05	77.65±0.45	94.42±0.17
GCN	65.83±0.25	84.53±0.11	65.43±0.20	86.23±0.11	80.69±0.08	91.40±0.01	70.82±0.28	84.30±0.23
GCN$_{RUS}$	66.59±0.44	86.64±0.62	65.21±1.05	87.13±0.36	78.93±0.78	90.34±0.73	66.23±0.47	81.29±0.27
GAT	71.18±0.32	92.57±0.13	67.16±1.23	86.01±0.56	81.50±0.08	93.23±0.04	77.52±0.70	93.45±0.46
DR-GCN$_{gan}$	66.13±0.39	85.55±0.13	66.31±0.17	86.68±0.05	80.69±0.50	92.89±0.45	70.86±0.03	84.36±0.01
DR-GCN$_{dist}$	72.54±1.38	92.79±0.35	67.07±1.00	88.81±0.24	80.92±0.39	92.92±0.57	76.90±0.35	92.92±0.10
DR-GCN	**74.09±0.51**	**93.66±0.33**	**67.71±0.49**	**89.19±0.12**	**81.69±0.32**	**93.39±0.36**	**78.86±0.12**	**94.93±0.60**

Table 3: Performance of class-imbalanced node classification on Cora, Citeseer, Pubmed and DBLP.

Metrics	Acc	Precision	F1	NMI
DeepWalk	31.42	28.48	16.13	2.23
Graph-LSTM	63.90	64.75	62.28	52.36
GCN	63.55	51.58	49.27	42.32
GCN$_{RUS}$	66.17	61.26	57.98	46.56
GAT	**71.14**	61.27	62.29	50.99
DR-GCN$_{gan}$	69.24	65.24	63.23	52.55
DR-GCN$_{dist}$	69.31	65.24	63.27	53.16
DR-GCN	69.70	**65.63**	**63.49**	**53.44**

Table 4: Clustering results on Cora.

Metrics	Acc	Precision	F1	NMI
DeepWalk	34.96	33.22	30.38	4.77
Graph-LSTM	60.54	58.43	53.87	37.09
GCN	64.13	58.84	55.89	38.20
GCN$_{RUS}$	66.17	58.21	55.84	37.97
GAT	63.41	55.15	55.16	35.42
DR-GCN$_{gan}$	**64.98**	59.09	**56.83**	**38.67**
DR-GCN$_{dist}$	63.90	58.82	56.23	38.17
DR-GCN	64.19	**59.17**	56.48	38.24

Table 5: Clustering results on Citeseer.

and $\mathbf{L}_6$ as minority classes and $\mathbf{L}_0$ as majority class. Similarly, we assume $\mathbf{L}_3$ as minority class and $\mathbf{L}_1$ and $\mathbf{L}_2$ as majority classes for Citeseer. Then, we vary the ratio of training samples from minority classes and the ratio for majority classes is changed accordingly (e.g., in Table 2 for Cora the original training ratios for minority and majority classes are (9+7) and 29 percents, respectively. When the ratio for minority classes increases to 18 percent, thereby the ratio for majority classes is reduced to 27 percent). We can observe in Figure 3 that DR-GCN significantly outperforms CGN with various minority class ratios, which demonstrates our model shows better robustness for class-imbalanced graph learning.

Parameter analysis. Figure 4 shows the impact of parameter α to balance the standard convolution learning and the distribution alignment training in Eq. (14). On both Cora and Citeseer datasets, the classification performances first increase and then tend to decline with larger values of α.

5.2 Graph Clustering

Table 4 and Table 5 report the clustering results of all baselines. On both the Cora and Citeseer datasets, we can observe that DR-GCN consistently outperforms GCN, which again verifies the benefit of our regularized learning process. Although GAT has slightly better Acc performance than that of others on Cora, it has a rather poor performance w.r.t. Precision, F1, and NMI compared with DR-GCN on both Cora and Citeseer datasets. The clustering results demonstrate the superiority of our proposed learning frameworks.

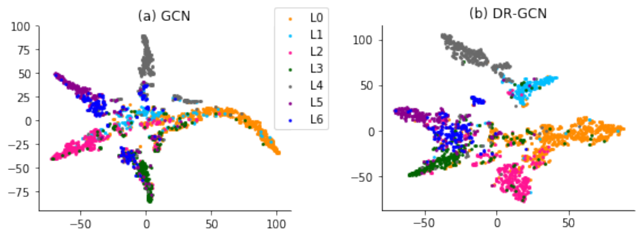


Figure 5: Graph visualization on the Cora dataset.

5.3 Graph Visualization

Figure 5 shows the 2-D node embedding visualization results on the Cora dataset. Compared with GCN, we can observe that DR-GCN learns more discriminative node embeddings, especially for minority classes, such as $\mathbf{L}_1$ and $\mathbf{L}_6$, which account for 9% and 7% node population, respectively.

6 Conclusion

Real-world graph structured data usually present highly-skewed class distributions. The most critical challenge, when learning from class-imbalanced graphs, is that the nodes have strong topological interdependence, causing existing network representation learning methods to underperform on minority classes. In this paper, we proposed a novel dual-regularized graph convolutional network that contains a conditional adversarial training to enhance the separation of nodes from different classes and a distribution alignment training to enforce balanced learning between majority and minority classes.

We conducted extensive comparative studies to evaluate the proposed framework for both node classification and unsupervised graph clustering. The validations, visualizations, and comparisons from the experimental results demonstrated that the proposed DR-GCN model is effective to handle graph data with naturally imbalanced class distributions.

Acknowledgements

This research was sponsored in part by the U.S. National Science Foundation through Grant Nos. IIS-1763452 and CNS-1828181.

References

[Ahrendt, 2005] Peter Ahrendt. The multivariate gaussian probability distribution. *Technical University of Denmark, Tech. Rep*, 2005.

[Chawla *et al.*, 2002] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[Chen *et al.*, 2019] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 5177–5186, 2019.

[Dong *et al.*, 2018] Qi Dong, Shaogang Gong, and Xiatian Zhu. Imbalanced deep learning by minority class incremental rectification. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 41(6):1367–1381, 2018.

[Drummond *et al.*, 2003] Chris Drummond, Robert C Holte, et al. C4. 5, class imbalance, and cost sensitivity: why under-sampling beats over-sampling. In *Workshop on learning from imbalanced datasets II*, volume 11, pages 1–8, 2003.

[Hamilton *et al.*, 2017] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[He and Garcia, 2009] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

[Huang *et al.*, 2016] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. Learning deep representation for imbalanced classification. In *Proc. of the IEEE conf. on Computer Vision and Pattern Recognition*, pages 5375–5384, 2016.

[Japkowicz and Stephen, 2002] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intell. Data Anal.*, 6(5):429–449, 2002.

[Joyce, 2011] James M Joyce. Kullback-leibler divergence. *Inl. Ency. of Statistical Science*, pages 720–722, 2011.

[Kipf and Welling, 2016] Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Intl. Conf. on Learning Representations*, 2016.

[Lee *et al.*, 2018] John Boaz Lee, Ryan A Rossi, Xiangnan Kong, Sungchul Kim, Eunyee Koh, and Anup Rao. Higher-order graph convolutional networks. *arXiv preprint arXiv:1809.07697*, 2018.

[Lin and Cohen, 2010] Frank Lin and William W Cohen. Semi-supervised classification of network data using very few labels. In *Intl. IEEE Conf. on Advances in Social Networks Analysis and Mining*, pages 192–199, 2010.

[Liu *et al.*, 2008] Xu-Ying Liu, Jianxin Wu, and Zhi-Hua Zhou. Exploratory undersampling for class-imbalance learning. *IEEE Trans. on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 39(2):539–550, 2008.

[Mikolov *et al.*, 2013] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *ICLR Workshop*, 2013.

[Mirza and Osindero, 2014] Mehdi Mirza and Simon Osindero. Conditional generative adversarial nets. *CoRR, abs/1411.1784*, 2014.

[Pan and Zhu, 2013] Shirui Pan and Xingquan Zhu. Graph classification with imbalanced class distributions and noise. In *Proc. of the 23rd AAAI Conf. on Artificial Intelligence*, pages 1586–1592, 2013.

[Perozzi *et al.*, 2014] Bryan Perozzi, Rami Al-Rfou, Steven Skiena, Rami Al-Rfou, and Steven Skiena. Deepwalk: Online learning of social representations. In *Proc. of the ACM SIGKDD Intl. Conf.*, pages 701–710, 2014.

[Sun *et al.*, 2009] Yanmin Sun, Andrew KC Wong, and Mohamed S Kamel. Classification of imbalanced data: A review. *Intl. Journal of Pattern Recognition and Artificial Intelligence*, 23(04):687–719, 2009.

[Tai *et al.*, 2015] Kai Sheng Tai, Richard Socher, and Socher Richard Manning, Christopher D, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. In *Proc. of the 53rd Annual Meeting of the Association for Computational Linguistics*, page 1556–1566, 2015.

[Veličković *et al.*, 2018] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Lio, and Yoshua Bengio. Graph attention networks. In *Intl. Conf. on Learning Representations*, 2018.

[Wang *et al.*, 2019] Hongwei Wang, Miao Zhao, Xing Xie, Wenjie Li, and Minyi Guo. Knowledge graph convolutional networks for recommender systems. In *Proc. of the intl. conf. on World Wide Web*, pages 3307–3313, 2019.

[Wu *et al.*, 2020] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. A comprehensive survey on graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 2020.

[Xia *et al.*, 2014] Rongkai Xia, Yan Pan, Lei Du, and Jian Yin. Robust multi-view spectral clustering via low-rank and sparse decomposition. In *Proc. of the 28th AAAI conf. on Artificial Intelligence*, 2014.

[Yao *et al.*, 2019] Liang Yao, Chengsheng Mao, and Yuan Luo. Graph convolutional networks for text classification. In *Proc. of the 33rd AAAI Conf. on Artificial Intelligence*, pages 7370–7377, 2019.

[Zhang *et al.*, 2019] Yingxue Zhang, Soumyasundar Pal, Mark Coates, and Deniz Ustebay. Bayesian graph convolutional neural networks for semi-supervised classification. In *Proc. of the 33rd AAAI Conf. on Artificial Intelligence*, pages 5829–5836, 2019.

[Zhang *et al.*, 2020] Daokun Zhang, Jie Yin, Xingquan Zhu, and Chengqi Zhang. Network representation learning: A survey. *IEEE Trans. on Big Data*, 6(1):3–28, 2020.