ImVerde: Vertex-Diminished Random Walk for Learning Imbalanced Network Representation

Jun Wu
Arizona State University
junwu6@asu.edu

Jingrui He
Arizona State University
jingrui.he@asu.edu

Yongming Liu
Arizona State University
Yongming.Liu@asu.edu

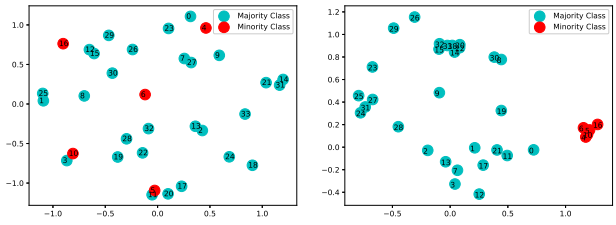
***Abstract*—Imbalanced data widely exist in many high-impact applications. An example is in air traffic control, where among all three types of accident causes, historical accident reports with 'personnel issues' are much more than the other two types ('aircraft issues' and 'environmental issues') combined. Thus, the resulting data set of accident reports is highly imbalanced. On the other hand, this data set can be naturally modeled as a network, with each node representing an accident report, and each edge indicating the similarity of a pair of accident reports. Up until now, most existing work on imbalanced data analysis focused on the classification setting, and very little is devoted to learning the node representations for imbalanced networks. To bridge this gap, in this paper, we first propose Vertex-Diminished Random Walk (VDRW) for imbalanced network analysis. It is significantly different from the existing Vertex Reinforced Random Walk by discouraging the random particle to return to the nodes that have already been visited. This design is particularly suitable for imbalanced networks as the random particle is more likely to visit the nodes from the same class, which is a desired property for learning node representations. Furthermore, based on VDRW, we propose a semi-supervised network representation learning framework named *ImVerde* for imbalanced networks, where context sampling uses VDRW and the limited label information to create node-context pairs, and balanced-batch sampling adopts a simple under-sampling method to balance these pairs from different classes. Experimental results demonstrate that *ImVerde* based on VDRW outperforms state-of-the-art algorithms for learning network representations from imbalanced data.**

***Index Terms*—Network representation, random walk, imbalanced data**

I. Introduction

Nowadays, big data is being generated across multiple high impact application domains. Often times, the target data set is imbalanced in terms of the proportion of examples from various classes. For example, in air traffic control, the large number of historical flight accident reports can be used to analyze and identify the leading indicators of various accident causes. According to National Transportation Safety Board[1], there are three major types of accident causes, including aircraft issues, personnel issues, and environmental issues. Historical records with 'personnel issues' are much more than the other two types combined, thus creating an imbalanced data set. For example, of the accident reports between aircraft control and directional control, aircraft control of them are due to 'personnel issues', which dominate the entire data set.

[1]https://www.ntsb.gov/Pages/default.aspx



(a) Planetoid (b) *ImVerde*

Fig. 1: Network embedding in 2-dimensional space with two imbalanced classes from Zachary's Karate network [28] (shown in Figure 3(a)) using: (a) Planetoid [2] (b) proposed *ImVerde*. (t-SNE [29] is adopted to visualize the network embedding; best viewed in color)

On the other hand, such a data set can be naturally modeled as a network, where each node corresponds to a historical accident report, and each edge reflects the similarity between two reports.

Despite the extensive existing work on imbalanced classification [9]–[11], up until now, very little (if any) effort has been devoted to learning the node representation from imbalanced networks. Most existing work on network embedding either works in the unsupervised fashion, i.e., not considering the label information at all, or is implicitly assuming a balanced setting, i.e., not suitable for the imbalanced setting. More specifically, unsupervised embedding approaches [5], [13], [30], [36] aimed to learn the node representations which preserve the graph topological structure in the feature space; and semi-supervised methods [1], [2], [32] implicitly assumed that the labeled set consisted of roughly equal number of examples from each class. However, in the presence of imbalanced labeled set, these methods will inevitably suffer from imbalanced node-context pairs, resulting in node representations not amiable to the following classification task.

To bridge this gap, in this paper we propose a generic framework named *ImVerde* for learning node representations from imbalanced networks. It is based on a novel random walk model named Vertex-Diminished Random Walk (VDRW), where the basic idea is that the transition probability to one node decreases each time it is visited by the random particle. Notice that when learning the embedding vectors from imbalanced networks, compared with the existing Vertex Reinforced Random Walk [22], VDRW can capture the node-

context pairs from the minority class with a higher accuracy.

Furthermore, we sample the context using both graph structure and label information. In order to extract node-context pairs using the two types of information jointly, we propose a jumping scheme to combine them into one unified framework. In other words, if a node is labeled, it would have a certain probability jumping to other nodes with the same labels in one step rather than walking randomly to neighbors. Finally, in order to balance the extracted node-context pairs in imbalanced networks, we use a simple under-sampling method for mini-batch sampling. Figure 1 shows that compared with Planetoid [2], the network embedding produced by the proposed *ImVerde* leads to better separability between the two classes in the imbalanced data set. To validate the effectiveness of the proposed network representation framework, we conduct experiments on several real network data sets with promising results.

The main contributions of this paper can be summarized as follows:

- We propose a novel random walk model, named Vertex-Diminished Random Walk (VDRW), which adjusts the transition probability each time a node is visited by the random particle. It leads to an effective context sampling method for imbalanced networks, and its convergence properties are analyzed.
- Based on VDRW, we propose a semi-supervised network representation framework named *ImVerde*, in which two strategies are introduced to improve the quality of node-context pairs: ***Context Sampling*** and ***Balanced-batch Sampling***. ***Context Sampling*** extracts node-context pairs based on both graph structure and label information, and ***Balanced-batch Sampling*** aims to keep the extracted pairs balanced.
- The proposed framework is evaluated on several data sets, with experimental results demonstrating its superior performance over state-of-the-art techniques. These results are also consistent with our analysis on imbalanced data in terms of, e.g., high-quality node-context pairs, separability of different classes, etc.

The rest of the paper is organized as follows. We review the related work in Section II. VDRW is introduced in Section III and Section IV presents our proposed semi-supervised network representation framework. The extensive experimental results and discussion are provided in Section V. Finally we conclude the paper in Section VI.

II. Related Work

In this section, we briefly review the related work on imbalanced data analysis and network representation learning.

A. Imbalanced Data Analysis

Imbalanced data mining has attracted significant attention in many high-impact applications since standard algorithms implicitly assume the balanced distribution in data and may fail to analyze class-imbalanced data [38]. And in most cases, the data annotated by workers [42], [43] hardly ensure the balanced distribution. Up until now, over-sampling [11], [16] and under-sampling [39] are two mainstream approaches to balance the instance distribution in different classes via adding replicated instances from minority class or reducing instances from majority class. To solve the over-fitting in over-sampling, SMOTE [11] created synthetic samples between each minority instance and its nearest neighbors. Tomek links [39] were defined as a pair of samples in opposite classes with minimal distance. By removing or cleaning the overlapping instances, Tomek links usually integrated with other sampling methods to target imbalanced problems. Instead of altering data distribution, cost-sensitive methods [17] considered the penalty for misclassifying instances to another class. Specifically, misclassified minority instances bear heavier penalty in order to alleviate imbalanced data distribution. Recently deep learning models [10], [37] are also adopted to deal with imbalanced data. Additionally, minority classes or rare categories normally play important roles in the database, such as financial fraud in bank systems. Rare category detection algorithms [18]–[21] aimed to identify those special minority groups from imbalanced data. Compared with the existing work on imbalanced data analysis, in this paper, we study a novel problem setting of learning network representation from imbalanced networks, whereas existing work mainly focuses on the classification problem or the active learning setting.

B. Network Representation Learning

The objective of network representation is to learn a low-dimensional dense vector for each node in the network. One of the common assumptions in most of the existing work [8], [13], [41], [44] is that nearby nodes have similar embeddings in the feature space. Among these algorithms, there are mainly three categories: factorization-based, random walk based, and CNN-based. The factorization-based algorithms can be traced back to classic dimensionality reduction approaches LLE [3] and ISOMAP [4], which preserved the manifold properties in the low-dimensional space. Recently, to capture high-order proximities relationship among the nodes, LINE [6] and GraRep [7] were proposed to learn network representation unsupervisedly. For random walk based models, DeepWalk [5] firstly showed that truncated random walk follows the similar power-law distribution as skip-gram model does in word frequency. Subsequently, many (unsupervised or semi-supervised) random work based approaches [1], [2], [23], [24] were proposed to learn node representation from homogeneous networks. And Qiu et al. [35] proposed that some of those embedding approaches, e.g., DeepWalk [5] and node2vec [23], can be transformed into the matrix factorization problems. In addition, motivated by convolutional neural network architecture, Graph Convolutional Networks (GCN) [14], [15], [25] extended convolution operation to graph-structured data using spectral graph theory for semi-supervised node classification. Similarly, GraphSAGE [26] aimed to learn a function that aggregates embeddings from nodes' local neighborhood for inductive representation learning. To capture the highly non-linear network structure, SDNE [27] used a semi-supervised

deep neural network model with non-linear activation func tions to preserve first- and second-order proximities among th nodes. However, very little effort has been devoted to learnin the node representation from imbalanced networks, which wil result in imbalanced node-context pairs. Our proposed *ImVerd* framework is designed to address this issue through VDRW, novel context sampling method, as well as the balanced-batc sampling method.

III. Vertex-Diminished Random Walk

In this section, we present the Vertex-Diminished Randon Walk (VDRW) for imbalanced network analysis. It resembles the existing Vertex Reinforced Random Walk (VRRW) [22] in terms of the dynamic nature of the transition probabilities, as well as some convergence properties. However, it is fundamentally different from VRRW in which the transition probabilities to one node increase each time it is visited. As our analysis will show, VDRW is particularly suitable for analyzing imbalanced networks as it encourages the random particle to walk within the same class, which in turn will lead to separable node representations in the embedding space between the majority and the minority classes in the imbalanced data set.

A. Notation

Suppose that an undirected (or directed) network is denoted $G = (V, E)$ where V is the node set consisting of n nodes, and E is the edge set consisting of m edges. Each edge $e = (v_i, v_j) \in E$ is associated with a positive weight $w_{ij} > 0$ if v_i and v_j are connected in the network. Otherwise, $w_{ij} = 0$. For this network, let $\mathbf{R}$ denote the non-negative $n \times n$ transition matrix, whose element in the i^{th} row and j^{th} column is $\mathbf{R}_{i,j} = \frac{w_{ij}}{\sum_j w_{ij}}$. Let $Y_t \in \{1, \ldots, n\}$ be the state of the random particle at step t and $\mathbf{S}(t) = [\mathbf{S}_1(t), \cdots, \mathbf{S}_n(t)] \in \mathbb{R}^n$ denote the number of visits to each node up to time t where $\mathbf{S}_i(0) = 0$ for $i = 1, \cdots, n$. That is, $\mathbf{S}_i(t+1) = \mathbf{S}_i(t) + \delta_{Y_{t+1},i}$ where $\delta_{Y_{t+1},i} = 1$ if v_i is visited at step $t+1$, namely $Y_{t+1} = i$; otherwise, $\delta_{Y_{t+1},i} = 0$.

B. VDRW: A Novel Random Walk Model

As illustrated in DeepWalk [5], a short random walk follows a power-law distribution as word frequency in language modeling. By implicitly assuming the balanced setting in labeled training examples [1], [2], most nodes can find a path within the same class using short random walk. Nevertheless, random walk does not necessarily work for forcing nodes to walk within the same class on an imbalanced network because random walk might follow the node-degree distribution to create walking path. That is, the nodes in minority class are likely to walk to majority class instead of walking within the same class since node degrees of majority class are higher than that of minority class in most cases. As a result, there is not much difference between the nodes coming from the majority and minority classes in terms of their walking paths, leading to non-separable representations in the embedding space.

To address this problem, in this paper, we propose a novel random walk model named Vertex-Diminished Random Walk

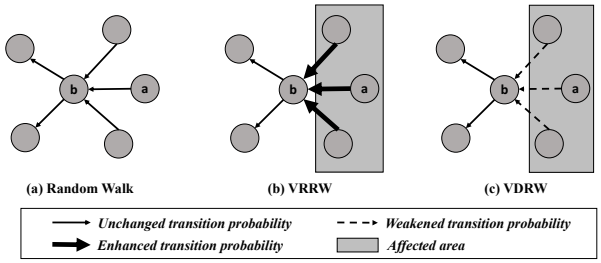


Fig. 2: Comparison among Random Walk, VRRW [22] and VDRW

(VDRW) based on the idea that the transition probability to one node decreases each time it is visited. It resembles the existing Vertex-Reinforced Random Walk (VRRW) [22] in terms of the dynamic nature of the transition probabilities. But the transition probabilities to visit one node increase each time it is visited in VRRW. Figure 2 illustrates the difference among random walk, VRRW, and VDRW in a directed graph. When node b is visited by node a, there is no change regarding the transition probability in random walk. However, all the transitions to node b are affected in VRRW and VDRW when it is visited by node a. Those transitions would be weakened in VDRW, whereas VRRW forces them to be enhanced after one node is visited each time. Later, we will explain why VDRW is adopted in imbalanced network analysis instead of random walk or VRRW.

Now we first present a generalized random walk in which the transition probability changes when one node is visited by the random particle. We introduce a visiting function $f(\mathbf{S}_i(t))$ to capture the change behavior of the transition probabilities with respect to visited times $\mathbf{S}_i(t)$ for node v_i. Therefore, after moving for t steps, $f(\mathbf{S}_i(t))$ keeps track of how the transition probabilities to v_i change. In general, the transition probability to v_j at time $t+1$ is defined as follows:

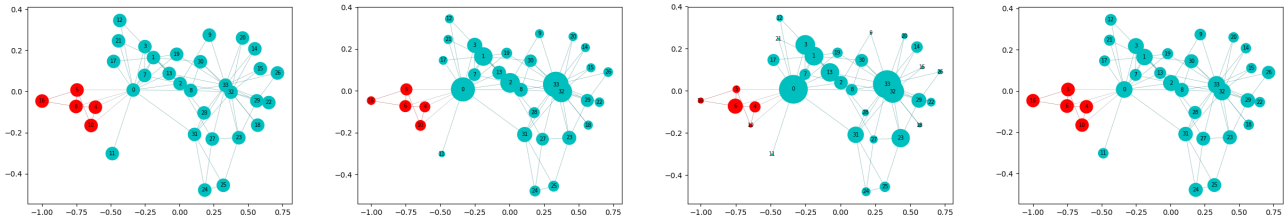
$$P(Y_{t+1} = j | \mathscr{F}_t) = \frac{\mathbf{R}_{Y_t,j} f(\mathbf{S}_j(t))}{\sum_i \mathbf{R}_{Y_t,i} f(\mathbf{S}_i(t))} \tag{1}$$

where $\mathscr{F}_t$ is the σ-field[2] generated by $\{Y_k\}_{1 \le k \le t}$. The visiting function $f(\mathbf{S}_i(t))$ determines how the transition probability changes over time. Obviously, it has two special cases: $f(\mathbf{S}_i(t)) = 1$ for random walk, and $f(\mathbf{S}_i(t)) = \mathbf{S}_i(t) + 1$ for VRRW. In other words, the transition probability is always constant in traditional random walk, whereas linearly relying on the visited times of one node in VRRW.

VDRW is a stochastic process in which the transition probability to one node is decreased each time it is visited, resulting in more likely to explore the unvisited nodes compared to random walk or VRRW. It is inversely proportional to the visited times $\mathbf{S}_i(t)$ at time t. In this paper, we define the visiting function $f(\mathbf{S}_i(t))$ for VDRW as follows:

$$f(\mathbf{S}_i(t)) = \alpha^{\mathbf{S}_i(t)} \tag{2}$$

[2] σ-field on a set X is a collection of the whole subsets of X that includes the empty subset.



(a) Karate Network (b) Random Walk (c) VRRW (d) VDRW

Fig. 3: The node visited frequency distribution after walking 2000 steps on Zachary's Karate network [28] (The node size corresponds to the visited times and the colors corresponds to different node classes)

where $0 < \alpha < 1$ is a specific hyperparameter. Intuitively, when node v_i is visited, the transition to v_i from all other nodes will be diminished as shown in Figure 2(c).

Of importance of this algorithm is to change the transition probabilities with respect to the number of times one node has been visited. Nevertheless, it is fundamentally different from VRRW. In VRRW, the transition probabilities to one node increase each time it is visited. High-degree nodes would be visited frequently because the transition probabilities to those nodes are more likely to be increased. On the contrary, VDRW encourages the random particle to explore the unvisited or less-frequently visited nodes in the network. That is, the visited frequency distribution using VDRW would be denser than that using VRRW or random walk. Figure 3 is the visited frequency distribution for all the nodes on Zachary's Karate network [28] after walking 2000 steps (best viewed in color). It can be observed that the visited frequency in random walk follows the node degree distribution on the original graph. And it is much sparser in VRRW than random walk because VRRW implicitly forces the high-degree nodes to be easily visited by gradually increasing the transition probabilities to those nodes. On the contrary, VDRW would encourage all the nodes to be visited smoothly as shown in Figure 3(d). Besides, the experiments in Section V-B demonstrate that the nodes tend to walk within the same class using short VDRW.

The intuition of adopting VDRW in our network embedding framework can be simply explained as follow. Short VDRW decreases the transition probabilities to one node when it is visited each time, leading to explore unvisited or less-frequently visited nodes instead of high-degreed nodes. That is, nodes in the imbalanced network would have smooth visited frequency distribution. And the transition to the majority class would be more likely to be decreased. That forced the nodes in the minority class to walk within the same class rather than explore the nodes in the majority class. Thus high-quality node-context pairs can be extracted for our network representation framework using short VDRW.

C. Convergence Analysis

In this subsection, we analyze the convergence properties of the function $\mathbf{V}(t) \in \mathbb{R}^n$ with respect to step t, in which the i-th element is defined as $\mathbf{V}_i(t) = f(\mathbf{S}_i(t))/\sum_i f(\mathbf{S}_i(t))$. Obviously, $\mathbf{V}(t)$ belongs to the $(n-1)$-simplex[3] $\Delta \subseteq \mathbb{R}^n$.

[3] k-simplex is defined as a k-dimensional polytope with $k+1$ vertices.

To analyze the convergence of VDRW, we use the same definitions as [22]. For any vector $\mathbf{v} \in \Delta$ with $H(\mathbf{v}) = \sum_{ij} \mathbf{R}_{ij} v_i v_j > 0$, Markov transition matrix $\mathbf{M} \in \mathbb{R}^{n \times n}$ is defined as:

$$\mathbf{M}_{ij}(\mathbf{v}) = \frac{\mathbf{R}_{ij} v_j}{\sum_j \mathbf{R}_{ik} v_k} \tag{3}$$

And let $\mathcal{C} \subseteq \Delta$ be the set of points $\mathbf{v}$ with $\pi(\mathbf{v}) = \mathbf{v}$ where the vector $\pi(\mathbf{v}) \in \Delta$ is defined as:

$$\pi_i(\mathbf{v}) = \frac{\sum_j \mathbf{R}_{ij} v_i v_j}{H(\mathbf{v})} \tag{4}$$

Let $\mathcal{C}_0 \subseteq \Delta$ be the set of points $\mathbf{v}$ for which $\mathbf{M}(\mathbf{v})$ is reducible. The following theorem provides our main results regarding the convergence of VDRW.

Theorem 1. *With probability one, dist* $(\mathbf{V}(t), \mathcal{C} \cup \mathcal{C}_0) \longrightarrow 0$ *where dist* $(x, A) = \inf\{|x - y| : y \in A\}$.

To prove this theorem, we introduce the following lemma regarding the convergence conditions of VDRW. When the defined visiting function satisfies this lemma, it can be proven that Theorem 1 holds for VDRW.

More specifically, for any t, let $\mathbf{M}_t(t), \mathbf{M}_t(t+1), \cdots$ denote a Markov chain with fixed transition matrix $\mathbf{M}(\mathbf{V}(t))$ beginning at Y_t at time t, $\mathbf{S}'(t) = \mathbf{S}(t)$ and $\mathbf{S}'(i) = \mathbf{S}'(i-1) + e_{\mathbf{M}_t(i)}$ for $i > t$ where e_j denotes the j-th standard basis vector. Let $\mathcal{N}$ be a closed subset of a simplex with $\mathcal{N} \cap (\mathcal{C} \cup \mathcal{C}_0) = \emptyset$.

Lemma 2. *If* $\{f(\mathbf{S}'(i))\}$ *has Markov property, and* $\big(f(\mathbf{S}'(t+L)) - f(\mathbf{S}'(t))\big)/\big(\sum_i f(\mathbf{S}'_i(t+L)) - \sum_i f(\mathbf{S}'_i(t))\big)$ *approaches a point-mass at* $\pi(\mathbf{v})$ *in distribution as* L *increases, then there exists* $c > 0$ *such that* $\mathbf{E}\big(H(\mathbf{V}(n+L))|\mathbf{V}(n)\big) > H\big(\mathbf{V}(n)\big) + c/n$ *whenever* $\mathbf{V}(n) \in \mathcal{N}$.

Proof. The detailed proof is omitted here for brevity. □

When $f(\mathbf{S}(t)) = 1 + \log\alpha \cdot \mathbf{S}(t)$, we have the following result.

$$\begin{aligned}
&\frac{f(\mathbf{S}'(t+L)) - f(\mathbf{S}'(t))}{\sum_i f(\mathbf{S}'_i(t+L)) - \sum_i f(\mathbf{S}'_i(t))} \\
&= \frac{(1 + \log\alpha \cdot \mathbf{S}'(t+L)) - (1 + \log\alpha \cdot \mathbf{S}'(t))}{(n - \log\alpha \cdot \sum_i \mathbf{S}'_i(t+L)) - (n - \log\alpha \cdot \sum_i \mathbf{S}'_i(t))} \\
&= \frac{\log\alpha \cdot (\mathbf{S}'(t+L) - \mathbf{S}'(t))}{\log\alpha \cdot L} \\
&= \frac{\mathbf{S}'(t+L) - \mathbf{S}'(t)}{L}
\end{aligned}$$

As shown in [22], $(\mathbf{S}'(t+L) - \mathbf{S}'(t))/L$ approaches a point-mass at $\pi(\mathbf{v})$ in distribution as L increases. Therefore, $f(\mathbf{S}(t)) = 1 + \log\alpha \cdot \mathbf{S}(t)$ meets the basic condition in the Lemma 2 above. It is obvious that the function $f(\mathbf{S}(t))$ defined in Equation (2) has the following Taylor expansion:

$$f(\mathbf{S}(t)) = \alpha^{\mathbf{S}(t)} = 1 + \log\alpha \cdot \mathbf{S}(t) + O(\log\alpha \cdot \mathbf{S}(t)) \quad (5)$$

Thus, we use $f(\mathbf{S}(t)) = \alpha^{\mathbf{S}(t)}$ to define the visiting function in VDRW. Moreover, the following Corollary 3 presents the conditions under which the convergence of $\mathbf{V}(t)$ is almost surely.

Corollary 3. *If all the off-diagonal entries of* $\mathbf{R}$ *are positive and all the principal minors of* $\mathbf{R}$ *are invertible,* $\mathbf{V}(t)$ *converges almost surely.*

Proof. The detailed proof is omitted here for brevity. □

Regarding the convergence properties of VDRW, we would like to make the following two remarks.

Remark 1: As shown in [22], VRRW enjoys similar convergence properties as VDRW. However, the completely different definition of the visiting function makes the proposed VDRW more suitable for imbalanced networks.

Remark 2: We would like to point out that for real networks, the graph structure hardly meets the convergence conditions in Corollary 3. However, as we will show in the experimental results, short VDRW is still an effective way to extract node-context pairs in learning network representation from imbalanced networks, especially in comparison with state-of-the-art techniques. In Section V-B, we further investigate the convergence of VDRW on the real networks.

IV. The Proposed Embedding Framework

In this section, we present the semi-supervised network representation framework based on Vertex-Diminished Random Walk. Currently most of the existing semi-supervised network representation approaches (e.g., [1], [2]) aim to learn the network embedding implicitly assuming that the labeled set has roughly equal examples for each class in the network. However, when applied on imbalanced networks, i.e., some classes having significantly more examples (vertices) than the other classes, these approaches will suffer from suboptimal performance. Take Figure 3(a) as an example. Karate network has two imbalanced classes where one has 5 labeled examples and the other one has 29 examples. On such an imbalanced network, the embedding learned using [2] would make the two classes non-separable from each other, as shown in Figure 1(a). This is mainly due to inaccurate node-context pairs generated in these methods (i.e., the center node and the context node belong to different classes), which can be greatly improved using the proposed VDRW.

Formally, the imbalanced network embedding problem can be defined as follows.

Problem 1. ***(Imbalanced Network Embedding)***

Input: *(i) a (directed or undirected) network* $G = (V, E)$*, and (ii)* ***imbalanced class labels*** *for nodes in* V

Output: *a low-dimensional vector representation for each vertex* $v \in V$*, so that the minority classes are naturally separated from majority class in the embedding feature space.*

When the label information of only a small portion of the entire network is given, the above problem of imbalanced network embedding would be semi-supervised in nature, as the learned network representation would preserve both the graph structure and the label information in the low-dimensional feature space.

A. Semi-supervised Network Representation Learning

Suppose that we are given the network $G = (V, E)$ with l labeled nodes $\{\mathbf{x}_i\}_{i=1}^{l}$ associated with label $\{\mathbf{y}_i\}_{i=1}^{l}$[4] and $n-l$ unlabeled nodes $\{\mathbf{x}_i\}_{i=l+1}^{n}$ where $\mathbf{x}_i$ is the feature vector of node v_i. The goal is to learn a d-dimensional vector e_i for each node v_i using the class label information and the graph structure. The semi-supervised network representation framework can be formulated as an optimization problem with the following loss function:

$$\mathcal{L} = \mathcal{L}_s + \lambda\mathcal{L}_u \quad (6)$$

where the first term denotes the prediction loss with respect to the labeled examples, and the second term measures the consistency on the graph. λ is a user-defined constant factor that balances between the two terms.

Since the node label can be predicted using both the original features on the node and the embedding features, we use two feed-forward neural network models to train based on the node features and the embedding features, respectively [2], and then concatenate them to predict the label. Therefore, the supervised loss on the labeled nodes can be expressed as follows.

$$\mathcal{L}_s = -\frac{1}{l}\sum_{i=1}^{l}\log\frac{\exp([\mathbf{h}^{l_1}(\mathbf{x}_i), \mathbf{h}^{l_2}(\mathbf{e}_i)]^T\mathbf{y}_i)}{\sum_{\mathbf{y}'}\exp([\mathbf{h}^{l_1}(\mathbf{x}_i), \mathbf{h}^{l_2}(\mathbf{e}_i)]^T\mathbf{y}')} \quad (7)$$

where l_1 and l_2 represent the number of hidden layers in the neural network models using node features and embeddings, respectively. $[\cdot,\cdot]$ concatenates two column vectors. The function $\mathbf{h}(\cdot)$ denotes the output of the neural network model.

Assuming that nearby nodes have the similar embeddings, the graph-based loss function is formulated as follow [26].

$$\mathcal{L}_u = -\sum_{(v_i,v_c)}(\log\sigma(\mathbf{w}_c^T\mathbf{e}_i) + k\cdot\mathbb{E}_{v_n\sim P_n(v_c)}\log(\sigma(-\mathbf{w}_n^T\mathbf{e}_i))) \quad (8)$$

where (v_i, v_c) denotes the node-context pair sampled from the network w.r.t. node v_i. $\mathbf{w}_c \in \mathbb{R}^d$ is a vector representation of v_c when v_c acts as the context. Node v_n is randomly selected from a negative sampling distribution P_n [12], and k is the number of negative samples. $\sigma(x) = 1/(1+e^{-x})$ is the sigmoid function.

Of key importance for unsupervised loss is the extraction of node-context pairs in the imbalanced network. As mentioned

[4] $\mathbf{y}_i \in \mathbb{R}^C$, where C denotes the number of classes in the network.

before, VDRW can capture effectively the node-context pairs for the minority class. Based on that, a ***Context Sampling*** method is presented to integrate known class-label information with graph structure for extracting node-context pairs. In addition, mini-batch training method is used in our model training. During the embedding learning, a batch of nodes is randomly selected as the initial points $\mathcal{S}$ to extract node-context pairs at each iteration. Majority class has the higher probability to be selected according to the imbalanced distribution than minority class. Thus, ***Balanced-batch Sampling*** is introduced to balance those pairs with respect to class-label in model training.

The training process in our network representation learning framework is illustrated in Algorithm 1. It is given the graph G with node features $\mathbf{x}_{1:n}$ and partial labels $y_{1:l}$, and the training model parameters (batch iterations $T1$ and $T2$, the negative sampling size k, embedding size d) as input, and outputs the embedding vector $\mathbf{e}_i$ for each node v_i. Lines 1-8 demonstrate the training process in learning the embedding vector for each node in the imbalanced network. Lines 9-12 correspond to the process of predicting the label using node feature $\mathbf{x}_i$ and embedding vector $\mathbf{e}_i$. And stochastic gradient descent (SGD) is adopted to train our model.

Algorithm 1 ***Model Training***

Input: Graph G with features $\mathbf{x}_{1:n}$ and labels $y_{1:l}$
batch iterations $T1$ and $T2$
negative sampling size k
embedding size d
Output: Embedding vector $\{\mathbf{e}_i\}_{i=1}^{n}$

1: **for** $t \leftarrow 1$ to $T1$ **do**
2: Initialize $\mathcal{S}$ using ***Balanced-batch Sampling***;
3: **for** $s \leftarrow 1$ to $|\mathcal{S}|$ **do**
4: Sample positive pairs using ***Context Sampling***;
5: Sample k negative pairs;
6: **end for**
7: Update the model parameters in Eq. (8) using SGD;
8: **end for**
9: **for** $t \leftarrow 1$ to $T2$ **do**
10: Sample a batch of $\{\mathbf{x}_i, y_i\}$ from labeled instances;
11: Update the model parameters in Eq. (7) using SGD;
12: **end for**
13: Return the embedding vector $\mathbf{e}_i \in \mathbb{R}^d$ for each node.

B. Context Sampling

In node-context pairs extraction, DeepWalk [5] has proven the effectiveness of random walk in exploring the context for each node in the network. The core idea in random walk is to choose one neighbor based on the transition probabilities. As discussed in Section III, random walk might not capture the high-quality node-context pairs, especially for the nodes within minority class. Thus, VDRW is introduced to explore the context for nodes by adjusting the transition probabilities with respect to their visited times. Based on the proposed VDRW, we present a ***Context Sampling*** method integrating label information with network topological structure.

In semi-supervised learning, it is implicitly assumed that examples from the same class would be close to each other in the feature space. That is the reason why labeled nodes can naturally be the contexts for each other if they have identical labels. Planetoid [2] also exploited this assumption. However, they extracted node-context pairs using labels and graph structure separably. That is, context is either determined by random walk or by labels, but not both. Here we adopt a jumping strategy in which one node may jump directly to another node during random walk if they have the same labels. That would encourage one node to explore the context though they are not close in the network. To some extent, it enlarges the size of the neighborhood by using known label information.

The ***Context Sampling*** method for extracting node-context pairs is illustrated in Algorithm 2. If one node is labeled, it can visit another node with the same label directly with probability r $(0 < r < 1)$. With probability $1 - r$, it selects the neighbors based on the transition matrix. After each step, the transition matrix is updated using VDRW. The node-context pairs can be extracted from the walking path as did in DeepWalk [5].

Algorithm 2 ***Context Sampling***

Input: Graph G with transition matrix $\mathbf{R}$, labels $y_{1:l}$
initial node v_s
jumping probability r
length of walking sequences T
Output: Node-context pairs (v_i, v_c)

Initialize: Add v_s to walking path p;
for $t = 0$ to $T - 1$ **do**
if Node is labeled **then**
With r, select one node v with the same label;
With $1 - r$, select its neighbor v based on $\mathbf{R}$;
else
Select its neighbor v based on $\mathbf{R}$;
end if
Update $\mathbf{R}$ using VDRW;
Update the walking path p;
end for
Return the node-context pairs sampled from path p.

C. Balanced-batch Sampling

Mini-batch training is commonly used in building neural network models [1], [2] assuming that the training instances are class-balanced. In our framework, however, the number of node-context pairs in different classes would be typically imbalanced if we randomly select a batch of nodes as the initial nodes. In other words, the imbalance characteristic is preserved in those mini-batches even though we could find node-context pairs correctly. To avoid this issue, we adopt a simple ***Balanced-batch Sampling*** method for mini-batch selection at each iteration.

Considering the binary imbalanced learning as an example, there are three different sets of nodes on the training model: labeled majority S_{maj}, labeled minority S_{min} and unlabeled instances S_u. In most cases, the batch size is significantly larger than the number of labeled instances as only a very small portion of nodes are labeled. Balanced-batch sampling randomly selects a subset S'_{maj} from the labeled majority class such that $|S'_{maj}| = |S_{min}|$, and then sampled a subset S'_u from the unlabeled data. Consequently, the set given by ***Balanced-batch Sampling*** $S = S_{min} \cup S'_{maj} \cup S'_u$ will be used as initial nodes in model training.

It is close to the under-sampling method which reduces the instance to balance data distribution. In general, random under-sampling may cause the training model to miss important details related to the majority class [9]. Nevertheless, it would be alleviated in our mini-batch training because it considers the entire graph nodes via iterative training. Note that in mini-batch model training, the examples in the minority class might be adopted at each iteration by combining with randomly selected examples in the majority class if they are less than the batch size $T1$.

V. Experiments

In this section, we conduct the experiments on several real-world networks to validate the effectiveness of the proposed *ImVerde* by comparing to several state-of-the-art methods. Vertex-Diminished Random Walk (VDRW) is proposed to extract the node-context pairs for our imbalanced network embedding framework, which is denoted as *ImVerde-VDRW*. Additionally, Vertex-Reinforced Random Walk (VRRW) [22] is also considered in our framework (denoted as *ImVerde-VRRW*) for comparison.

In particular, we focus on the following questions: (1) How effective is short VDRW for exploring the context on the imbalanced networks compared with short random walk and VRRW? (2) Compared to other state-of-the-art methods, how effective is the *ImVerde* framework for learning network representation from imbalanced data? (3) What are the impacts of class-imbalanced distribution for the *ImVerde*?

A. Setup

Data sets: We use several node classification data sets to validate the effectiveness of the proposed method, including Karate, Cora, Citeseer, Pubmed and NTSB data sets. Table I listed the detailed data description. The model effectiveness on those data sets can be evaluated by predicting the correct label for nodes in the minority class.

- Karate network [28]. It is a social network of the karate club, which represents the social relationship of 34 members in the karate club. Karate network has 34 nodes and 156 edges. To validate the effectiveness of VDRW on imbalanced data, We consider it as a binary imbalanced network as shown in Figure 3(a) where the minority class contains only 5 nodes and the other 29 nodes belong to the majority class.
- Cora [33], Citeseer [34] and Pubmed [40] networks. All of them are the citation networks. For those data sets, each node corresponds to one scientific publication and the edge represents the citation relationship between two publications. And each publication is described by a sparse bag-of-words feature vector. Originally, there are several classes for every data set where each class has the roughly equal number of examples (nodes). In our experiments, we reconstruct them for binary imbalanced network analysis. Specifically, we select 20 nodes from one class and 120 nodes from the rest classes as the labeled training examples. Besides, 1000 nodes are chosen as the test examples and others are unlabeled training examples. Take Cora data set as an example, seven imbalanced binary data sets are possible to be reconstructed considering choosing one of them as the minority class (marked as Cora_1, $\cdots$, Cora_7, respectively). The imbalance ratio of the number of examples in minority class over that in majority class is 20:120 for each reconstructed Cora data set.
- NTSB network. It is built from the historical accident reports published in National Transportation Safety Board database[5]. Those reports can be naturally modeled as a network in which each node represents one accident report and each edge reflects the similarity between two reports. It contains 1473 nodes and 11784 edges where each node is associated with a 4424-dimensional TF-IDF feature vector [31] extracted from the accident report. For NTSB data set, the imbalance ratio is 5:5:5:200. Therefore, it is a significantly imbalanced network with 4 classes. The number of labeled training instances in the majority class is 40 times more than that of each minority class. And there are 500 test instances. This data set is used to validate the effectiveness of the proposed method on multi-class classification from imbalanced data.

TABLE I: Data statistics

Dataset	Karate	Cora	Citeseer	Pubmed	NTSB
Nodes	34	2,708	3,327	19,717	1,473
Edges	156	5,429	4,732	44,338	11,784
Classes	2	7	6	3	4
Features	—	1,433	3,703	500	4,424

Baselines: Four baseline methods are adopted in our experiments to validate the effectiveness of the proposed approach.

- DeepWalk [5]. It is an unsupervised network embedding method which first considered to use truncated random walk to explore the context as word embedding models. The learned node representations encoded the graph topological structure in a continuous vector space.
- node2vec [23]. It is a semi-supervised model for learning node representation in networks. In order to smoothly interpolate between BFS and DFS, it designed a biased random walk for neighborhood definition where two parameters p and q were adopted to balance the depth

[5]https://www.ntsb.gov/Pages/default.aspx

and width of node neighborhood. And these parameters would be learned using the labeled training data in a semi-supervised fashion.

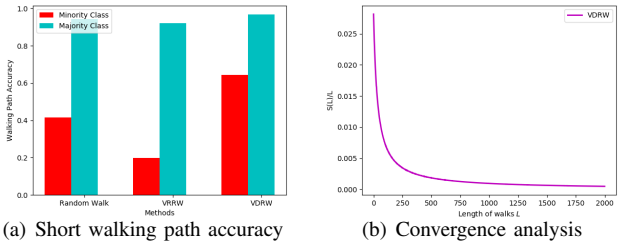
- Planetoid [2]. It is also a semi-supervised learning framework where each node's class label and neighborhood context would be jointly predicted. It used both random walk on the graph and node label information to define the neighborhood context based on the assumption that nearby nodes or nodes with the same class label have the similar embeddings in the feature space. The authors provided both transductive and inductive versions of Planetoid (Planetoid-T and Planetoid-I) for learning node representation. In this paper, we only consider the transductive version (denoted as Planetoid in our experiments instead of Planetoid-T in [2]). The inductive *ImVerde* can be modeled using the similar strategy as Planetoid-I.
- SMOTE [11]. It is an over-sampling approach for imbalanced data, in which the synthetic samples are created between each minority instance and its nearest neighbors. Therefore, we use this method to simply balance the data distribution on the original feature space. It assumed that the data are independent and identically distributed without considering any graph properties.

B. Case Study: Zachary's Karate Network

In Section III, we observed that VDRW encourages the nodes to be visited with similar frequency. This characterization would force short VDRW to walk within the same class. We conduct a case study on Zachary's Karate network [28] to validate the conclusion empirically. The strategies including random walk, VRRW and VDRW are used to walk for short steps starting from each node. The walking length and walking times are set as 10 and 100, respectively. We statistic the walking path accuracy, which defines how many nodes in the walking path are located within the same class. Figure 4(a) gives the average walking accuracy in both minority and majority class for those methods. It can be seen that VDRW improves the walking accuracy in both classes. That is, it encourages then nodes to walk within the same class. As the node-context pairs are extracted from the walking path in *ImVerde*, the high-quality node-context pairs would improve the embedding performance for imbalanced networks. The convergence of VDRW is also evaluated on Karate network. We statistic the difference of the visited times $\mathbf{S}(L)/L$ after each step. As shown in Figure 4(b), the difference keeps decreasing with respect to the walking steps L. Additionally, the node representations learned by *ImVerde* and Planetoid are visualized in the 2-dimensional space using t-SNE as shown in Figure 1.

C. Node Classification

We conduct the node classification experiments on four real networks to evaluate the proposed framework *ImVerde*. In VDRW, we select the parameters $\alpha = 0.7$ and $r = 0.2$, which is discussed in details later. And we use the same random walk parameters for all the used methods: walking length $T = 10$, negative samples $k = 10$, embedding size $d = 50$. The logistic regression classifier is used to evaluate the node representations learned from DeepWalk, node2vec and SMOTE in our experiments.



(a) Short walking path accuracy (b) Convergence analysis

Fig. 4: The analysis of VDRW on Karate network

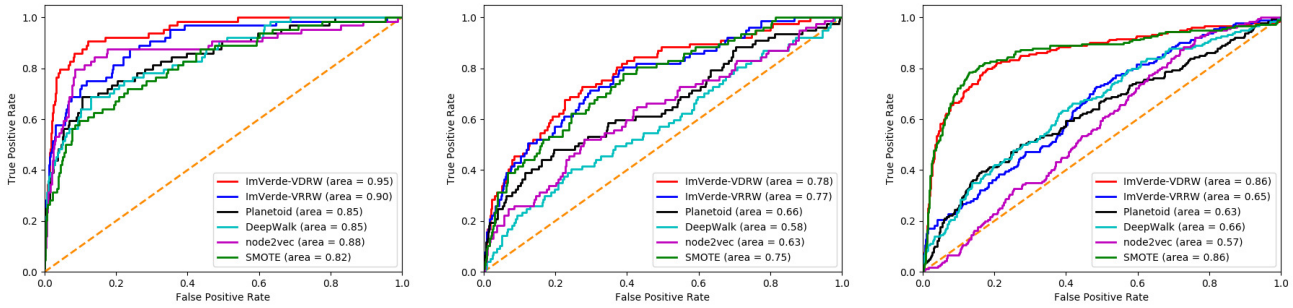
TABLE II: Average precision for the binary node classification (The best results are indicated in bold)

	Planetoid	DeepWalk	node2vec	SMOTE	*ImVerde-VDRW*
Cora_1	0.639	0.650	0.612	0.473	**0.720**
Cora_2	0.749	0.778	0.761	0.664	**0.852**
Cora_3	0.900	0.863	0.840	0.707	**0.910**
Cora_4	0.783	0.657	0.761	0.723	**0.826**
Cora_5	0.637	0.610	0.471	0.641	**0.769**
Cora_6	0.557	0.722	0.505	0.584	**0.805**
Cora_7	0.501	0.474	0.503	0.404	**0.657**
Citeseer_1	0.238	0.116	0.169	0.252	**0.304**
Citeseer_2	0.396	0.225	0.250	**0.550**	0.513
Citeseer_3	0.651	0.616	**0.693**	0.605	0.666
Citeseer_4	0.608	0.352	0.415	0.729	**0.732**
Citeseer_5	0.706	0.512	0.610	**0.790**	0.785
Citeseer_6	0.591	0.310	0.432	0.644	**0.651**
Pubmed_1	0.298	0.335	0.203	0.631	**0.631**
Pubmed_2	0.454	0.632	0.497	**0.669**	0.663
Pubmed_3	0.489	**0.736**	0.462	0.685	0.653

TABLE III: Multi-class text classification on NTSB data set

	Accuracy
Planetoid	0.311
DeepWalk	0.458
node2vec	0.408
ImVerde-VRRW	0.416
ImVerde-VDRW	**0.460**

Figure 5(a) gives the Receiver Operating Characteristic (ROC) curve and AUC value for predicting the minority class on Cora_7, which means that the 7-th class is selected as the minority while other classes are the majority on Cora. We observe the similar results on other reconstructed binary data sets of Cora, but omit them due to the limited space. Figure 5(b) and Figure 5(c) give the ROC curves and AUC values on Citeseer_6 and Pubmed_1, respectively. More specifically, Table II lists the average precisions on Cora, Citeseer and Pubmed data sets where the best results are highlighted in bold. Besides, we conduct the multi-class node classification on NTSB data set with class-imbalanced labeled training instances. The multi-classification accuracy on NTSB data set is given in Table III. It shows that *ImVerde-VDRW* outperforms *ImVerde-VRRW* in these data sets. Short VDRW encourages the node-context pairs to be extracted within the same class. On the contrary, short VRRW would force one node to visit other high-degree nodes. The quality of node-context pairs is



(a) Cora (b) Citeseer (c) Pubmed

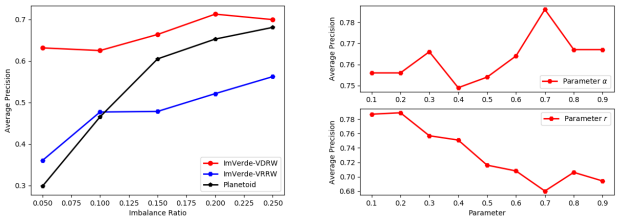
Fig. 5: The ROC curve and AUC value for node classification on Cora, Citeseer and Pubmed data sets (best seen in color)

crucial to the random walk based embedding approaches when the labeled data is class-imbalanced.

We achieve the significant improvement over DeepWalk on Cora, Citeseer and Pubmed. DeepWalk[6] adopted short random walk to extract the context for each node from the network. Node2vec[7] designed another random walk strategy to interpolate between BFS and DFS. It stated that the likelihood of immediately revisiting a node decreases. That idea is similar to VDRW which decreases the transition probability each step to explore the neighborhood. The short random walk in DeepWalk can be seen as a special case for both node2vec and VDRW ($p = q = 1$ for node2vec, $\alpha = 1$ for VDRW). But there are some significant differences between them. First, the affected area is different. When node v_j is visited from v_i, node2vec considers updating the probability of links v_j will visit next time. In contrast, the probability of any transition to v_j would decrease in VDRW. Second, the transition probability fluctuates temporarily in node2vec, whereas VDRW in *ImVerde* preserves the diminished probability for the whole node-context extraction process. Planetoid[8] is a semi-supervised learning framework which used both random walk and labels to extract node-context. But it does not perform well on imbalanced data classification due to the node-context pairs extracted using random walk. On the other hand, SMOTE[9] simply balanced the data distribution using attributes without graph information. Thus the node attributes have a great impact on its classification performance. It achieves the worst performance on Cora data set, but gets excellent classification results on other data sets. In such cases, the embedding representation learned from *ImVerde-VDRW* is still competitive.

D. Imbalance Analysis

Since we aim to learn the network representation from imbalanced data, the imbalance ratio impacts on the embedding performance for our framework. We consider three semi-supervised learning frameworks: Planetoid, *ImVerde-VRRW* and *ImVerde-VDRW*. Figure 6(a) gives the binary node classification result on Pubmed_1 with different imbalance ratio using different frameworks. When the imbalance ratio ranges from 0.050 to 0.250, it shows that the average precision of *ImVerde-VDRW* is slightly affected compared to Planetoid and *ImVerde-VRRW*. And *ImVerde-VDRW* outperforms other two methods when the network is extremely class-imbalanced. The classification performance of Planetoid is heavily relying on the imbalance ratio of labeled training data because it implicitly assumes that the labeled instances have the approximately balanced distribution.



(a) The imbalance ratio analysis (b) Parameter sensitivity analysis

Fig. 6: Model analysis

E. Parameter Sensitivity

There are two crucial hyper-parameters (α and r) in our ***Context Sampling*** strategy of *ImVerde* where α ($0 < \alpha < 1$) controls how the transition probability decreases w.r.t. visited times in VDRW and r is the probability of jumping to another node with the same label in ***Context Sampling***. Take Cora_1 data set as an example, we investigate the node-classification performance with different parameter values. Figure 6(b) shows the average precision of the proposed framework using different parameter values, where we fix one parameter when investigating the classification performance w.r.t the other one. It can be observed that both α and r affect the classification performance of *ImVerde*.

VI. Conclusion

In this paper, we present a novel Vertex-Diminished Random Walk model, which reduces the transition probability to one node each time it is visited. This property makes it particularly suitable for analyzing imbalanced networks, as it encourages the random particle to walk within the same

[6]https://github.com/phanein/deepwalk
[7]https://snap.stanford.edu/node2vec
[8]https://github.com/kimiyoung/planetoid
[9]https://github.com/scikit-learn-contrib/imbalanced-learn

class, thus generating more accurate node-context pairs for the following network embedding task. Then based on VDRW, we propose *ImVerde*, a semi-supervised network representation learning framework for imbalanced networks. In this framework, we use VDRW and the label information to capture node-context pairs by assuming nodes with the same context or label have the similar embedding feature. Furthermore, we adopt a simple under-sampling method to balance those pairs from different classes. We compare *ImVerde* with state-of-the-art techniques on several imbalanced networks. The experimental results demonstrate the effectiveness of the proposed framework on node classification, especially when the labeled data is extremely class-imbalanced.

ACKNOWLEDGMENT

This work is supported by the United States Air Force and DARPA under contract number FA8750-17-C-0153, National Science Foundation under Grant No. IIS-1552654, Grant No. IIS-1813464 and Grant No. CNS-1629888, NASA under Grant No. NNX17AJ86A, the U.S. Department of Homeland Security under Grant Award Number 17STQAC00001-02-00, and an IBM Faculty Award. The views and conclusions are those of the authors and should not be interpreted as representing the official policies of the funding agencies or the government.

REFERENCES

[1] J. Liang, P. Jacobs, J. Sun and S. Parthasarathy, "Semi-supervised embedding in attributed networks with outliers." In SDM, 153-161, 2018.

[2] Z. Yang, W. W. Cohen, and R. Salakhutdinov, "Revisiting semi-supervised learning with graph embeddings." In ICML, 40-48, 2016.

[3] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding." Science, 290(5500), 2323-2326, 2000.

[4] J. B. Tenenbaum, V. De Silva and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction." Science, 290(5500), 2319-2323, 2000.

[5] B. Perozzi, R. Al-Rfou and S. Skiena, "DeepWalk: Online learning of social representations." In SIGKDD, 701-710, 2014.

[6] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan and Q. Mei, "LINE: Large-scale information network embedding." In WWW, 1067-1077, 2015.

[7] S. Cao, W. Lu and Q. Xu, "GraRep: Learning graph representations with global structural information." In CIKM, 891-900, 2015.

[8] X. Wang, P. Cui, J. Wang, J. Pei, W. Zhu and S. Yang, "Community Preserving Network Embedding." In AAAI, 203-209, 2017.

[9] H. He and E. A. Garcia, "Learning from imbalanced data." IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263-1284, 2009.

[10] C. Huang, Y. Li, C. L. Chen and X. Tang, "Learning deep representation for imbalanced classification." In CVPR, 5375-5384, 2016.

[11] N. V. Chawla, K. W. Bowyer, L. O. Hall and W. P. Kegelmeyer, "SMOTE: synthetic minority over-sampling technique." Journal of artificial intelligence research, 16, 321-357, 2002.

[12] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality." In NIPS, 3111-3119, 2013.

[13] W. L. Hamilton, R. Ying and J. Leskovec, "Representation Learning on Graphs: Methods and Applications." IEEE Data Engineering Bulletin, 2017.

[14] T. N. Kipf, M. Welling, "Semi-supervised classification with graph convolutional networks." In ICLR, 2017.

[15] J. Chen, T. Ma and C. Xiao, "FastGCN: fast learning with graph convolutional networks via importance sampling." In ICLR, 2018.

[16] H. He, Y. Bai, E. A. Garcia and S. Li, "ADASYN: Adaptive synthetic sampling approach for imbalanced learning." In IJCNN, 1322-1328, 2008.

[17] C. Elkan, "The foundations of cost-sensitive learning." In IJCAI, 17(1), 973-978, 2001.

[18] J. He, Y. Liu and R. Lawrence, "Graph-based rare category detection." In ICDM, 833-838, 2008.

[19] D. Zhou, J. He, K. S. Candan and H. Davulcu, "MUVIR: Multi-View Rare Category Detection." In IJCAI, 4098-4104, 2015.

[20] D. Zhou, J. He, H. Yang and W. Fan, "SPARC: Self-Paced Network Representation for Few-Shot Rare Category Characterization." In SIGKDD, 2018.

[21] D. Zhou, K. Wang, N. Cao, J. He, "Rare category detection on time-evolving graphs." In ICDM, 1135-1140, 2015.

[22] R. Pemantle, "Vertex-reinforced random walk." Probability Theory and Related Fields, 92(1), 117-136, 1992.

[23] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks." In SIGKDD, 855-864, 2016.

[24] C. Yang, Z. Liu, D. Zhao, M. Sun and E. Y. Chang, "Network Representation Learning with Rich Text Information." In IJCAI, 2111-2117, 2015.

[25] M. Defferrard, X. Bresson and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering." In NIPS, 3844-3852, 2016.

[26] W. Hamilton, Z. Ying and J. Leskovec, "Inductive representation learning on large graphs." In NIPS, 1025-1035, 2017.

[27] D. Wang, P. Cui and W. Zhu, "Structural deep network embedding." In SIGKDD, 1225-1234, 2016.

[28] W. W. Zachary, "An information flow model for conflict and fission in small groups." Journal of Anthropological Research, 33(4), 452-473, 1977.

[29] L. V. D. Maaten and G. Hinton, "Visualizing data using t-SNE." Journal of Machine Learning Research, 9(Nov), 2579-2605, 2008.

[30] Y. A. Lai, C. C. Hsu, W. H. Chen, M. Y. Yeh and S. D. Lin, "PRUNE: Preserving Proximity and Global Ranking for Network Embedding." In NIPS, 5257-5266, 2017.

[31] G. Salton, and C. Buckley, "Term-weighting approaches in automatic text retrieval." Information Processing & Management, 24(5), 513-523, 1988.

[32] J. Tang,M. Qu, and Q. Mei, "PTE: Predictive text embedding through large-scale heterogeneous text networks." In SIGKDD, 1165-1174, 2015.

[33] A. K. McCallum, K. Nigam, J. Rennie, and K. Seymore, "Automating the construction of internet portals with machine learning." Information Retrieval, 3(2), 127-163, 2000.

[34] C. L. Giles, K. D. Bollacker, S. Lawrence, "CiteSeer: An automatic citation indexing system." In Proceedings of the third ACM conference on Digital libraries, 89-98, 1998.

[35] J. Qiu, Y. Dong, H. Ma, J. Li, K. Wang, and J. Tang, "Network embedding as matrix factorization: Unifying DeepWalk, LINE, PTE, and node2vec." In WSDM, 459-467, 2018.

[36] A. G. Duran, M. Niepert, "Learning Graph Representations with Embedding Propagation." In NIPS, 5119-5130, 2017.

[37] S. H. Khan, M. Hayat, M. Bennamoun, F. A. Sohel, R. Togneri, "Cost-sensitive learning of deep feature representations from imbalanced data." IEEE transactions on Neural Networks and Learning Systems, 29(8), 3573-3587, 2017

[38] V. López, A. Fernández, S. García, V. Palade, and F. Herrera, "An insight into classification with imbalanced data: Empirical results and current trends on using data intrinsic characteristics." Information Sciences, 250, 113-141, 2013.

[39] I. Tomek, "Two modifications of CNN". IEEE Trans. Systems, Man and Cybernetics, 6, 769-772, 1976.

[40] G. Namata, B. London, L. Getoor, and B. Huang, "Query-driven active surveying for collective classification." In 10th International Workshop on Mining and Learning with Graphs, 2012.

[41] P. Cui, X. Wang, J. Pei, and W. Zhu, "A survey on network embedding." IEEE Transactions on Knowledge and Data Engineering, 2018.

[42] Y. Zhou, J. He, "Crowdsourcing via Tensor Augmentation and Completion." In IJCAI, 2435-2441, 2016.

[43] Y. Zhou, L. Ying, J. He, "MultiC^2: an optimization framework for learning from task and worker dual heterogeneity." In SDM, 579-587, 2017.

[44] X. Huang, J. Li, and X. Hu, "Label informed attributed network embedding." In WSDM, 731-739, 2017.