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

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 exists in many high-impact applications. An example is in air traffic control, where we aim to identify the leading indicators for each type of accident cause from historical records. Among all three types of accident causes, historical records with 'personnel issues' are much more than the other two types ('aircraft issues' and 'environmental issues') combined. Thus, the resulting dataset is highly imbalanced, and can be naturally modeled as a network. Up until now, most existing work on imbalanced data analysis focused on the classification setting, and very little is devoted to learning the node representation from imbalanced networks. To address this problem, in this paper, we propose Vertex-Diminished Random Walk (VDRW) for imbalanced network analysis. The key idea is to encourage the random particle to walk within the same class by adjusting the transition probabilities each step. It resembles the existing Vertex Reinforced Random Walk in terms of the dynamic nature of the transition probabilities, as well as some convergence properties. However, it is more suitable for analyzing imbalanced networks as it leads to more separable node representations in the embedding space. Then, based on VDRW, we propose a semi-supervised network representation learning framework named ImVerde for imbalanced networks, in which context sampling uses VDRW and the label information to create node-context pairs, and balanced-batch sampling adopts a simple under-sampling method to balance these pairs in different classes. Experimental results demonstrate that ImVerde based on VDRW outperforms state-of-the-art algorithms for learning network representation from imbalanced data.

**KEYWORDS**

Network representation, random walk, imbalanced data

**1 INTRODUCTION**

Nowadays, big data is being generated across multiple application domains. Often times, the target data set is imbalanced in terms of the proportion of examples from various classes, or categories. 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 Board¹, there are three major types of accident causes, including aircraft issues, personnel issues, and environment issues. Historical records with 'personnel issues' are much more than the other two types combined, thus creating an imbalanced data set. For such data sets which can be modeled as networks (e.g., each node corresponds to a historical accident, and each edge reflects the similarity between two accidents), despite the extensive existing work on imbalanced classification [3, 8, 11, 13, 21], up until now, very little (if any) effort has been devoted to learning the node representation from imbalanced networks.

On the other hand, 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., the labeled set consisting of roughly equal number of examples from each class. More specifically, traditional unsupervised embedding algorithms [1, 24] are preserving the local neighborhood relationship among nodes without any label information. But they fail to capture high-order graph structure and have extremely high time complexity for analyzing large-scale networks. To solve these problems, recently some works [2, 23, 26] define different high-order proximities between nodes to capture the graph structure. Besides, inspired by word2vec model [17], neural network based node embedding algorithms [6, 9] have been proposed to learn low-dimensional network representations based on the assumption that nodes in similar contexts have similar embeddings. Among these algorithms, short random walk is commonly adopted to automatically detect the node neighborhood as it has been shown that random walk follows the similar power law distribution as word frequency in language modeling [19]. Then, it is extended to semi-supervised learning setting [14, 22], implicitly assuming that the labeled set consists 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 imbalance.

¹https://www.ntsb.gov/Pages/default.aspx
pairs, resulting in node representations not amiable to the following classification task, as shown in Figure 1(a).

To address this problem, in this paper we propose a general framework named ImVerde for learning network representation from imbalanced data. 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. Within the proposed framework, when learning the embedding vectors from imbalanced networks, VDRW can capture the node-context pairs from the minority class with a higher accuracy.

Furthermore, as demonstrated in [28], label information is very effective for obtaining node-context pairs since nodes with the same label are likely to have similar properties. Therefore, we also sample the context using both graph structure and label information. But instead of extracting node-context pairs using the two types of information separately, we propose a jumping scheme to combine them into one 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, to balance the extracted node-context pairs in imbalanced networks, we use a simple under-sampling method for mini-batch sampling. Figure 1(b) shows an example where the proposed ImVerde generates network representation that is naturally separable between the majority and minority classes. To validate the effectiveness of the proposed network representation framework, we conduct experiments on four real network data sets for text classification.

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

- We propose 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 context sampling method extracts node-context pairs based on both graph structure and label information, and balanced-batch sampling method aims to keep the extracted pairs balanced.
- The proposed framework is evaluated on four datasets, with experimental results demonstrating its superior performance over the state-of-the-art techniques. These results are also consistent with our analysis in terms of imbalanced data analysis, separability of different classes, etc.

The rest of the paper is organized as follows. We review the related work in Section 2. VDRW is introduced in Section 3 and Section 4 presents our proposed semi-supervised network representation framework. The extensive experimental results and discussion are provided in Section 5. Finally we conclude the paper in Section 6.

2 RELATED WORK

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

2.1 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 tackle these problems. Up until now, random sampling and synthetically creating instances are two mainstream methods to balance the samples in different classes. Typically, random over-sampling and under-sampling are used to balance the data distribution via adding replicated instances from the minority class or reducing instances from majority class. To solve the overfitting in over-sampling, SMOTE [3] created synthetic samples between each minority instance and its nearest neighbors, and it has been successfully applied to various applications. Another strategy commonly used in imbalanced data is data clean. Tomek links [10] are defined as a pair of samples in opposite classes with minimal distance. By removing or cleaning the overlapping instances, Tomek links usually integrate with other sampling methods to target imbalanced problems. Instead of altering data distribution, cost-sensitive methods [5] consider the penalty for misclassifying instances to another class. And typically misclassified minority instances bears heavier penalty in order to avoid imbalanced problems. Recently deep learning models are adopted to deal with imbalanced data. Integrating with re-sampling and cost-sensitive learning, CNN-based approach [13] was proposed to learn deep representation for imbalanced classification. Additionally, minority classes or rare categories normally play important roles in the database, such as financial fraud in bank systems. Motivated by manifold ranking algorithm, GRADE [12] detected the minority classes based on probability density change in similarity ranking. And MUVIR [29] made use of the relationship among multiple views to determine the probability of belonging to minority classes. 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 focus on the classification problem or the active learning setting.

2.2 Network Representation

The goal of network representation is to learn a low-dimensional dense vector for each node in the network. The basic assumption in most of the existing works is that nodes with similar contexts in the network have the similar embeddings. Normally node representation should preserve the graph structure or other properties. Recently many excellent network representation methods have been proposed. DEEPWALK [19] firstly showed that short random walk follows the same power-law distribution as skip-gram model does in word frequency. And then DEEPWALK was proven to be equivalent to matrix factorization. Based on this, text-associated DEEPWALK [27] was presented by incorporating text features into network representation. LINE [23] was designed to preserve the first- and second-order proximities respectively, and then concatenated them for network embedding. Similarly, GraRep [2] captured k-step relational information for preserving global structural information, and also concatenated all k-step representations. In addition, some work [4, 15, 20] assumes that nodes with similar structural roles should be similar in embedding space. Inspired by this intuition, struc2vec [20] used a hierarchical degree sequence to measure role similarity and then constructed a multilayer graph.
to encode node similarities. GRAPHWAVE [4] learned the low-
dimensional embeddings of nodes to capture the local network
neighborhoods using spectral graph wavelet diffusion patterns.
And it is then extended to semi-supervised network representation
learning with label information by implicitly assuming a balanced
label setting. To capture the highly non-linear network structure,
SDNE [25] used a semi-supervised deep neural network model
with non-linear functions and also preserved first- and second-
order proximities. In Planetoid [28], both label information and
graph structure are used to extract contexts for each node on model
training. And CNN-based deep learning architectures [14] also
have been presented for network representation analysis. However,
very little effort has been devoted to learning the node representa-
tion from imbalanced networks, which will result in imbalanced
node-context pairs. Our proposed ImVerde framework is designed
to address this issue through VDRW, a novel context sampling
method, as well as the balanced-batch sampling method.

3 VERTEX-DIMINISHED RANDOM WALK
In this section, we present the Vertex-Diminished Random Walk
(VDRW) for imbalanced network analysis. It resembles the ex-
isting Vertex Reinforced Random Walk (VRRW) [18] in terms of
the dynamic nature of the transition probabilities, as well as some
convergence properties. However, it is fundamentally different
from VRRW, as it encourages the random particle to walk within
the same class, which in turn will lead to separable node repre-
sentations in the embedding space between the majority and the
minority classes.

3.1 Notation
Suppose that a 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 \( R \) denote the non-negative \( n \times n \)
transition matrix, whose element in the \( j \)-th row and \( k \)-th column
\( \delta_{ij} = \sum_j w_{ij} \). Let \( Y_t \in \{1, \ldots, n\} \) be the state of the random
particle at step \( t \) of the random walk, and let \( S_i(t) \) denote the number
of visits to node \( v_i \) up to time \( t \). That is, \( S_i(t+1) = S_i(t) + \delta_{Y_t,i} \),
where \( \delta_{Y_t,i} = 1 \) if \( Y_t = i \), otherwise, \( \delta_{Y_t,i} = 0 \), and \( S_i(0) = 0 \)
for \( i = 1, \ldots, n \). We define a visiting function \( f(S_i(t)) \) as the change
magnitude of the transition probabilities with respect to visited
times \( S_i(t) \) of node \( v_i \). The current transition probability \( R_{i,j} \) can be
expressed as \( R_{i,j} = R_{i,j} + f(S_j(t)) \) at time \( t \). Therefore, after \( t \)
steps of random walk, \( f(S_i(t)) \) keeps track of how the transition
probabilities to visit \( v_j \) change. For example, it is obvious that
\( f(S_i(t)) = 1 \) in random walk because the transition probabilities
are constant all the time. In VDRW, the transition probability of
visiting node \( v_j \) from node \( v_i \) decreases each time when node \( v_j \) is
visited.

3.2 VDRW: A Novel Random Walk Model
As proven in DEEPWALK [19], a short random walk follows a
power-law distribution as word frequency in language modeling.
Planetoid [28] also utilizes short random walks as well as the label
information to extract node-context pairs as the input to model
training. By implicitly assuming the balanced setting in labeled
training examples, most nodes can find a path within the same class
using short random walk. Only the boundary nodes have constant
probabilities to visit the same class or the opposite class after a short
random walk. However, there are some limitations when applying
random walk for imbalanced network analysis. Most nodes in the
minority class are boundary nodes due to the small percentage
of this class in the entire network. In other words, random walk
starting from minority class nodes have a high probability of tran-
slating into the majority class. As a result, there is not much
difference between the nodes coming from the majority and minor-
ity 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 ran-
dom walk model named Vertex-Diminished Random Walk (VDRW)
that addresses the intuition that the transition probability to one node
decreases each time it is visited. It resembles the existing Vertex-
Reinforced Random Walk (VRRW) [18] in terms of the dynamic
nature of the transition probabilities. In VRRW, the transition
probabilities of visiting one node increase each time it is visited.
Therefore, nodes with high degrees are more likely to be visited,
and the number of visits to these nodes tends to be even more
skewed than the other nodes after a long walk. Therefore, VRRW
actually encourages the random particle to visit the nodes with
high degrees. For example, in Figure 2, there are two classes in the
imbalanced network where nodes \( v_1, v_2 \) and \( v_3 \) belong to the
majority class (red line means the walking path). The key challenge is
how to find the correct path for sampling node-context pairs within
the minority class using a random walk based method. Regular
random walks select one neighboring node at each time based on
a fixed transition matrix. Thus walking one step from \( v_3 \) to
\( v_3 \) in Figure 2(b), the random particle will visit node \( v_2 \) or \( v_4 \) with
equal probability. Note that from \( v_3 \), the random particle always
has the same probabilities to select the correct or incorrect path.
In VRRW [18], the transition probabilities to node \( v_2 \) increase each
time \( v_2 \) is visited. That is, \( v_4 \) with a high degree may have high
probabilities to be visited than \( v_2 \) after a short walk. So \( v_3 \) is more
likely to visit node \( v_4 \) as shown in Figure 2(c). Similar to VRRW,
the proposed VDRW also considers the probability change when
node \( v_2 \) is visited each time. To encourage nodes in minority class
to walk within the same class, the transition probabilities in VRRW
are inversely proportional to the visited times of the nodes. Since
the nodes from the majority class usually have a higher degree
than the nodes from the minority class in an imbalanced network,
the probabilities to visit those nodes in majority class after a short
walk are more likely to decrease. So in Figure 2(d), \( v_3 \) has a lower
probability to be visited than \( v_2 \) or \( v_1 \).

VDRW is a stochastic process in which the probability of visiting
one node is decreased each time it is visited. So VDRW is more
likely to explore the unvisited nodes compared to VRRW or regular
random walk. Based on this intuition, in VDRW, we define the
transition probability as follows:

\[
P(Y_{t+1} = j | Y_t) = \frac{R_{Y_t,j} f(S_j(t))}{\sum_k R_{Y_t,k} f(S_k(t))}
\]

where \( Y_t \) is the \( \sigma \)-field generated by \( Y_k \), \( 1 \leq k \leq t \).

The visiting function \( f(S_i(t)) \) determines how the transition
probability matrix changes. Obviously, there are two special cases:
Figure 2: Comparison among random walk, VRRW [18] and VDRW. (a) Imbalanced network with two classes (the green nodes are from the minority class); (b) fixed transition probability; (c) the transition to node \( v_k \) is reinforced each time it is visited; (d) in contrast, the transition probability from \( v_3 \) to \( v_4 \) decreases over time in VDRW.

\[
f(S_t(t)) = 1 \text{ for random walk, and } f(S_t(t)) = S_t(t) + 1 \text{ for VRRW.}
\]

As for VDRW, it is inversely proportional to the visited times \( S_t(t) \) at time \( t \). In this paper, we define the visiting function \( f(S_t(t)) \) as follows:

\[
f(S_t(t)) = \alpha S_t(t)
\]

where \( 0 < \alpha < 1 \) is the parameter. When one node is visited frequently, the probability to visit this node gradually decreases. That means it would visit other neighboring nodes with high probabilities next time. But normally it is difficult to learn the optimal parameter \( \alpha \) above. Therefore, here we also introduce an alternative definition for the visiting function \( f(S_t(t)) \) as follows, which does not require the parameter:

\[
f(S_t(t)) = f(S_t(t-1)) \cdot (1 - \frac{1}{\exp[S_t(t)]})
\]

The detailed algorithm for VDRW is presented in Algorithm 1. It is given the graph \( G \) with adjacent matrix \( R \), the parameter \( \alpha \) and the length of walking sequences as input, and outputs the walking path. In the first step, it adds initial node \( v \in V \) to path \( p \). And then it starts to walk during step 3-6 for \( T \) walking steps. For each step, it computes the transition probabilities for \( v \) at time \( t \) using Equation (1). Based on those probabilities, one node \( v_j \) is randomly selected to be visited by the random particle, and added to the path \( p \). After that, it updates the visiting function \( f(S_t(t)) \) using Equation (2) or (3), and starts to walk at time \( t + 1 \).

The key idea of this algorithm is to change the transition probabilities with respect to the number of times a node has been visited. It encourages the random particle to explore the unvisited or less-frequently visited nodes in the network. To extract node-context pairs in imbalanced networks especially for the minority class, VDRW is applied in our semi-supervised network representation learning framework, which will be introduced in Section 4. Assuming that there are very few links between the minority and majority classes in the network, minority nodes are likely to explore nodes within the same category once these links have been visited. Otherwise, minority nodes may be regarded as one part of majority class for node representation learning if the links between the minority and majority classes are very dense. Figure 3 shows the context distribution for \( v_1 \) with a short walk from node \( v_1 \). For random walk, minority class nodes have the fixed probability to select the path in the next step, so the walking path may cover several majority class nodes. Since the main idea in VRRW is to increase the probability of transitioning to \( v_j \) each time \( v_j \) is visited, nodes with a high degree will be visited more frequently. That is, \( v_1 \) may also visit nodes with the high degree in the majority class frequently. In contrast, VDDW encourages the random particle to walk within the same class by adjusting the transition probabilities each step. So we use VDRW to extract node-context pairs for learning network representation in this paper.

**Algorithm 1** Vertex-Diminished Random Walk

**Input:** Graph \( G \) with adjacency matrix \( R \), parameter \( \alpha \) if the visiting function is defined as in Equation (2) length of walking sequences \( T \)

**Output:** Walking path \( p \)

1. \textbf{Initialize:} path \( p \) with initial node \( v \in V \)
2. for \( t = 0 \) to \( T - 1 \) do
3. \hspace{1em} Compute \( P(Y_{t+1} = j | F_t) \), \( j = 1, \ldots, n \), using Equation (1)
4. \hspace{1em} Walk for one step based on the transition probabilities
5. \hspace{1em} Add the selected node \( v_j \in V \) to path \( p \)
6. \hspace{1em} Update the visiting function \( f(S_t(t)) \) using Equation (2) or Equation (3)

7. end for

Figure 3: The walking path distribution using random walk, VRRW and VDRW from \( v_1 \) in minority class

**3.3 Convergence Analysis**

In this subsection, we analyze the convergence properties of the function \( V(i) \in \mathbb{R}^n \), in which the \( i \)-th element is defined as \( V(i) = f(S_t(t))/\sum_i f(S_t(t)) \). It measures the model learning behavior in the sense that the transition probabilities do not change significantly when \( V(i) \) converges. Obviously, \( V(i) \) belongs to the \((n-1)\)-simplex \( \Delta \subseteq \mathbb{R}^n \). To discuss the convergence of VDRW, we first give several definitions [18]. For \( v \in \Delta \) with \( H(v) = \sum_{i,j} R_{ij} v_i v_j > 0 \), the Markov transition matrix \( M \) is defined as:

\[
M_{ij}(v) = \frac{R_{ij} v_j}{\sum_k R_{ik} v_k}
\]
And let $C \subseteq \Lambda$ be the set of points $v$ with $\pi(v) = v$ where the vector $\pi(v) \in \Lambda$ is defined as:

$$
\pi_i(v) = \frac{\sum_j R_{ij} v_{ij} v_j}{H(v)}
$$

Let $C_0 \subseteq \Lambda$ be the set of points $v$ for which $M(v)$ is reducible.

**Theorem 3.1.** [18] With probability one, $dist(V(t), C \cup C_0) \rightarrow 0$ where $dist(x, A) = \inf \{|x - y| : y \in A\}$.

Base on this theorem, we introduce the following lemma regarding the convergence conditions of VDRW. For any $t$, let $M_1(t), M_2(t+1), \ldots$ denote a Markov chain with fixed transition matrix $M(V(t))$ beginning at $Y_t$ at time $t$, $\pi(t) = [S(t), \cdots, S_n(t)]^T$ and $\pi(t) = S(i - 1) + c_{m(i)}$ for $i > t$ where $c_j$ denotes the $j$-th standard basis vector. Let $N$ be a closed subset of a simplex with $N \cap (C \cup C_0) = \emptyset$.

**Lemma 3.2.** If $\{f(\pi'(t))\}$ has Markov property, and $(f(\pi'(t + L)) - f(\pi'(t))/(\sum_i f(\pi'(t + L)) - \sum_i f(\pi'(t))))$ approaches a point-mass at $\pi(v)$ in distribution as $L$ increases, then there exists $c > 0$ such that $E[H(V(n + L))/V(n)] > H(V(n)) + c/n$ whenever $V(n) \notin N$.

The proof of this lemma is similar to Lemma 3.1 in [18]. But we define the visiting function $f(\pi'(t))$ in VDRW to give the basic conditions for this lemma. When $f(\pi'(t)) = 1 + \log \alpha \cdot S(t)$, it has the following conclusion.

$$
f(\pi'(t + L)) = f(\pi'(t))
$$

As shown in [18], $f(\pi'(t + L)) = f(\pi'(t))$ approaches a point-mass at $\pi(v)$ in distribution as $L$ increases. Therefore, $f(\pi'(t)) = 1 + \log \alpha \cdot S(t)$ meets the basic condition in Lemma 3.6 above. And since the function $f(\pi'(t))$ defined in Equation (2) has the following Taylor expansion:

$$
f(\pi'(t)) = \alpha^{S(t)} = 1 + \log \alpha \cdot S(t) + O(\log \alpha \cdot S(t))
$$

In VDRW, we use $f(\pi'(t)) = \alpha^{S(t)}$ to define the visiting function. And moreover, the following Corollary 3.7 presents the conditions under which the convergence of $V(t)$ is almost surely.

**Corollary 3.3.** [18] If all the off-diagonal entries of $R$ are positive and all the principal minors of $R$ are invertible, $V(t)$ converges almost surely.

We would like to point out that for real networks, the graph structure hardly meets the convergence conditions in Corollary 3.7. However, as we will show in the experimental results, VDRW is still an effective way to sample node-context pairs in learning network representation from imbalanced networks, especially in comparison with state-of-the-art techniques.

## 4 THE PROPOSED FRAMEWORK

In this section, we present the semi-supervised network representation framework based on Vertex-Diminished Random Walk. Similar to Planetoid [28], we extract node-context pairs using both the graph structure and the label information. Nevertheless, there are significant differences between our framework and Planetoid. First, we adopt the novel VDRW to keep the random walk within the same class, especially for the minority classes, whereas traditional random walk is used for context extraction in Planetoid. Secondly, Planetoid uses a preset parameter to control the ratio of graph-based and label-based contexts, but we use both graph and label to determine the context in walking path. Finally, to balance the node-context pairs from different classes, we sample balanced initial points in mini-batch training process, which is also not considered in Planetoid designed for balanced networks.

### 4.1 Semi-supervised Network Representation

Following the notation in [28], suppose that there are $L$ labeled nodes $(x_i, y_i)_{i=1}^L$ and $U$ unlabeled nodes $(x_i)_{i=L+1}^U$, where $x_i \in R^l$ is the $l$-dimensional feature vector of node $v_i$. And Each node $v_i$ is embedded into a $d$-dimensional vector $e_i$. The semi-supervised network representation framework is formulated as

$$
\mathcal{L} = \frac{1}{T} \sum_{i=1}^L \log p(y_i | x_i, e_i) + \lambda \sum_{i,c} \log p(c(i))
$$

where the first term is the prediction loss using the attributes of labeled nodes; the second term denotes the unsupervised loss based on the graph structure; and $\lambda$ is a constant balancing these two terms.

We adopt the feed-forward neural network with $l_1$ hidden layers based on feature $x_i$ and $l_2$ hidden layers based on embedding vector $e_i$ to predict the label. The supervised loss can be expressed as:

$$
p(y_i | x_i, e_i) = \exp(h_i^T(x_i)T, h_i^T(e_i)T | v_y)
$$

where $[\cdot, \cdot]$ concatenates two row vectors and $v_y \in R^C$ denotes the vectorized expression of the label $y$ ($C$ is the number of classes in the network). By assuming that nodes in similar contexts have similar embeddings, the unsupervised loss is defined with the node-context pairs $(i, c)$ as:

$$
p(c(i)) = \sigma(w_c^T e_i) + k \cdot \mathbb{E}_{y(i,c)} \log (-\sigma(w_c^T e_i))
$$

where $c$ is the randomly sampled negative context for node $v_i$ and $k$ is the number of negative samples $c$. $\sigma(x)$ is defined as the sigmoid function $\sigma(x) = 1/(1 + e^{-x})$.

The training process in our network representation learning framework is illustrated in Algorithm 2. It is given the graph $G$ with adjacent matrix $R$, the node features $x_{1:L+U}$ with labels $y_{1:L}$, and the training model parameters (batch iterations $T_1$ and $T_2$, the negative sampling size $k$, embedding size $d$) as input, and outputs the embedding vector $e_i$ for each node $v_i$. Lines 1-8 demonstrate the training process in learning the embedding vector for each node in the imbalanced network. Mini-batch training method is used in this model, and we randomly sample a batch of nodes $S$ from all the training data as the initial points in context sampling. Then node-context pairs are extracted in the walking paths learned by
both VDRW and label information. Lines 9-12 correspond to the process of predicting the label using node feature $x_i$ and embedding vector $e_i$. Finally, stochastic gradient descent (SGD) is adopted to train our model.

Algorithm 2 Model Training

Input: Graph $G$ with adjacent matrix $R$, $x_i \in L + U$, $y_i \in L$
batch iterations $T1$ and $T2$
negative sampling size $k$
embedding size $d$

Output: Embedding vector $\{e_i\}_{i=1}^{L+U}$

1: for $t \leftarrow 1$ to $T1$ do
2: Sample a batch of balanced initial points $S$
3: for $s \leftarrow 1$ to $|S|$ do
4: Sample positive pairs $(i, c)$ using VDRW
5: Sample $k$ negative pairs for each $(i, c)$
6: end for
7: Update $e_i$ and $w_c$ using SGD
8: end for
9: for $t \leftarrow 1$ to $T2$ do
10: Sample a batch of $(x_i, y_i)$ from labeled instances
11: Update $W^k$ and $b^k$ using SGD
12: end for

Notice that there are two crucial steps in our framework: context sampling and balanced-batch sampling. The goal of context sampling is to select correct node-context pairs using both the graph structure and label information, and balanced-batch sampling is to balance the number of pairs from different classes for imbalanced networks.

4.2 Context Sampling

In this subsection, we present the context sampling method in our framework using both label and graph structure. In context sampling, DEEPWALK [19] has proven the effectiveness of random walk in exploring the context for each node in the network. And Planetoid [28] also adopts random walk to detect graph context for learning graph structure.

The core idea in random walk is to choose one neighboring node based on the transition probabilities. However, it does not work well for context sampling in imbalanced networks. For the minority class, small sampled incorrect contexts will greatly affect the performance of embeddings. To address the problem of context sampling in imbalanced networks, we propose a novel sampling method based on VDRW as VDRW encourages the random particle to walk within the same class by adjusting the transition probabilities each step.

The context sampling method is presented using both graph structure and label information as illustrated in Algorithm 3. If the current node is labeled, it will visit another node with the same label with probability $r$. And with probability $1 - r$, it selects the walking path based on the transition matrix. After each step, the transition matrix is updated as presented in Algorithm 1. Finally, the node-context pairs can be extracted from the walking path.

Algorithm 3 Context Sampling

Input: Graph $G$ with adjacent matrix $R$
initial node $v_s$
jumping probability $r$

Output: Node-context pairs $(i, c)$

1: Initialize: Add $v_s$ to walking path $p$
2: while $t > 0$ do
3: if Node is labeled then
4: With $r$, select node $v_i$ with the same label
5: With $1 - r$, select node $v_j$ using VDRW
6: else
7: Select node $v_j$ using VDRW
8: end if
9: Add node $v_i$ to path $p$
10: Update the transition probabilities in graph
11: end while
12: Sample the node-context pairs from path $p$

Figure 4: Mini-batch sampling where ‘l’ represents labeled node and ‘u’ is unlabeled. If one labeled green node from the minority class has been selected, one labeled blue node from the majority class will also be randomly selected, after which other unlabeled nodes will be randomly selected. Based on these initial nodes, node-context pairs are extracted using Algorithm 3.

4.3 Balanced-batch Sampling

After node-context pairs are extracted correctly from imbalanced networks, the number of pairs in different classes are typically imbalanced for model training. For example, if we adopt the mini-batch sampling strategy as in [28], every node has the same chance to be used as an initial point when sampling node-context pairs. In other words, most of the node-context pairs are sampled starting from the majority class. To avoid this problem, we adopt a simple under-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 training model: labeled majority $S_{maj}$, labeled minority $S_{min}$ and unlabeled examples $S_u$. Balanced-batch sampling randomly selects a subset $S'_{maj}$ from labeled majority class so that $|S'_{maj}| = |S_{min}|$, and then randomly selects a subset $S'_{u}$ from the whole unlabeled data. Consequently, the sampled nodes given by mini-batch sampling are $S = S_{min} \cup S'_{maj} \cup S'_{u}$, which are used as initial points in subsequent context sampling. The sampling process is illustrated in Fig. 4. All the node-context pairs are sampled based on the balanced initial points selected using Balanced-batch sampling method.
In general, random under-sampling may cause the training model to miss important details related to the majority class [11]. Nevertheless, this typically will not happen in our mini-batch sampling because the iterative mini-batch training actually considers all the training examples. Note that in mini-batch model training, the minority class examples are adopted at each iteration by combining with randomly selected equal number of majority class examples.

5 EXPERIMENTS

In this section, we present the experimental results comparing the proposed ImVerde with state-of-the-art network embedding and imbalanced classification methods. Among the network embedding methods used for comparison, DEEPWALK [19] and node2vec [7] are random walk based network representation learning methods without any label information. Planetoid [28] extends these methods to the semi-supervised setting using both the graph structure and label information. In our experiments, we consider the transductive model, i.e., Planetoid-T, as the baseline to learn node representations for imbalanced networks. We also compared with GraRep [2], which captures k-step relational information from the graph structure. For imbalanced classification, ADASYN [10] creates synthetic samples as additional examples from the minority class. Here we consider three versions of random walks mentioned above for our framework where ImVerde-a and ImVerde-e use the VDRW using visiting function (2) and (3), respectively, and ImVerde-r adopts VRRW for our framework. By comparing their performance in node classification, we validate the effectiveness of ImVerde for imbalanced network analysis. The model parameters in our experiments are set to $\alpha = 0.7$, batch iterations $T_1 = T_2 = 200$, $r = 0.1$ and $d = 50$.

The data sets used for our experiments are summarized in Table 1. Cora, Citeseer and Pubmed are citation network data sets with bag-of-words representation and citation links for each data set. These links are treated as undirected and unweighted edges in these networks. Following the data setting in Planetoid\footnote{https://github.com/kimiyoung/planetoid}, there are 20 labeled training samples from each class and 1000 test samples in every data set. Since they are balanced networks with equal labeled training samples for each class, we split these labeled examples into two parts with imbalanced size. That is, we choose one class as an individual category, and all of the other classes as the other category to construct the imbalanced binary networks. For NTSB data set, we downloaded 2409 flight accident reports from National Transportation Safety Board\footnote{https://www.ntsb.gov/Pages/default.aspx} to analyze and identify the leading indicators of different accident causes. Each report can be represented using a 1000-dimensional TF-IDF feature vector. These reports are modeled as an imbalanced network in which each node corresponds to a historical accident report and each edge reflects the cosine similarity between TF-IDF features in two reports.

For imbalanced network representation learning, the main goal is to learn the low-dimensional embedding vectors for each node in such a way that the majority and minority classes are well separated. For the test set, we evaluate the classification precision and recall with respect to the minority class by sorting the prediction probabilities belonging to this class, as the minority class examples are often of more interest as compared to majority class examples.

| Dataset | Nodes | Edges | Classes | Features |
|---------|-------|-------|---------|----------|
| Cora    | 2,708 | 5,429 | 7       | 1,433    |
| Citeseer| 3,327 | 4,732 | 6       | 3,703    |
| Pubmed  | 19,717| 44,338| 3       | 500      |
| NTSB    | 2,409 | 4,793 | 2       | 1,000    |

5.1 Embedding Visualization

We use t-SNE [16] to visualize the embedding vectors in 2-D spaces on the Cora dataset. The results are shown in Figure 5. Since there are seven classes on Cora dataset with equal training examples from each class, we choose the 7th class as the minority class, and all the other classes as the majority to construct an imbalanced network, although similar results have been observed for other settings. We adopt DEEPWALK, GraRep, Planetoid and the proposed ImVerde to learn the node representations. The embedding vectors of test examples are visualized in Figure 5. It can be seen that compared with DEEPWALK, GraRep and Planetoid, ImVerde has the best performance in terms of separating the minority class (black dots) from the majority class (red ones).

DEEPWALK and Planetoid use short random walk to extract node-context pairs for their network learning models, so their node-context pairs focused on the majority class due to the imbalanced property of the nodes in the network. Planetoid adopts the label information to create node-context pairs in the semi-supervised learning setting with implicitly assuming that the training data is balanced. This might be the reason why planetoid shows relatively better separability between the two classes in the embedding space than DEEPWALK as shown in Figure 5(a) and (c). Instead of adopting random walk, GraRep defines k-step neighborhoods to capture the structure information so that the minority class nodes are still clustered in the embedding space. But GraRep might capture the neighboring nodes within the opposite class when extracting high-order proximities. However, in the proposed ImVerde-e, starting from a minority class node, VDRW in context sampling encourages the random particle to walk within the minority class, with a lower probability to walk to the opposite majority class. Balanced-batch sampling in ImVerde-e balances the extracted node-contexts pairs in different classes on training model. Therefore, most nodes in the minority class can be separated from the majority class as shown in Figure 5(d).

5.2 Imbalance Analysis

Most of the existing network representation learning algorithms have been successfully applied to balanced networks. The imbalance ratio in the networks significantly affects the embedding performance of these algorithms. Table 2 lists the node classification recall@k on the Pubmed data set where $k$ is set to the number of test instances in minority class. There are three classes in this dataset with 20 training examples from each class, and 1000 test examples in total. In this experiment, we choose the 1st class as the minority and other two classes as the majority class (40 labeled training examples). By randomly selecting several samples (2, 8, 14 or 20) from the 1st class as the minority training data, the imbalance ratio changes.
It can be seen from Table 2 that ImVerde-e and ImVerde-a outperform other methods in terms of both classification recall and robustness to the imbalance ratio. When taking only 2 training examples from the minority class, it is an extremely imbalanced network. The recall value of Planetoid is significantly decreased compared to that in the relatively balanced network with 20 labeled examples from the minority class. Both node2vec [7] and DEEPWALK do not perform well on Pubmed data set, especially when training samples are imbalanced. Without exploiting any label information, they use random walk to explore the neighborhood and thus result in less discrimination for the minority class in the embedding space. Typically, the minority class tend to be misclassified as the majority class because the context differences between them using random walk are decreased. Node2vec has the similar intuition about random walk that the transition probability matrix is not constant. The likelihood of immediately revisiting a node decreases because node2vec encourages breadth and deep search in random work. And node2vec has more robust performance than DEEPWALK and Planetoid for learning imbalanced network representation. But there are still significant differences between ImVerde and node2vec. When node $v_j$ is visited from $v_i$, node2vec considers adjusting the probability of links $v_j$ will visit next time. In contrast, the probability of a transition to $v_j$ decreases in ImVerde. Besides, the balanced-batch sampling in ImVerde keeps the node-context pairs balanced, which is beneficial for node discrimination when the network is extremely imbalanced. The iterative training process reduces the information loss of the majority class in under-sampling.

| Methods     | # nodes in minority class |
|-------------|---------------------------|
|             | 2  | 8  | 14 | 20 |
| DEEPWALK    | 0.327 | 0.444 | 0.45 | 0.455 |
| Node2vec    | 0.528 | 0.617 | 0.589 | 0.622 |
| GraRep      | 0.522 | 0.605 | 0.594 | 0.628 |
| Planetoid   | 0.372 | 0.528 | 0.661 | 0.733 |
| imVerde-t   | 0.650 | 0.672 | 0.728 | 0.761 |
| imVerde-e   | 0.611 | 0.694 | 0.711 | 0.756 |
| imVerde-a   | 0.650 | 0.711 | 0.744 | 0.767 |

### 5.3 Text Classification

We use Cora and Citeseer data sets to test the performance of the proposed framework in text classification. There are 20 training examples from each class in these data sets. Both the precision and recall curves are shown in Figure 6. For Cora data set, the 2nd and 7th classes are selected as the minority class, respectively, and other classes as the majority class. ImVerde-e and ImVerde-a outperform other network representation and imbalanced classification algorithms. ADASYN4 samples the synthetic data for the minority class according to their difficulty in learning. We use node features to adaptively create synthetic examples based on ADASYN for the minority class in Cora data set. And then logistic regression is adopted to train on these balanced data. But it is shown in Figure 6 that ADASYN does not perform well on this data set because it does not utilize any graph structure or other properties of the nodes. ImVerde-t adopts VRRW to extract node-context pairs in the imbalanced networks. But as mentioned before, VRRW would constrain nodes to walk around special nodes with a high degree. That may cause the problem that the random particle walks around these special nodes regardless of their categories. As a result, incorrect node-context pairs definitely affects the separability of embedding

4https://github.com/scikit-learn-contrib/imbalanced-learn
vectors in a negative way. Figure 7 shows the classification results on Citeseer data set. These results further validate our analysis.

![Figure 7: Precision and recall for text classification on Citeseer data set](image)

We also use NTSB accidental reports to construct a binary imbalanced network for accident cause analysis. Here we use cosine similarity of their TF-IDF features to determine the edges for each node. In this experiment, we only consider node embedding vectors for classification without node features. Table 3 lists the classification results on NTSB data set. It shows that ImVerde-a outperforms all the other methods in terms of recall@k where k is set to the number of test instances in minority class.

| Methods     | Recall@k |
|-------------|----------|
| DEEPWALK    | 0.500    |
| Node2vec    | 0.467    |
| GraRep      | 0.516    |
| Planetoid   | 0.472    |
| ImVerde-t   | 0.522    |
| ImVerde-e   | 0.500    |
| ImVerde-a   | 0.538    |

6 CONCLUSIONS

In this paper, we present a novel Vertex-Diminished Random Walk for imbalanced network analysis, which encourages the random particle to walk within the same class by adjusting the transition probability each step. Compared with regular random walk, VDRW reduces the probability of the transition probability to one node each time it is visited. Then based on VDRW, we propose ImVerde, a semi-supervised network representation learning framework for imbalanced networks. In this framework, we use VDRW and a small number of node labels to capture node-context pairs by assuming nodes with the same context or label have the similar embeddings. Besides, we adopt a simple under-sampling method to balance these pairs from different classes. We compare ImVerde with other state-of-the-art algorithms on three social networks and one flight accidental report network. The experimental results demonstrate the effectiveness of the proposed method, especially when the data set is extremely imbalanced.

REFERENCES

[1] Mikhail Belkin and Partha Niyogi. 2003. Laplacian eigenmaps for dimensionality reduction and data representation. Neural computation 15, 6 (2003), 1373–1396.
[2] Shaosheng Cao, Wei Lu, and Qiongkai Xu. 2015. Gzrep: Learning graph representations with global structural information. Proceedings of the 24th ACM International Conference on Information and Knowledge Management (2015), 891–900.
[3] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. 2002. SMOTE: synthetic minority over-sampling technique. Journal of artificial intelligence research 16 (2002), 321–357.
[4] Claire Donnat, Marinka Zitnik, David Hallac, and Jure Leskovec. 2017. Spectral Graph Wavelets for Structural Role Similarity in Networks. arXiv preprint arXiv:1710.10321 (2017).
[5] Charles Elkan. 2001. The foundations of cost-sensitive learning. International Joint Conference on Artificial Intelligence 17, 1 (2001), 973–978.
[6] Palash Goyal and Emilio Ferrara. 2017. Graph embedding techniques, applications, and performance: A survey. arXiv preprint arXiv:1705.02801 (2017).
[7] Aditya Grover and Jure Leskovec. 2016. Node2vec: Scalable feature learning for networks. Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining (2016), 855–864.
[8] Guo Haixiang, Li Yijing, Jennifer Shang, Gu Mingyun, Huang Yuanyue, and Gong Bing. 2017. Learning from class-imbalanced data: Review of methods and applications. Expert Systems with Applications 73 (2017), 226–239.
[9] William L Hamilton, Rex Ying, and Jure Leskovec. 2017. Representation Learning on Graphs: Methods and Applications. arXiv preprint arXiv:1709.05584 (2017).
[10] Haibo He, Yang Bai, Edward A Garcia, and Shituo Li. 2008. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. Neural Networks, 2008. IJCNN 2008. (IEEE World Congress on Computational Intelligence). IEEE International Joint Conference on (2008), 1322–1328.
[11] Haibo He and Edward A Garcia. 2009. Learning from imbalanced data. IEEE Transactions on knowledge and data engineering 21, 9 (2009), 1263–1284.
[12] Jingrui He, Yan Liu, and Richard Lawrence. 2008. Graph-based rare category discovery and data mining. Data Mining, 2008. ICDM’08. Eighth IEEE International Conference on (2008), 833–838.
[13] Chen Huang, Yining Li, Chen Change Loy, and Xiaoou Tang. 2016. Learning deep representation for imbalanced classification. (2016), 5375–5384.
[14] Thomas N Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. ICLR (2017).
[15] Tianshu Lyu, Yuan Zhang, and Yan Zhang. 2017. Enhancing the Network Embedding Quality with Structural Similarity. Proceedings of the 2017 ACM on Conference on Information and Knowledge Management (2017), 147–156.
[16] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-SNE. Journal of machine learning research 9, Nov (2008), 2579–2605.
[17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. Advances in neural information processing systems (2013), 3111–3119.
[18] Robin Pemantle. 1992. Vertex-reinforced random walk. Probability Theory and Related Fields 92, 1 (1992), 117–136.
[19] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. 2014. Deepwalk: Online learning of social representations. Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining (2014), 701–710.
[20] Leonardo FR Ribeiro, Pedro HP Saverese, and Daniel R Figueiredo. 2017. Struc2vec: Learning node representations from structural identity. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2017), 385–394.
[21] Yannmin Sun, Andrew KC Wong, and Mohamed S Kamel. 2009. Classification of imbalanced data: A review. International Journal of Pattern Recognition and Artificial Intelligence 23, 04 (2009), 687–719.
[22] Jian Tang, Meng Qu, and Qiaozhu Mei. 2015. PTE: Predictive text embedding through large-scale heterogeneous text networks. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2015), 1165–1174.
[23] Jian Tang, Meng Qu, Minghe Wang, Ming Zhang, Jun Yan, and Qiaozhu Mei. 2015. Line: Large-scale information network embedding. Proceedings of the 24th International Conference on World Wide Web (2015), 1067–1077.
[24] Joshua B Tenenbaum, Vin De Silva, and John C Langford. 2000. A global geometric framework for nonlinear dimensionality reduction. science 290, 5500 (2000), 2319–2323.
[25] Daxin Wang, Peng Cui, and Wenwu Zhu. 2016. Structural deep network embedding. Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining (2016), 1225–1234.
[26] Xiao Wang, Peng Cui, Jing Wang, Jian Pei, Wenwu Zhu, and Shiqiang Yang. 2017. Community Preserving Network Embedding. AAAI (2017), 203–209.
[27] Cheng Yang, Zhiyuan Liu, Deli Zhao, Maosong Sun, and Edward Y Chang. 2015. Network Representation Learning with Rich Text Information. IJCAI (2015), 2111–2117.
[28] Zhilin Yang, William W Cohen, and Ruslan Salakhutdinov. 2016. Revisiting semi-supervised learning with graph embeddings. ICML (2016).

[29] Dawei Zhou, Jingrui He, K Seluk Candan, and Hasan Davulcu. 2015. MUVIR: Multi-View Rare Category Detection. IJCAI (2015), 4098–4104.