Large Scale Graph Construction and Label Propagation

Z. Ibrahim[1], A. Bosaghzadeh[3][a] and F. Dornaika[1,2,*][b]
[1]University of the Basque Country UPV/EHU, San Sebastian, Spain
[2]IKERBASQUE, Basque Foundation for Science, Bilbao, Spain
[3]Shahid Rajaee Teacher Training University, Tehran, Iran
fi

Keywords: Scalable Graph Construction, Semi-Supervised Learning, Topology Imbalance, Large Scale Databases, Reduced Flexible Manifold Embedding.

Abstract: Despite the advances in semi-supervised learning methods, these algorithms face three limitations. The first is the assumption of pre-constructed graphs and the second is their inability to process large databases. The third limitation is that these methods ignore the topological imbalance of the data in a graph. In this paper, we address these limitations and propose a new approach called Weighted Simultaneous Graph Construction and Reduced Flexible Manifold Embedding (W-SGRFME). To overcome the first limitation, we construct the affinity graph using an automatic algorithm within the learning process. The second limitation concerns the ability of the model to handle a large number of unlabeled samples. To this end, the anchors are included in the algorithm as data representatives, and an inductive algorithm is used to estimate the labeling of a large number of unseen samples. To address the topological imbalance of the data samples, we introduced the Renode method to assign weights to the labeled samples. We evaluate the effectiveness of the proposed method through experimental results on two large datasets commonly used in semi-supervised learning: Covtype and MNIST. The results demonstrate the superiority of the W-SGRFME method over two recently proposed models and emphasize its effectiveness in dealing with large datasets.

1 INTRODUCTION

In recent years, various machine learning systems have integrated supervised learning methods, demonstrating impressive outcomes across diverse tasks and domains. However, the reliance of these methods on substantial amounts of labeled data introduces significant human involvement in the modeling process and potentially high costs for data annotation. Addressing these challenges, graph-based semi-supervised learning (GSSL) serves as a theoretical framework, capitalizing on insights derived from unlabeled data. This approach involves a dataset and a graph illustrating connections between labeled and unlabeled elements. GSSL operates under two key assumptions: the clustering assumption, which pertains to the data's nature, and the manifold assumption, which relates to its spatial distribution (Belkin et al., 2006). The majority of graph-based semi-supervised learning (GSSL) approaches rely on a pre-existing graph, treating graph construction and label propagation as distinct tasks (Qiu et al., 2019; Sindhwani and Niyogi, 2005; Song et al., 2022). For instance, in the work of (Bosaghzadeh et al., 2013), an adaptive KNN algorithm is employed to establish the graph, while in (Wang et al., 2010), the affinity matrix is formed based on a data representation algorithm. However, in more recent methodologies, these two tasks are integrated to simultaneously create the graph and predict labels (Tu et al., 2022; Wang et al., 2022; Wu et al., 2019).

An additional significant concern pertains to the issue of imbalanced data. While previous studies have primarily addressed imbalances arising from unevenly distributed labeled examples across classes (set imbalance) (Chen et al., 2021), we posit that graph data introduce a unique form of imbalance due to the asymmetric topological characteristics of labeled nodes. Specifically, labeled nodes differ in their structural roles within the graph (topology imbalance) (Chen et al., 2021). This phenomenon has been explored within the realm of data analysis, particularly in the field of topological data analysis (TDA). The

[a] https://orcid.org/0000-0002-0372-6144
[b] https://orcid.org/0000-0001-6581-9680
*Corresponding author

Ibrahim, Z., Bosaghzadeh, A. and Dornaika, F.
Large Scale Graph Construction and Label Propagation.
DOI: 10.5220/0012430100003660
Paper published under CC license (CC BY-NC-ND 4.0)
In *Proceedings of the 19th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2024) - Volume 2: VISAPP*, pages 702-709
ISBN: 978-989-758-679-8; ISSN: 2184-4321
Proceedings Copyright © 2024 by SCITEPRESS – Science and Technology Publications, Lda.

mapper algorithm stands out as a prominent approach in this domain, and various algorithms, such as the fuzzy mapper algorithm (Bui et al., 2020) and Shape Fuzzy C-Means (SFCM) (Bui et al., 2021), have been proposed to address this aspect of topology imbalance.

In tasks involving label propagation, labeled samples situated near the decision boundaries between different classes are more prone to generating conflicts in information. Conversely, labeled samples positioned farther away from these boundaries do not encounter such conflict issues (Chen et al., 2021; Chen et al., 2019).

A significant limitation of Graph-Based Semi-Supervised Learning (GSSL) is the scalability issue (Collobert et al., 2006; Zhu and Lafferty, 2005). Despite notable advancements in semi-supervised methods, particularly for smaller datasets, many of these approaches struggle to scale effectively when confronted with large, unlabeled datasets common in practical applications (Sindhwani et al., 2005; Wang et al., 2019). The challenges in scalability primarily manifest in the graph generation and label evaluation phases of graph-based SSL solutions (Long et al., 2019; Qiu et al., 2019; Song et al., 2022)

Another challenge with Semi-Supervised Learning (SSL) methods lies in predicting the labels of test samples. Transductive methods require the repetition of the whole procedure, including graph construction and label estimation, to predict labels for unseen test samples. Conversely, inductive approaches define a projection that maps test samples from the feature space to the label space, enabling reliable label estimation for test samples (Qiu et al., 2019; Sindhwani et al., 2005).

This article introduces the W-SGRFME model, an inductive semi-supervised framework that addresses challenges associated with large datasets through the incorporation of anchor points. Additionally, it tackles topological imbalance in the data by assigning weights to labeled nodes. The model can simultaneously predict the projection matrix, anchor affinity matrix, and labels for unlabeled data. Furthermore, it offers a method for estimating the labels of test samples through a linear transformation. The contributions of this work include:

- Expanding the idea of graph topology imbalance to large data sets.
- Incorporating weights of labeled samples into the unified scalable semi-supervised model.
- Showing the effectiveness of the proposed method through experimental results on two large datasets in the context of semi-supervised learning.

The subsequent sections of the paper are structured as follows: Section 2 provides an overview of Graph-Based Semi-Supervised Learning (GSSL) approaches. Section 3 explains some fundamental concepts along with the proposed algorithm. The experimental results of the method are outlined in Section 4, and the paper concludes with Section 5.

2 RELATED WORK

The use of prefabricated graphs poses a significant challenge in GSSL algorithms, as highlighted by (Cui et al., 2018; Dornaika et al., 2021; Hamilton et al., 2017; Nie et al., 2010). Prefabricated graphs, especially for large datasets, can be impractical, containing inappropriate connections. This issue becomes pronounced with very large datasets due to computational impracticality and memory space concerns, given the quadratic scaling of the graph matrix with the number of nodes.

Recent research emphasizes the interconnected nature of graph building and learning tasks, advocating for their simultaneous consideration (Kang et al., 2021; Nie et al., 2017; Yuan et al., 2021).

Weighting labeled samples has shown promise in improving classifcation accuracy (Aromal et al., 2021; Chen et al., 2021), suggesting that assigning lower weights to samples near class boundaries is beneficial. Combining the node effect shift phenomenon with label propagation, (Chen et al., 2021) presents a unified approach to analyze quantitative and topological imbalance problems. The ReNode method (Chen et al., 2021) flexibly reweights the effects of labeled nodes based on their positions relative to class boundaries, providing a model-neutral solution.

This paper presents an algorithm that combines label transfer and graph generation into a unified operation. The incorporation of labels throughout the process of graph generation contributes to a more comprehensive evaluation of data diversity. The proposed approach is inductive, capable of handling large amounts of previously unseen data, and scalable, managing extensive training databases using anchors. Furthermore, assigning different weights to labeled nodes using the ReNode algorithm enhances the robustness of the proposed model.

3 PROPOSED METHOD

In any semi-supervised classification problem, we have N data samples in a data matrix $\mathbf{X} = \{\mathbf{X}_l, \mathbf{X}_u\} =$

$\{\mathbf{x}_1,...,\mathbf{x}_l,\mathbf{x}_{l+1}...,\mathbf{x}_{l+u}\} \in \mathbb{R}^{d\times N}$, where l, u, $N = l + u$, and d are the numbers of labeled and unlabeled and total number of training samples, and the dimensionality of each sample, respectively. For each labeled sample $\mathbf{x}_i$, there exist a label vector as $\mathbf{y}_i$ where $y_{ij} = 1$ if sample $\mathbf{x}_i$ corresponds to the j^{th} class and 0 otherwise. Consequently, we have a label matrix for the training data as $\mathbf{Y} = [\mathbf{Y}_l,\mathbf{Y}_u] = [\mathbf{y}_1;\mathbf{y}_2;....;\mathbf{y}_l;\mathbf{y}_{l+1};...;\mathbf{y}_{l+u}] \in \mathbb{R}^{N\times C}$, where C is the number of classes. Moreover, there is the soft-label matrix $\mathbf{F} = [\mathbf{f}_1;\mathbf{f}_2;....;\mathbf{f}_N] \in \mathbb{R}^{N\times C}$, where F_{ij} shows the probability that the sample $\mathbf{x}_i$ is a member of the j^{th} class.

Also, we have a similarity graph as $\mathbf{W} \in \mathbb{R}^{N\times N}$ where W_{ij} shows the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. Furthermore, the Laplacian matrix is defined as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, $\mathbf{D}$ being the degree matrix of the graph.

Moreover, we have the matrix of anchors as $\mathbf{Z} = [\mathbf{z}_1,\mathbf{z}_2,...,\mathbf{z}_m] \in \mathbb{R}^{d\times m}$ where m is the number of anchors and $m << N$. The affinity matrix $\mathbf{B} \in \mathbb{R}^{N\times m}$ represents the similarity between the anchors and the training samples. It is worth noting that, using the label of anchors (i.e., $\mathbf{A} \in \mathbb{R}^{m\times C}$), one can predict the label of the training data using Eq. (1).

$$\mathbf{F} = \mathbf{B}\,\mathbf{A}. \tag{1}$$

3.1 Brief Description of SGRFME

The reduced FME (r-FME) (Qiu et al., 2019) technique was proposed to solve the limitations of the former FME algorithm which was the problem of handling large databases. r-FME algorithm solves this problem by using anchors that can serve as representative samples for a collection of nodes (Qiu et al., 2019).

However, r-FME method treated the graph construction and label propagation as two separate tasks. Hence, in (Ibrahim et al., 2023), we proposed the Simultaneous graph construction and Reduced RFME method that jointly estimates the r-FME unknowns and the anchor-to-anchor graph similarity matrix. In other words, the SGRFME method simultaneously estimates the anchor-to-anchor graph matrix and the r-FME model variables (i.e., Soft label matrix, projection matrix, and bias vector). Hence, the anchor-to-anchor graph is not fixed a priori as in r-FME (Qiu et al., 2019). Moreover, it uses both the feature of anchors and the online predicted labels of unlabeled samples.

One of the main issues in the SGRFME method is that it treats the whole samples equally and does not consider the topology importance of the nodes. The main obstacle in solving the topology imbalance problem is how to evaluate the relative topological position of the labeled node to its class.

3.2 Node Weighting in Large Scale Databases

Renode algorithm (Chen et al., 2021) was proposed to use the node topology information and calculate the weights of the available labeled samples. The calculated weights are called Topology Relative Location measure (Totoro). This algorithm provides high weights of any labeled sample using the available labels and the graph topology seen by that labeled sample. These weights are used in the loss function of some deep semi-supervised classifiers.

However, the problem is that the Renode algorithm requires an $N \times N$ affinity graph which is not feasible to be constructed for large scale databases due to memory limitations. Hence, our solution is to adapt it for large scale databases using anchor nodes. Instead of using the whole training database to calculate the weights of the labeled nodes, we use the anchors as data representatives for unlabeled data (which builds a large portion of training data).

First, we select m anchors from the unlabeled data (i.e., $\mathbf{Z} \in \mathbb{R}^{d\times m}$). We put these anchors which are unlabeled data representatives along labeled data and construct a new data matrix as $\mathbf{X}_T \in \mathbb{R}^{d\times(l+m)}$ such that $\mathbf{X}_T = [\mathbf{X}_l,\ \ \mathbf{Z}]$. Then, we build the affinity matrix $\mathbf{O} \in \mathbb{R}(l+m)\times(l+m)$ which shows the similarity between the $l+m$ samples, where $l+m << N$. It is worth mentioning that the $\mathbf{O}$ affinity matrix can be efficiently computed using any graph construction method. In this paper, we use the well-known KNN method with K =10 to find the neighbors of a node, and the similarity between the nodes is calculated using the Gaussian function.

We then feed this affinity matrix in the Renode algorithm and calculate the weights for the labeled samples (i.e., $w_k, k = 1,...,l$). The calculated weights show the topological location and importance of the labeled samples.

3.3 Proposed Weighted SGRFME

As we explained before, the SGRFME algorithm (Ibrahim et al., 2023) has two drawbacks: First, it considers the anchors equally and second, it does not weight the labeled samples.

The objective function of the SGRFME is

$$\min_{\mathbf{A},\mathbf{Q},\mathbf{b},\mathbf{S}} \quad Tr(\mathbf{A}^T\mathbf{L}\mathbf{A}) + \lambda Tr(\mathbf{Z}\mathbf{L}\mathbf{Z}^T) + Tr((\mathbf{B}\mathbf{A}-\mathbf{Y})^T\mathbf{U}(\mathbf{B}\mathbf{A}-\mathbf{Y})) + \frac{\rho}{2}||\mathbf{S}||_2^2 + \mu(||\mathbf{Q}||^2 + \gamma||\mathbf{Z}^T\mathbf{Q} + \mathbf{1}\mathbf{b}^T - \mathbf{A}||^2) \tag{2}$$

where $\mathbf{S}$ is the anchor-to-anchor matrix of the graph and $\mathbf{L}$ is the Laplacian matrix of this graph. The first term is the smoothness of the anchors' labels, the second term measures the smoothness of the anchors' features, the third term is the error of the weighted labeling estimate over the labeled samples, the fourth term is a ℓ_2 regularization of the graph $\mathbf{S}$, and the fifth term regularizes the projection matrix and the estimated fitting error of the anchors over the projection matrix (regression error). The λ, ρ, μ, and γ are balanced parameters.

Our solution to insert the calculated weights (i.e., w_k) into the SGRFME objective function is to use the third term in Eq.4 which is the labeling error calculated over the labeled samples and propose a weighted label fitting term.

In other words, we extend our previous work (Ibrahim et al., 2023) and present a weighted simultaneous graph construction and a reduced flexible branching model that can adapt appropriate weights to nodes and also manage large datasets using anchor points. This method dynamically calculates the weights of labeled nodes using the ReNode algorithm and then uses those weights to improve the model.

Without loss of generality, we assume that the first l rows of the $\mathbf{B}$ matrix contain the labeled samples. Hence, the first l elements in the $\mathbf{U}$ matrix contain a fixed value and the rest are zero.

We define a new diagonal matrix $\mathbf{V}$ given by Eq. (3), which indicates the importance of the labeled samples.

$$\mathbf{V} = \begin{pmatrix} w_1 & \cdots & & & & \\ \vdots & \ddots & & & & \vdots \\ & & w_l & & & \\ & & & \ddots & & \\ & & & & 0 & \vdots \\ & & & & \cdots & 0 \end{pmatrix} \tag{3}$$

Thus, in any minimization problem that aims to recover the unknowns of the model, the important or relevant nodes that have large weights will receive more importance compared to the labeled nodes with low weights. Hence, our proposed objective function will become as:

$$\min_{\mathbf{A},\mathbf{Q},\mathbf{b},\mathbf{S}} \quad Tr(\mathbf{A}^T\mathbf{L}\mathbf{A}) + \lambda Tr(\mathbf{Z}\mathbf{L}\mathbf{Z}^T) + Tr((\mathbf{B}\mathbf{A}-\mathbf{Y})^T\mathbf{V}(\mathbf{B}\mathbf{A}-\mathbf{Y})) + \frac{\rho}{2}||\mathbf{S}||_2^2 + \mu(||\mathbf{Q}||^2 + \gamma||\mathbf{Z}^T\mathbf{Q} + \mathbf{1}\mathbf{b}^T - \mathbf{A}||^2) \tag{4}$$

where $\mathbf{V}$ is the diagonal matrix of the labeled sample weights (Eq. (3)) and the unknown variable are those of SGRFME.

In the next section, we explain the solution of the proposed objective function.

3.4 Optimization

The proposed method has a first step of initialization followed by the solution to obtain the unknowns.

In the initialization step, we determine the anchors and calculate the weight of labeled samples. To determine the anchors, we use the well-known Kmeans clustering method and determine m centroids and set them as anchors (i.e. $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, ..., \mathbf{z}_m]$). These anchors allows us to estimate the weight of labeled samples and moreover, we use them in the semi-superivsed model depicted in the objective function 4.

To calculate the weights of labeled nodes, we follow the procedure explained in Section 3.2. We build a new data matrix as $\mathbf{X}_T = [\mathbf{X}_l, \mathbf{Z}]$ by linking the labeled samples and the anchors. Then, we construct the affinity matrix for the $\mathbf{X}_T$ matrix using the well-known KNN method. The we use the algorithm presented in (Wang et al., 2016) to construct the $\mathbf{B}$ matrix and calculate the initial anchor-to-anchor graph by setting $\mathbf{S} = \mathbf{B}^T\mathbf{B}$.

Next step is how to solve the optimization function introduced in Eq.4 to calculate the unknowns (i.e., $\mathbf{S}$, $\mathbf{A}, \mathbf{Q}$, $\mathbf{b}$). Since the proposed objective function does not have a closed form solution, we adopt an iterative algorithm to solve it. In other words, we fix some variables and solve for other variables.

Fix S and estimate $\mathbf{A}, \mathbf{Q}$, b

By fixing the $\mathbf{S}$ matrix, the objective function is reduced to

$$\min_{\mathbf{A},\mathbf{Q},\mathbf{b}} Tr(\mathbf{A}^T\mathbf{L}\mathbf{A}) + Tr(\mathbf{B}\mathbf{A}-\mathbf{Y})^T\mathbf{V}(\mathbf{B}\mathbf{A}-\mathbf{Y}) + \mu(||\mathbf{Q}||^2 + \gamma||\mathbf{Z}^T\mathbf{Q} + \mathbf{1}\mathbf{b}^T - \mathbf{F}||^2) \tag{5}$$

that is similar to the objective function of r-FME method (Qiu et al., 2019), hence, the solution is similar to the solution of r-FME variables as

$$\mathbf{A} = [\mathbf{L} + \mathbf{B}^T\mathbf{V}\mathbf{B} + \mu\mathbf{H}_a - \mu\mathbf{H}_a\mathbf{Z}^T(\mathbf{Z}\mathbf{H}_a\mathbf{Z}^T + \gamma\mathbf{I})^{-1}\mathbf{Z}\mathbf{H}_a]^{-1}\mathbf{B}^T\mathbf{V}\mathbf{Y} \tag{6}$$

$$\mathbf{Q} = (\mathbf{Z}\mathbf{H}_a\mathbf{Z}^T + \gamma\mathbf{I})^{-1}\mathbf{Z}\mathbf{H}_a\mathbf{A} \tag{7}$$

$$\mathbf{b} = \frac{1}{m}(\mathbf{A}^T 1 - \mathbf{Q}^T \mathbf{Z} 1) \tag{8}$$

Fix $\mathbf{A}, \mathbf{Q}$, b and estimate S

Next step we fix all variables and solve the objective function to estimate the **S**. Doing so, the objective function can be reduced to

$$\min_{\mathbf{S}} \quad Tr(\mathbf{A}^T \mathbf{L} \mathbf{A}) + \lambda Tr(\mathbf{Z} \mathbf{L} \mathbf{Z}^T) + \frac{\rho}{2} ||\mathbf{S}||_2^2 \tag{9}$$

We have the following in the area of spectral analysis:

$$Tr(\mathbf{A}^T \mathbf{L} \mathbf{A}) = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} ||\mathbf{A}_{i,.} - \mathbf{A}_{j,.}||_2^2 S_{ij} \tag{10}$$

where $\mathbf{A}_{i,.}$ is the i^{th} row of the matrix **A**. Thus, by expanding the two trace terms, the minimization problem of Eq. (9) can be written as

$$\min_{\mathbf{S}} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \left(||\mathbf{A}_{i,.} - \mathbf{A}_{j,.}||_2^2 + \lambda ||\mathbf{Z}_{.,i} - \mathbf{Z}_{.,j}||_2^2 \right) S_{ij} \tag{11}$$

$$+ \frac{\rho}{2} \sum_{i=1}^{m} ||\mathbf{s}_i||_2^2$$

$$\equiv \min_{\mathbf{S}} \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} g_{ij} S_{ij} + \frac{\rho}{2} \sum_{i=1}^{m} ||\mathbf{s}_i||_2^2$$

where $g_{ij} = (||\mathbf{A}_{i,.} - \mathbf{A}_{j,.}||_2^2 + \lambda ||\mathbf{Z}_{.,i} - \mathbf{Z}_{.,j}||_2^2)$, and $\mathbf{Z}_{.,i}$ is the ith column of the matrix **Z**.

Eq. (11) may be subdivided into m sub-problems $(i = 1, ..., m)$, each of which can be used to estimate a row of the similarity matrix, $\mathbf{s}_i$. So, we have:

$$\min_{\mathbf{s}_i} \sum_{j=1}^{m} g_{ij} S_{ij} + \rho ||\mathbf{s}_i||_2^2, \;\; i = 1, \ldots m \tag{12}$$

The solution of Eq. (12) was introduced in (Nie et al., 2016; Nie et al., 2017) using a closed form solution by imposing three constraints on the m problems. First, the solution is non-negative (i.e., $\mathbf{s} > 0$). Second, their sum is 1 ($\sum_{i=1}^{m} s_{ij} = 1$). Third, the optimal solution $\mathbf{s}_i$ has exactly K nonzero values, where $K \leq 10$.

Till here, we solved the algorithm for one iteration. We repeat these two steps to calculate $\mathbf{A}, \mathbf{Q}$, **b** and **S** until the difference between two **S** matrices in subsequent iterations is less than a threshold or we reach 10 iterations.

After convergence, the problem is how to estimate the label of unlabeled samples in the training set and the test samples. For the unlabeled samples one can use Eq. (1) to determine the labels and for the test samples the labels can be estimate using

$$\mathbf{f} = \mathbf{Q}^T \mathbf{x}_{test} + \mathbf{b} \tag{13}$$

For the estimated soft label vector f_k the class label is obtained by

$$c = \arg\max_{k} f_k; k = 1, \ldots, C \tag{14}$$

4 EXPERIMENTAL RESULTS

In this section, we evaluate the performance of the proposed method compared to recently proposed methods. For this purpose, two algorithms namely r-fme (Qiu et al., 2019) and SGRFME (Ibrahim et al., 2023) is adopted.

For databases, we select two large-scale databases namely Covtype and MNIST databases.

MNIST. This database has 60,000 images, and we randomly select 1000 samples from each class for training and the rest for testing. We fed the images into the ResNet-50 (He et al., 2015) network and extracted the information in the Average Pooling layer as the image descriptor that forms a 2048-dimensional vector.

Covtype. This database[1] contains the forest cover type for 30 x 30 meter cells obtained from US Forest Service data. It contains 581,012 instances and 54 attributes. We arbitrarily selected 80% of the data for training and the remaining 20% for testing.

To reduce the dimensionality of the data, we applied PCA and kept the top 50 dimensions.

For both databases, o samples of each class in the training set are selected as labeled and the rest as unlabeled. Having C classes, we can conclude to have $l = o \times C$. To reduce the dependency of results on a specific set of labeled data, we created 20 random combinations of labeled and unlabeled sets and reported the average of results.

We use Matlab version R2018a and a PC with a i9-7960@2.80 GHz CPU and 128 GB RAM.

4.1 Parameter Evaluation

The proposed method has several parameters i.e., μ, γ, λ, ρ, α, and w_{max}. The values adopted for these parameters can has a high impact on the accuracy of the proposed method, hence their values should be cleverly selected.

To evaluate the effect these parameters, we take the five labeled samples from the MNIST database and vary the parameters and report the accuracy.

In the first experiment, we evaluate the effect of the parameter α. Fig. 1 shows the variation of accuracy respect to the variation of the parameter α. As we observe, the accuracy for the test samples is relatively fix and does not vary when α varies, however, the unlabeled data has a higher variance. Also, the accuracy has its best values when α is 0.7.

In the second experiment, the effect of w_{max} is evaluated. Fig. 2 shows the effect of varying this parameter on the accuracy. We can see that the accuracy

[1] http://archive.ics.uci.edu/ml/datasets/Covertype

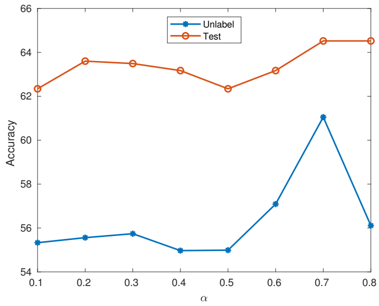

Figure 1: Performance of the proposed method versus α parameter on the MNIST database with five labeled samples.

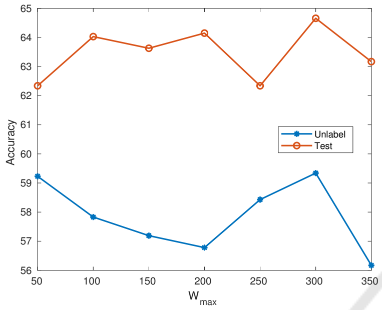

Figure 2: Performance of the proposed method versus w_{max} parameter on the MNIST database with five labeled samples.

has its highest value when w_{max} is 300. Moreover, we observe that the accuracy does not follow the same trend specially when w_{max} is lower than 250.

Thirdly, we evaluate the effect of varying number of the anchors. In Fig. 3, we have plotted the accuracy when the number of anchors varies in the range [10 20 50 100 200 500 1000 2000]. As we can see the accuracy increases as the number of anchors increases, however, we observe that the accuracy suddenly drops as we use more than 1000 anchors.

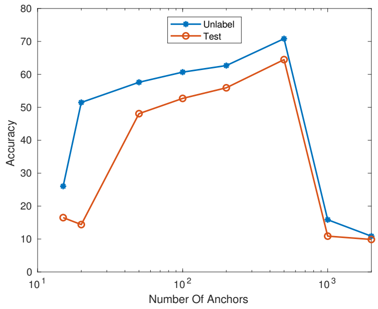

Figure 3: Performance of the proposed method versus the number of anchors on the MNIST database with five labeled samples.

4.2 Comparison with Other Methods

For comparison, we used two recently proposed methods, r-FME(Qiu et al., 2019) and SGRFME(Ibrahim et al., 2023). Since we are in semi-supervised context, in the training data we have labeled and unlabaled samples. Hence, for the number of labeled samples per class, in the Covtype database, we set l to $30, 50$, and 70 and for MNIST we set l to 5 and 20.

In these experiments, we fixed w_{min} to one, α to 0.7, and w_{max} to 300. For the diagonal U_l matrix, we set the first l elements to the weights obtained by Totoro values (i.e., w_k) and the rest to zero. For μ, γ, λ, and ρ, we select one split of labeled and unlabeled data and scan the parameters to find the best combinations. Then we fix the obtained parameters for the rest of the experiments.

Table 1 shows the average and standard deviation for 20 random combinations of labeled and unlabeled samples on Covtype an MNIST databases. We use bold font for the highest accuracy. As we observe, the average accuracy of the proposed method is higher compared to other competing methods including SGRFME, which shows the effect of the adaptive weighting of SGRFME algorithm. Also, the standard deviation of the proposed method is lower, showing a more stable accuracy compared to other methods. We observe this behavior for both databases which shows that there is no bias toward a specific database for the outperformance of the proposed method. Also, we can see that even though the training database is the same, when we increase the number of labeled samples, we have an increase in the accuracy for all methods. However, the proposed method keeps its outperformance in different numbers of labeled samples.

4.3 Confusion Matrix

We calculate the confusion matrix of the proposed method. We select one split of MNIST database with 5 labeled samples and estimate the labels of the unlabeled set and test set. We then plot the confusion and report the precision, recall, and F1 score for each class in Table 2. Moreover, we report the macro precision, macro recall, and macro F1 score. Macrometrics (Precision, Recall, and F1 score) are calculated by averaging the metrics across all classes.

5 CONCLUSION

In this article, we tackle the problem of topology imbalance in graphs. We adopted the Renode technique to assign weights to the labeled samples. To do so,

Table 1: Average accuracy with standard deviation for the proposed method and two recent methods (r-FME and SGRFME) obtained on 20 random combinations of labeled and unlabeled samples.

Covtype						
Method	**30 labeled samples**		**50 labeled samples**		**70 labeled samples**	
	Unlabeled	Test	Unlabeled	Test	Unlabeled	Test
r-FME (Qiu et al., 2019)	47.70 ± 3.20 $(10^{15}, 10^{0})$	45.88 ± 3.87 $(10^{9}, 10^{-3})$	49.54 ± 1.78 $(10^{24}, 10^{6})$	50.01 ± 3.14 $(10^{9}, 10^{-3})$	51.89 ± 2.08 $(10^{9}, 10^{-3})$	53.36 ± 2.74 $(10^{9}, 10^{-3})$
SGRFME (Ibrahim et al., 2023)	51.00 ± 2.02 $(10^{24}, 10^{-12}$ $10^{18}, 10^{-12})$	49.62 ± 2.39 $(10^{24}, 10^{-12}$ $10^{-12}, 10^{18})$	52.39 ± 1.82 $(10^{12}, 10^{24}$ $10^{24}, 10^{12})$	52.23 ± 1.97 $(10^{6}, 10^{3}$ $10^{12}, 10^{0})$	54.62 ± 0.95 $(10^{18}, 10^{6}$ $10^{3}, 10^{24})$	54.11 ± 1.24 $(10^{6}, 10^{3}$ $10^{12}, 10^{24})$
W-SGRFME	$\mathbf{52.70 \pm 2.58}$ $(10^{24}, 10^{-12}$ $10^{-12}, 10^{18})$	$\mathbf{52.91 \pm 2.26}$ $(10^{24}, 10^{-12}$ $10^{-12}, 10^{18})$	$\mathbf{53.38 \pm 0.84}$ $(10^{12}, 10^{24}$ $10^{24}, 10^{12})$	$\mathbf{53.41 \pm 1.74}$ $(10^{6}, 10^{3}$ $10^{12}, 10^{0})$	$\mathbf{55.16 \pm 0.76}$ $(10^{18}, 10^{6}$ $10^{3}, 10^{24})$	$\mathbf{54.77 \pm 0.66}$ $(10^{6}, 10^{3}$ $10^{12}, 10^{24})$

MNIST				
Method	**5 labeled samples**		**20 labeled samples**	
	Unlabeled	Test	Unlabeled	Test
r-FME (Qiu et al., 2019)	64.47 ± 2.24 $(10^{21}, 10^{-12})$	57.82 ± 4.31 $(10^{0}, 10^{3})$	70.39 ± 1.05 $(10^{21}, 10^{-12})$	67.61 ± 3.07 $(10^{9}, 10^{0})$
SGRFME (Ibrahim et al., 2023)	65.29 ± 1.82 $(10^{9}, 10^{0}$ $10^{-24}, 10^{15})$	58.27 ± 4.42 $(10^{-24}, 10^{3}$ $10^{-12}, 10^{9})$	71.22 ± 0.83 $(10^{9}, 10^{0}$ $10^{-24}, 10^{15})$	68.08 ± 2.95 $(10^{-24}, 10^{3}$ $10^{-12}, 10^{9})$
W-SGRFME	$\mathbf{66.09 \pm 1.42}$ $(10^{9}, 10^{0}$ $10^{-24}, 10^{15})$	$\mathbf{59.17 \pm 4.12}$ $(10^{-24}, 10^{3}$ $10^{-12}, 10^{9})$	$\mathbf{71.72 \pm 1.01}$ $(10^{9}, 10^{0}$ $10^{-24}, 10^{15})$	$\mathbf{69.17 \pm 2.89}$ $(10^{-24}, 10^{3}$ $10^{-12}, 10^{9})$

Table 2: Confusion Matrix, Precision, Recall, and F1 of the proposed method on the MNIST database (5 labeled samples).

Unlabeled samples													
	Predicted										Precision	Recall	F1
Class 1 (Actual)	846	0	6	4	0	1	33	2	14	59	0.85	0.87	0.86
Class 2 (Actual)	0	1050	0	0	2	0	3	28	0	2	0.93	0.96	0.95
Class 3 (Actual)	9	1	547	95	15	146	40	22	74	120	0.55	0.51	0.53
Class 4 (Actual)	3	0	48	307	1	170	3	2	7	12	0.30	0.55	0.38
Class 5 (Actual)	7	43	92	9	900	31	28	306	14	29	0.92	0.61	0.73
Class 6 (Actual)	2	0	71	552	4	498	6	3	8	27	0.55	0.42	0.48
Class 7 (Actual)	84	0	101	18	2	34	779	9	71	114	0.79	0.64	0.70
Class 8 (Actual)	3	29	28	0	25	5	13	661	2	11	0.63	0.85	0.72
Class 9 (Actual)	10	0	24	18	0	3	7	0	645	50	0.66	0.85	0.74
Class 10 (Actual)	23	1	76	19	25	16	74	11	140	568	0.57	0.59	0.58
Macro											0.69	0.68	0.67

Test samples													
	Predicted										Precision	Recall	F1
Class 1 (Actual)	4679	1	60	21	14	40	805	71	18	581	0.94	0.74	0.83
Class 2 (Actual)	7	4981	37	0	41	19	198	884	33	40	0.88	0.79	0.84
Class 3 (Actual)	2	4	1184	37	35	63	56	54	21	195	0.23	0.71	0.35
Class 4 (Actual)	59	1	1168	4070	56	2827	178	75	80	752	0.79	0.43	0.56
Class 5 (Actual)	37	620	683	49	4213	87	175	1072	81	264	0.86	0.57	0.69
Class 6 (Actual)	15	0	499	696	7	1092	296	10	24	89	0.24	0.40	0.30
Class 7 (Actual)	33	0	67	21	0	39	1920	8	4	31	0.38	0.90	0.54
Class 8 (Actual)	16	11	409	27	454	126	82	3035	10	123	0.58	0.70	0.63
Class 9 (Actual)	87	0	818	148	36	76	1154	10	4598	1484	0.94	0.54	0.69
Class 10 (Actual)	1	0	40	40	12	148	68	2	7	1398	0.28	0.81	0.41
Macro											0.61	0.66	0.58

we used anchors as data representatives and modified the Renode method to extend the idea of topological imbalance for large scale databases. Then we adopted the label estimation error term to insert these weights into our objective function. Our experimental results on two large databases show that the proposed method has a higher average accuracy and lower standard deviation compared to two recently proposed methods namely r-FME and SGRFME. On the other hand, due to the iterative nature of the proposed method, it has a higher computational complexity than its competitive methods. Moreover, the relatively large number of parameters to be tuned is another weakness of the proposed method. Hence, as a future work, we focus on the automatic tuning of these parameters. Also, reduction of the running time is another track to follow.

REFERENCES

Aromal, A., M. Rasool, A., Dubey, A., and Roy, B. N. (2021). Optimized weighted samples based semi-supervised learning. In *2021 Second International Conference on Electronics and Sustainable Communication Systems (ICESC)*, pages 1311–1318.

Belkin, M., Niyogi, P., and Sindhwani, V. (2006). Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *The Journal of Machine Learning Research*, 7:2399–2434.

Bosaghzadeh, A., Moujahid, A., and Dornaika, F. (2013). Parameterless local discriminant embedding. *Neural Processing Letters*, 38.

Bui, Q.-T., Vo, B., Do, H.-A. N., Hung, N. Q. V., and Snasel, V. (2020). F-mapper: A fuzzy mapper clustering algorithm. *Knowledge-Based Systems*, 189:105107.

Bui, Q.-T., Vo, B., Snasel, V., Pedrycz, W., Hong, T.-P., Nguyen, N.-T., and Chen, M.-Y. (2021). Sfcm: A fuzzy clustering algorithm of extracting the shape information of data. *IEEE Transactions on Fuzzy Systems*, 29(1):75–89.

Chen, D., Lin, Y., Zhao, G., Ren, X., Li, P., Zhou, J., and Sun, X. (2021). Topology-imbalance learning for semi-supervised node classification. *Advances in Neural Information Processing Systems*, 34:29885–29897.

Chen, X., Yu, G.-X., Tan, Q., and Wang, J. (2019). Weighted samples based semi-supervised classification. *Applied Soft Computing*, 79:46–58.

Collobert, R., Sinz, F., Weston, J., and Bottou, L. (2006). Large scale transductive svms. *Journal of Machine Learning Research*, 7:1687–1712.

Cui, B., Xie, X., Hao, S., Cui, J., and Lu, Y. (2018). Semi-supervised classification of hyperspectral images based on extended label propagation and rolling guidance filtering. *Remote Sensing*, 10(4).

Dornaika, F., Baradaaji, A., and El Traboulsi, Y. (2021). Semi-supervised classification via simultaneous label and discriminant embedding estimation. *Information Sciences*, 546:146–165.

Hamilton, W. L., Ying, R., and Leskovec, J. (2017). Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 1025–1035, Red Hook, NY, USA. Curran Associates Inc.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Ibrahim, Z., Bosaghzadeh, A., and Dornaika, F. (2023). Joint graph and reduced flexible manifold embedding for scalable semi-supervised learning. *Artificial Intelligence Review*, 56:9471–9495.

Kang, Z., Peng, C., Cheng, Q., Liu, X., Peng, X., Xu, Z., and Tian, L. (2021). Structured graph learning for clustering and semi-supervised classification. *Pattern Recognition*, 110:107627.

Long, Y., Li, Y., Wei, S., Zhang, Q., and Yang, C. (2019). Large-scale semi-supervised training in deep learning acoustic model for asr. *IEEE Access*, 7:133615–133627.

Nie, F., Cai, G., and Li, X. (2017). Multi-view clustering and semi-supervised classification with adaptive neighbours. In *Thirty-First AAAI Conference on Artificial Intelligence*.

Nie, F., Wang, X., Jordan, M. I., and Huang, H. (2016). The constrained laplacian rank algorithm for graph-based clustering. In *AAAI Conference on Artificial Intelligence*.

Nie, F., Xu, D., Tsang, I. W., and Zhang, C. (2010). Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction. *IEEE Transactions on Image Processing*, 19(7):1921–1932.

Qiu, S., Nie, F., Xu, X., Qing, C., and Xu, D. (2019). Accelerating flexible manifold embedding for scalable semi-supervised learning. *IEEE Transactions on Circuits and Systems for Video Technology*, 29(9):2786–2795.

Sindhwani, V. and Niyogi, P. (2005). Linear manifold regularization for large scale semi-supervised learning. In *Proc. of the 22nd ICML Workshop on Learning with Partially Classified Training Data*.

Sindhwani, V., Niyogi, P., Belkin, M., and Keerthi, S. (2005). Linear manifold regularization for large scale semi-supervised learning. *Proc. of the 22nd ICML Workshop on Learning with Partially Classified Training Data*.

Song, Z., Yang, X., Xu, Z., and King, I. (2022). Graph-based semi-supervised learning: A comprehensive review. *IEEE Transactions on Neural Networks and Learning Systems*, pages 1–21.

Tu, E., Wang, Z., Yang, J., and Kasabov, N. (2022). Deep semi-supervised learning via dynamic anchor graph embedding in latent space. *Neural Networks*, 146:350–360.

Wang, J., Yang, J., Yu, K., Lv, F., Huang, T., and Gong, Y. (2010). Locality-constrained linear coding for image classification. In *IEEE Conference on Computer Vision and Pattern Recognition*.

Wang, M., Fu, W., Hao, S., Tao, D., and Wu, X. (2016). Scalable semi-supervised learning by efficient anchor graph regularization. *IEEE Transactions on Knowledge and Data Engineering*, 28(7):1864–1877.

Wang, Z., Wang, L., Chan, R. H., and Zeng, T. (2019). Large-scale semi-supervised learning via graph structure learning over high-dense points.

Wang, Z., Zhang, L., Wang, R., Nie, F., and Li, X. (2022). Semi-supervised learning via bipartite graph construction with adaptive neighbors. *IEEE Transactions on Knowledge and Data Engineering*, pages 1–1.

Wu, X., Zhao, L., and Akoglu, L. (2019). A quest for structure: Jointly learning the graph structure and semi-supervised classification.

Yuan, Y., Li, X., Wang, Q., and Nie, F. (2021). A semi-supervised learning algorithm via adaptive laplacian graph. *Neurocomputing*, 426:162–173.

Zhu, X. and Lafferty, J. (2005). Harmonic mixtures: combining mixture models and graph-based methods for inductive and scalable semi-supervised learning. In *Machine Learning, Proceedings of the Twenty-Second International Conference (ICML 2005), Bonn, Germany, August 7-11, 2005*.