Semi-supervised Learning with Deterministic Labeling and Large Margin Projection

Ji Xu, Gang Ren, Yao Xiao, Shaobo Li, Guoyin Wang, Senior Member, IEEE

Abstract—The centrality and diversity of the labeled data are very influential to the performance of semi-supervised learning (SSL), but most SSL models select the labeled data randomly. This study first construct a leading forest that forms a partially ordered topological space in an unsupervised way, and select a group of most representative samples to label with one shot (differs from active learning essentially) using property of homeomorphism. Then a kernelized large margin metric is efficiently learned for the selected data to classify the remaining unlabeled.sample. Optimal leading forest (OLF) has been observed to have the advantage of revealing the difference evolution along a path within a subtree. Therefore, we formulate an optimization problem based on OLF to select the samples. Also with OLF, the multiple local metrics learning is facilitated to address multi-modal and mix-modal problem in SSL, especially when the number of class is large. Attribute to this novel design, stableness and accuracy of the performance is significantly improved when compared with the state-of-the-art graph SSL methods. The extensive experimental studies have shown that the proposed method achieved encouraging accuracy and efficiency. Code has been made available at https://github.com/alankuji/DeLaLA.

Index Terms—Deterministic labeling, homeomorphism, large margin, optimal leading forest, semi-supervised learning.

I. INTRODUCTION

SEMI-SUPERVISED learning (SSL) has been attracting the attention of researchers for a long history [1] [2] and keeps gaining more popularity in the big data era (e.g., [3] [4] [5] [6], just name a few), because labeling data is a laborious or expensive task while unlabeled data are relatively easy to collect. Different from various inductive SSL, including self-training [7], co-training [8] [9], etc., graph SSL (GSSL) has the reputation of good interpretability, competitive accuracy and scalability to large scale data [10]. However, traditional GSSLs have the limitation of relatively low efficiency due to their iterative optimization in label propagation and training the entire dataset. There is a bunch of state-of-the-art research aiming to solve the low-efficiency problem of GSSL, among which are efficient anchor graph regularization (EAGR) [11] optimal bipartite graph-based SSL (OBGSSL) [5], label propagation in optimal leading forest (LaPOLLeaf) [6], etc. As can be shown in our study later, there is still some room for improvement of learning efficiency of GSSL. In addition, traditional GSSLs are subjected to accuracy unstableness because of their random labeling strategy.

Randomly choosing samples to label may suffer performance drop when the chosen data are of high alike andness and there is a domain-shift between labeled and unlabeled data. This phenomenon that there are many clusters within a class and multiple classes may locate in one cluster has been referred to as “multimodal” and “mixmodal”, respectively [12]. A collection of research on this regard is called domain adaptation, which aims to generalize the training model to accurately predict or classify the data from non-identical distribution [13]. Domain adaption methods usually have sophisticated formulations and a relatively heavy computational cost. To this end, choosing a few most representative samples to label is a desirable alternative to save labeling expenditure and achieve stable accuracy.

There have been quite a few research works on selecting data in machine learning (e.g., [14], [15], [16]), most of which fall into the paradigm of active learning [17] and follow a progressive principle in classification. To our best knowledge, there has no research that deterministically choosing the samples to label with one shot in the field of SSL. Inspired by topological meaning of a leading tree path [6], we consider select most representative samples in a path that can approximately represent the underlying manifold of input data. Then it can be expected to achieve a stably superior accuracy over random labeling strategy and higher efficiency over active learning.

After the deterministic labeling, a kernelized large margin component analysis (KLMCA) [18] method is applied to non-linearly project the data into a lower dimensional intrinsic manifold. The advantage of learning a supervised dimension reduction to improve the classification has been recognized for a long time [19] [20]. Recently, contrastive learning has attracted wide interest from deep learning society [21] [22], which shares the same idea with large margin learning. When compared to the exist supervised contrastive learning approaches, our method needs only a few selected labeled data to training and therefore more efficient and economical. If the data have many classes (for example, more than ten classes) or the inner clusters with a class are too many, then learning a global metric may not be adequate to describe the full distribution. In this case, we adapt a multiple local metric learning policy inspired by [23].

We presented in this paper a semi-supervised learning method with Deterministic Labeling and Large Margin projection that is termed as DeLaLA. The proposed model leverages a structure named as optimal leading forest OLF [24], which
we have developed for multi-granular clustering subjected to the principle of justifiable granulation [25].

The SSL society has reached a consensus that inappropriately including more unlabeled data may degenerate the performance [26] [27]. Another advantage of DeLaLA is it can avoid including harmful samples in training and degenerating the SSL model. By training on just a few selected samples, one can bypass the risk of including some harmful unlabeled data in training.

An illustrative example of DeLaLA is shown in Fig. 1. DeLaLA first constructs OLF from the input data without labels, then the most divergent and representative samples, which are recognized as the root and leaves with maximum depth in a subtree, are chosen to label. Thirdly, the labeled exemplar data are used to train a series of large margin metric with KLMCA. After the nonlinear projection is learned, each class of exemplar samples will be pull to a compact region and different class of data are pushed far away from each other. Finally, the ultimate classification is fulfilled by applying the learned metrics to the unlabeled data and assigned the labels to the projected representations according to a simple 1NN policy.

The main contributions of DeLaLA are as follows.

1) Deterministically (rather than randomly as the existing approaches) selecting the most representative and the most divergent samples to label via topological homeomorphism, which guarantees the stability of the SSL performance. The idea combines the robustness and uncertainty policies in active learning, but the selection is performed with only one shot.

2) The training of DeLaLA is highly efficient. The construction of an OLF is easy to accelerate and the learning of a nonlinear projection onto a lower dimensional space is fast because it is trained on just a few selected samples.

3) The core structure of DeLaLA is the leading tree, which is more natural and stable to reflect the true neighborhoods in data than k-nearest neighbor. The micro-clusters formed with leading tree are stable while those formed by k-means are changeable when the number k or the initial center guess is different.

4) The leading tree structure is convenient to present a hierarchical clustering result [28]. Therefore, DeLaLA can learn a succession of global-local metrics to address the issue of multi-modal and mix-modal phenomenon [12] in many-class classification.

In addition, DeLaLA shows encouraging accuracy and actual running time in the empirical validation, which is in accordance with the theoretical analysis.

The rest of the paper is organized as follows. Section II briefly reviews the related work. The model of DeLaLA is presented in details in Section III. Section IV analyzes the computation complexity of DeLaLA and discusses the relationship to other researches, and Section V describes the experimental study. The conclusion is given in Section VI.

A. Notations

Let $\mathcal{X} = \{x_1, \ldots, x_l, x_{l+1}, \ldots, x_{l+n}\} \in \mathbb{R}^{n \times d}$ denotes a dataset that consists of l labeled and u unlabeled samples, $n = l + u$. In addition, we denote $\mathcal{U} = \{x_1, \ldots, x_l\}$ and $\mathcal{L} = \{x_{l+1}, \ldots, x_{l+n}\}$. Usually, we have $l \ll n$. The labels of the labeled samples are $\mathcal{Y} = \{y_1, y_2, \ldots, y_l\}$, $y_j \in \{1, \ldots, C\}$. C is the number of class. $D = \{d_{ij}\}_{i,j=1}^{n}$ denotes the distance in the $\mathbb{R}^d$ space and kernelized distance between each pair of $x_i$ and $x_j$, respectively.

B. Optimal leading forest

Optimal leading forest has been shown to be effective in multi-granular granulation [24] and label propagation in GSSL [6]. An OLF is a collection of leading trees obtained by disconnecting the centers from their parent in the entire leading tree corresponding to the dataset $\mathcal{X}$. We describe here the matrix and vectors to compute to construct an entire leading tree from original data.

a) distance matrix $D$. Although the complexity for computing $D$ is $O(n^2)$, it is convenient to accelerate the computing thanks to its good parallelizability [6].

b) local density of each data point

$$
\rho_i = \sum_{j \neq i} \exp(-d_{ij}^2/\sigma^2),
$$
where $\sigma$ is a bandwidth parameter.

c) leading node (also known as the parent in the leading tree) of each sample, except for the point with greatest local density

$$
P_a_i = \arg \min_{x_j} \{d_{ij} | \rho_j > \rho_i\}.
$$

In the course of determining $P_a_i$, we also record the distance

$$
\delta_i = \min \{d_{ij} | \rho_j > \rho_i\}
$$
for further use.

Once the vector $Pa_i$ is computed, the leading tree can be constructed from the dataset $\mathcal{X}$. By leading tree, it means that a node $x_i$ tends to be led by $Pa_i$ to join the same cluster (or class) that $P_a_i$ belongs to.

The leading tree structure has two advantages. One is its centers are characterized by high values of centrality

$$
\gamma_i \triangleq \rho_i \delta_i,
$$
therefore the clustering result are simply achieved by disconnecting the nodes with highest $\gamma$ parameter from their corresponding parents, this procedure do not involve iterative optimization. The other advantage is the structure can reflect the difference evolution within a given class, in the form of a path from the root to a leaf node with greatest depth [6].

C. Large margin component analysis

A milestone of distance metric learning is the large margin nearest neighbor method (LMNN) [20]. However, LMNN has slow convergence and overfitting issues since it learns the matrix $M = L^\top L$ and needs to project the intermediate result of $M_t$ onto a positive semidefinite cone in each iteration.

II. RELATED STUDIES
Apart from accuracy improvement, LMCA accelerates the computation of LMNN by directly optimizing \( L \) rather than \( M \). LMCA offers a linear and a kernelized version of the lower-dimension projection algorithm. Because of its stronger modeling capacity, we are especially interested in the kernelized version.

LMNN and LMCA share the same optimization objective as in Eq. (5).

\[
\epsilon(L) = \sum_{ij} \eta_{ij} L_{ij} + c \sum_{ijm} \eta_{ij}(1 - y_{im}) h(L_{ij} - L_{im} + 1),
\]

where \( L_{ij} = \|L(x_i - x_j)\|^2 \) is the squared distance between \( x_i \) and \( x_j \) in the projected space, \( \eta_{ij} \in \{0, 1\} \) indicates whether \( x_j \) is a \( k \)-nearest neighbor of \( x_i \) and \( y_i = y_j \), \( c \) is a positive parameter. \( y_{im} \in \{0, 1\} \) is 0 iff \( y_i \neq y_m, i, m \leq l \), and \( h(s) = \max(s, 0) \) is the hinge function.

Kernelized MLCA (KLMCA) unifies the Mahalanobis distance metric learning and dimension reduction in one framework. It replaces \( x_i \) in Eq. (5) with a kernel mapping \( \phi_i = \phi(x_i) \), but one needs not compute \( \phi(x_i) \) explicitly. If we denote \( \phi^{ij} = (\phi_i - \phi_j)(\phi_i - \phi_j)^\top \) for short, then the gradient of \( \epsilon(L) \) can be written as

\[
\frac{\partial \epsilon(L)}{\partial L} = 2L \sum_{ij} \eta_{ij} \phi^{ij} + 2cL \sum_{ijm} \eta_{ij}(1 - y_{im}) h'(s_{ijm}) \phi^{ij} - \phi^{im},
\]

where \( s_{ijm} = (\|\phi_i - \phi_j\|^2 - \|\phi_i - \phi_m\|^2 + 1) \). Denote \( [\phi_1, ..., \phi_n]^\top \) as \( \Phi \) and consider \( L \) as a combination of the feature points \( \{\phi_i\} \), then one can parameterize \( L \) as \( L = \Omega \Phi \). This parameterization allows us to update \( \Omega \) rather than \( L \) in each iteration to learn the lower-dimension projection as in Eq. (7).

\[
L^{(t+1)} = L^{(t)} - \lambda \frac{\partial \epsilon(L)}{\partial L} \bigg|_{L=L^{(t)}} = \left( \Omega^{(t)} - \lambda \Gamma^{(t)} \right) \Phi = \Omega^{(t+1)} \Phi,
\]

in which \( \Gamma \) is a function of (\( \Omega \)) defined as

\[
\Gamma = 2\Omega \sum_{ij} \eta_{ij} \epsilon_{ij} + 2c\Omega \sum_{ijm} \eta_{ij} h'(s_{ijm})(1 - y_{im}) \left[ \epsilon_{ij} - \epsilon_{im} \right],
\]

where \( \epsilon_{ij} = (E_i^{k_j - k_i} - E_j^{k_j - k_i}) \), \( E_i^v = [0, ..., v, 0, ..., 0] \) is a \( n \times n \) matrix with the \( v \)-th column equals \( v \) and all other columns equal to zero vector, and \( k_i = \Phi \phi_i = [\kappa(1, i), ..., \kappa(n, i)] \).

According to Eq. (7) and (8), we can learn \( \Omega \) by the updating rule \( \Omega \leftarrow \Omega - \lambda \Gamma \) until convergence, and the ultimate projection of \( x_q \) can be computed via \( L\phi_q = \Omega \Phi \phi_q = \Omega k_q \).

We denote the dimensionality of \( x_q \) after the large margin projection as \( p \), usually \( p \ll d \).

III. SSL WITH DETERMINISTIC LABELING AND LARGE MARGIN PROJECTION

A. Homeomorphism between input data and OLF

From the perspective of topology, a leading tree constructed from a dataset forms a partially ordered discrete topological space \( (G, \mathcal{T}_G) \) [29], where \( G \) is the set of node in the leading forest and \( \mathcal{T}_G \) the topology on \( G \). The relationship between the data points in the Euclidean space and the topological space is illustrated in Fig. 2. By the homeomorphism between the topological space \( (X, \mathcal{T}_X) \) and \( (G, \mathcal{T}_G) \), we can select the most representative samples by exploiting the centrality, density and layer index of each node in the leading tree. The evolution along a path we observed in the leading tree definitely has its equivalence in the original sample space because of the bijective correspondence [30], based on which our sample-selecting strategy is mathematically grounded. Because in the original sample space, it is intractable to select the data points that are representative in the sense of centrality or divergence with one shot. So we turn to the other topological space in the form of leading forest, in which the information of centrality or divergence lies in paths of a subtree.
Fig. 2: Consider we want to insert a node $k$ in the leading tree, then we actually need to add a dot in the possible region located between $a$ and $b$. The possible region is an open set. Therefore, the one-to-one mapping between an open set in the Euclidean space and a node of the leading tree indicates that the two topological spaces $(X, \mathcal{T}_X)$ and $(G, \mathcal{T}_G)$ are homeomorphic.

B. Objective Function

Ideally, we would like to design a discrete optimization objective to select a small sized subset, satisfying the following desiderata:

a) It selects both typical and divergent samples in each class;

b) At least one typical sample in each class are included;

c) The proportions of typical sample and divergent sample are adaptive. That is, if multi-modal phenomenon exists in a class, then multiple typical samples are emphasized; if the distribution of the samples within a class is relatively centered, then the divergent samples laying on the margin zone are emphasized.

Let $\mathcal{L}_c$ be the labeled samples of class $c$. $\mathcal{L}_{cen}$ and $\mathcal{L}_{div}$ denote the labeled samples expressing the centrality and divergence of each class, respectively. Mathematically, $\{\mathcal{L}_c\}_{c=1}^{C}$ and $\{\mathcal{L}_{cen}, \mathcal{L}_{div}\}$ are two different ways to partition $\mathcal{L}$. Combining the two partitions, we can further define $\mathcal{L}_{cen,c}$ and $\mathcal{L}_{div,c}$ as the samples of representativeness and divergence of class $c$, respectively. To satisfy the desiderata mentioned above, the optimization objective is proposed as in Eq. (9), in which $\alpha$ is a parameter to adjust the weight between typicalness and divergence of labeled data.

$$\min J(\mathcal{L}) = \alpha \sum_{x_i \in \mathcal{L}_{cen}} h(\gamma_i) + (1 - \alpha) \sum_{x_j \in \mathcal{L}_{div}} \frac{\rho_j}{|layer_j|},$$

$$s.t. \ |\mathcal{L}| = l,$$

$$|\mathcal{L}_c| \geq k,$$

(9)

where $|\cdot|$ is the cardinality operator, and $l$ the number of total samples to be labeled. $h(\gamma_i)$ is a decreasing function that is set as $\frac{1}{\log(\gamma_i)}$ in this study. $k$ is the number of nearest neighbor to learn a KLMCA projection whose value is typically set 2 or 3.

The rational of the objective is obvious. Assume the expenditures of labeling a sample in different classes are the same and amount of labeled samples $l$ are given, it would be reasonable to label more data points of a multi-modal class than that of single modal class. The design of right hand side of Eq. (9) can achieve this goal.

We investigate the objective and analyze the solution to it. $\gamma$ vector as a fully ordered parameter that indicates the potential of an object to be selected as a center, essentially reflect the most typical pattern of a class or a mode within a class. The second item that indicates how rare it is and how far a sample diverge from its center can also be regarded as a fully ordered vector. Therefore, to find a solution to Eq. (9), we only need to balance the two items of typicalness and divergence. The problem is solved through zero-order optimization. Because both $\gamma_i$ and $\frac{\rho_i}{|layer_i|}$ are nonnegative, the constraints are met by firstly choosing exactly $k$ samples in each class, then balancing the cardinality of $\mathcal{L}_{cen,c}$ and $\mathcal{L}_{div,c}$ according to parameter $\alpha$.

C. DeLaLA Algorithm

1) Selecting representative samples: Considering a sample to be selected to label can be either in $\mathcal{L}_{cen}$ or in $\mathcal{L}_{div}$ (but no way in both sets), the problem (9) can be solved with a continuous version of logical XOR operation. So, by defining $\psi_i = \frac{\rho_i}{|layer_i|}$, the objective is reformulated as:

$$\max J(\mathcal{L}) = \sum_{i=1}^{l} (\alpha h(\gamma_i)(1 - \psi_i) + (1 - \alpha)[1 - h(\gamma_i)]\psi_i),$$

$$s.t. \ |\mathcal{L}_c| \geq k.$$  

(10)

Remark: the $\gamma$, $\rho$ and layer parameters of a sample are coupled rather than independent, therefore choosing labeled samples must consider these parameters jointly.

The continuous XOR logical operation on two coupled arrays ($h(\gamma)$ and $\psi$ in this case) is implemented as in Algorithm 2.

2) Training and Applying KLMCA: We use KPCA [31] as the initial guess of $\Omega$ matrix, and then use Eq. (7) and (8) to update $\Omega$ until convergence. When applying the result of KMLCA, we first reduce the dimension of $\mathcal{U}$ to $p$ by computing $\tilde{\mathcal{U}} = \Omega\mathcal{K}_{l \rightarrow u}$, where $\mathcal{K}_{l \rightarrow u}$ is the kernel matrix between $\mathcal{L}$ and $\mathcal{U}$. The computing of $\mathcal{K}_{l \rightarrow u}$ can be accelerated by reusing the distance matrix $D$.

D. DaLaLA With Multi-metric Learning

For a large scale dataset with many classes, the mixmodal (that is, a cluster may contain samples of different classes) and multimodal (that is, samples of a class may form several clusters) situations usually exist. In this case, just one metric may not be adequate to capture the non-linear mapping between features and the label. Therefore, a multi-metric learning framework for DeLaLA is designed to learn multiple metrics for different subtrees (Algorithm 3). Because a subtree is indeed a local structure to reflect a particular mode of the samples, so the overall accuracy of classification is enhanced by exploiting this local structure, as demonstrated in the experiment studies. This approach shares the similar idea with [32].
Algorithm 1: DeLaLA Algorithm

Input: Dataset $\mathcal{X}$, number of class $C$, $k$ for LMCA, size of samples to be labeled $l$

Output: Samples to be labeled $\mathcal{L}$

// Stage 1: Initialization
1 Compute $\rho, \gamma$ using Eqs. (1), (3) and (4)
2 Compute layer $\psi$ for each $x_i$ (Algorithm 3)
3 $\psi = \left\{ \frac{\rho_z}{\text{layer}_z} \text{ for } x_j \in \mathcal{X} \right\}$
4 $h(\gamma) = \frac{1}{\log(\gamma)}$

// Stage 2: Select $l$ samples to label $S_{\text{Inds}} = \text{SelectC_Small}(h(\gamma), \psi, \text{rootW})$ (Alg. 2)
6 labeled$N = 0$
7 while labeled$N < l$
8 $i = \text{Pop}(S_{\text{Inds}})$
9 $c$ = the label of $x_i$
10 if $|\mathcal{L}_c| < k$ then
11 $\mathcal{L}_c = \mathcal{L}_c \cup \{x_i\}$
12 labeled$N = $ labeled$N + 1$
13 end
14 if $p > 0$ and $|\mathcal{L}_c| < p$ then
15 $\mathcal{L}_c = \mathcal{L}_c \cup \{x_i\}$
16 labeled$N = $ labeled$N + 1$
17 end
18 else
19 Skip $x_i$
20 end
21 end
22 end
24 $\mathcal{L} = \bigcup_{c=1}^{C} \mathcal{L}_c \cup \mathcal{L}_G$

// Stage 3: Train and apply KLMA to feed 1NN classification
25 Use $\mathcal{L}$ and $\mathcal{Y}$ to train a KLMA projection $P$
26 Classify $\mathcal{U}$ by applying 1NN w.r.t. $P(\mathcal{U})$ and $P(\mathcal{L})$

E. DaLaLA-Select as A Generic Preprocessing for SSLs

Stage 1 and Stage 2 in Algorithm 1 can be regarded as an independent module called DeLaLA-Select. This module can server as a general labeled samples selection method. By postponing the labeling operation after the structure of the target data is established, one can select the most representative samples to label. This strategy would significantly enhance the performance and stableness of most SSLs. We have verified this assumption via corresponding experiments (Section V-D).

IV. DISCUSSION

A. Relationship to Other Research Works

1) GSSL Perspective: DeLaLA constructs a leading forest that can be regarded as a graph to express the affinity between samples. But the difference lying behind is the structure can reflect the centrality and divergence with a class, so it is used to select instances to train the LMCA transformation. We can think the central and marginal samples describes an entire manifold given an adequately extensive sampling of a class. Therefore, it is sufficient to decide a sample’s label by applying low dimension projection and 1NN policy rather than a label propagation scheme.

2) Sparse LMCA Perspective: Traditional large margin metric learning techniques use full dataset to train, thus cause relatively low efficiency. DeLaLA selects very a few most representative samples to present all data, and achieves preferable performance. The idea is quite similar to the anchor family in GSSL [33] [11] [34].

3) Data Selection Perspective: In SSL field, feature selection appears more frequently than instance selection [15]. There are also some studies combining feature selection and instance selection in one framework [16]. DeLaLA differs from them in two ways. First, DeLaLA chooses instances via a partial-order leading forest structure, while others choose instances in a peer-to-peer neighborhood in the form of clusters or bins. Second, DeLaLA learns a kernelized lower dimension projection to achieve a large margin goal rather than selects the most relevant features.

4) An Enhanced Version of LaPOLeAF: The first model to employ leading forest in SSL is LaPOLeAF [6]. The advantage of LaPOLeAF lies in its high efficiency in label propagation. But due to random labeling strategy and the lack of metric learning, the accuracy attained by LaPOLeAF is just comparable. DeLaLA deterministically select most representative samples to train a large margin projection, thus the accuracy
is greatly improved over LaPOLeaF, and the high efficiency merit is retained.

B. Complexity analysis

The pipeline of DeLaLA consists of three major stages: a) OLF construction, b) Samples selection and labeling, c) KLMCA training, applying and 1NN classification.

Among the three stages, OLF construction costs the most time due to distance matrix requires \(O(n^2)\) complexity. However, because the computing of \(D\) can be transformed into matrix computation instead of computing each pairwise-distance \(d_{ij}\), the computing of OLF is fast in practice.

In samples selection and labeling, the most time-consuming part is using continuous XOR to select most representative samples because of the sort operation. The overall complexity in this stage is \(O(n \log n)\).

In the third KMLCA stage, the complexity of learning the projection matrix \(\Omega\) is \(O(T(dl + T^2))\), where \(T\) is the iteration number (typically \(10 < T < 100\)). Applying \(\Omega\) to \(K\) needs \(O(plu)\) complexity and the ultimate 1NN between \(U\) and \(L\) costs \(O(lpu)\).

Because the matrix computation has been made very efficient by modern data analysis languages such as Python, Matlab, etc., the entire DeLaLA is efficient in implementation when compared to other competing methods.

V. EXPERIMENTAL STUDIES

All the experiments run on a laptop with Intel i7-12700H CPU, 32G DDR5 RAM and 1TB SSD. An RTX 3060 laptop GPU is also available for fast computing of distance matrix. The programming language is Python 3.7.

A. Datasets Descriptions

To evaluate the performance of DeLaLA, we choose 19 datasets that are grouped according to their usage and characteristics: GP1={DS1, DS2, DS3}, GP2={DS4, ..., DS12}, GP3={DS13, ..., DS17}, GP4={DS18, DS19}. The details of the datasets are provided in Table I.

In GP1, Iris and Wine are small sized datasets with a class number of three. The two datasets can be classified directly using DeLaLA and superior accuracies are observed over the competing methods. Yeast data is also small sized, but with more classes. So it is classified with Multi-metrics-DeLaLA. The previous experimental results on datasets in GP1 are reported in [35] and [6]. The datasets in GP2 are characterized by small instance size and dimension with two or three classes. The two groups of GP2 and GP3 are studied in [4]. Dataset group GP3 contains two text datasets, one handwriting digits and two face datasets, among which Baseball and PC_mac need feature extraction preprocessing and the two face datasets require the multi-granularity iterative DeLaLA described in Algorithm 3. GP4 consists of the large sized datasets Letter and MNIST, which are widely adopted for empirical study in SSL research.

B. Competing methods

There are many semisupervised classification methods, we choose the classic models (LDA [36], NCA [37], SDA [38], LGC [39]), most relevant models (FLP [35], EAGR [11]) and latest methods (LaPOLeaF [6], AWSSL [4], OBGSSL [5]) as the competing methods. After describing the experimental results, we will discuss the strength and limitations of DeLaLA later in conclusion section.

C. DeLaLA Performance

1) Accuracy: We first compare the accuracy of DeLaLA on three UCI datasets of GP1 with the aforementioned competing methods, as shown in Table II.

TABLE II: Accuracy comparisons on the GP1 datasets. \(|\mathcal{L}_c| = 2\) for Iris and Wine; \(|\mathcal{L}| = 111\) for Yeast.

| Method  | Iris       | Wine       | Yeast      |
|---------|------------|------------|------------|
| LDA     | 66.91 ± 25.29 | 62.05 ± 12.67 | 26.00 ± 3.75 |
| NCA     | 92.28 ± 3.24  | 83.10 ± 9.70  | 35.78 ± 6.32 |
| SDA     | 89.41 ± 5.40  | 90.89 ± 5.39  | 39.03 ± 6.89 |
| LGC     | 91.13 ± 0.47  | 93.12 ± 0.24  | 43.25 ± 0.13 |
| FLP     | 93.45 ± 3.09  | 93.13 ± 3.32  | 40.03 ± 5.40 |
| EAGR    | 84.18 ± 2.75  | 90.47 ± 2.19  | 32.95 ± 2.50 |
| LaPOLeaF| 94.86 ± 4.57  | 91.01 ± 1.12  | 42.28 ± 2.36 |
| DeLaLA (Ours) | **96.53** | **97.09** | **51.49** |
We depict the DeLaLA-selecting and KLMCA training result of the datasets Iris and Wine in Fig. 3, from which one can clearly see the effectiveness of DeLaLA.

![Iris and Wine KLMCA 2D projection](image)

**Fig. 3:** The results of KLMCA 2D projection of the datasets Iris and Wine after selecting the samples with DeLaLA. The bigger markers with colored face are the labeled samples and the smaller markers without filled color represent the unlabeled samples.

The performance comparison between DeLaLA and others with regarding to the datasets in GP2 is shown in Table III. Apart from LGC, AWSSL, EAGR, the competing methods include GFHF [40], 1-semi [41], Semi-Supervised Low-Rank Representation (S^2LRR) [42], Semi-supervised Learning with adaptive neighbors (ANSSL, another version of AWSSL that does not consider the weight of data point). Among a total 24 times of comparison, DeLaLA wins the first place for 20 times. Even when it loses the first place, it still holds the second place (three times) or third place (once) with a small gap. Also, from the performance report, one can find out that DeLaLA constantly excels when the value of $\mathcal{L}_c$, i.e., the number of selected labeled samples per class, is small (say 2 or 3). And when $\mathcal{L}_c$ grows, the accuracy of DeLaLA does not improve accordingly. The possible reason is that if too many data are used to train the large margin projection, an over-fitting may have occurred.

In GP3, Baseball and PC_mac are two subsets in dataset 20_newsgroups. The Baseball dataset includes 1000 texts of topic “rec.sport.baseball” and 993 texts sampled from “comp.sys.mac.hardware”; and the PC_mac dataset includes 1000 text of topic “comp.sys.mac.hardware” and 943 texts sampled from “rec.sport.baseball”. Both Baseball and PC_mac are preprocessed using TF-IDF representation and obtain a vector of dimension 2000 for each text. DeLaLA produced much higher accuracy than the competing models on Baseball and PC_mac. Digits dataset is also directly classified with just one global metric and achieved superior accuracy. Because of the large number of class, multi-modal and mix-modal situations in ORL and Yale dataset, Algorithm 3 for multi-metric DeLaLA is applied to classify the two face datasets. The intermediate structures of the sub-trees in classifying Yale and ORL are depicted in Fig. 4 and Fig. 5, respectively. The performance comparison between DeLaLA and others on the datasets in GP3 is shown in Table IV.

For GP4, because of the relatively large size and many modes in each class, Algorithm 3 is applied. We perform three layers of OLF construction for each dataset. For Letter dataset, the first layer contains 45 subtrees, among which 34 have fewer classes and can be directly classified. Among the remaining 11 subtrees, only four have more than 10 classes. This makes the majority of subtrees complete the classification at the first or second layer. On the third layer, regardless of the number of classes of each subtree, we directly classify it. For MNIST dataset, the method is the same as with that of Letter. The number of subtrees is 385 and 1482 samples are labeled. The accuracy comparison with other methods on the two datasets is shown in Table V, from which we can see that DeLaLA obtained encouraging results.

**D. DeLaLA-Select for downstream SSLs**

As mentioned before, DeLaLA-Select can serve as a general policy to select the labeled data for SSLs. We evaluate the superiority of DeLaLA-Select over random selecting with two SSLs, namely, LGC and EAGR, on DS1 ~ DS12. The accuracy is considerably improved after DeLaLA-Select module is plugged in both LGC and EAGR methods as a preprocessing step, as shown in Table VI, from which we see that DeLaLA-Select can enhance the accuracy of LGC and EAGR on the twelve datasets by a large rate of over 7%.

**E. Running time**

We compare the running time of DeLaLA with LGC and EAGR on several datasets because LGC and EAGR are two typical efficient models of GSSLs. The datasets include Breast, ORL, Yale, Baseball, Digits, and Letter, for which the efficiency of single-metric and multi-metric DeLaLA are evaluated. Table VII shows the competitive efficiency of DeLaLA.

**F. The Parameters Selection**

There are four main parameters needs to be tuned in DeLaLA: bandwidth $\sigma$ in computing local density for constructing leading forest, target dimensionality $p$ in KLMCA projection, Length-Scale in computing Gaussian kernel matrix for KLMCA, and the centrality weight $\alpha$ in DeLaLA-Select. The parameters of single metric DaLaLA for DS1 ~ DS15 are given in Table VIII. Besides, the parameters in multi-metric DaLaLA for classifying the datasets Yeast, Yale and ORL are shown in Table IX.
### TABLE III: Accuracy (%) comparisons on DS4 ~ DS11.

| Datasets | $|L_c|$ | L1-semi | LGC | GFHF | $S^2$LRR | ANSSL | EAGR | AWSSL | DeLaLA (ours) |
|----------|------|--------|-----|------|---------|-------|------|-------|--------------|
| Breast   | 2    | 62.56  | 95.25 | 60.79 | 40.39   | 63.68 | 97.04 | 70.58 | 97.20        |
|          | 8    | 65.93  | 96.23 | 79.19 | 43.49   | 66.12 | 96.91 | 82.38 | 97.15        |
|          | 12   | 65.97  | 96.85 | 83.09 | 43.72   | 66.46 | 96.74 | 70.58 | 96.97        |
| Crx      | 3    | 44.44  | 74.45 | 36.92 | 46.55   | 47.64 | 45.37 | 51.1  | 86.38        |
|          | 6    | 46.96  | 75.28 | 54.23 | 51.36   | 51.20 | 45.31 | 64.75 | 83.13        |
|          | 9    | 49.06  | 77.81 | 63.14 | 57.88   | 50.46 | 45.27 | 69.05 | 82.65        |
| German   | 7    | 54.06  | 49.99 | 50.07 | 28.81   | 52.06 | 67.51 | 58.32 | 70.26        |
|          | 14   | 59.26  | 49.28 | 51.23 | 33.90   | 53.85 | 50.40 | 68.52 | 69.78        |
|          | 21   | 58.98  | 50.40 | 45.89 | 37.47   | 55.85 | 60.21 | 70.88 | 69.46        |
| Heart    | 2    | 50.34  | 66.69 | 56.82 | 48.88   | 53.27 | 55.27 | 64.39 | 80.08        |
|          | 3    | 54.58  | 63.97 | 64.50 | 50.04   | 61.81 | 75.65 | 70.23 | 79.55        |
|          | 4    | 55.81  | 65.29 | 67.05 | 51.89   | 58.52 | 76.75 | 70.16 | 79.01        |
| Ionosphere | 2   | 35.37  | 75.34 | 56.20 | 35.22   | 48.99 | 55.27 | 61.62 | 72.33        |
|          | 4    | 53.23  | 79.70 | 59.85 | 50.40   | 53.41 | 64.43 | 66.71 | 80.17        |
|          | 6    | 61.95  | 81.49 | 64.60 | 65.57   | 67.67 | 64.60 | 78.29 | 82.01        |
| Monkl    | 2    | 50.21  | 50.61 | 54.32 | 50.35   | 50.87 | 50.00 | 58.69 | 79.44        |
|          | 4    | 50.94  | 49.72 | 54.95 | 52.03   | 51.85 | 50.00 | 65.57 | 86.32        |
|          | 6    | 51.19  | 50.43 | 55.48 | 57.21   | 52.78 | 50.00 | 66.43 | 81.67        |
| Pima     | 5    | 34.70  | 65.02 | 48.88 | 34.28   | 48.41 | 60.13 | 52.64 | 68.36        |
|          | 10   | 43.99  | 65.54 | 53.46 | 47.68   | 53.86 | 61.34 | 61.10 | 65.24        |
|          | 15   | 59.08  | 66.25 | 56.40 | 49.08   | 51.11 | 62.16 | 65.72 | 65.58        |
| Vote     | 5    | 49.86  | 76.88 | 50.25 | 43.59   | 44.31 | 85.12 | 60.51 | 86.59        |
|          | 10   | 55.59  | 82.01 | 52.89 | 56.90   | 52.74 | 86.71 | 68.42 | 86.75        |
|          | 15   | 57.34  | 83.29 | 54.35 | 58.73   | 60.15 | 86.92 | 69.65 | 87.90        |

![Fig. 5: The diagram of multi-metric learning procedure with DeLaLA on the ORL face dataset. The meaning of MC is the same as in Fig. 4.](image)

The parameter sensitivity analysis is shown in Fig. 6. Combining Fig. 6 and Tables VIII and IX, one can tell that although the four parameters are influential to the accuracy, $\sigma$ and Length-Scale are more sensitive than $p$ and $\alpha$. We shall introduce the heuristics for tuning the parameters as follows.

1) $\sigma$ in leading tree: Typical $\sigma$ value lies around the 2% percentile smallest element in the distance matrix. If the micro-clusters are many in the same class, we need to investigate more into the local structure of neighbors. In this case, smaller
TABLE IV: Accuracy (%) comparisons on DS13 ∼ DS17.

| Datasets    | $|L_c|$ | L1-semi | LGC   | GFHF   | S$^2$LRR | ANSSL  | EAGR   | AWSSL  | DeLaLA(ours) |
|-------------|------|---------|-------|--------|----------|--------|--------|--------|--------------|
| Baseball    |      | 9       | 49.01 | 50.43  | 50.89    | 44.47  | 50.57  | 50.16  | 52.35        | 92.46        |
|             | 18   | 52.92   | 51.11 | 51.15  | 51.07    | 50.16  | 52.53  | 96.12  |              |              |
|             | 27   | 52.09   | 51.42 | 62.89  | 53.29    | 48.52  | 67.89  | 97.42  |              |              |
| PC_mac      |      | 9       | 48.94 | 50.74  | 50.41    | 49.55  | 49.71  | 48.52  | 49.97        | 93.25        |
|             | 18   | 50.49   | 51.23 | 51.11  | 50.55    | 50.45  | 48.48  | 52.18  | 97.54        |              |
|             | 27   | 51.14   | 51.32 | 51.42  | 50.86    | 50.99  | 48.44  | 52.68  | 97.94        |              |
| Digits      |      | 3       | 45.87 | 49.71  | 49.80    | 46.27  | 76.00  | 59.68  | 53.25        | 80.75        |
|             | 5    | 54.20   | 64.11 | 54.12  | 54.94    | 82.48  | 67.70  | 68.11  | 86.15        |              |
|             | 7    | 64.71   | 67.58 | 56.22  | 66.43    | 84.66  | 64.59  | 70.82  | 89.11        |              |
| ORL         |      | 4       | 90.42 | 92.92  | 91.58    | 93.67  | 89.42  | 78.55  | 94.17        | 95.09        |
| Yale        |      | 5       | 61.18 | 62.23  | 68.48    | 79.00  | 50.89  | 46.17  | 79.39        | 83.50        |

σ will be helpful. Although σ affects the ultimate accuracy when it changes in a large range, but it also remains robust to the result when it falls into an equivalent interval [6].

2) $p$ in KLMCA: The projected dimension in KLMCA sometimes affects the final result accuracy, but not frequently. Fig. 6b shows the accuracy changing along with $p$ increases on the datasets Wine, Hear and Vote.

3) Length-scale in computing kernel: The length scale in computing the kernel matrix for $X$ is denoted as $\tilde{\sigma}^2$, in which $\tilde{\sigma}$ is another bandwidth parameter whose value is set referring to the distance distribution in $D$. This parameter can decrease the accuracy considerably if improperly set. However, this parameter also shows robustness within a reasonable range (as shown in Fig. 6c).

4) $\alpha$ in selecting $L$: The $\alpha$ in selecting labeled samples reflects the balance between choosing most central samples (roots) and most divergent ones (leaves). The larger $\alpha$ implies that we tend to choose more leaves, and smaller $\alpha$ means that roots are preferable. The $\alpha$ will certainly affect the ultimate classification accuracy, but the extend limited. For example, the sensitivity of $\alpha$ on the three datasets (Wine, Iris and Breast) is illustrated in Fig. 6d.

VI. CONCLUSIONS

These paper proposed a method to deterministically select samples to label, taking into account the centrality and diversity revealed by the paths in a homeomorphic topological space, to train an SSL model. Because the most representative samples are selected and identified to describe a manifold of data distribution and a subsequent large margin projection is efficiently learned based on the small number of labeled samples, DeLaLA shows competitive efficiency and superior accuracy when compared to the state-of-the-art SSLs. Furthermore, the DeLaLA-Select module itself can be regarded as a preprocessing step for any SSL methods. Two downstream SSL methods (LGC and EAGR) are evaluated for this purpose.

TABLE V: Accuracy comparisons on Letter and MNIST datasets.

| Dataset  | Method | $|L|$ | Accuracy   |
|----------|--------|-----|------------|
| Letter   | LGC    | 988 | 81.46±0.59 |
|          | EAGR   | 988 | 82.81±0.67 |
|          | OBGSSL | 2000| 66.03±1.02 |
|          | DeLaLA(ours) | 980 | 84.62       |
| MNIST    | AGR    | 2000| 91.45±1.55 |
|          | EAGR   | 2000| 90.68±1.17 |
|          | OBGSSL | 2000| 92.56±0.62 |
|          | DeLaLA(ours) | 1842| 92.22       |

Fig. 6: Parameter sensitivity.
TABLE VI: DeLaLA-Select enhances the performance of downstream SSLs.

| Dataset ID | Name | $|L_c|$ | LGC Accuracy (%) | EAGR Accuracy (%) | Improve Rate |
|------------|------|------|------------------|-------------------|--------------|
| DS1        | Iris | 2    | 89.95 ±0.47      | 95.14 ±0.19      | 5.77         |
| DS2        | Wine | 2    | 93.12 ±0.24      | 95.93 ±0.19      | 3.02         |
| DS3        | Yeast| 2    | 33.41 ±0.36      | 29.37 ±0.19      | -12.1        |
| DS4        | Breast| 2  | 95.26 ±0.17      | 97.05 ±0.19      | 1.88         |
| DS5        | Crx | 3    | 74.45 ±1.86      | 86.42 ±0.19      | 14.54        |
| DS6        | German | 2  | 48.83 ±0.56      | 51.1 ±0.19       | 4.65         |
| DS7        | Heart| 2    | 66.69 ±1.10      | 79.70 ±0.19      | 19.51        |
| DS8        | Ionosphere | 5  | 74.45 ±1.05      | 83.57 ±0.19      | 12.25        |
| DS9        | Monkl| 10   | 50.40 ±0.08      | 58.91 ±0.19      | 18.67        |
| DS10       | Pima | 2    | 60.83 ±0.10      | 63.61 ±0.19      | 4.57         |
| DS11       | Vote | 2    | 72.02 ±1.29      | 81.21 ±0.19      | 12.76        |
| DS12       | Newthyroid | 3  | 85.69 ±0.78      | 87.56 ±0.19      | 2.18         |

Summary: Average Improve Rate (%) 7.31

TABLE VII: Running time (milliseconds for the first three datasets and seconds for the rest) comparisons on different size of datasets. The number below the dataset name in the second row is the value of $|L_c|$.

| Method | Breast | ORL | Yale | Baseball | Digits | Letter |
|--------|--------|-----|------|----------|--------|--------|
|        | 2      | 8   | 12   | 4        | 5      | 9      | 18    | 27    | 3      | 5      | 7      | 38    |
| LGC    | 375    | 380 | 371  | 372      | 121    | 5.78   | 5.98  | 5.77  | 2.62   | 2.73   | 2.74   | 2119.70 |
| EAGR   | 319    | 342 | 362  | 888      | 592    | 10.86  | 12.59 | 13.39 | 0.97   | 1.05   | 1.15   | 206.73  |
| DeLaLA | 133    | 967 | 968  | 634      | 1564   | 2.39   | 2.49  | 2.82  | 1.97   | 5.51   | 12.28  | 109.96  |

TABLE VIII: Parameter settings of DeLaLA (Single-Metric).

| ID | Name | #LT | $\sigma$ | $|L_c|$ | L | $\alpha$ | length-scale | $k$ | $p$ |
|----|------|-----|----------|-------|---|---------|--------------|----|----|
| DS1 | Iris | 6   | 0.1      | 2     | 6 | 0.5     | 0.7          | 1  | 2  |
| DS2 | Wine | 8   | 0.2      | 2     | 6 | 0.5     | 1.5          | 1  | 2  |
| DS4 | Breast | 8  | 0.1      | 2     | 4 | 0.5     | 1.8          | 1  | 2  |
| DS5 | Crx | 8   | 3       | 2     | 4 | 0.5     | 0.1          | 1  | 2  |
| DS6 | German | 8  | 1.1      | 2     | 6 | 0.5     | 1.5          | 1  | 2  |
| DS7 | Heart | 10  | 0.06     | 2     | 4 | 0.5     | 0.11         | 1  | 2  |
| DS8 | Ionosphere | 10   | 0.8      | 5     | 10 | 0.5     | 0.09         | 2  | 2  |
| DS9 | Monkl | 8   | 0.09     | 2     | 4 | 0.5     | 0.1          | 1  | 2  |
| DS10 | Pima | 8   | 0.3     | 5     | 10 | 0.5     | 3            | 1  | 3  |
| DS11 | Vote | 10  | 1       | 4     | 12 | 0.5     | 1.1          | 2  | 2  |
| DS12 | Newthyroid | 10  | 3     | 3     | 9 | 0.5     | 1            | 2  | 2  |
| DS13 | Baseball | 5   | 1.3     | 9     | 18 | 0.5     | 0.6          | 1  | 100 |
| DS14 | PC_mac | 5   | 1.3     | 9     | 18 | 0.5     | 0.54         | 1  | 100 |
| DS15 | Digits | 6   | 1.5     | 7     | 70 | 0.5     | 0.13         | 2  | 10 |

TABLE IX: Parameter settings of DeLaLA (Multi-Metric). $\tilde{C}$ denotes the number of class in a subtree.

| Name | SubsetID | #LT | $\sigma$ | $|L_c|$ | L | $\alpha$ | length-scale | $k$ | $p$ |
|------|----------|-----|----------|-------|---|---------|--------------|----|----|
| Breast | Layer0 | 5   | 0.3     | 3     | 27 | 0.5     | 2.5          | 2  | 14 |
| ORL | Layer1 | 3   | 0.3     | 2     | 6 | 0.5     | 2.5          | 1  | 14 |
| Digits | Layer2 | 3   | 0.3     | 2     | 8 | 0.8     | 6.0          | 0  | 14 |
| Letter | Layer3 | 3   | 0.3     | 2     | 8 | 0.8     | 6.0          | 1  | 14 |
| | Layer4 | 3   | 0.3     | 2     | 4 | 0.8     | 2.5          | 1  | 14 |
| | Layer5 | 3   | 0.18    | 2     | 6 | 0.8     | 0.18         | 1  | 3  |

ACKNOWLEDGMENT

This work has been supported by the National Key Research and Development Program of China under grant 2018AAA0101800, the National Natural Science Foundation of China under grant 61966005 and 61936001.
REFERENCES

[1] X. Zhu, “Semi-supervised learning literature survey,” Tech. Rep. 1530, 2008.

[2] J. E. Van Engelen and H. H. Hoos, “A survey on semi-supervised learning,” Machine Learning, vol. 109, no. 2, pp. 373–440, 2020.

[3] Z. Zhang, C. Xue, A. Han, J. Yang, and C. Gong, “Universal semi-supervised learning,” Advances in Neural Information Processing Systems, vol. 34, 2021.

[4] F. Nie, S. Shi, and X. Li, “Semi-supervised learning with auto-weighting feature and adaptive graph,” IEEE Transactions on Knowledge and Data Engineering, vol. 32, no. 6, pp. 1167–1178, 2020.

[5] F. He, F. Nie, R. Wang, H. Hu, W. Jia, and X. Li, “Fast semi-supervised learning with optimal bipartite graph,” IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 9, pp. 3245–3257, 2021.

[6] J. Xu, T. Li, Y. Wu, and G. Lapolleau, “Label propagation in an optimal leading forest,” Information Sciences, vol. 575, pp. 133–154, 2021.

[7] D. Wu, M. Shang, X. Luo, J. Xu, H. Yan, W. Deng, and G. Wang, “Self-training semi-supervised classification based on density peaks of data,” Neurocomputing, vol. 275, pp. 180–191, 2018.

[8] M. Wang, W. Fu, S. Hao, H. Liu, D. Tao, and X. Wu, “Scalable semi-supervised learning by efficient anchor graph regularization,” IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 7, pp. 1864–1877, 2016.

[9] Q. Zhang, T. Chu, and C. Zhang, “Semi-supervised graph based embedding with non-convex sparse coding techniques,” IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 5, pp. 2193–2207, 2021.

[10] J. Li and Y. Cui, “Semi-supervised learning: A comprehensive review,” IEEE Transactions on Neural Networks and Learning Systems, 2022.

[11] M. Wang, W. Fu, S. Hao, D. Tao, and X. Wu, “Scalable semi-supervised learning by efficient anchor graph regularization,” IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 7, pp. 1864–1877, 2016.

[12] Z. Song, X. Yang, Z. Xu, and I. King, “Graph-based semi-supervised learning: A comprehensive review,” IEEE Transactions on Neural Networks and Learning Systems, 2022.

[13] M. Wang, W. Fu, S. Hao, D. Tao, and X. Wu, “Scalable semi-supervised learning by efficient anchor graph regularization,” IEEE Transactions on Knowledge and Data Engineering, vol. 28, no. 7, pp. 1864–1877, 2016.

[14] Q. Zhang, T. Chu, and C. Zhang, “Semi-supervised graph based embedding with non-convex sparse coding techniques,” IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 5, pp. 2193–2207, 2021.

[15] M. Pini and C. Viral, “Domain adaptation on graphs by learning aligned graph bases,” IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 2, pp. 587–600, 2022.

[16] X. Zhu, J. Laflerty, and Z. Ghahramani, “Combining active learning and semi-supervised learning using gaussian fields and harmonic functions,” in ICML 2003 workshop on the continuum from labeled to unlabeled data in machine learning and data mining, vol. 3, 2003.

[17] R. Zhang and A. I. Rudnickiy, “A new data selection principle for semi-supervised incremental learning,” in 18th International Conference on Pattern Recognition (ICPR’06), vol. 2, pp. 780–783, IEEE, 2006.

[18] K. Benabdeselam, D. E. F. Kansouri, and R. Makkhongkaw, “Sos: Semi-supervised co-selection by a similarity preserving approach,” IEEE Transactions on Knowledge and Data Engineering, vol. 34, no. 6, pp. 2899–2911, 2022.

[19] C. C. Aggarwal, X. Kong, Q. Gu, J. Han, and S. Y. Philip, “Active learning: A survey,” in Data Classification, pp. 599–634, Chapman and Hall/CRC, 2014.

[20] L. Torresani and K.-c. Lee, “Large margin component analysis,” Advances in neural information processing systems, vol. 19, 2006.

[21] X. Geng, D.-C. Zhan, and Z.-H. Zhou, “Supervised nonlinear dimensionality reduction for visualization and classification,” IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 35, no. 6, pp. 1098–1107, 2005.

[22] K. Q. Weinberger and L. K. Saul, “Distance metric learning for large margin nearest neighbor classification,” Journal of machine learning research, vol. 10, no. 2, 2009.

[23] T. Wang and P. Isola, “Understanding contrastive representation learning through alignment and uniformity on the hypersphere,” in ICML, vol. 119 of Proceedings of Machine Learning Research, pp. 9929–9939, PMLR, 03–08 Jul 2020.

[24] M. Lopez-Martin, A. Sanchez-Eguilevillas, J. I. Arrabas, and B. Carro, “Supervised contrastive learning over prototype-label embeddings for network intrusion detection,” Information Fusion, vol. 79, pp. 200–228, 2022.

[25] M. Dong, Y. Wang, X. Yang, and J.-H. Xue, “Learning local metrics and influential regions for classification,” IEEE transactions on pattern analysis and machine intelligence, vol. 42, no. 6, pp. 1522–1534, 2020.

[26] J. Xu, G. Wang, T. Li, and W. Pedrycz, “Local-density-based optimal granulation and manifold information granule description,” IEEE transactions on cybernetics, vol. 48, no. 10, pp. 2795–2808, 2018.

[27] W. Pedrycz and W. Homenda, “Building the fundamentals of granular computing: A principle of justifiable granularity,” Applied Soft Computing, vol. 13, no. 10, pp. 4209–4218, 2013.

[28] Y.-F. Li and Z.-H. Zhou, “Towards making unlabeled data never hurt,” IEEE transactions on pattern analysis and machine intelligence, vol. 37, no. 1, pp. 175–188, 2014.

[29] A. Simard, R. Nowak, and J. Zhu, “Unlabeled data: Now it helps, now it doesn’t,” Advances in neural information processing systems, vol. 21, 2008.

[30] J. Xu, G. Wang, and W. Deng, “DenPEHC: Density peak based efficient hierarchical clustering,” Information Sciences, vol. 373, pp. 200–218, 2016.

[31] W. Tholen, “Ordered topological structures,” Topology and its Applications, vol. 156, no. 12, pp. 2148–2157, 2009.

[32] J. R. Munkres, Topology, vol. 2. Prentice Hall Upper Saddle River, 2000.

[33] B. Schölkopf, A. Smola, and K.-R. Müller, “Kernel principal component analysis,” in International conference on artificial neural networks, pp. 583–588, Springer, 1997.

[34] S. Hu, D. Miao, and W. Pedrycz, “Multi granularity based label propagation with active learning for semi-supervised classification,” Expert Systems with Applications, vol. 192, p. 116276, 2022.

[35] W. Liu, J. He, and S.-F. Chang, “Large graph construction for scalable semi-supervised learning,” in Proceedings of the 27th international conference on machine learning (ICML-10), pp. 679–686, 2010.

[36] M. Wang, W. Fu, S. Hao, H. Liu, and X. Wu, “Learning on big graph: Label inference and regularization with anchor hierarchy,” IEEE Transactions on Knowledge and Data Engineering, vol. 29, no. 5, pp. 1101–1114, 2017.

[37] B. Ni, S. Yan, and A. Kassim, “Learning a propagable graph for semisupervised learning: Classification and regression,” IEEE Transactions on Knowledge and Data Engineering, vol. 24, no. 1, pp. 114–126, 2012.

[38] P. Belhumeur, J. Hespanha, and D. Kriegman, “Eigenfaces vs. fisherfaces: recognition using class specific linear projection,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 19, no. 7, pp. 711–720, 1997.

[39] S. Roweis, G. Hinton, and R. Salakhudinov, “Neighbourhood component analysis,” Adv. Neural Inf. Process. Syst.(NIPS), vol. 17, no. 5, pp. 520, 2004.

[40] D. Cai, X. He, and J. Han, “Semi-supervised discriminant analysis,” in ICCV 2007, 2007.

[41] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, “Learning with local and global consistency,” in Advances in neural information processing systems, pp. 321–328, 2004.

[42] X. Zhu, Z. Ghahramani, and J. D. Lafferty, “Semi-supervised learning using gaussian fields and harmonic functions,” in Proceedings of the 20th International Conference on Machine Learning (ICML-03), pp. 912–919, 2003.

[43] F. Nie, H. Wang, H. Huang, and C. Ding, “Unsupervised and semi-supervised learning via n-norm graph,” in 2011 International Conference on Computer Vision, pp. 2268–2273, IEEE, 2011.

[44] C.-G. Li, Z. Lin, H. Zhang, and J. Guo, “Learning semi-supervised representation towards a unified optimization framework for semi-supervised learning,” in Proceedings of the IEEE International Conference on Computer Vision, pp. 2767–2775, 2015.
Gang Ren received the B.E degree from East China Jiaotong University, Nanchang, China in 2019. He is currently a graduate student with the State Key Laboratory of Public Big Data, Guizhou University. His current research interests include granular computing and machine learning.

Yao Xiao received the B.E degree from Shanxi University, Taiyuan, China in 2017. He is currently working toward the M.E degree with the State Key Laboratory of Public Big Data in Guizhou University. His research interests include machine learning and granular computing.

Shaobo Li received the Ph.D. degree in computer software and theory from the Chinese Academy of Sciences, China, in 2003. From 2007 to 2015, he was the Vice Director of the Key Laboratory of Advanced Manufacturing Technology of Ministry of Education, Guizhou University (GZU), China. He is currently the director of the State Key Laboratory of Public Big Data, GZU. He is also a part-time doctoral supervisor with the Chinese Academy of Sciences. He has published more than 200 papers in major journals and international conferences. His current research interests include big data of manufacturing and intelligent manufacturing.

Guoyin Wang received the B.S., M.S., and Ph.D. degrees from Xi’an Jiaotong University, Xi’an, China, in 1992, 1994, and 1996, respectively. He worked at the University of North Texas, USA, and the University of Regina, Canada, as a visiting scholar during 1998-1999. Since 1996, he has been working at the Chongqing University of Posts and Telecommunications, where he is currently the vice-president of the university. He was the President of International Rough Sets Society (IRSS) 2014-2017. He is the Chairman of the Steering Committee of IRSS and the Vice-President of Chinese Association of Artificial Intelligence. He is the author of 15 books, the editor of dozens of proceedings of international and national conferences, and has over 200 reviewed research publications. His research interests include rough set, granular computing, knowledge technology, data mining, neural network, and cognitive computing.