Semi-Supervised Ensemble Learning in the Framework of Data 1-D Representations with Label Boosting

Jianzhong Wang

College of Sciences and
Engineering Techonology
Sam Houston State University
Huntsville, TX 77341-2206, USA
Email: jzwang@shsu.edu

Huiwu Luo, Yuan Yan Tang

Faculty of Science and Technology
University of Macau, Macau, China
Email: luohuiwu@gmail.com
yytang@umac.mo

Abstract—**The paper introduces a novel ensemble method for semi-supervised learning. The method integrates the regularized classifier based on data 1-D representation and label boosting in a serial ensemble. In each stage, the data set is first smoothly sorted and represented as a 1-D stack, which preserves the data local similarity. Then, based on these stacks, an ensemble labeler is constructed by several 1-D regularized weak classifiers. The 1-D ensemble labeler extracts a newborn labeled subset from the unlabeled set. United with this subset, the original labeled set is boosted and the enlarged labeled set is utilized into the next semi-supervised learning stage. The boosting process is not stopped until the enlarged labeled set reaches a certain size. Finally, a 1-D ensemble labeler is applied again to construct the final classifier, which labels all unlabeled samples in the data set. Taking the advantage of ensemble, the method avoids the kernel trick that is the core in many current popular semi-supervised learning methods such as Transductive Supported Vector Machine and Semi-Supervised Manifold Learning. Because the proposed algorithm only employs relatively simple semi-supervised 1-D classifiers, it is stable, effective, and applicable to data sets of various types. The validity and effectiveness of the method are confirmed by the experiments on data sets of different types, such as handwritten digits and hyperspectral images. Comparing to several other popular semi-supervised learning methods, the results of the proposed one are very promising and superior to others.**

Keywords–Data smooth sorting; one-dimensional embedding; regularization; label boosting; ensemble classification; semi-supervised learning.

I. Introduction

In this paper, we introduce a novel ensemble method for semi-supervised learning (SSL) based on *data 1-D representation* and *label boosting*, which is abbreviated to ESSL1dLB. A preliminary discussion of the topic has been present in the conference presentation [1]. The purpose of this paper is to provide an extension with some new developments.

A standard SSL problem can be briefly described as follows: Assume that the samples (or members, points) of a given data set $X = \{\vec{x}_i\}_{i=1}^n \subset \mathbf{R}^m$ belong to c classes and $\mathcal{B} = \{b_1, \cdots, b_c\}$ is the class-label set. Let the labels of the samples in X be $y_1, y_2, \cdots, y_n$, respectively. When X is observed, only the samples of a subset, say, $X_\ell = \{\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_{n_0}\} \subset X$ have the known labels $Y_\ell = \{y_1, y_2, \cdots, y_{n_0}\} \subset \mathcal{B}$, while the labels of the samples in its complementary set $X_u = \{\vec{x}_{n_0+1}, \vec{x}_{n_0+2}, \cdots, \vec{x}_n\} = X \setminus X_\ell$ are unknown. A function $f : X \to \mathcal{B}$ is called a *classifier* (or *labeler*) if it predicts the labels for all samples in X_u. The *classification error* usually is measured by the number of the misclassified samples:

$$E(f) = |\{\vec{x}_i \in X|\ f(\vec{x}_i) \neq y_i,\ 1 \leq i \leq n\}|,$$

where $|S|$ denotes the cardinality of a set S. Then, the quality of a classifier is measured by the *classification error rate* (CErrRate) $E(f)/|X|$. The task of SSL is to find a classifier f with the CErrRate as small as possible.

In a SSL problem, if the samples of X only belong to two classes, say, Class A and Class B, it is called a *binary classification problem*. In this case, we may assign the sign-labels 1 and -1 to Classes A and B, respectively. In a binary classification problem, the error of a classifier f can be estimated by the ℓ_0 error:

$$E(f) = \sum_{k=1}^{n} \text{sign}(|f(\vec{x}_i) - y_i|).$$

It is worth to point out that the binary classification is essential in SSL. When the samples of X belong to more than two classes, we can recursively apply binary classification technique to achieve multi-classification [2], [3]. In a binary classification model, the classifier f on X usually is designed to a continuous real-valued function. The sign $f(\vec{x})$ then gives the class label of $\vec{x}$ so that the decision boundary is determined by the level-curve $f(\vec{x}) = 0$

SSL models make use some assumptions. The main one is the *smoothness assumption*, which asserts that the samples in the same class are similar while those in different classes are dissimilar. It enables us to design classifiers in a smooth function space, say, Soblev space. Its special case is the *cluster assumption*, which asserts that the data tend to form discrete clusters, and points in the same cluster are most likely in the same class. Clustering models are based on this assumption. When the dimension of data is high, due to the curse of dimensionality [4], [5], most computations on the data become inaccurate and unstable. According to the *manifold assumption*, the high-dimensional data lie approximately on a manifold of much lower dimension. Therefore, the dimensionality reduction technique should be utilized in SSL models.

Many statistical and machine learning methods for SSL were proposed in the last two decades. The monograph [6] and the survey paper [7] gave a comprehensive review of various SSL methods. Geometrically, the main difficult in SSL is the nonlinearity of the decision boundary. In general, it is a combination of several disjoint surfaces in $\mathbf{R}^m$. To overcome the difficult, many popular methods, such as transductive support vector machines, manifold regularization, and various graph-based methods, utilize so-called *kernel trick* to linearize the decision boundary [8], [9]. That is, in such a model, with the help of a kernel function, one constructs a reproducing kernel Hilbert space (RKHS) [10], where the classifier is a linear function so that it can be constructed by a regularization method. The success of a kernel-based method strongly depends on the exploration of data structure by the kernel. However, it is often difficult to design suitable kernels, which precisely explore the data features. Therefore, recently researchers try to establish new SSL models, in which classifiers are constructed without kernel technique. These models include the data-tree based method [11], [12], SSL using Gaussian fields [13], and others.

In most of the models above, a single classifier is constructed for a given SSL task. However, when a data set has a complicate intrinsic structure, a single classifier usually cannot complete the task satisfactorily. The *multiple classifier systems* (MCSs) offer alternatives. The ensemble methodology in MCSs builds a single *strong classifier* by integrating several *weak classifiers*. Although each weak classifier is slightly correlated with the true classification, the strong classifier is well-correlated with the true one. MCSs perform information fusion at different levels to overcome the limitations of the traditional methods [14]–[16]. In MCSs two canonical topologies, parallel and serial ones, are employed in the ensemble (see Fig. 4 in [15]). In the parallel topology, all weak classifiers are built on the same data set and the strong classifier is made by a combination of their outputs. On the other hand, in the serial topology, the weak classifiers are applied in sequence, such that the output of the predecessor turns to be the input of the successor, and the final label prediction comes from the last weak classifier. Originally, ensemble algorithms are developed for supervised learning. A well-known parallel ensemble algorithm is bagging (bootstrap aggregating) [14], [17]. Boosting algorithms, such as AdaBoost [18], LPBoost, LSBoost, RobustBoost, and GentleBoost, apply serial ensemble. Due to the flexibility, MCSs open a wide door for developing various ensemble SSL algorithms.

The novelty of the introduced ensemble SSL method is the following: It adopts the framework of data 1-D representation, in which the data set is represented by several different 1-D sequences, then a labeler is constructed as an ensemble of several weak classifiers, which are built on these 1-D sequences. Here, we are partial to data 1-D models because 1-D decision boundary reduces to a set of points on a line, which has the simplest topological structure. As a result, the weak classifiers can be easily constructed by standard 1-D regularization methods without using kernel trick. Furthermore, the simplicity of 1-D models makes the algorithm more reliable and stable. Hence, the core of our method is an ensemble binary classification algorithm for SSL, whose architecture and technological process are described in the following.

1) **Making data 1-D (shortest path) representation.** The data set X first is smoothly sorted and mapped to several 1-D sets $\{T^i\}_{i=1}^k$, of which each preserves the local similarity of members in X. Correspondingly, the couple $\{X_\ell, X_u\}$ is mapped to $\{T_\ell^i, T_u^i\}$ such that $T_\ell^i \cup T_u^i = T^i$. The 1-D sets $\{T^i\}_{i=1}^k$ provide a framework of our method.
2) **Constructing ensemble labeler in the 1-D framework.** Based on T^i, a weak classifier g^i on X is constructed by a 1-D regularization method. Then an ensemble labeler is built from these weak classifiers. From the unlabeled set X_u, the labeler extracts a *feasibly confident subset* L, which contains the samples, whose predicted labels are accurate with high confidence.
3) **Developing label boosting algorithm.** A label selection function is constructed to select the *newborn labeled subset* S from the feasibly confident subset L to reduce the misclassification error. It computes *class weights* of the members of L for selection decision. Then, the initial labeled set X_ℓ is boosted to $X_\ell^{new} = X_\ell \cup S$. The process is repeated and not terminated until the boosted labeled set X_ℓ^{new} reaches a certain size.
4) **Building the final (strong) classifier.** Finally, several weak classifiers g^i are constructed based on the final updated labeled set X_ℓ^{new}. The final classifier f is defined as the mean of these weak classifiers.

Our strategy in the binary classification algorithm above adopts *Model-guided Instance Selection* approach to boosting [14]. But it is slightly different from the standard boosting algorithms [19] in the sense that they boost the misclassified weights on X_u, while our method boosts the labeled subset X_ℓ. The preliminary work of the proposed method can be found in [20]–[23].

In this paper, we employ the well-known *One-Against-All* strategy [2] to deal with multi-classification using a combination of binary classifications.

The paper is organized as follows: In Section II, we introduce our ensemble SSL method in details and present the corresponding ESSL1DLB algorithm. In Section III, we demonstrate the validity of our method in examples and give the comparison of our results with those obtained by several popular methods. The conclusion is given in Section IV.

II. Ensemble SSL Method in Framework of Data 1-D Representation with Label Boosting

In this section, we introduce the novel ensemble SSL method based on data 1-D representation and label boosting. The main steps of the method has been introduced in the previous section. We now introduce the method and corresponding algorithm in details.

A. Data 1-D representations by shortest path sorting

Assume that the data set X is initially arranged in a stack $\mathbf{x} = [\vec{x}_1, \cdots, \vec{x}_n]$. Let $w(\vec{x}, \vec{y})$ be a distance-type weight function on $X \times X$ that measures the dissimilarity between the samples $\vec{x}$ and $\vec{y}$. Let π be an index permutation of the index sequence $[1, 2, \cdots, n]$, which induces a permutation P_π on the initial stack $\mathbf{x}$, yielding a stack of X headed by $\vec{x}_{\pi(1)}$: $\mathbf{x}_\pi = P_\pi \mathbf{x} = [\vec{x}_{\pi(1)}, \cdots, \vec{x}_{\pi(n)}]$. We denote the set of all permutations of X with the head $\vec{x}_\ell$ by

$$\mathcal{P}_\ell = \{P_\pi; \quad \pi(1) = \ell\}.$$

According to [24], the *shortest-path sorting of X headed by $\vec{x}_\ell$* is the stack $\mathbf{x}_\pi$ that minimizes the path starting from $\vec{x}_\ell$ and though all points in X, i.e., $\mathbf{x}_\pi = P_\pi \mathbf{x} = [\vec{x}_{\pi(1)}, \vec{x}_{\pi(2)}, \cdots, \vec{x}_{\pi(n)}]$, where

$$P_\pi = \underset{P \in \mathcal{P}_\ell}{\arg\min} \sum_{j=1}^{n-1} w((P\mathbf{x})_j, (P\mathbf{x})_{j+1}). \tag{1}$$

Define the 1-D sequence $\mathbf{t} = [t_1, \cdots, t_n]$ by

$$t_1 = 0, \ t_{j+1} - t_j = \frac{w(\vec{x}_{\pi(j)}, \vec{x}_{\pi(j+1)})}{\sum_{k=1}^{n-1} w(\vec{x}_{\pi(k)}, \vec{x}_{\pi(k+1)})}. \tag{2}$$

Then, the 1-D stack $\mathbf{t}$ provides the *1-D (shortest-path) representation* of X headed by $\vec{x}_\ell$. We call the function $h_\ell : X \to [0,1], h_\ell(\vec{x}_{\pi(j)}) = t_j$ the (isometric) *1-D embedding* of X headed by $\vec{x}_\ell$. When the index ℓ is not stressed, we will simplify h_ℓ to h.

The sorting problem (1) essentially is a traveling salesman problem, which has NP computational complexity. To reduce the complexity, approximations of P_π in (1) are adopted. For instance, a greedy algorithm for the approximation was developed in [24] for sorting all patches in an image. In this paper, we slightly modify the algorithm in [24] so that it works for data sets that are represented by weighted data graphs. We first see how to construct weighted graphs for two popular types of data sets:

1) The data set forms a point cloud $X \subset \mathbf{R}^m$ equipped with a metric. To construct a weighted graph $[X, E, W]$ on X, we identify a point $\vec{x}$ with a node in the graph so that X can be considered as a set of nodes in the graph. By the metric on X, we derive a distance-type weight function $d(\vec{x}, \vec{y})$ on $X \times X$, which measures the dissimilarity between the samples in X. For any $\vec{x} \in X$, the k nearest neighbors (kNN) of the node $\vec{x}$ is denoted by $\mathcal{N}_{\vec{x}} \subset X$. Assume that $|X| = n$, and set $\mathcal{I} = \{1, 2, \cdots n\}$. Then the edge set in the graph is $E = \{(i,j) \in \mathcal{I} \times \mathcal{I};\ \vec{x}_j \in \mathcal{N}_{\vec{x}_i}\}$. Finally, we define the weight matrix $W = [w_{ij}]_{i,j=1}^n$, where $w_{i,j} = d(\vec{x}_i, \vec{x}_j)$.
2) The data set is a hyperspectral image (HSI) represented as an imaginary cube $X \in \mathbf{R}^{m \times n \times s}$. In the cube, the (i,j)-pixel, $X(i,j,:)$, is a spectral vector; and the spatial neighbors of $X(i,j,:)$ usually is defined as the pixel-square centered by $X(i,j,:)$: $\mathcal{N}_{(i,j)} = \{X(k,l,:);\ |k-i| \le q, |l-j| \le q\}$, where q is a preset positive integer. For a given HSI cube X, we construct the weighted graph $[X, E, W]$ as follows: We first map the double index (i,j) to the single one $k = i + (j-1)m$. Then we write $\vec{x}_k = X(i,j,:)$ and convert the 2-D neighborhood to the 1-D one: $\mathcal{N}_{\vec{x}_k} = \mathcal{N}_{(i,j)}$, which defines the edge set E. For a HSI data set, there are various ways to define the distance-type weights on edges. We propose the spectral-spatial weights introduced in the paper [22]. Similar to the first case, the node set $X = \{\vec{x}_k;\ 1 \le k \le nm\}$, the edge set E and the weight set W form a weight graph $[X, E, W]$ on the HSI cube X.enumerate

Note that the edge set E in the graph $[X, E, W]$ induces an index neighbor set from a node neighbor. For instance, the index set $\mathcal{N}_k = \{j;\ (k,j) \in E\}$ is corresponding to the neighbor set $\mathcal{N}_{\vec{x}_k}$. Using index neighbors to replace the node neighbors can simplify code writing. If the graph $[X, E, W]$ is complete, then the neighbor set of each node $\vec{x}$ is the set $X \setminus \{\vec{x}\}$, which leads to the global search scheme in the greedy algorithm.

Adopting weighted data graph $[X, E, W]$ as the input of the greedy sorting algorithm enables us to apply the algorithm to various data sets. For instance, it can be applied to the data set X, whose samples cannot be digitally represented by vectors, but the similarity between them can be measured. For this type of data, although X is not digitized, the algorithm works. Many data sets obtained by social survey are in this category.

The pseudocode of our data 1-D (shortest-path) representation (1dSPR) algorithm is presented at **Algorithm 1,** in which ϵ is called the *path selection parameter*. Since the algorithm is a slight modification of that one in [24], we omit the details of the explanation of the parameter settings here.

Algorithm 1 1dSPR Algorithm

```
Require: Data graph $[X, E, W]$; probability vector $\tilde{\mathbf{p}} = [\tilde{p}_1, \tilde{p}_2, \cdots, \tilde{p}_n]$, where $\tilde{p}_i \in (0,1)$, and $n = |X|$.
Initialization of Output: $\pi$: empty index stack; $\mathbf{t}$: $n$-dimensional zero vector.
Set $\pi(1) \leftarrow j$, $j$: random index; and set $t_1 = 0$.
Define $\mathcal{I} = \{1, 2, \cdots, n\}$.
for $k = 1, 2, \cdots, n-2$ do
    • set $\mathcal{N}^c_{\pi(k)} = \mathcal{N}_{\pi(k)} \setminus \pi$; $\mathcal{I}^c = \mathcal{I} \setminus \pi$
    • if   $|\mathcal{N}^c_{\pi(k)}| = 1$
      —  $\pi(k+1) \leftarrow j \in \mathcal{N}^c_{\pi(k)}$
    • else
      — if   $|\mathcal{N}^c_{\pi(k)}| \ge 2$
            * Find $j_1 \in \mathcal{N}^c_{\pi(k)}$ such that $\vec{x}_{j_1}$ is the nearest neighbor to $\vec{x}_{\pi(k)}$ in $\mathcal{N}^c_{\vec{x}_{\pi(k)}}$
            * Find $j_2 \in \mathcal{N}^c_{\pi(k)}$ such that $\vec{x}_{j_2}$ is the second nearest neighbor to $\vec{x}_{\pi(k)}$ in $\mathcal{N}^c_{\vec{x}_{\pi(k)}}$
      —  elseif   $|\mathcal{N}^c_k| = 0$
            * Find $j_1 \in \mathcal{I} \setminus \pi$ such that, in all nodes with indices in $\mathcal{I} \setminus \pi$, $\vec{x}_{j_1}$ is the nearest node to $\vec{x}_{\pi(k)}$
            * Find $j_2 \in \mathcal{I} \setminus \pi$ such that, in all nodes with indices in $\mathcal{I} \setminus \pi$, $\vec{x}_{j_2}$ is the second nearest node to $\vec{x}_{\pi(k)}$
      —  endif
    • endif
    Compute $q_k$:
        $$q_k = \frac{1}{1 + \exp\left(\frac{w(\vec{x}_{\pi(k)}, \vec{x}_{j_1}) - w(\vec{x}_{\pi(k)}, \vec{x}_{j_2})}{\epsilon}\right)} \tag{3}$$
    Set $\pi(k+1) = \begin{cases} j_2 & \text{if } q_k < \tilde{p}_{\pi(k)} \\ j_1 & \text{otherwise.} \end{cases}$
    Set $t_{k+1} = t_k + w(\vec{x}_{\pi(k)}, \vec{x}_{\pi(k+1)})$.
end for
Set $\pi(n) \leftarrow j \in \mathcal{I} \setminus \pi$, $t_n = t_{n-1} + w(\vec{x}_{\pi(n-1)}, \vec{x}_{\pi(n)})$.
Normalize vector $\mathbf{t}$: $t_j \leftarrow t_j / t_n$
Ensure: $\mathbf{t}$; $\pi$.
```

Because sorting scheme is a serial process, it is a bias in the sense of smoothness. That is, in general the difference $\Delta t_j =$

$t_{j+1} - t_j$ is increasing with respect to j, i.e., earlier selected adjacent pairs are more similar than the later selected ones. This bias impacts cluster preserving when X is represented by T.

Enlightened by the spinning technique [25], we introduce multiple 1-D embedding of X to reduce the sorting dias. Let $\{\vec{x}_{j_1}, \vec{x}_{j_2}, \cdots, \vec{x}_{j_k}\}$ be a subset of X selected at random, and h_i be the 1-D embedding headed by $\vec{x}_{j_i}$. We call k the *spinning number* and the vector-valued function $\vec{h} = [h_1, h_2, \cdots, h_k]$ a k-ple 1-D embedding of X. Then $\vec{h}(X)$ gives a k-ple 1-D representation of X.

B. Construction of ensemble labeler from weak classifiers on data multiple 1-D representation

Let h be a (single) 1-D embedding of X (with $|X| = n$). We write $T_\ell = h(X_\ell)$ and $T_u = h(X_u)$. Then, T_ℓ is a labeled set and T_u is an unlabeled set, and a labeler on T induces a labeler on X. Since T is a 1-D set, its class decision boundary is reduced to a point set in the line segment $[0, 1]$. Therefore, no kernel trick is needed for constructing a labeler on T. Instead, a simple regularization scheme works.

As we mentioned above, a single 1-D representation may not truly preserve the data similarity because the sorting bias. In the proposed method, we create a multiple 1-D representation of X, and construct a weak labeler based on each of them. Then, from these weak labelers, we build an ensemble labeler (1dEL), which better predicts the labels of the samples in the unlabeled set X_u. The following is the details of **1dEL** Algorithm.

Let $\vec{h} = [h_1, \cdots, h_k]$ be a k-ple 1-D embedding of X with the head stack $[\vec{x}_{j_1}, \vec{x}_{j_2}, \cdots, \vec{x}_{j_k}]$. Let P_i be the permutation operator corresponding to h_i and $\mathbf{x}_{\pi_i} = P_i\mathbf{x}$. The embedding h_i produces a 1-D representation of X: $\mathbf{t}^i = h_i(\mathbf{x}_{\pi_i})$, and any function f on X through h_i derives a function $s^i = f \circ h_i^{-1}$ on $\mathbf{t}^i$. Equivalently, $f = s^i \circ h_i$, which given the relation of a labeler on $\mathbf{t}^i$ and a labeler on X. Since $\mathbf{t}^i$ is a discrete set, we can represent the function s^i on $\mathbf{t}^i$ in the vector form $\mathbf{s}^i = [s^i_1, \cdots, s^i_n]$, where $s^i_j = s(t^i_j)$.

To construct the labelers on $\mathbf{t}^i$, we define the first-order difference of s^i (at t^i_j) by $\Delta s^i_j = s(t^i_{j+1}) - s(t^i_j)$, and the first-order difference quotient by $Ds_j = (s(t^i_{j+1}) - s(t^i_j))/(t^i_{j+1} - t^i_j)$. Inductively, we define the k^{th}-order difference of s^i (at t^i_j) by $\Delta^k s^i_j = \Delta^{k-1}s^i_{j+1} - \Delta^{k-1}s^i_j$ and the k^{th}-order difference quotient by $D^k s^i_j = (D^{k-1}s^i_{j+1} - D^{k-1}s^i_j)/(t^i_{j+k} - t^i_j)$. They describe various smoothness of s^j. Let $T^i_\ell = h_i(X_\ell)$ and $T^i_u = h_i(X_u)$. As we have mentioned, a weak labeler g^i on X can be constructed as the composition $g^i = q^i \circ h_i$, where q^i is a labeler on $\mathbf{t}^i$. We construct q^i using one of the following 1-D regularization models:

1. Least-square regularization. Let q^i be the solution of the following unconstrained minimization problem:

$$q^i = \arg\min \frac{1}{n_0}\sum_{j=1}^{n_0}\left(s^i(h_i(\vec{x}_j)) - y_j\right)^2 + \frac{\lambda}{2}\sum_{j=1}^{n-1}(Ds^i_j)^2, \quad (4)$$

where λ is the standard *regularization parameter*. We denote by I_{n_0} the $n \times n$ diagonal matrix, in which only $(\pi^i(j), \pi^i(j))$-entries are 1, $1 \le j \le n_0$, but others are 0. Set $w_0 = w_n = 0, w_j = 1/(t^i_{j+1} - t^i_j)^2$, and denote by $D = [D_{i,j}]$ the $n \times n$ three-diagonal matrix, in which

$$\begin{cases} D_{j,j} = w_{j-1} + w_j & 1 \le j \le n, \\ D_{j,j+1} = D_{j+1,j} = -w_j & 1 \le j \le n-1, \end{cases}$$

Then, the vector representation of q^i on the stack $\mathbf{t}^i$ is

$$\mathbf{q}^i = \left(I_{n_0} + n_0\lambda D\right)^{-1}\vec{y}. \quad (5)$$

Assume that the class distribution on X_u is the same as on X_ℓ. Let $M = \frac{1}{n_0}\sum_{j=1}^{n_0} y_j$. Then we may add the constraint

$$\frac{1}{n}\sum_{j=1}^{n} s^i(t^i_j) = M$$

to the minimization problem (4). Correspondingly, the solution (5) is modified to

$$\mathbf{q}^i = \left(I_{n_0} + n_0\lambda D\right)^{-1}\left(\vec{y} + \mu\vec{1}\right) \quad (6)$$

with

$$\mu = \frac{M - \mathcal{E}\left((I_{n_0} + n_0\lambda D)^{-1}\vec{y}\right)}{\mathcal{E}\left((I_{n_0} + n_0\lambda D)^{-1}\vec{1}\right)},$$

where $\vec{1}$ denotes the vector whose all entries are 1 and $\mathcal{E}(\vec{v})$ the mean value of the vector $\vec{v}$.

Remark: In the Least-square regularization model (4), the difference quotient term $(Ds^i_j)^2$ may also be replaced by the difference term $(\Delta s^i_j)^2$. This replacement is equivalent to using the equal-distance sequence $\mathbf{t}^i$ in (4).

2. Regularization by interpolation. Let q^i be the solution of the following constrained minimization problem:

$$q^i = \arg\min \sum_{j=1}^{n-2}(D^2 s^i_j)^2 \quad (7)$$

subject to

$$s^i(h_i(\vec{x}_j)) = y_j, \quad 1 \le j \le n_0. \quad (8)$$

Write $\hat{t}_j = h_i(\vec{x}_j)$. Then q^i is the cubic spline that has the nodes at $\{\hat{t}_j\}_{j=1}^{n_0}$ and takes the values as in (8).

Let i run through 1 to k. Then we obtained k 1-D labelers $q^1, \cdots, q^k$. They further derive k weak labelers $g^i = q^i \circ h_i^{-1}, 1 \le i \le k$, on X. We will use $[g^1, \cdots, g^k]$ in two cases. Firstly, in the label boosting precess, they are used to construct the *feasibly confident subset*. Recall that each g^i predicts the label $\text{sign}(g^i(\vec{x}))$ for $\vec{x} \in X_u$. Let

$$g(\vec{x}) = \frac{1}{k}\sum_{i=1}^{k}\text{sign}(g^i(\vec{x})), \quad \vec{x} \in X_u, \quad (9)$$

and define

$$L^+ = \{\vec{x} \in X_u;\ g(\vec{x}) = 1\}, \quad L^- = \{\vec{x} \in X_u;\ g(\vec{x}) = -1\}.$$

Then we call L^+ the *feasibly confident subset of Class A*, L^- the *feasibly confident subset of Class B*, and $L = L^+ \cup L^-$ the *feasibly confident subset*. In a great chance, a sample in L^+ is in Class A, while a sample in L^- in Class B. For convenience, we denote the set operator that create the feasibly confident subset L from X_u by $\mathbf{G} : \mathbf{G}(X_u) = L$.

Secondly, we use $[g^1, \cdots, g^k]$ to construct the final classifier f in the last step of our algorithm as follows:

$$f(\vec{x}) = \frac{1}{k}\sum_{i=1}^{k} g^i(\vec{x}), \quad \vec{x} \in X_u. \tag{10}$$

C. Label boosting

To further eliminate the misclassification in L^+ and L^-, we will construct a subset $S^+ \subset L^+$ and a subset $S^- \subset L^-$ as follows: Let $X_\ell^+ \subset X_\ell$ be the subset that contains all Class-A members and $X_\ell^- \subset X_\ell$ the subset that contains all Class-B members. For each $\vec{x} \in L$, define

$$w^+(\vec{x}) = \frac{\sum_{\vec{y} \in X_\ell^+} w(\vec{x}, \vec{y})}{|X_\ell^+|}$$

and

$$w^-(\vec{x}) = \frac{\sum_{\vec{y} \in X_\ell^-} w(\vec{x}, \vec{y})}{|X_\ell^-|}.$$

We now create the *class weight* function by

$$w(\vec{x}) = \frac{w^+(\vec{x})}{w^+(\vec{x}) + w^-(\vec{x})}.$$

It is obvious that a greater value of $w(\vec{x})$ indicates that $\vec{x}$ is nearer the points in X_ℓ^-. Therefore, it is more likely in Class B. Let the set S^+ contain the half of members of L^+ with the smallest class weights and S^- contain the half of members in L^- with the greatest class weights. We call $S = S^+ \cup S^-$ the *newborn labeled subset*, call the operator $\mathbf{S} : \mathbf{S}(L) = S$ a *newborn labeled subset selector*, and call the composition $\mathbf{M} = \mathbf{S} \circ \mathbf{G}$ a *newborn labeled subset generater*. Therefore, we have the newborn labeled set $S = \mathbf{S}(L) = \mathbf{S}(\mathbf{G}(X_u)) = \mathbf{M}(X_u)$.

The *Label Boosting Algorithm* iteratively adds the newborn labeled subset to the original labeled set so that the labeled set is cumulatively boosted. In detail, let the initial labeled set x_ℓ and the unlabeled set $X - u$ be re-written as X_ℓ^0 and X_u^0, respectively. We apply the newborn labeled subset generater $\mathbf{M}_1$ on X_u^0 to create a newborn labeled set $S^1 = \mathbf{M}_1(X_u^0)$, which is united with X_ℓ^0 to produce $X_\ell^1 = X_\ell^0 \cup S^1$. Meanwhile, we set $X_u^1 = X_u^0 \setminus S_1$. Repeating the procedure for N times, the labeled set will be cumulatively boosted to a enlarged labeled set

$$X_\ell^N = X_\ell^0 \bigcup_{j=1}^{N} S_j, \quad S_j = \mathbf{M}_j(X_u^{j-1}). \tag{11}$$

We set a *boosting-stop parameter* $p, 0 < p < 1$. The process will not be terminated until the labeled set X_ℓ^N reaches the size $|X_\ell^N| \geq p|X|$. We call N the *label boosting times*.

D. Construction of the final classifier

Finally, we apply 1dEL algorithm on the couple $\{X_\ell^N, X_u^N\}$ to construct the final classifier f by (10). Then each $\vec{x} \in X$ is labeled by $\operatorname{sign} f(\vec{x})$.

The whole algorithm that creates the final classifier f is called **ESSL1dLB**.

E. One-Against-All strategy for multi-classification

Many strategies are proposed in literature for handling multi-classification using binary ones [26]–[29]. We apply the well-known *One-Against-All* strategy for multi-classification tasks [2], [30], [31]. In the paper, we choose the simplest one, which is briefly described in the following:

Assume that X consists of c-classes ($c > 2$): Class 1 to Class c. Using **ESSL1dLB,** we create c binary classifier $\{f_1, f_2, \cdots, f_c\}$, where f_i classifies two classes: Class A is identical with Class i, and Class B contains all of other classes, as we described above. A simple one-vs-all classification strategy is the following: Let f be the multi-classifier. Then

$$f(\vec{x}) = \arg\max_{1 \leq i \leq c} f_i(\vec{x}).$$

III. Experiments on Hyperspectral Images

In this section, we evaluate our ensemble SSL method in the experiments on hyperspectral images. An earlier method in the ensemble SSL framework for the classification of hyperspectral images has been reported in [22], where we used the interpolation splines as 1-D weak labelers (see (7)) and adopted the following simpler label boosting method: Choosing the newborn labeled subset at random. The obtained results are still very promising and superior over many other popular methods. In this section, we apply **ESSL1dLB** algorithm for the multi-classification of hyperspectral images. There are two main differences between **ESSL1dLB** and the algorithm used in [22]: Firstly, the **ESSL1dLB** algorithm uses Least-square regularization for the construction of weak labelers (see (4)). Secondly, it uses the class-weight method for label boosting.

In this section, we first introduce the data formats and the metrics of the data sets used in our experiments. Then we tune the parameters in the **ESSL1dLB** algorithm. Finally, we report the results of the experiments and comparisons.

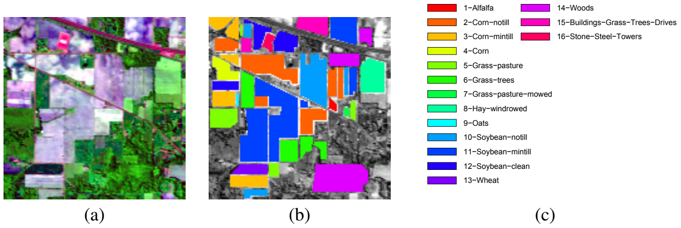


Figure 1. RGB composition and classification map for AVIRIS Indian Pines 1992 scenario. (a) Pseudocolor image. (b) Ground truth map. (c) Class labels.

A. Data Collection and Experiment Design

All of the data sets used in the experiments are published for research usage only. Three hyperspectral images are chosen for our experiments, which are particularly designed.

1) Data sets: The first data set used in our experiments is the *AVIRIS Indian Pines* 1992, which was gathered by the National Aeronautics and Space Administration's (NASA) Airborne Visible/Infrared Imaging Spectrometer (AVIRIS) sensor over the northwestern Indian Pines test site in 1992. The raw calibrated data are available on-line from [32] with the ground-truth class map. This data set consists of 145×145 pixels and 224 spectral reflectance bands in the wavelength range

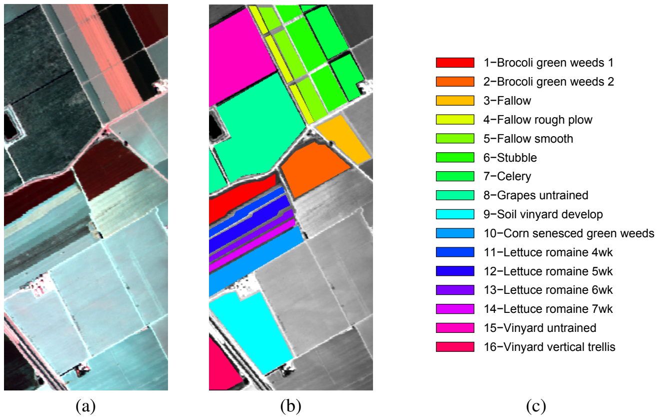


Figure 2. RGB composition and classification map for Salinas scenario. (a) Pseudocolor image. (b) Ground truth map. (c) Class labels.

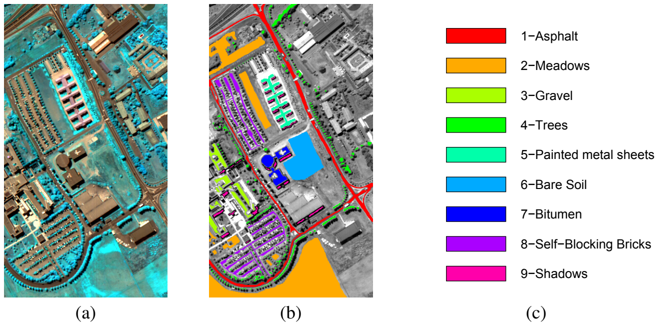


Figure 3. RGB composition and classification map for Pavia University scene scenario. (a) Pseudocolor image. (b) Ground truth map. (c) Class labels.

$0.4 \times 10^{-6} \sim 0.6 \times 10^{-6}$ meters, representing a vegetation-classification scenario. Among all pixels, two thirds are agriculture and one third is forest and other natural perennial vegetation. The image also contains two major dual lane highways, a rail line, some houses and other buildings with low density, and a few local roads. These objects are treated as background so that they will not be classified. When the image was captured, the main crops of soybean and corn are in their early stage of growing. We use the no-till, min-till, and clean-till denote the different growing status of the crops. The water absorption bands (104-108, 150-163, 220) are removed before experiment since they are useless bands for the classification. Hence, the exact 204 spectral bands are used. In the experiments, totally 10,249 (labeled) pixels are employed to form the data set X, in which about 10% are selected in the labeled set X_ℓ and the remains form the unlabeled set X_u in the test. The ground truth image contains 16 classes. Fig. 1 consists of three sub-images: (a) the pseudocolor image of Indian Pines; (b) the ground true map of the classifications; and (c) the color bar of 16 classes.

The second data set used in our experiments is the *AVIRIS Salinas* scenario, which was captured by the AVIRIS sensor over Salinas Valley, California, USA, with a spatial resolution of 3.7 meter per pixels. This data set has totally 224 bands of size 512 × 217. The 20 watered absorption bands (108-112, 154-167, 224) are excluded in experiment. Moreover, this scene was available only as at-sensor radiance data. It includes vegetables, bare soils, and vineyard fields. Totally 16 classes are included in this data set. Fig. 2 shows (a) the color composite of the Salinas image, (b) the ground truth map, and (c) the color bar of 16 classes.

The third data set *Pavia University* scene was captured by the Reflective Optics System Imaging Spectrometer (ROSIS-03) optical satellite sensor, which provides 115 bands HSI data during a flight campaign over the Pavia of the northern Italy. The size of Pavia University scene is 610×340 with 115 bands. In the experiment, 12 polluted bands are removed since they have no contribution for the classification. Likewise, some of the samples are treated as background since they are not in the classes we need to determine. The geometric resolution of the scenes is 1.3 meters per pixel, covering the wave ranges from 0.43 μm to 0.86 μm. The pixels of the HSI image cover 9 classes excluding the background. Fig. 3 shows the pixels used in the experiment in (a) pseudo-color image, (b) the ground truth map, and (c) the class bar of all classes, respectively.

2) Metrics on the data sets: It is a common sense that the performance of a classification scheme for HSI images is heavily relied on the quality of metric on HSI data [33]. Many experiences show that the standard Euclidean distance between the spectral vectors (pixels) of a HSI image may not represent the exact similarity. The main reason is that the spectral vectors in the HSI image are departed from their truthes by the noise. Note that a pixel in the spatial neighborhood of a pixel $\vec{x}$ is most likely in the same class as $\vec{x}$. Since the spatial positions of pixels are not impacted by noise, merging the spatial distance into the spectral one can correct the derivation caused by noise. In this paper, we adopt the following spectral-spatial affinity metric:

$$w_{ij}(\vec{x}_i, \vec{x}_j) = w^r_{ij}(\vec{x}_i, \vec{x}_j) + \mu w^s_{ij}(\vec{x}_i, \vec{x}_j), \qquad (12)$$

where w^r_{ij} and w^s_{ij} are radian weight (or spectral distance) and spatial weight (a distance-type weight), respectively, and $0 \le \mu \le 1/2$ is the *weight balance parameter* that measures the strength of the spatial prior.(in the paper, we set $\mu = 1/2$).

The radian weight w^r_{ij} is defined by the following:

$$w^r_{ij}(\vec{x}_i, \vec{x}_j) = 1 - \exp\left(-\frac{\|\vec{x}_i - \vec{x}_j\|^2}{\rho_i \rho_j}\right) \qquad (13)$$

where ρ_i denotes the local scaling parameter with respect to $\vec{x}_i$ defined by

$$\rho_i = \|\vec{x}_i - \vec{x}^s_i\|, \qquad (14)$$

where $\|*\|$ denotes the l_2 norm, $\vec{x}^s_i$ is the s-th nearest neighbor of $\vec{x}_i$, and s is a preset positive odd integer (in our experiment, $s = 5$), called *spectral-weight parameter*. The distance in (13) is call a *diffusion distance*. It is more consistent of the manifold structure of data. More details on the spectral distance design are refer to [34].

The spatial weight in the paper is defined as follows: We first construct a spatial neighborhood system on HSI. Let $\vec{x}_i$ and $\vec{x}_j$ have the 2D index $i = (i_1, i_2)$ and $j = (j_1, j_2)$ in the HSI image X, respectively. Let $r > 0$ be the size of a spatial neighborhood system on X. (In this paper, we fix $r = 2$.) We define the spatial neighborhood of $\vec{x}_i$ by

$$\mathcal{N}_i = \{j;\ \max(|i_1 - j_1|, |i_2 - j_2|) \le r, \quad j \ne i\}$$

Aid with the neighborhood system, we formulate the spatial (distance-type) weight between two pixels $\vec{x}_i$ and $\vec{x}_j$ as the following:

$$w_{ij}^s = \begin{cases} -1, & \text{if } j \in \mathcal{N}_i \\ 0. & \text{otherwise} \end{cases} \tag{15}$$

Note that we use spatial weight -1 for the pixel in the neighborhood to shorten the spectral-spatial distance between neighbored pixels.

3) Optimization of parameters in the algorithm: The process for optimizing the free parameters in **ESSL1dLB** algorithm is very similar to that for the **SS1DME** one in [22]. For shortening the length of the paper, we only give a brief description of the process. The following parameters are tested and optimized: (a) regularization parameter λ in (4), (b) weight balance parameter μ in (12), (c) path selection parameter ϵ in (3), (d) the spinning number K in (9), and (e) the label boosting times N in (11).

The regularization parameter λ balances the fidelity term and smoothness one in the regularization algorithm. It can be learned in a standard way. The tuning process shows that it is insensitive in the range $[0.3, 5]$. We fix it to 0.5 in all experiments. The weight balance parameter μ impacts the definition of similarity between pixels. Its selection should not be dependent on an individual SSL method. The experiments show that it can be chosen in the range $[0.3, 0.7]$. The quantitative analysis for μ can bo found in Fig. 12-13 in [22].

The path selection parameter ϵ impacts the approximation quality of the greedy method presented in **Algorithm 1.** It is uniform for all SLL methods based on data 1-D representation, but possibly dependent on the data set. Fortunately, although differen data sets have different optimal ϵ, it is an insensitive parameter. The parameter tuning experiment results almost are similar when ϵ is selected in a very wide range, say in $[50, 500]$ (see Fig. 14 in [22]). Hence, the values used in [22] can also be applied to the proposed method in the paper. In our experiments, we choose $\epsilon = 100$.

To investigate how the spinning number K impacts the classification output. We use the HSI image "AVIRIS Indian Pines"in the parameter tuning experiments, where other parameters are fixed as following: $\lambda = 0.5, \mu = 0.5, \epsilon = 100$, and the boosting number $N = 5$; but the values of K are chosen in the integer range $[3, 10]$. The experiment is repeated 5 times for each value of K, and the average scores of OA, AA, and κ are reported. Their meanings are explained in the next subsection. The results are shown in Fig. 4 and Tab. I. We observe that the spinning parameter K in (9) is relatively insensitive too. In our experiments, we will set $K = 7$.

Finally, we test the effectiveness of the number of label boosting times N, which determines the enlargement of the labeled set. A greater number of the boosting times usually yields a larger size of the boosted set X_ℓ^N at the last step in the construction of the final classifier. Again, we use the HSI image "AVIRIS Indian Pines"as the train set, where other parameters are fixed as following: $\lambda = 0.5, \mu = 0.5, \epsilon = 100, K = 7$, but the number of label boosting times N is chosen in the integer range $[4, 10]$. The experiment is repeated 5 times for each value of N, and the average scores of OA, AA, and κ are reported in Fig. 5 and Tab. II. The experiment results indicate that the number of label boosting times can be chosen from the integer range $N \geq 4$.

All of the tests above indicate the stability and reliability of the **ESSL1dLB** algorithm: Although the algorithm is a multi-parametric one, all of parameters are relatively insensitive so that each of them can be chosen in a wide range without great deviation.

B. Measurements of performances of experiments

The maps of the thematic land covering, which are generated by different classification methods, are used in a variety of applications for data analysis. In this paper, each experiment contains five repeated tests at random using the same parameter settings. The quality of the output of the experiment is evaluated in the standard way commonly used in the classification of HSI images [35]. That is, the performance will be measured by *overall accuracy* (OA), *average accuracy* (AA), and *Kappa coefficient* (κ) of the five tests. As their names indicate, OA, one of the simplest and most popular accuracy, measures the accuracy of the classification weighted by the proportion of testing samples of each class in the total training set, AA measures the average accuracy of all classes, and Kappa measures the agreement of the tests, of which each classifies n samples into C mutually exclusive classes.

C. Experiment comparison settings

Similar to [22], three widely used hyperspectral data sets, the Indian Pines 1992 scene, the Salinas scene, and the University of Pavia scene are used in experiments to evaluate the classification performance. The pseudocolor images, the ground truth maps, and the class label bar of these HSI images are shown in Fig. 1–3, respectively. For comparison, we assess our proposed **ESSL1dLB** algorithm with several spectral-based and spectral-spatial extended methods, LDA [36], LDA with multi-logistic prior (LDA-MLL) [37], SVM [38], [39], Laplacian SVM (LapSVM) [40], SVM with component kernel (SVM$_{\text{CK}}$) [41], orthogonal matching pursuit (OMP) [42], simultaneous OMP (SOMP) [43], MLR*sub* [37], MLR*sub*-MLL [37], *semi*MLR-MLL [44], WT-EMP [45] for hyperspectral image classification. These methods are well established in the hypersepctral remotely sensing community. In the comparison, we also add SS1DME [22], which was an earlier work in the ensemble SSL framework developed by the author and his colleagues.

In all of the mentioned methods, the LapSVM and the *semi*MLR-MLL approaches are usually considered to be the reference benchmarks for semi-supervised learning in hyperspectral image classification, as summarized in [44], [46], [47].

For fair comparison, five experiments with different randomly sampled data are carried out for each data set to enhance the statistical significance. In the comparison, the experiment results of other methods are either directly obtained from the authors' papers, or obtained by running the code provided by the authors with the optimal parameters. In all of the following experiments, we use the unconstraint 1-D least-square regularization model, set $\lambda = \mu = 0.5, \epsilon = 100, N = 5$, and choose the spinning number $K = 7$ in the label boosting process and set $K = 10$ in the last spinning for producing the final classifier.

D. Experiment 1– AVIRIS Indian Pines Data Set

The first experiment is conducted on the AVIRIS Indian Pines data set, whose format and data structure information

TABLE I. THE RESULTS OF VARIOUS SPINNING NUMBERS USED IN THE EXPERIMENT FOR CLASSIFICATION OF AVIRIS INDIAN PINES. IN THE EXPERIMENT $\lambda = 0.5, \mu = 0.5, \epsilon = 100$ AND $N = 5$ ARE FIXED, BUT K ARE SELECTED IN THE INTEGER RANGE $[3, 10]$.

K	3	4	5	6	7	8	9	10
OA mean(%)	99.10	99.11	98.83	99.08	99.22	99.14	98.89	99.08
OA std	0.32	0.21	0.29	0.18	0.17	0.21	0.23	0.25
AA mean(%)	99.33	99.40	99.17	99.32	99.46	99.34	99.13	99.36
AA std	0.24	0.11	0.14	0.11	0.09	0.14	0.13	0.12
κ(%)	98.97	98.98	99.17	98.95	99.11	99.02	98.73	98.95
κ std	0.36	0.24	0.33	0.20	0.20	0.24	0.27	0.29

TABLE II. THE RESULTS OF VARIOUS SPINNING NUMBERS USED IN THE EXPERIMENT FOR CLASSIFICATION OF AVIRIS INDIAN PINES. IN THE EXPERIMENT, $\lambda = 0.5, \mu = 0.5, \epsilon = 100$, AND $K = 7$ ARE FIXED, BUT N ARE CHOSEN FROM THE INTEGER RANGE $[4, 10]$.

N	4	5	6	7	8	9	10
OA mean(%)	99.04	99.20	99.14	99.02	99.23	98.94	99.05
OA std	0.35	0.20	0.24	0.25	0.21	0.23	0.30
AA mean(%)	99.14	99.43	99.36	99.26	99.46	99.27	99.35
AA std	0.38	0.11	0.12	0.11	0.12	0.22	0.21
κ(%)	98.89	99.10	98.95	98.89	99.11	98.93	98.89
κ std	0.51	0.20	0.33	0.30	0.20	0.22	0.31

TABLE III. NUMBER OF TRAINING AND TEST SAMPLES FOR THREE HSIS.

ID	Indian Pines			Salinas			University of Pavia		
	Class Name	Train	Test	Class Name	Train	Test	Class Name	Train	Test
1	Alfalfa	20	26	Brocoli Green Weeds 1	144	1865	Asphalt	553	6078
2	Corn-notill	134	1294	Brocoli Green Weeds 2	200	3526	Meadows	1161	17488
3	Corn-mintill	75	755	Fallow	151	1825	Gravel	304	1795
4	Corn	44	193	Fallow Rough Plow	135	1259	Trees	328	2736
5	Grass-pasture	49	434	Fallow Smooth	159	2519	Painted metal sheets	261	1084
6	Grass-trees	56	674	Stubble	209	3750	Bare Soil	440	4589
7	Grass-pasture-mowed	17	11	Celery	192	3387	Bitumen	263	1067
8	Hay-windrowed	59	419	Grapes Untrained	404	10867	Self-Blocking Bricks	379	3303
9	Oats	11	9	Soil Vinyard Develop	282	5921	Shadows	232	715
10	Soybean-notill	95	877	Corn Senesced Green Weeds	179	3099			
11	Soybean-mintill	209	2246	Lettuce-Romaine-4wk	121	947			
12	Soybean-clean	65	528	Lettuce-Romaine-5wk	150	1777			
13	Wheat	29	176	Lettuce-Romaine-6wk	118	798			
14	Woods	104	1161	Lettuce-Romaine-7wk	129	941			
15	Buildings-Grass-Trees-Drives	37	349	Vinyard Untrained	289	6979			
16	Stone-Steel-Towers	20	73	Vinyard Vertical Trellis	138	1669			
	Total	1024	9225	Total	3000	51129	Total	3921	38855

TABLE IV. CLASSIFICATION ACCURACIES OBTAINED BY DIFFERENT METHODS FOR AVIRIS INDIAN PINE SCENE (%).

Class Name	LDA	LDA-MLL	SVM	LapSVM	SVM_{CK}	OMP	SOMP	MLR*sub*	MLR*sub*-MLL	*semi*MLR-MLL	WT-EMP	SS1DME	**ESSL1dLB**
Alfalfa	**100.00**	96.43	89.29	82.62	**100.00**	45.95	81.25	85.19	81.48	**100.00**	92.52	**100.00**	**100.00**
Corn-notill	61.21	92.61	69.92	78.70	92.07	56.13	92.09	65.09	89.15	86.20	90.30	99.30	**99.34**
Corn-mintill	56.80	77.97	57.78	66.69	97.30	56.33	89.20	47.56	**99.87**	76.05	93.15	98.68	99.12
Corn	71.72	**100.00**	71.43	76.91	**100.00**	44.74	96.65	72.78	99.40	82.13	79.76	99.48	98.92
Grass-pasture	91.76	94.85	90.39	91.54	97.88	84.46	92.65	87.85	93.69	91.41	94.79	96.31	**98.22**
Grass-trees	94.95	98.33	94.52	97.49	98.80	89.73	98.67	94.81	97.47	98.35	98.04	**100.00**	**100.00**
Grass-pasture-mowed	92.31	**100.00**	85.71	95.83	**100.00**	63.64	75.00	61.54	**100.00**	**100.00**	94.46	90.91	**100.00**
Hay-windrowed	96.80	**100.00**	96.98	98.56	98.33	95.81	**100.00**	99.29	**100.00**	99.53	96.83	**100.00**	**100.00**
Oats	**100.00**	**100.00**	88.89	98.89	**100.00**	50.00	44.44	77.78	**100.00**	**100.00**	97.78	**100.00**	**100.00**
Soybean-notill	59.07	82.08	75.17	77.61	91.72	69.15	87.53	65.32	95.96	83.72	87.68	99.54	98.16
Soybean-mintill	65.21	98.63	84.57	83.79	95.67	75.15	97.16	69.04	96.87	92.24	92.58	99.33	**99.46**
Soybean-clean	68.30	74.63	74.95	82.18	87.29	50.00	87.09	73.85	97.23	91.89	88.73	**98.30**	98.08
Wheat	98.87	99.44	97.16	99.45	99.44	95.12	**100.00**	99.31	100.00	99.42	98.25	**100.00**	**100.00**
Woods	91.47	92.47	96.62	94.70	99.22	91.50	99.74	93.05	98.03	96.43	98.01	**99.91**	99.72
Buildings-Grass-Trees-Drives	62.28	**100.00**	54.94	68.75	95.93	41.42	99.71	52.08	97.42	89.41	89.92	99.43	99.70
Stone-Steel-Towers	95.77	85.33	93.65	89.33	**100.00**	90.54	98.61	83.87	**100.00**	84.29	98.61	**100.00**	**100**
OA (mean)	71.54	91.98	80.74	84.11	94.94	71.38	94.42	73.64	94.95	89.32	92.80	99.05	**99.13**
OA (std)	0.25	0.15	0.62	0.37	0.66	0.32	0.18	0.37	0.50	0.95	0.19	0.13	0.24
AA (mean)	79.53	87.63	83.83	86.44	96.21	67.13	91.65	75.42	95.09	86.48	93.21	98.72	**99.36**
AA (std)	1.10	0.33	0.38	0.65	0.53	1.29	2.09	2.02	1.97	3.22	0.78	0.95	0.17
κ (mean)	67.62	90.76	78.03	81.81	94.22	67.32	93.62	69.78	94.18	93.77	91.78	98.92	**99.00**
κ (std)	0.38	0.17	0.70	0.43	0.75	0.38	0.20	0.42	0.58	2.76	0.21	0.15	0.28

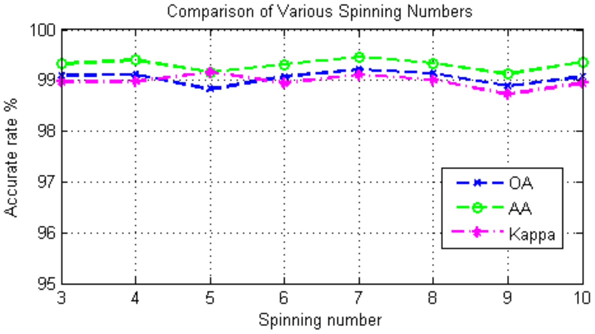


Figure 4. Sensitivity analysis of the spinning number K.

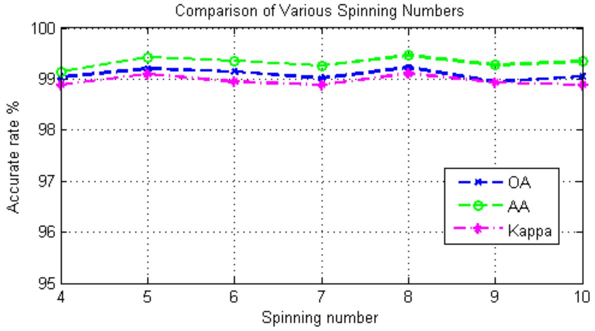


Figure 5. Sensitivity analysis of the label boosting times N.

are given in Subsection III-A. The number of training samples and test samples are given in Tab. III. Figure 6 shows the classification pseudo-color maps that are obtained by different methods along with the corresponding OA score. Among all of the methods, LDA-MLL, SVM_{CK}, SOMP, MLR*sub*MLL, *semi*MLR-MLL, WT-EMP, SS1DME, and **ESSL1dLB** yield high accuracy. Comparing with the all other methods method, the proposed **ESSL1dLB** method wins the best performance in all of OA, AA, and Kappa coefficient. We note that the classification accuracies of **ESSL1dLB** exceeds 98% for all of 16 classes.

Remark. In the Fig. 6, the OA score is slightly different from that in Tab. III. Because the OA score in Tab. III is the average of 5 experiments, while Fig. 6 is for one of the experiments selected at random. The same remark is also valid for the following two experiments.

E. Experiment 2—AVIRIS Salinas Data Set

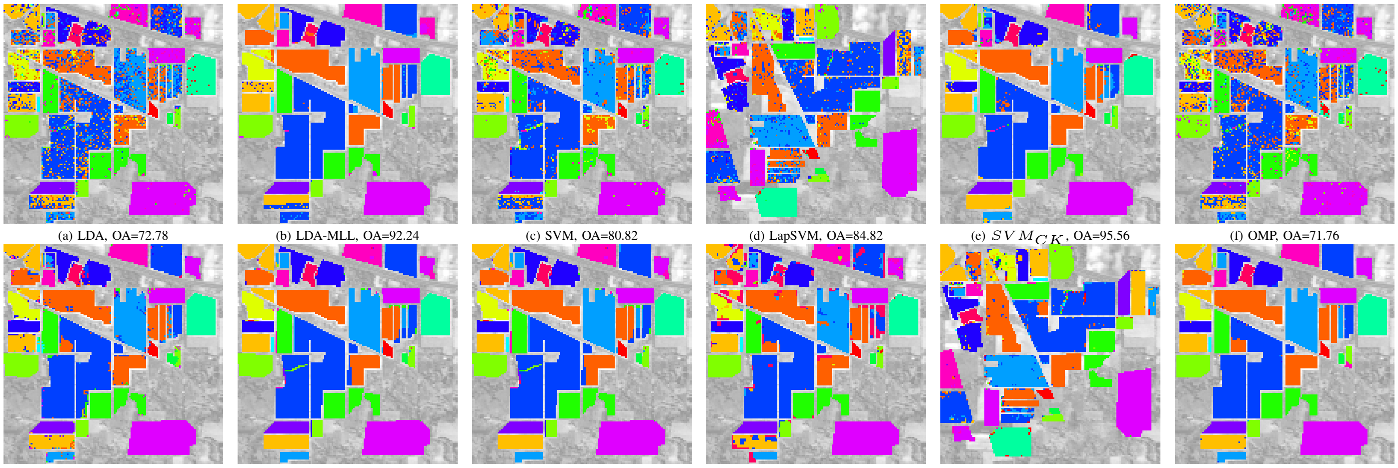
The second experiment was performed on the AVIRIS Salinas hyperspectral image. The number of training and testing samples for the image are given in Tab. III, where the training set contains about 5.25% of all the labeled samples, chosen at random. Because the image size is too large to be treated on a Laptop, we divide the data set into 8 blocks in the experiment. A visual perspective of these methods are presented in Fig. 7. The quantitative results are presented in Tab. V. Similar to the AVIRIS Indian Pines image, it can be seen that the proposed **ESSL1dLB** beats the classification performances of other methods in terms of OA, AA and Kappa coefficient.

F. Experiment 3—ROSIS University of Pavia Data Set

The third experiment is conducted on the data set of *ROSIS University of Pavia scene*. In this experiment, we use the randomly chosen 3,921 labeled samples for training, which count about 8.4% of all labeled pixels, while the remains are used for testing. Detailed numbers for training and testing can be found in Tab. III. Because the data set has $512 \times 217 = 111104$ pixels, this size is too large to be treated on a Laptop too. Hence, we divide it into 8 disjoint blocks, then apply the proposed algorithm on each block. The classification maps obtained by different methods and the associated OA scores are presented in Fig. 8. Meanwhile, the quantitative results (means and standard deviations over the experiments on randomly selected five different training sets) are listed in Tab. VI. It can be observed that the proposed **ESS1DLB** algorithm again performs better than other methods significantly in both of quantitative results and visual qualities. For example, our algorithm obtains more than 99% accuracy for all classes. Particularly, for the *Gravel, Trees, Self-Blocking Bricks* classes, the classification accuracies obtained by most methods are not very satisfactory, but our method still produces a super result.

IV. Experiments on Handwritten Digits

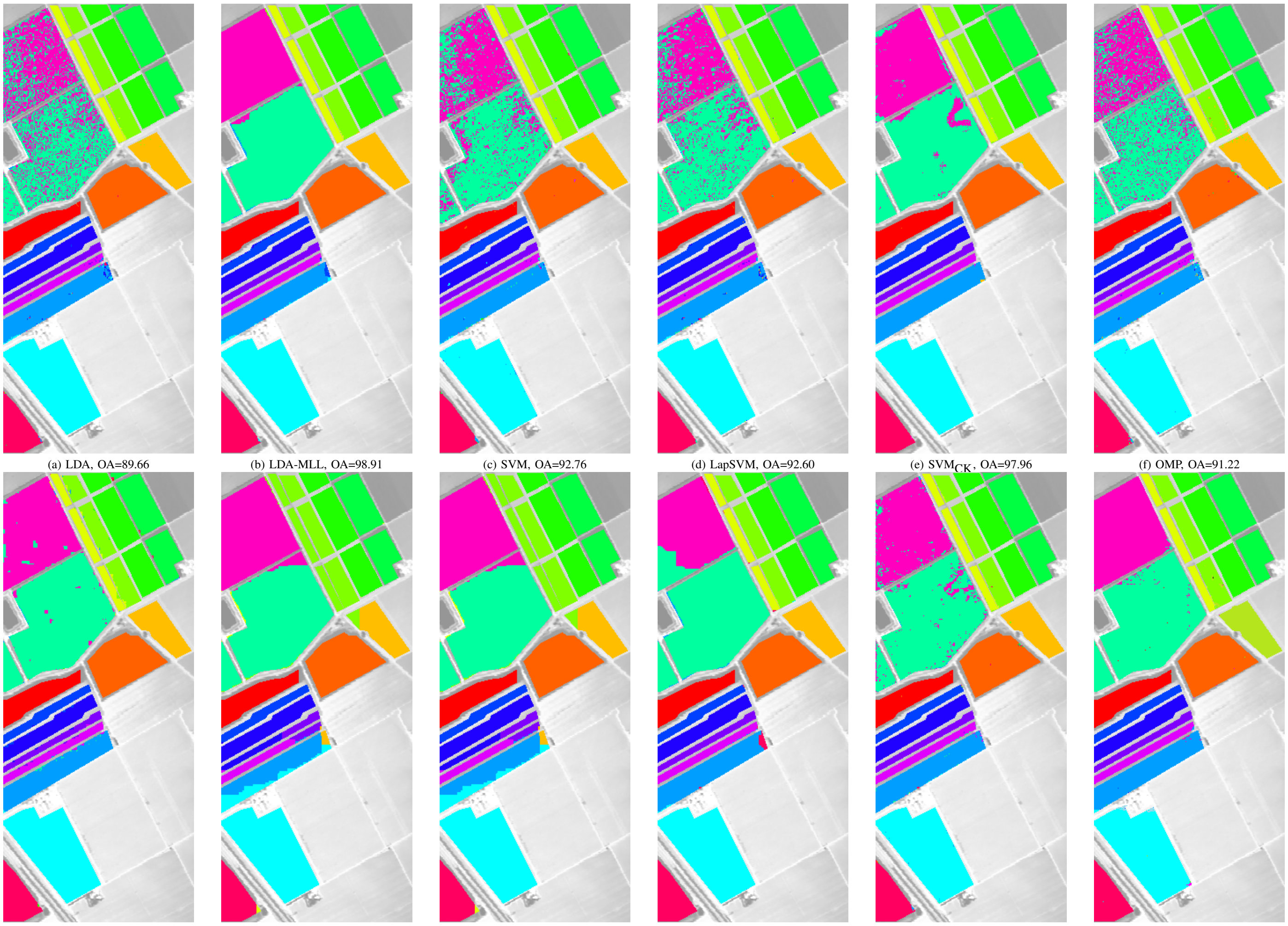
In this section, we evaluate our ensemble SSL method in the experiments on handwritten digits. We use two benchmark databases of handwritten digits, MNIST [48] and USPS [49] in the experiments to present the validity and effectiveness of the proposed method. In the literature of machine learning, MNIST is often used to test the error rate of classifiers obtained by supervised learning. The best result for the error rate up to 2012 was 0.23%, reported in [50] by using the convolutional



(a) LDA, OA=72.78 (b) LDA-MLL, OA=92.24 (c) SVM, OA=80.82 (d) LapSVM, OA=84.82 (e) SVM_{CK}, OA=95.56 (f) OMP, OA=71.76

(g) SOMP, OA=94.65 (h) MLR*sub*, OA=73.90 (i) MLR*sub*MLL,OA=96.20 (j) *semi*MLR-MLL,OA=90.27 (k) WT-EMP, OA=92.99 (l) **ESSL1dLB**, OA=99.13

Figure 6. Classification pseudocolor map obtained by different methods for the AVIRIS Indian Pines data set, where the value of OA is given in percent.



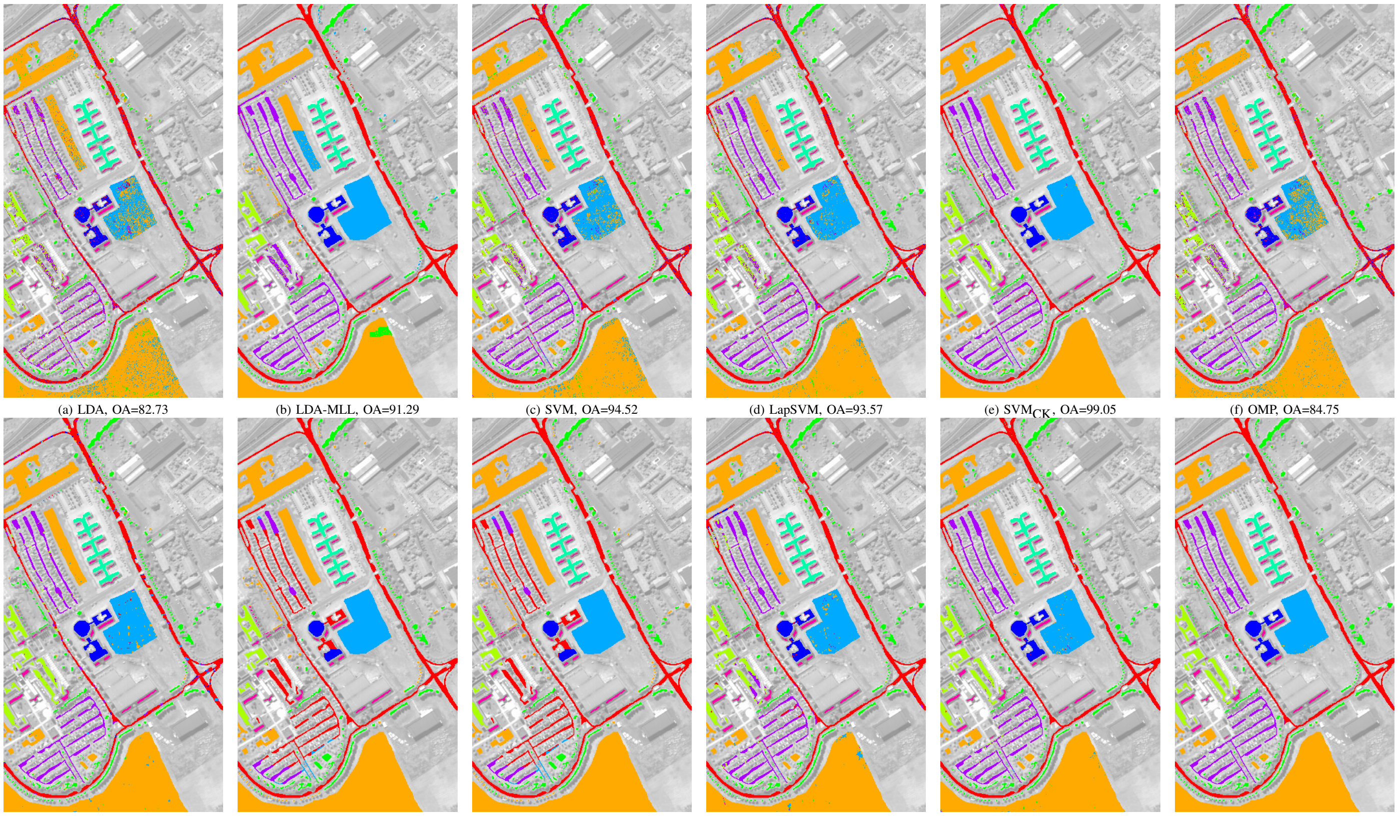
(a) LDA, OA=89.66 (b) LDA-MLL, OA=98.91 (c) SVM, OA=92.76 (d) LapSVM, OA=92.60 (e) SVM_{CK}, OA=97.96 (f) OMP, OA=91.22

(g) SOMP, OA=98.44 (h) MLR*sub*, OA=76.54 (i) MLR*sub*MLL, OA=92.98 (j) *semi*MLR-MLL, OA=97.13 (k) WT-EMP, OA=97.31 (l) **ESSL1dLB**, OA=99.56

Figure 7. Classification pseudocolor map obtained by different methods for the AVIRIS Salinas hyperspectral image, where the value of OA is given in percent.

TABLE V. CLASSIFICATION ACCURACIES OBTAINED BY DIFFERENT METHODS FOR AVIRIS SALINAS SCENE (%).

Class Name	LDA	LDA-MLL	SVM	LapSVM	SVM$_{\text{CK}}$	OMP	SOMP	MLR*sub*	MLR*sub*-MLL	*semi*MLR-MLL	WT-EMP	SS1DME	**ESSL1dLB**
Brocoli Green Weeds 1	99.78	**100.00**	99.57	99.84	99.84	99.13	**100.00**	99.78	**100.00**	**100.00**	99.59	99.95	**100.00**
Brocoli Green Weeds 2	99.94	**100.00**	99.83	99.76	**100.00**	99.43	**100.00**	88.44	**100.00**	**100.00**	99.79	99.89	99.98
Fallow	99.56	**100.00**	99.73	99.62	98.42	99.87	**100.00**	61.77	90.27	**100.00**	99.80	99.67	**100.00**
Fallow Rough Plow	99.60	99.44	99.44	99.41	99.92	99.64	97.94	10.85	**100.00**	99.20	99.43	99.60	99.60
Fallow Smooth	98.44	99.12	99.52	99.09	99.09	97.62	95.49	99.88	**100.00**	99.17	98.70	99.40	99.38
Stubble	99.89	99.89	99.89	99.89	99.97	99.94	99.89	99.66	99.97	**100.00**	99.85	99.97	99.92
Celery	99.62	99.91	99.74	99.58	99.82	99.76	98.91	99.85	99.94	**99.97**	99.57	99.88	99.91
Grapes Untrained	75.11	97.50	86.67	85.24	97.68	80.83	98.52	59.04	95.75	99.00	94.29	**99.05**	98.97
Soil Vinyard Develop	99.92	**100.00**	99.43	99.91	99.81	99.76	**100.00**	99.35	100.00	99.98	99.57	99.58	99.66
Corn Senesced Green Weeds	96.00	95.23	96.76	96.64	97.09	96.38	97.35	48.53	68.91	95.61	97.89	**98.84**	99.07
Lettuce-Romaine-4wk	99.26	94.60	99.25	99.00	99.89	99.77	99.37	93.87	99.43	99.68	98.89	**100.00**	99.82
Lettuce-Romaine-5wk	99.38	**100.00**	99.83	**100.00**	**100.00**	**100.00**	96.68	88.39	99.77	**100.00**	**100.00**	**100.00**	**100.00**
Lettuce-Romaine-6wk	99.24	99.37	98.73	98.94	99.87	98.23	95.11	99.04	39.64	99.49	99.72	**100.00**	99.81
Lettuce-Romaine-7wk	96.60	98.94	98.93	97.73	**99.79**	95.68	94.80	73.73	71.46	97.89	98.24	98.72	99.39
Vinyard Untrained	66.85	99.56	71.05	73.27	54.30	69.50	96.88	56.05	99.35	83.33	93.21	99.47	**99.54**
Vinyard Vertical Trellis	99.28	99.58	98.92	99.15	99.04	98.41	99.64	98.52	98.27	**100.00**	99.10	99.94	**100.00**
OA (mean)	89.59	97.48	92.67	92.78	97.24	90.96	97.93	76.37	91.79	96.55	97.44	99.45	**99.55**
OA (std)	0.26	0.88	0.09	0.06	0.61	0.18	0.47	0.09	1.27	0.38	0.26	0.04	0.05
AA (mean)	95.57	98.46	96.60	96.71	98.75	95.68	97.69	82.33	86.85	91.83	98.60	99.64	**99.69**
AA (std)	0.16	0.35	0.10	0.12	0.24	0.10	0.74	2.06	1.09	0.17	0.13	0.02	0.05
κ (mean)	87.95	97.18	91.81	91.93	96.92	89.93	97.68	73.75	90.83	96.36	97.15	99.39	**99.50**
κ (std)	0.30	0.99	0.10	0.06	0.68	0.20	0.53	0.09	1.40	0.47	0.29	0.04	0.05



(a) LDA, OA=82.73

(b) LDA-MLL, OA=91.29

(c) SVM, OA=94.52

(d) LapSVM, OA=93.57

(e) SVM$_{\text{CK}}$, OA=99.05

(f) OMP, OA=84.75

(g) SOMP, OA=96.11

(h) MLR*sub*, OA=62.60

(i) MLR*sub*MLL, OA=89.31

(j) *semi*MLR-MLL, OA=96.49

(k) WT-EMPs, OA=98.72

(l) **ESSL1dLB**, OA=99.82

Figure 8. Classification pseudocolor map obtained by different methods for University of Pavia scene data set, where the value of OA is given in percent.

TABLE VI. CLASSIFICATION ACCURACIES OBTAINED BY DIFFERENT METHODS FOR ROSIS UNIVERSITY OF PAVIA SCENE(%).

Class Name	LDA	LDA-MLL	SVM	LapSVM	SVM_{CK}	OMP	SOMP	MLR*sub*	MLR*sub*-MLL	*semi*MLR-MLL	WT-EMP	SS1DME	**ESSL1dLB**
Asphalt	79.91	87.58	92.40	89.14	98.51	79.74	86.12	98.43	**99.87**	96.04	98.39	99.57	99.84
Meadows	90.48	92.53	97.83	97.44	**99.99**	95.64	99.44	98.78	98.96	98.65	99.49	99.98	**99.99**
Gravel	69.54	65.67	87.44	79.28	95.24	59.14	98.16	63.46	64.60	85.32	96.71	**99.94**	**99.94**
Trees	86.95	77.62	96.03	95.79	98.24	86.17	94.96	69.99	76.99	96.58	98.02	99.12	**99.48**
Painted metal sheets	99.91	99.82	99.72	99.30	**100.00**	99.54	**100.00**	99.89	**100.00**	99.53	99.86	**100.00**	**100.00**
Bare Soil	64.03	**100.00**	91.00	91.09	99.50	59.43	95.32	**100.00**	99.96	95.89	97.78	99.76	**100.00**
Bitumen	81.54	99.35	90.20	90.49	99.63	78.20	99.72	36.65	59.89	95.78	97.45	**100.00**	99.90
Self-Blocking Bricks	67.97	98.73	86.94	87.69	96.49	80.62	96.11	2.46	26.01	91.47	96.57	99.18	**99.81**
Shadows	99.29	93.90	99.86	99.63	**100.00**	96.17	92.72	98.05	99.80	99.30	99.95	99.86	99.69
OA (mean)	81.30	90.79	94.32	93.51	97.96	84.60	95.97	62.39	88.90	95.98	98.60	99.74	**99.91**
OA (std)	0.11	0.30	0.13	0.04	1.44	0.16	0.11	0.16	0.38	0.57	0.18	0.02	0.03
AA (mean)	83.02	90.15	93.59	92.21	96.98	81.80	95.89	75.48	82.84	83.90	98.25	99.70	**99.85**
AA (std)	0.22	0.49	0.11	0.21	2.53	0.20	0.11	0.87	3.45	0.41	0.30	0.02	0.04
κ (mean)	74.51	87.73	92.37	91.26	97.26	79.31	94.57	52.66	84.60	94.57	98.11	99.65	**99.87**
κ (std)	0.13	0.39	0.17	0.06	1.92	0.21	0.14	0.24	0.55	0.77	0.24	0.03	0.04

TABLE VII. ERROR RATE OF THE PROPOSED ESSL1DLB FOR 50 RANDOMLY SELECTED SUBSETS FROM MNIST WITH $|X| = 1000$.

$\lvert X_0 \rvert$	10	20	30	40	50	60	70	80	90	100
Mean%	7.84	7.80	4.58	3.06	2.91	1.91	1.93	1.97	1.23	1.27
Min%	7.60	3.10	3.80	1.90	2.90	1.90	1.90	1.90	1.20	1.20
Max%	19.4	7.90	4.60	3.10	3.50	2.50	2.60	3.90	2.80	3.30
STD	1.65	0.67	0.11	0.19	0.08	0.08	0.14	0.35	0.22	0.37

TABLE VIII. ERROR RATE OF THE PROPOSED ESSL1DLB FOR 50 RANDOMLY SELECTED SUBSETS FROM USPS WITH $|X| = 1500$.

$\lvert X_0 \rvert$	10	20	30	40	50	60	70	80	90	100
Mean%	3.07	1.933	1.55	1.49	1.28	1.38	1.37	1.39	1.34	1.20
Min%	3.00	1.27	1.53	1.07	1.27	1.20	1.07	1.33	0.80	1.20
Max%	3.73	2.87	1.67	1.53	1.40	1.40	1.40	1.40	1.40	1.20
STD	0.22	0.76	0.04	0.14	0.04	0.06	0.10	0.02	0.18	0.00

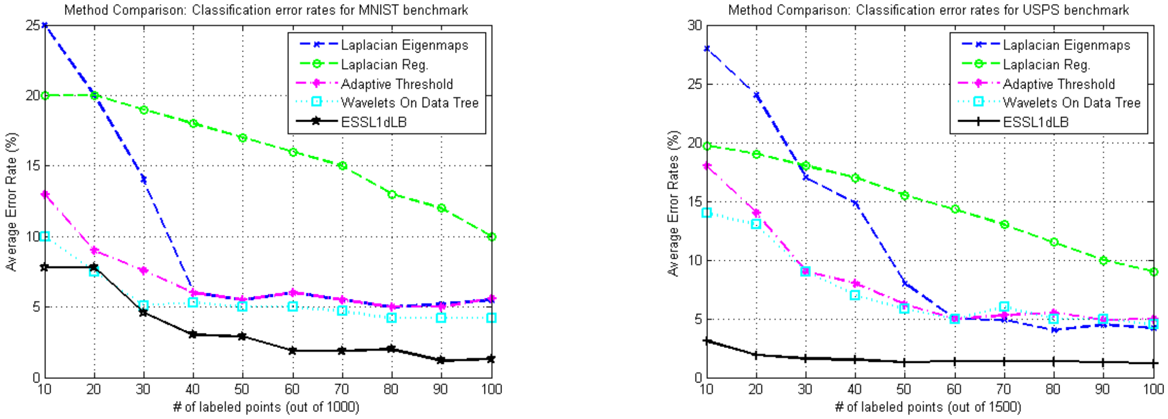


Figure 9. Result comparison with different SSL models.

neural network technique. In 2013, the authors of [51] claimed to achieve 0.21% error rate using **DropConnect** method, which is based on regularization of neural networks. Because in SSL no large training set is available for producing classifiers, the error rates obtained by SSL methods usually are much higher than the claimed error rates obtained by supervised learning. Besides, the error rates of SSL are strongly dependent the size of the initial label set X_ℓ. In general, the smaller the size of X_ℓ, the higher the error rate. Hence, it is unfair to compare the error rates obtained by SSL methods to the above recorded ones.

The parameters are tuned in the similar way as we have done above. Once again, the tuning experiments show the insensitivity of the parameters. Since the tuning process is very similar to that in Subsection III-A3, we omit the details here. In all of our experiments, the balance parameter in the least-square regularization is set to $\lambda = 0.5$. The spin number $K = 3$ is used for constructing 1DEL algorithm, while $K = 20$ is chosen for building the final classifier. The boosting-stop parameter p is set to 0.7, which yields 6 times of label boosting in most cases.

For comparison, we choose the same data setting as in [12]: In MINST, for each of the digits $\{3, 4, 5, 7, 8\}$, 200 samples are selected at random so that the cardinality of the data set is $|X| = 1000$, where the digit 8 is assigned to Class A, and others belong to Class B. In USPS, for each of the digits $0-9$, 150 samples are selected at random so that $|X| = 1500$, where the digits 2 and 5 are assigned to Class A, and others belong to Class B. In all experiments, the initial labeled set X_0 is preset to 10 various sizes of $10, 20, \cdots, 100$, respectively, and the labeled digits are distributed evenly on each chosen digit.

Note that a vector $\vec{x} \in X$ is originally represented by a $c \times c$ matrix $[x_{i,j}]_{i,j=1}^{c}$, where $c = 20$ for MNIST and $c = 16$ for USPS. To reduce the shift-variance, we define the 1-*shift distance* between two digit images [1]:

$$d(\vec{x}, \vec{y}) = \min_{\substack{|i'-i|\le 1\\ |j'-j|\le 1}} \sqrt{\sum_{i=2}^{c-1}\sum_{j=2}^{c-1} \left(x_{i,j} - y_{i',j'}\right)^2}.$$

In the first experiment, we run our **ESSL1dLB** algorithm on 50 subsets, of which each has with 1000 members, randomly chosen from the MNIST database, where the regularization parameter λ in (4) is chosen to be 0.5. The experiment results are shown in Table VII, where the first row is the number of samples in X_ℓ, and the $2^{nd} - 5^{th}$ rows are the mean, minimum, maximum, and standard deviation of the classification error rates of the 50 tests, respectively.

In the second experiment, we run our **ESSL1dLB** algorithm for USPS in a similar way: 50 subsets, of which each has 1500 members, are randomly chosen from USPS database. The test results are shown in Tab. VIII.

Tab. VII and Tab. VIII show that the standard deviations of the error rates are quite small. This indicates the high stability of the proposed algorithm.

In Fig. 9, we give the comparison of the average error rates (of 50 tests) of our 1-D based ensemble method **ESSL1dLB** to Laplacian Eigenmaps (Belkin & Niyogi, 2003 [8]), Laplacian Regularization (Zhu et al., 2003 [13]), Laplacian Regularization with Adaptive Threshold (Zhou and Belkin, 2011 [52]), and Haar-Like Multiscale Wavelets on Data Trees (Gavish et al., 2011 [12]) on the subsets randomly chosen from MNIST and USPS databases, respectively.

The results show that our method achieves competitive results comparing to others.

V. Conclusion

We propose a new ensemble SSL method (**ESSL1dLB**) based on data 1-D representations and label boosting, which enables us to construct ensemble classifiers assembled from several weak-classifiers for the same data set using classical 1-D regularization technique. Furthermore, a label boosting technique is applied for robustly enlarging the labeled set to a certain size so that the final classifier is built based on the boosted labeled set. The experiments show that the performance of the proposed method is superior to many popular SSL methods. The method also exhibits a clear advantage for learning the classifier when only a small labeled set is given. Because the method is independent of the data dimensionality, it can be applied to various types of data. Since the algorithm in the proposed method only employs 1-D regularization technique, avoiding the complicate kernel trick, they are simple and stable. The experiments also indicate that the parameters in the algorithm is relatively insensitive that makes the algorithm more controllable and reliable. The algorithm has been tested on various types of data sets, such as handwritten digits and hyperspectral images. The experimental results are very promising, showing that our method is superior to other existent methods. It can be expected that the created 1-D framework in this paper will be applied to the development of more machine learning methods for different purposes. In the algorithm, the most time-consuming step is data (shortest path) sorting. In the future work, we will study how to accelerate the sorting algorithm in 1-D embedding and consider to integrate the data-driven wavelets with the proposed method.

Acknowledgment

In this work, the first author was supported by SHSU-2015 ERG Grant-250711. The second and third authors were supported by the Research Grants of MYRG2015-00049-FST, MYRG2015-00050-FST, RDG009/FST-TYY/2012, 008-2014-AMJ from Macau; and the National Natural Science Foundation of China 61672114.

The authors would like to thank the anonymous reviewers for their highly insights and helpful suggestions.

References

[1] J. Wang, “Semi-supervised learning in the framework of data multiple 1-d representation,” IMMM 2016, The Sixth International Conference on Advances in Information Mining and Management, 2016, pp. 1–5.

[2] R. K. Eichelberger and V. S. Sheng, “Does one-against-all or one-against-one improve the performance of multiclass classifications?” in Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, Bellevue, Wash, USA, 2013.

[3] C. Yu, L. Jinxu, Z. Fudong, B. Ran, and L. Xia, “Comparative study on face recognition based on SVM of one-against-one and one-against-rest methods,” in Future Generation Communication and Networking (FGCN), 2014 8th International Conference on, Dec 2014, pp. 104–107.

[4] R. Bellman, Adaptive Control Processes: A Guided Tour. Princeton: Princeton University Press, 1961.

[5] J. Wang, Geometric Structure of High-Dimensional Data and Dimensionality Reduction. Higher Education Press and Springer, 2012.

[6] O. Chapelle, B. Schölkopf, and A. Zien, Semi-supervised learning. MIT press Cambridge, 2006.

[7] X. Zhu, “Semi-supervised learning literature survey,” University of Wisconsin-Madison, Computer Sciences TR-1530, July 2008.

[8] M. Belkin and P. Niyogi, “Using manifold structure for partially labeled classification,” Advances in Neural Information Processing Systems, vol. 15, 2003, pp. 929–936.

[9] T. Joachims, “Transductive learning via spectral graph partitioning,” in Proceedings of ICML-03, 20th International Conference on Machine Learning, 2003, pp. 290–297.

[10] V. Vapnik, Statistical Learning Theory. Wiley-Interscience, New York, 1998.

[11] R. Coifman and M. Gavish, “Harmonic analysis of digital data bases,” Applied and Numerical Harmonic Analysis (Special issue on Wavelets and Multiscale Analysis), 2011, pp. 161–197.

[12] M. Gavish, B. Nadler, and R. R. Coifman, “Multiscale wavelets on trees, graphs and high dimensional data: Theory and applications to semi supervised learning,” in Proceedings of the 27th International Conference on machine Learning, 2010.

[13] X. Zhu, Z. Ghahramani, and J. Lafferty, “Semi-supervised learning using Gaussian fields and harmonic functions,” in Proceedings of the 20th International Conference on Machine Learning, 2003.

[14] L. Rokach, “Ensemble-based classifiers,” Artif. Intell. Rev., vol. 33, 2010, pp. 1–39.

[15] M. Wozniak, M. Grana, and E. Corchado, “A survey of multiple classifier systems as hybrid systems,” Information Fusion, vol. 16, 2014, pp. 3–17.

[16] Z.-H. Zhou, J. Wu, and W. Tang, “Ensembling neural networks: Many could be better than all,” Artificial Intelligence, vol. 137, 2002, pp. 239–263.

[17] Breiman, “Bagging predictors,” Machine Learning, vol. 24, no. 2, 1996, pp. 123–140.

[18] Y. Freund and R. E. Schapire, “A short introduction to boosting,” Journal of Japanese Society for Artificial Intelligence, vol. 14, no. 5, 1999, pp. 771–780.

[19] D. Z. Li, W. Wang, and F. Ismail, “A selective boosting technique for pattern classification,” Neurocomputing, vol. 156, 2015, pp. 186–192.

[20] J. Wang, “Semi-supervised learning using multiple one-dimensional embedding based adaptive interpolation,” International Journal of Wavelets, Multiresolution and Information Processing (Special Issue on Semi-Supervised Learning and Data Processing in the Framework of Data Multiple One-Dimensional Representation), vol. 14, no. 2, February 2016, pp. 1 640 002: 1–11.

[21] ——, “Semi-supervised learning using ensembles of multiple 1d-embedding-based label boosting,” International Journal of Wavelets, Multiresolution and Information Processing (Special Issue on Semi-Supervised Learning and Data Processing in the Framework of Data Multiple One-Dimensional Representation), vol. 14, no. 2, 2016, pp. 164 001: 1–33.

[22] H. Luo, Y. Y. Tang, Y. Wang, J. Wang, C. Li, and T. Hu, “Hyperspectral image classification based on spectral-spatial 1-dimensional manifold,” submitted to IEEE Transaction of Geoscience and Remote Sensing.

[23] Y. Wang, Y. Y. Tang, L. Li, and J. Wang, “Face recognition via collaborative representation based multiple one-dimensional embedding,” International Journal of Wavelets, Multiresolution and Information Processing (Special Issue on Semi-Supervised Learning and Data Processing in the Framework of Data Multiple One-Dimensional Representation), vol. 14, no. 2, 2016, pp. 1 640 003: 1–15.

[24] I. Ram, M. Elad, and I. Cohen, “Image processing using smooth ordering of its patches,” IEEE Trans. on Image Processing, vol. 22, no. 7, July 2013, pp. 2764–2774.

[25] ——, “Image denoising using nl-means via smooth patch ordering,” in Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on, May 2013, pp. 1350–1354.

[26] T. Hastie and R. Tibshirani, “Classification by pairwise coupling,” The Annals of Statistics, vol. 26, 1998, pp. 451–471.

[27] E. Allwein, R. Schapire, and Y. Singer, “Reducing multiclass to binary: A unifying approach for margin classifiers,” Journal of Machine Learning Research, vol. 1, 2000, pp. 113–141.

[28] T. Hamamura, H. Mizutani, and B. Irie, “A multiclass classification method based on multiple pairwise classifiers,” in International Conference on Document Analysis and Recognition, August 3-6 2003, pp. 809–813.

[29] D. Lu and Q. Weng, “A survey of image classification methods and techniques for improving classification performance,” Int. J. Remote Sens., vol. 28, no. 5, 2007, pp. 823–870.

[30] J. Li, X. Huang, P. Gamba, J. Bioucas-Dias, L. Zhang, J. Atli Benediktsson, and A. Plaza, “Multiple feature learning for hyperspectral image classification,” IEEE Trans. Geosci. Remote Sens., vol. 53, no. 3, March 2015, pp. 1592–1606.

[31] Z. Ye, H. Li, Y. Song, J. Wang, and Jon, “A novel semi-supervised learning framework for hyperspectral image classification,” International Journal of Wavelets, Multiresolution, vol. 14, no. 2, February 2016, pp. 164 005–1–17.

[32] Aviras databases. Acceptable on November 17, 2016. [Online]. Available: https://engineering.purdue.edu/ biehl/MultiSpec/

[33] Y. Gu and K. Feng, “Optimized laplacian svm with distance metric learning for hyperspectral image classification,” IEEE J. Sel. Topics Appl. Earth Observ., vol. 6, no. 3, June 2013, pp. 1109–1117.

[34] M. Sugiyama, “Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis,” J. Mach. Learn. Res., vol. 8, May 2007, pp. 1027–1061.

[35] C. Liu, P. Frazier, and L. Kumar, “Comparative assessment of the measures of thematic classification accuracy,” Remote Sens. Environ., vol. 107, no. 4, 2007, pp. 606 – 616.

[36] T. Bandos, L. Bruzzone, and G. Camps-Valls, “Classification of hyperspectral images with regularized linear discriminant analysis,” IEEE Trans. Geosci. Remote Sens., vol. 47, no. 3, March 2009, pp. 862–873.

[37] J. Li, J. Bioucas-Dias, and A. Plaza, “Spectral-spatial hyperspectral image segmentation using subspace multinomial logistic regression and Markov random fields,” IEEE Trans. Geosci. Remote Sens., vol. 50, no. 3, March 2012, pp. 809–823.

[38] G. Mercier and M. Lennon, “Support vector machines for hyperspectral image classification with spectral-based kernels,” in Geoscience and Remote Sensing Symposium, 2003. IGARSS ’03. Proceedings. 2003 IEEE International, vol. 1, July 2003, pp. 288–290 vol.1.

[39] F. Melgani and L. Bruzzone, “Classification of hyperspectral remote sensing images with support vector machines,” IEEE Trans. Geosci. Remote Sens., vol. 42, no. 8, Aug 2004, pp. 1778–1790.

[40] M. Belkin, P. Niyogi, and V. Sindhwani, “Manifold regularization: A geometric framework for learning from labeled and unlabeled examples.” J. Mach. Learn. Res., vol. 7, 2006, pp. 2399–2434.

[41] G. Camps-Valls, L. Gomez-Chova, J. Munoz-Mari, J. Vila-Frances, and J. Calpe-Maravilla, “Composite kernels for hyperspectral image classification,” IEEE Geosci. Remote Sens. Lett., vol. 3, no. 1, Jan 2006, pp. 93–97.

[42] Y. Chen, N. Nasrabadi, and T. Tran, “Sparsity-based classification of hyperspectral imagery,” in Geoscience and Remote Sensing Symposium (IGARSS), 2010 IEEE International, July 2010, pp. 2796–2799.

[43] ——, “Hyperspectral image classification using dictionary-based sparse representation,” IEEE Trans. Geosci. Remote Sens., vol. 49, no. 10, Oct 2011, pp. 3973–3985.

[44] J. Li, J. Bioucas-Dias, and A. Plaza, “Semisupervised hyperspectral image segmentation using multinomial logistic regression with active learning,” IEEE Trans. Geosci. Remote Sens., vol. 48, no. 11, Nov 2010, pp. 4085–4098.

[45] P. Quesada-Barriuso, F. Arguello, and D. Heras, “Spectral-spatial classification of hyperspectral images using wavelets and extended morphological profiles,” IEEE J. Sel. Topics Appl. Earth Observ., vol. 7, no. 4, April 2014, pp. 1177–1185.

[46] W. Kim and M. Crawford, “Adaptive classification for hyperspectral image data using manifold regularization kernel machines,” IEEE Trans. Geosci. Remote Sens., vol. 48, no. 11, Nov 2010, pp. 4110–4121.

[47] H. Yang and M. Crawford, “Spectral and spatial proximity-based manifold alignment for multitemporal hyperspectral image classification,” IEEE Trans. Geosci. Remote Sens., vol. 54, no. 1, Jan 2016, pp. 51–64.

[48] Y. LeCun, C. Cortes, and C. J. C. Burges. The mnist database of handwritten digits. Accepted November 17, 2016. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[49] Usps handwritten digit data. Accepted November 17, 2016. [Online]. Available: http://www.gaussianprocess.org/gpml/data/ fide

[50] C. Dan, U. Meier, and J. Schmidhuber, “Multi-column deep neural network for image classification,” 2012, pp. 3642–3649.

[51] W. Li, M. Zeiler, S. Zhang, Y. L. Cun, and R. Fergus, “Regularization of neural networks using dropconnect,” Journal of Machine Learning Research, vol. 28, no. 3, 2013, pp. 1058–1066.

[52] X. Zhou and M. Belkin, “Semi-supervised learning by higher order regularization,” in Proceedings of the 14th International Conference on Artificial Intelligence and Statistics (AISTATS). Fort Lauderdale, FL, USA: W&CP, 2011, pp. 892–900, volume 15 of JMLR.