Semi-Supervised Learning Using Multiple One-Dimensional Embedding Based Adaptive Interpolation

Jianzhong Wang
Department of Mathematics and Statistics, Sam Houston State University
Huntsville, TX 77341, USA
jzwang@shsu.edu.

Received 12 November 2014
Revised 15 March 2015
Accepted 20 March 2015

Abstract We propose a novel semi-supervised learning scheme using adaptive interpolation on multiple one-dimensional (1-D) embedded data. For a give high dimensional data set, we smoothly map it onto several different one-dimensional (1-D) sequences, so that the labeled subset is converted to a 1-D subset for each of these sequences. Applying the cubic interpolation of the labeled subset, we obtain a subset of unlabeled points, which are assigned to the same label in all interpolations. Selecting a proportion of these points at random and adding them to the current labeled subset, we build a larger labeled subset for the next interpolation. Repeating the embedding and interpolation, we enlarge the labeled subset gradually, and finally reach a labeled set with a reasonable large size, based on which the final classifier is constructed. We explore the use of the proposed scheme in the classification of handwritten digits and show promising results.

Keywords: multiple 1-D embedding; 1-D multi-embedding; interpolation; semi-supervised learning.

AMS Subject Classification: 22E46, 53C35, 57S20

1. Introduction

In recent years, analysis of high dimensional data is a major challenge for the statistics and machine learning communities. Many traditional statistical methods cannot directly applied to high dimensional data analysis. The usual machine learning framework often does not directly exploit the rich geometric structure of high dimensional data. To capitalize on the geometric structure, new learning approaches to high dimensional data have been developed. Their main theme is embedding high dimensional data into a lower dimensional manifold so that the analysis can be carried out on the reorganized data. Smooth ordering of image patches introduced in Ref. 16, 17, 18 originally was developed for image processing. The key idea of smooth-ordering is to deal with high-dimensional data in a one-dimensional framework. We find that this idea can also be applied to semi-supervised learning (SSL). The purpose of this paper is to introduce a novel SSL algorithm based on data 1-D representation. Let $X = \{\vec{x}_i\}_{i=1}^n \subset \mathbb{R}^m$ be a given data set in m-space,

where a distance $d(\cdot,\cdot)$ is defined. Assume that X is initially represented as a stack $\mathbf{x} = [\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_n]$. Let π be a permutation of the index set $\{1, 2, \cdots, n\}$, which induced a permutation operator $\mathbf{P}$ such that $\mathbf{Px} = [\vec{x}_{\pi(1)}, \vec{x}_{\pi(2)}, \cdots, \vec{x}_{\pi(n)}]$. Using the smooth ordering method proposed in Ref. 18, one finds a permutation operator $\mathbf{P}$, such that the stack $\hat{\mathbf{x}} = \mathbf{Px}$ has a nearly shortest path. Then the mapping $h : X \to \mathbb{R}$,

$$h(\vec{x}_{\pi(j)}) = t_j, \quad t_1 = 0, \quad t_{j+1} - t_j = d(\vec{x}_{\pi(j)}, \vec{x}_{\pi(j+1)}), \tag{1.1}$$

yields a 1-D embedding of X. For convenience, we usually re-scale the interval $[t_1, T_n]$ to $[0, 1]$ (by $t_j \to \frac{t_j}{t_n}$). Thanks to 1-D embedding, a function f on X can be substituted by the univariate function $\tilde{f} = f \circ h^{-1}$. Note that the 1-D embedding of X is not unique. To better reveal the features of f, we employ multiple 1-D embedding (or, shortly, 1-D multi-embedding) that consists of several different versions of 1-D embedding.

A typical problem of semi-supervised (binary) classification can be briefly described as follows: Assume that the members of a given data set X belong to two classes A and B, with the labels 1 and -1, respectively. Assume also that a labeled set, say $X_0 = \{\vec{x}_1, \vec{x}_2, \cdots, \vec{x}_{n_0}\}$, is given. That is, $f(\vec{x}_i) = y_i, 1 \le i \le n_0$, is known, where $y_j \in \{-1, 1\}$. We want to find a classifier $f : X \to \{-1, 1\}$ that labels all members in X. In practice, we extend f to a continuous function $f : X \to \mathbb{R}$, and use $\operatorname{sign} f(\vec{x})$ as the class label for $\vec{x} \in X$. The assumption of the continuity of f guarantees that similar members are most likely in the same class.

Based on the multiple 1-D embedding model, we propose the *Multiple One-Dimensional Embedding Based Adaptive Interpolation* (M1DEI) for semi-supervised learning. The main idea of M1DEI is the following: Let $T^i = \{t_1^i, t_2^i, \cdots, t_n^i\}, 1 \le i \le s$, be the i^{th} version of 1-D embedding of X, and $\tilde{f}_i$ be the univariate function on T^i, converted from f: $\tilde{f}_i = f \circ h_i^{-1}$. The labeled set X_0 now is mapped onto $T_0^i \subset T^i$. To estimate the target classifier f on X, we work on all s versions of univariate functions $\tilde{f}_i, 1 \le i \le s$. An interpolation of $\tilde{f}^i(T_0^i)$ on $[0, 1]$ produces an approximation of $\tilde{f}_i$, which is directly converted back to an approximation of f. If, at an unlabeled point $\vec{x} \in X$, all $\tilde{f}^i(h_i(\vec{x}))$ have the same sign, then $\vec{x}$ should have the label of that sign with a high probability. Adding a proportion of these points to the labeled set X_0, we build a larger labeled set X_1. Iterating the process, we will obtain a reasonable large updated labeled set, based on which the interpolation scheme produces a high-qualified classifier.

Many semi-supervised classification methods have been proposed in literature (see the monograph[4] and the survey paper[25]). Recently, transductive support vector machines, manifold regularization, and other graph-based methods become popular. They find the classifier as a solution of a minimization problem.[1,2,13] The methods based on data-tree are also very attractive.[7,12,15] The proposed scheme gives a different approach to semi-supervised learning.

The paper is organized as follows: In Section 2 we introduce the algorithm of multiple 1-D embedding based adaptive interpolation (M1DEI). In Section 3

we explore the use of the proposed approach to semi-supervised classification of handwritten digits, and show promising results comparing with other methods. In Section 4 we give the summary and the suggestions for future work.

2. Iterative Interpolation Scheme on Multiple 1-D Embedding

2.1. *Multiple 1-D Embedding*

The multiple 1-D embedding of $X \subset \mathbb{R}^m$ is a generalization of smooth sorting proposed in Ref. 18. Let the permutation operator $\mathbf{P}$ (derived from the index permutation π) give a solution of the following minimization problem:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{P}} \sum_{j=1}^{n-1} d(\hat{\vec{x}}_j, \hat{\vec{x}}_{j+1}), \quad \hat{\vec{x}}_i = \vec{x}_{\pi(i)}, \tag{2.1}$$

then the mapping h in (1.1) yields a 1-D embedding of X. In algorithms, we often re-scale the interval $[t_1, t_n]$ to $[0, 1]$, letting h be a function from X to $[0, 1]$.

The problem (2.1) has NP computational complexity. It can become very computationally expensive for large sets. As[18] does, we will apply the greedy algorithm to find an approximation of (2.1), in which the path search is restricted in neighborhood. Because our objects are more general than images, the neighborhood structure in this paper is different from Ref. 18. For an image patch, its neighbors are naturally defined as the spacial neighbors, that is, the pitches contain the nearby pixels. Since the data sets in SSL are data clouds in a space, we construct the neighborhood using the data graph.[7,3,22] Let the edge set E in the data graph $G = [X, E]$ be created by either k-nearest neighbor method or ϵ-neighbor method.[20] Then, the neighborhood of $\vec{x} \in X$ is the set $N(\vec{x}) = \{y \in X, (\vec{x}, \vec{y}) \in E\}$. When G is a complete graph, the neighborhood $N(\vec{x}) = X \setminus \{\vec{x}\}$ becomes the whole set X. This leads to the global greedy algorithm. If X is not too large, say $|X| \leq 10,000$, we suggest the global greedy algorithm, which usually provides a better approximation for the solution of (2.1).

Assume that a graph $G = [X, E]$ is constructed on X so that the neighborhood structure of the data set is well-defined. Then the greedy algorithm for finding a permutation operator P is very similar the that in Ref. 18. For readers' convenience, it is present here with the details. We first randomly create a probability vector $\mathbf{p} = [p_1, p_2, \cdots, p_n], 0 < p_i < 1$, for the path selection, whose role will be explained later. Then, we randomly select a node $\hat{\vec{x}}_1 = \vec{x}_{\pi(1)} \in X$ as the head for the embedding. Let Ω be the index set containing the indices of the selected nodes. Hence, $\Omega = \{\pi(1)\}$ at the initial step. Assume now that we already make $\Omega = \{\pi(1), \pi(2), \cdots, \pi(k)\}$, which is corresponding to the ordered stack $[\hat{\vec{x}}_1, \hat{\vec{x}}_2, \cdots, \hat{\vec{x}}_k]$, where $\hat{\vec{x}}_j = \vec{x}_{\pi(j)}$. Let $N(k)$ denote the neighborhood of $\hat{\vec{x}}_k$: $N(k) = \{\vec{x}; (\hat{\vec{x}}_k, \vec{x}) \in E\}$, and $N^c(k) = N(k) \setminus \Omega$. Let $|A|$ denote the cardinality of a set A. We now find $\hat{\vec{x}}_{k+1}$ as follows:

(1) If $|N^c(k)| \geq 2$, then we find the nearest node $\vec{y}_1$ and the second nearest node

$\vec{y}_2$ of $\hat{\vec{x}}_k$ in $N^c(k)$, and compute q_k using the following formula:

$$q_k = \frac{1}{1+\exp\left(\frac{d(\hat{\vec{x}}_k,\vec{y}_1)-d(\hat{\vec{x}}_1,\vec{y}_2)}{\alpha}\right)}, \tag{2.2}$$

where $\alpha > 0$ is a parameter. If $q_k < p_{\pi(k)}$, we select $\hat{\vec{x}}_{k+1} = \vec{y}_2$. Otherwise, select $\hat{\vec{x}}_{k+1} = \vec{y}_1$.

(2) If $|N^c(k)| = 1$, then we choose the only node in $N^c(k)$ as $\hat{\vec{x}}_{k+1}$.

(3) If $N^c(k) = \emptyset$, then we choose $\hat{\vec{x}}_{k+1}$ in the set $X^c = X \setminus \Omega$. If $|X^c| = 1$, then $k+1 = n$, so that only one node is unselected, we assign it as the last node $\hat{\vec{x}}_n$. Otherwise, in X^c we find the nearest and the second nearest nodes of $\hat{\vec{x}}_k$ and choose one of them as $\hat{\vec{x}}_{k+1}$ using the procedure described in Step (1).

After $\hat{\mathbf{x}}$ is constructed, the coordinate mapping h in (1.1) gives a 1-D embedding of X. If we select a different head $\hat{\vec{x}}_1$, we will get a different 1-D embedding. Let $h_i, 1 \le i \le s$, be s different coordinate mappings from X to $[0,1]$. The vector of mappings $\vec{h} = [h_1, h_2, \cdots, h_s]$ gives a multiple 1-D embedding of X.

Remark 2.1. When the dimension m is large, we often apply a dimensionality reduction (DR) algorithm to reduce the dimension of the data before we make the multiple 1-D embedding. Principal component analysis (PCA) is a popular linear method for DR. Recently, nonlinear DR methods were proposed and used widely in literature Ref.1, 5, 8, 10, 11, 14, 19, 21, 22, 23, 24. Note that DR can be considered as a pre-processing for semi-supervised learning. For convenience, in the paper, we will assume that the data set X is already dimensionally reduced by a certain DR algorithm.

The 1d embedding is not a embedding obtained by dimensionality reduction, but a data rearrangement. Readers an find the difference between (1-D) dimensionality reduction and (smooth) 1-D embedding.

2.2. *Finding Feasibly Confident Set Using Spinning Interpolation*

Let $\vec{h}$ be the multiple 1-D embedding described above, and $\mathbf{P}_i$ be the permutation operator corresponding to h_i. For the target function $f : X \to \mathbb{R}$, we define s univariate functions by $\tilde{f}_i = f \circ (\mathbf{P}_i^{-1} \circ h_i^{-1})$. Since f is given on the labeled set X_0, $\tilde{f}_i$ in given on the set $T_0^i = h_i(\mathbf{P}_i(X_0))$ by

$$\tilde{f}_i(t_j^i) = f(\vec{x}_{\pi^i(j)}), 1 \le j \le n_0.$$

We now choose an interpolation operator $\mathbf{L}$ on $[0,1]$, say the cubic interpolation, to interpolate each $\tilde{f}_i$ on the set T_0^i:

$$(\mathbf{L}\tilde{f}_i)(t_j^i) = \tilde{f}_i(t_j^i), 1 \le j \le n_0$$

which yields the following approximation of f:

$$f_i^* = (\mathbf{L}\tilde{f}_i) \circ (h_i \circ \mathbf{P}_i).$$

The vector $\vec{f}^* = [f_1^*, f_2^*, \cdots, f_s^*]$ records s different interpolations. We call $\vec{f}^*$ an *s-spinning interpolation* of f. Finally, we use the spinning average

$$f^*(\vec{x}) = E(\vec{f}^*(\vec{x})) = \frac{1}{s}\sum_{i=1}^{s} f_i^*(\vec{x}) \tag{2.3}$$

to approximate $f(\vec{x})$.

A single spinning interpolation usually does not provide a good enough approximation. Our proposed M1DEI robustly runs spinning interpolation serially for several times, of which each enlarges the labeled set in a certain scope. To choose the unlabeled points that should be added to the present labeled set, we construct the *feasibly confident set* for the $(k+1)^{th}$ spinning interpolation as follows: Let $\varepsilon > 0$ be a given tolerance and $X_k = \{\vec{x}_1, \cdots, \vec{x}_{n_k}\}$ be the present labeled set. Assume that after we run a term of spinning interpolation of f on X_k, we obtain the vector-valued function $\vec{f}^*$ on X. We denote by $\sigma(\vec{x})$ the standard deviation of the vector $\vec{f}^*(\vec{x})$. Then the set

$$L_k = \{\vec{x} \in X \setminus X_k;\ \sigma(\vec{x}) \leq \varepsilon\} \tag{2.4}$$

is called the *feasibly confident set* (of $\vec{f}^*$ with respect to X_k). We conjecture that, if $\vec{x} \in L_k$, then

$$|f^*(\vec{x}) - f(\vec{x})| \propto \varepsilon \tag{2.5}$$

with a high probability.

2.3. *Adaptive Interpolation Scheme*

M1DEI gradually enlarges the labeled set by iterating the spinning interpolation. If the scheme only adds *correctly labeled points* (on which (2.5) holds) to the present labeled set for the next spinning interpolation, then it will reduce the interpolation error. However, if some *mislabeled points* are added, then the error will be reinforced. To avoid this drawback, in M1DEI, we do not simply add the whole feasible confident set to the present labeled set. Instead, we pre-assign a probability $p, 0 < p < 1$, for "good point"selection. Then we randomly select a *newborn labeled subset* from the feasible confident set, say $U_k \subset L_k$ with $|U_k| = p|L_k|$, and build the new labeled set by $X_{k+1} = X_k \cup U_k$ for next spinning interpolation. We repeat the procedure above till the labeled set is large enough. Finally, the average of functions obtained from the last spinning interpolation gives the estimate of f.

Summary, an M1DEI consists of several serial terms of spinning interpolations, which produce the newborn labeled subsets to enlarge the labeled set gradually till it reaches a certain size. Finally the spinning interpolation based on the final updated labeled set gives the final classifier.

3. Classification of Handwritten Digits

We now apply the M1DEI to the classification of handwritten digits. Our experiments focus on two well-known handwritten digit data sets, MNIST and USPS. Let X be a dataset of n digits, randomly selected from either MNIST or USPS. In X only a small subset X_0 is labeled. In our experiments, we do not employ the DR as the pre-processing. Note that a digit in X is originally represented by a $c\times c$ matrix, where $c = 20$ for MNIST and $c = 16$ for USPS. To reduce the shift-variance, we define the (1-shift) distance between two digits $\vec{x} = (\vec{x}_{i,j})_{i,j=1}^{c}$ and $\vec{y} = (\vec{y}_{i,j})_{i,j=1}^{c}$ by

$$d(\vec{x},\vec{y}) = \min_{|i'-i|\le 1,|j-j'|\le 1} \sqrt{\sum_{i=2}^{c-1}\sum_{j=2}^{c-1}(\vec{x}_{i,j}-\vec{y}_{i',j'})^2}. \tag{3.1}$$

In the classification, we consider a digit in X as a $c\times c$ dimensional vector, which is extracted from the image column by column, starting from the most left one. Then, X can be represented as an $m\times n$ matrix ($m = c\times c$), whose columns are digits. We also write X as the stack $\mathbf{x} = [\vec{x}_1,\vec{x}_2,\cdots,\vec{x}_n]$, in which $\vec{x}_i$ is the vector form of i^{th} digit and the digits in the stack are arranged randomly.

3.1. *Classification of a subset of MNIST data set*

For comparison, we choose the same sizes of sample subsets from MNIST handwritten digits, as Ref. 12 does: For each of the digits $\{3,4,5,7,8\}$, 200 samples were selected at random so that $|X| = 1000$ and $X\subset\mathbb{R}^{400}$. Digits 8 were classified in Class A, labeled by $+1$, and the other digits were classified in Class B, labeled by -1. In each experiment, the initial labeled set X_0 has the size $10,20,\cdots,100$, respectively, and the labeled digits are distributed evenly on each digit. For example, when $|X_0| = 10$, two from each digit of $\{3,4,5,7,8\}$ are selected at random and labeled. We use the global greedy approximation for the multiple 1-D embedding of X, and set $\alpha = 1$ in (2.2), where $m = 400$.

In X, the ratio of digits in Class A to those in Class B is $r = 1:4$. In the experiment, we keep the same ratio in each updated labeled set. Assume that, after k terms of spinning interpolation, the labeled set is updated to X_k. In the $(k+1)^{th}$ term, interpolating f of date set X_k, we obtain vector-valued function $\vec{f}^{*} = [f_1^*,f_2^*,\cdots,f_s^*]$. We now define the feasible confident set L_k in a slightly different way as described in the previous section: We define $\sigma(\vec{x})$ by

$$\sigma(\vec{x}) = \frac{1}{s}\sum_{i=1}^{s}\sqrt{(\mathrm{sign}(f_i^*(\vec{x})) - E(\mathrm{sign}(\vec{f}^{*}(\vec{x}))))^2},$$

and define the feasible confident set of $\vec{f}^{*}$ with respect to X_k by

$$L_k = \{\vec{x}\in X\setminus X_k;\ \sigma(\vec{x}) = 0\}.$$

We divide L_k into two subsets as follows:

$$L_k^+ = \{\vec{x} \in L_k;\ E(\text{sign}(\vec{f}^*(\vec{x})) = 1\}, \tag{3.2}$$
$$L_k^- = \{\vec{x} \in L_k;\ E(\text{sign}(\vec{f}^*(\vec{x})) = -1\}. \tag{3.3}$$

According to a pre-assigned probability $p, 0 < p < 1$, we choose the subsets $U_k^+ \subset L_k^+$ and $U_k^- \subset L_k^-$ in such a way that

(a) the members in U_k^+ (U_k^-) are randomly selected from L_k^+ (L_k^-) with the cardinality restrictions:

$$|U_k^+| \le p|L_k^+|, \quad |U_k^-| \le p|L_k^-|; \tag{3.4}$$

and

(b) $|U_k^+| : |U_k^-| = r$.

Then, we construct the newborn labeled subset $U_k = U_k^+ \cup U_k^-$ and update the current labeled set to $X_{k+1} = X_k \cup U_k$ for next spinning interpolation. As mentioned in the previous section, in the last term of spinning interpolation, we compute the spinning average $E(\vec{f}^*)$ for obtaining the final classifier f.

Remark 3.1. The subsets L_k^+ in (3.2) and L_k^- in (3.3) can also be defined by

$$L_k^+ = \cap_{i=1}^s \{\vec{x} \in X \setminus X_k;\ f_i^*(\vec{x}) > 0\},$$
$$L_k^- = \cap_{i=1}^s \{\vec{x} \in X \setminus X_k;\ f_i^*(\vec{x}) < 0\}.$$

Remark 3.2. The ratio r for a given data set is known evidently. In a real-world problem, this rate usually can be learned statistically.

In the experiment, we choose $\alpha = 40$ in (2.2), spin 16 times for the last term of spinning interpolation, and spin 3 times for others. The iteration number (i.e. the terms of spinning interpolation) is decreasing as the initial labels increase from 10 to 100, whereas the parameter p in (3.4) is increasing as the initial labels increase. In every interpolation, the shape-preserving piecewise cubic interpolation is employed. We test our method over 50 subsets of MNIST, selected at random, and preset the results in Table 1, where the first column shows the numbers of initial labels, the second column gives the iteration numbers (i.e., the number of spinning interpolations), and the third column presents the relative errors, which are derived from the average of misclassified numbers over 50 randomly selected test subsets.

Fig. 3.1 gives the comparison of the average test errors of our method to Laplacian Eigenmaps (Belkin & Niyogi, 2003 Ref. 2), Laplacian Regularization (Zhu et al., 2003 Ref. 26), Laplacian Regularization with Adaptive Threshold (see Ref. 12), and Haar-Like Multiscale Wavelets on Data Trees (Gavish et al., 2010 Ref. 12). The results clearly show that our M1DEI method is superior to other methods on all cases.

Initial labels	Iterations	Parameter p	Misclassified (%)
10	26	0.50	7.94%
20	26	0.50	4.81%
30	25	0.55	4.33%
40	26	0.55	3.70%
50	21	0.60	3.63%
60	21	0.60	3.69%
70	21	0.60	3.04%
80	21	0.60	3.19%
90	18	0.65	3.01%
100	14	0.70	2.83%

Table 1. Test errors of M1DEI on the MNIST data subsets of 1000 digits

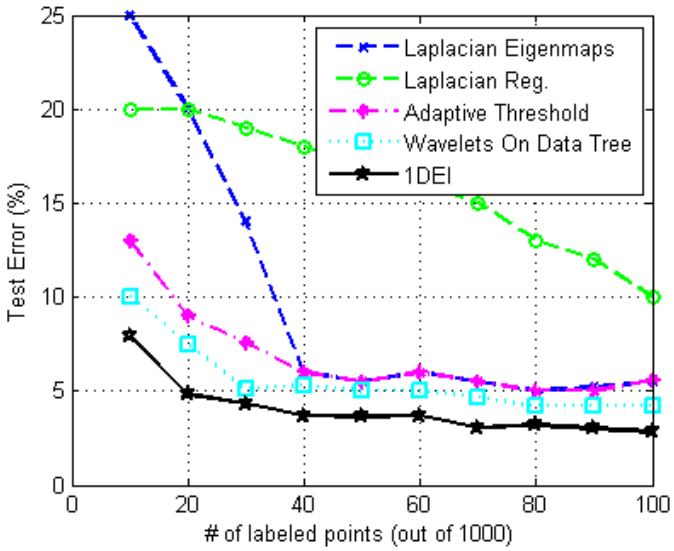

Fig. 1. Result Comparison of M1DEI to Other Methods on MNIST

3.2. *Classification of a subset of the USPS data set*

In the second experiment, we choose the sample subsets of USPS handwritten digits.[4,12] For each of the digits $0-9$, 150 samples are selected at random so that $|X| = 1500$. The size of the digit image is 16×16 so that $X \subset \mathbb{R}^{256}$. Our task is to distinguish the digits $\{2, 5\}$ from the rest, as[12] does. In the experiment, we still choose the size of the initial labeled set X_0 as $10, 20, \cdots, 100$, respectively, and the labeled digits are distributed evenly on each digit. The experiment is done in

the similar way as we do on MNIST subsets: The global greedy approximation is applied for 1-D embedding and the shape-preserving piecewise cubic spline is employed for every interpolation. In the experiment, we choose $\alpha = 1$ in (2.2), spin 16 times for the last term of spinning interpolation, and spin 3 times for others. The iteration number is decreasing as the initial labels increase from 10 to 100, whereas the parameter p in (3.4) is increasing as the initial labels increase. The test results are shown in Table 2.

Initial labels	Iterations	Parameter p	Misclassified (%)
10	29	0.40	9.70%
20	28	0.45	5.85%
30	28	0.50	3.87%
40	28	0.50	3.45%
50	21	0.50	3.39%
60	21	0.55	3.10%
70	21	0.55	2.37%
80	19	0.60	2.17%
90	19	0.72	2.04%
100	19	0.68	2.29%

Table 2. Test errors of M1DEI on USPS data subsets of 1500 digits

We compare our methods to Laplacian Eigenmaps, Laplacian Regularization, Laplacian Regularization with Adaptive Threshold, and Haar-Like Multiscale Wavelets on Data Trees. The comparison of the average test errors of the different methods are shown in Fig. 3.2. The results again show that our method is superior to other methods on all cases.

4. Conclusion

We propose a new scheme for semi-supervised learning based on data multiple 1-D embedding, which reveals data features from several different view angles in a relatively simple framework. We construct the multiple 1-D embedding by randomly spinning on the data points. M1DEI adaptively enriches the labeled set to achieve a high qualified classifier. The experiments show that the performance of the proposed M1DEI is superior to many popular methods in semi-supervised learning. The new scheme also exhibits a clear advantage for learning the classifier when only a small labeled set is given.

The proposed scheme can be also employed in the classification of other objects than handwritten digits, such as hyperspectral images and medical images. The applications of the scheme in image recognition and other areas will be in other papers written by the author and his collaborators.

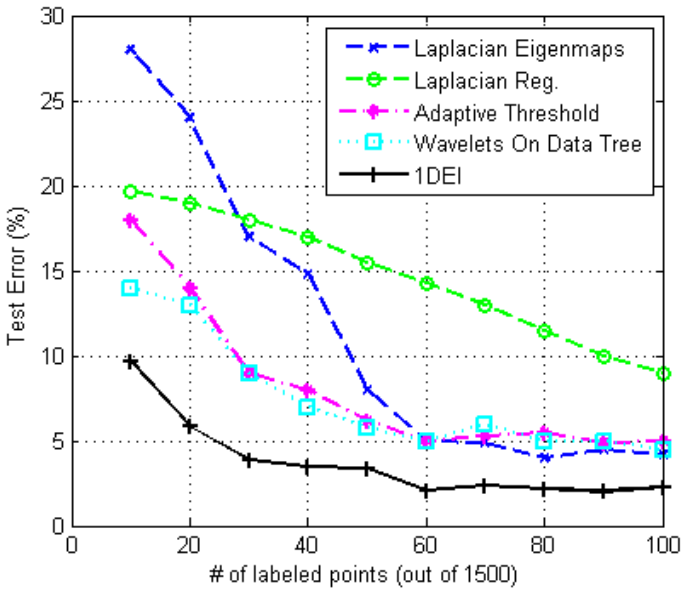

Fig. 2. Result Comparison of M1DEI to Other Methods on SPUS

References

1. M. Belkin and P. Niyogi, Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation* Vol. 15 No. 6, pp. 1373-1396, 2003.
2. M. Belkin and P. Niyogi, Using manifold structure for partially labeled classification. *Advances in Neural Information Processing Systems* Vol. 15, pp. 929-936, 2003.
3. J. Bondy and U. Murty, *Graph Theory*. Springer, 2008.
4. Olivier Chapelle, Alexander Zien, Bernhard Schölkopf (Eds.), *semi-supervised learning*. MIT Press, 2006.
5. C.K. Chui and J. Wang, Randomized Anisotropic Transform for Nonlinear Dimensionality Reduction, *International Journal on Geomathematics*, Vol. 1 No. 1, pp. 23-50, 2010.
6. R. R. Coifman and D. L. Donoho, *Wavelets and Statistics*. Springer- Verlag, 1995, ch. Translation-invariant de-noising, pp. 125-150.
7. R.R. Coifman and M. Gavish, Harmonic Analysis of Digital Data Bases. Preprint, 2011.
8. R.R. Coifman and S. Lafon, Diffusion maps. *Appl. Comput. Harmon. Anal.* Vol. 21, pp. 5-30, 2006.
9. T. H. Cormen, *Introduction to algorithms*. The MIT press, 2001.
10. T.F. Cox and M.A.A. Cox,*Multidimensional Scaling*. Chapman & Hall, London, New York, 1994.
11. D.L. Donoho and C. Grimes, Hessian eigenmaps: New locally linear embedding techniques for high-dimensional data. *Proc. Natl. Acad. Sci. USA* Vol. 100, pp. 5591-5596, 2003.
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. *Proceedings of the* 27^{th} *International Conference on Machine Learning*, Haifa, Israel, 2010.
13. T. Joachims, Transductive learning via spectral graph partitioning. *Proceedings of ICML-03, 20th International Conference on Machine Learning.* 2003.
14. I.T. Jolliffe,*Principal Component Analysis.* Springer Series in Statistics. Springer-Verlag, 1986.
15. C. Kemp, T. Griffiths, S. Stromsten, and J. Tenenbaum, semi-supervised learning with trees. *Advances in Neural Information Processing System* Vol. 16. 2003.
16. I. Ram, M. Elad and I. Cohen, Generalized Tree-Based Wavelet Transform,*IEEE Trans. Signal Processing*, Vol. 59, no. 9, pp. 4199-4209, 2011.
17. I. Ram, M. Elad and I. Cohen, Redundant Wavelets on Graphs and High Dimensional Data Clouds,*IEEE Trans. Signal Processing Letters*, Vol. 59, no. 5, pp. 291-294, May 2012.
18. I. Ram, M. Elad and I. Cohen, Image Processing using Smooth Ordering of its Patches,*IEEE Trans. on Image Processing*, Vol. 22, no. 7, pp. 2764-2774, July 2013.
19. S.T. Roweis and L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding. *Science* Vol. 290 No. 5500, pp. 2323-2326, 2000.
20. G. Shakhnarovich, T. Darrell, and P. Indyk,*Nearest-Neighbor Methods in Learning and Vision, Theory and Practice.* MIT Press, 2006.
21. J.B. Tenenbaum, V. de Silva, and J.C. Langford, A global geometric framework for nonlinear dimensionality reduction. *Science* Vol. 290 No. 5500, pp. 2319-2323, 2000.
22. J. Wang, *Geometric Structure of High-Dimensional Data and Dimensionality Reduction.* Springer, jointly published with Higher Education Express, 2011.
23. K.Q. Weinberger, B.D. Packer, and L.K. Saul, Nonlinear dimensionality reduction by semidefinite programming and kernel matrix factorization. In *Proc. of the 10th International Workshop on AI and Statistics*, 2005.
24. Z.Y. Zhang, and H.Y. Zha, Principal manifolds and nonlinear dimensionality reduction via tangent space alignment. *SIAM J. Sci. Comput.* Vol. 26 No. 1, pp. 313-338, 2004.
25. Xiaojin Zhu, Semi-supervised learning literature survey. *Computer Sciences TR-1530.* University of Wisconsin-Madison. Last modification. July 2008.
26. X. Zhu, Z.Ghahramani, and J. Lafferty, Semi-supervised learning using gaussian fields and harmonic functions. In *International Conference on Machine Learning*, Vol. 13, 2003.
27. X. Zhu, X., Kandola, J., Ghahramani, Z., and Lafferty, J. (2005). Nonparametric transforms of graph kernels for semi-supervised learning. In L. K. Saul, Y. Weiss and L. Bottou (Eds.), *Advances in neural information processing systems (nips)* 17. Cambridge, MA. MIT Press.