Embedding time series data for classification

Akira Hayashi[1], Yuko Mizuhara[2], and Nobuo Suematsu[1]

[1] Faculty of Information Sciences, Hiroshima City University
3-4-1 Ozuka-higashi, Asaminami-ku, Hiroshima 731-3194, Japan,
akira@im.hiroshima-cu.ac.jp
[2] Panasonic AVC Networks,
1-5 Matsuo-Cho, Kadoma, 571-8505, Japan

Abstract. We propose an approach to embed time series data in a vector space based on the distances obtained from Dynamic Time Warping (DTW), and to classify them in the embedded space. Under the problem setting in which both labeled data and unlabeled data are given beforehand, we consider three embeddings, embedding in a Euclidean space by MDS, embedding in a Pseudo-Euclidean space, and embedding in a Euclidean space by the Laplacian eigenmap technique.
We have found through analysis and experiment that the embedding by the Laplacian eigenmap method leads to the best classification result. Furthermore, the proposed approach with Laplacian eigenmap embedding shows better performance than k-nearest neighbor method.

1 Introduction

1.1 Classification of time series data

With the development of information technology, recognition of time series data, such as gesture recognition, video retrieval, online handwriting recognition, is becoming more important. Here, we consider the following 2 class classification problem for time series data.

A set of n time series data, $\mathcal{X} = \{X_1, \ldots, X_n\}$, is given, where, X_i $(1 \leq i \leq n)$ is a sequence of feature vectors whose length is l_i:$X_i = (\boldsymbol{x}_1^i, \ldots, \boldsymbol{x}_{l_i}^i)$. First s of the time series data, $\{X_i \mid 1 \leq i \leq s\}$, are labeled with a class label $y_i \in \{-1, +1\}$. The task is to estimate the labels of unlabeled data: $\{X_i \mid s+1 \leq i \leq n\}$.

Time series data are much more difficult to deal with than vector data with a fixed dimension, Many of the classification methods for time series data use generative models such as Hidden Markov Models (HMMs) [1]. When the true models are estimated correctly, these methods are accurate. But they needs lots of training data. Other classification methods such as k nearest neighbors are based on distances which are obtained from dynamic time warping (DTW) [1]. K nearest neighbors can express class boundaries which have complex shapes without assuming the form of probability densities, but they tend to be sensitive to noise in general .

1.2 Proposed approach

We propose a distance based approach which can be summarized as follows.

1. Compute the distances between time series data from DTW.
2. Map by $\boldsymbol{\Phi}$ the time series data into a vector space (a feature space) $\boldsymbol{\mathcal{F}}$, such that the DTW distances are preserved in some sense.

$$\begin{aligned} \boldsymbol{\Phi} : \mathcal{X} &\to \boldsymbol{\mathcal{F}} \\ X_i &\mapsto \boldsymbol{\Phi}(X_i) \end{aligned}$$

   Project $\boldsymbol{\Phi}(X_i)$ to a lower dimensional subspace, and obtain $\tilde{\boldsymbol{\Phi}}(X_i)$.
3. Train a classifier in $\boldsymbol{\mathcal{F}}$ using the labeled data $\{(\tilde{\boldsymbol{\Phi}}(X_i), y_i) \mid 1 \leq i \leq s\}$.
4. Classify the unlabeled data, $\{\tilde{\boldsymbol{\Phi}}(X_i) \mid s+1 \leq i \leq n\}$, using the classifier.

1.3 Embedding in a vector space

We consider three methods, i.e. three kinds of mapping $\boldsymbol{\Phi}$, for embedding time series data in a vector space. The first method is multidimensional scaling (MDS) [2], which embeds data in a Euclidean space. The second method, which is an extension of the first one, embeds data in a pseudo Euclidean space [3–5]. The third method uses the technique known as manifold learning [6, 7], and embeds time series data in a Euclidean space.

We consider the three embedding methods from the following reasons. MDS (the first method) is popular as an embedding method using distances between data. We consider the embedding in a pseudo Euclidean space (in the second method), because dynamic time warping distances do not satisfy the triangle inequality relationship for (Euclidean) distances. For the third embedding method, we have chosen, from several manifold learning techniques, one which can be applied to non vector data. Generally speaking, manifold learning embeds data on a manifold in a low dimensional space such that the geodesic distances are preserved. Manifold learning is gaining more and more attentions recently as a nonlinear dimensionality reduction method.

In order to analyze the three embedding methods, we employ the theoretical framework of kernel PCA [8], which is an extension of principal component analysis (PCA).

1.4 Related work

Shimodaira et al. [9] propose dynamic time alignment kernel for voice recognition, and report better classification accuracy than HMMs when the number of training data is small. Bahlmann et al.[10] propose GDTW kernel, which substitutes the distance term in the Gaussian kernel with DTW distance, and obtained classification accuracy comparable with that of HMMs for online handwritten characters. However, neither method can prove the positive definiteness of the corresponding kernel matrix which guarantees the existence of a feature space (a Hilbert space).

Graepel et al. [4] embed data in a pseudo Euclidean space on the basis of the similarity measures of pairs of data, and then classify the embedded data using SVM. They experimented with cat's cortex data and protein data, and obtained a favorable result when compared with k-nearest neighbors. Pekalska et al. [5] propose a similar method and report a good result in the experiment to classify offline handwritten characters / object shapes in binary images. But, neither of them consider the classification of time series data.

Belkin et al. [6, 7] propose Laplacian eigenmap method to embed data in a low dimensional vector space based on a similarity matrix. They have experimented with offline handwritten digit classification and report a better result than k-nearest neighbors. But they do not consider time series data, either.

1.5 Paper Organization

After briefly explaining DTW (Sec. 2) and kernel PCA (Sec. 3), we propose the embedding methods in Sec. 4, compare the distance between data before and after the embeddings in Sec. 5, and explain the classification including the classifier learning in Sec. 6. We report on the experiment to evaluate the proposed methods in Sec. 7. Sec. 8 concludes the paper.

2 DTW

The DTW distances $\{d^2(X_i, X_j) \mid 1 \le i, j \le n\}$ which we use in the paper are computed as follows ($\|\cdot\|$ is the Euclidean norm.)

1. Initialize: $g(0, 0) = 0$
2. Repeat: for $1 \le t_i \le l_i,\ \ 1 \le t_j \le l_j$

$$g(t_i, t_j) = \min \begin{cases} g(t_i - 1, t_j) + \|\boldsymbol{x}^i_{t_i} - \boldsymbol{x}^j_{t_j}\|^2 \\ g(t_i - 1, t_j - 1) + 2\|\boldsymbol{x}^i_{t_i} - \boldsymbol{x}^j_{t_j}\|^2 \\ g(t_i, t_j - 1) + \|\boldsymbol{x}^i_{t_i} - \boldsymbol{x}^j_{t_j}\|^2 \end{cases}$$

3. Finish: $d^2(X_i, X_j) = g(l_i, l_j)/(l_i + l_j)$

3 Kernel PCA

We explain kernel PCA by following [8]. Let $\mathcal{X} = \{X_1, X_2, \ldots\}$ be a finite or an infinite set (which need not be a subset of a vector space), and let k be a positive definite kernel function defined on $\mathcal{X} \times \mathcal{X}$. Then, there exists a mapping to a Hilbert space, $\boldsymbol{\Phi} : \mathcal{X} \to \boldsymbol{\mathcal{H}}$, and for any $X, X' \in \mathcal{X}$, $k(X, X') = \langle \boldsymbol{\Phi}(X), \boldsymbol{\Phi}(X') \rangle$ holds, where $\langle ., . \rangle$ stands for the inner product in the Hilbert space $\boldsymbol{\mathcal{H}}$.

Kernel PCA performs the principal component analysis of the set: $\{\boldsymbol{\Phi}(X_1)$, $\boldsymbol{\Phi}(X_2)$, $\ldots$, $\boldsymbol{\Phi}(X_n)\}$. Let $\boldsymbol{C}$ be the covariance matrix: $\boldsymbol{C} = \frac{1}{n} \sum_i \boldsymbol{\Phi}(X_i)\boldsymbol{\Phi}(X_i)^T$, and let λ and $\boldsymbol{v}$ be an eigenvalue and eigenvector of the covariance matrix:

$\boldsymbol{C}\boldsymbol{v} = \lambda\boldsymbol{v}$. Then, it can be shown that an eigenvector whose eigenvalue is not 0 is in the subspace spanned by $\{\boldsymbol{\Phi}(X_i) \mid 1 \le i \le n\}$, and hence can be expanded as $\boldsymbol{v} = \sum_{i=1}^{n} \alpha_i \boldsymbol{\Phi}(X_i)$. The expansion coefficients, $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n)^T$, are obtained from the eigenvectors of the Kernel matrix $\boldsymbol{K} : \boldsymbol{K}_{ij} = k(X_i, X_j)$ as follows.

$$n\lambda\boldsymbol{\alpha} = K\boldsymbol{\alpha} \tag{1}$$

As a consequence, it can be shown that m-th principal component of $\boldsymbol{\Phi}(X_i)$ can be computed by the following formula:

$$\langle \boldsymbol{v}^m, \boldsymbol{\Phi}(X_i)\rangle = \sqrt{n\lambda_m}\boldsymbol{\alpha}^m(i) \qquad (1 \le i \le n) \tag{2}$$

where λ_m is an eigenvalue of the matrix $\boldsymbol{K}$, $\boldsymbol{\alpha}^m(i)$ is the i-th element of the m-th eigenvector $\boldsymbol{\alpha}^m$. Note that we have assumed that $\frac{1}{n}\sum_{i=1}^{n} \boldsymbol{\Phi}(X_i) = \boldsymbol{0}$ in computing the covariance matrix, $\boldsymbol{C}$, but this can be easily realized by *centering* the kernel matrix $\boldsymbol{K}$.

4 Embedding in a vector space

4.1 First method: MDS

The first method uses MDS[2] to obtain a mapping $\boldsymbol{\Phi}_1 : \mathcal{X} \to \Re^n$ such that the following holds.

$$\|\boldsymbol{\Phi}_1(X_i) - \boldsymbol{\Phi}_1(X_j)\|^2 = d^2(X_i, X_j) \qquad (1 \le i, j \le n) \tag{3}$$

We abbreviate $\boldsymbol{\Phi}_1(X_i)$ as $\boldsymbol{z}_i$ in what follows. In order to centralize the kernel matrix, let $\overline{\boldsymbol{z}} = \frac{1}{n}\sum_i \boldsymbol{z}_i$, and consider the inner product: $k_1(X_i, X_j) = \langle \boldsymbol{z}_i - \overline{\boldsymbol{z}}, \boldsymbol{z}_j - \overline{\boldsymbol{z}}\rangle$. Using (3), we can obtain, after some manipulations, a formula to compute the kernel matrix from DTW distances.

$$\begin{aligned} k_1(X_i, X_j) = -\frac{1}{2}d^2(X_i, X_j) + \frac{1}{2n}\sum_{l=1}^{n} d^2(X_i, X_l) + \frac{1}{2n}\sum_{l=1}^{n} d^2(X_j, X_l) \\ -\frac{1}{2n^2}\sum_{l=1}^{n}\sum_{m=1}^{n} d^2(X_l, X_m) \end{aligned} \tag{4}$$

The kernel matrix $\boldsymbol{K} : \boldsymbol{K}_{ij} = k_1(X_i, X_j)$ is decomposed through eigenvalue analysis as follows.

$$\boldsymbol{K} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{U}^T \tag{5}$$

where $\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \ldots, \lambda_n), \lambda_1 \ge \ldots \ge \lambda_n$ is a diagonal matrix of the eigenvalues, and $\boldsymbol{U} = [\boldsymbol{e}^1, \ldots, \boldsymbol{e}^n]$ is a matrix of the eigenvectors.

When $\boldsymbol{K}$ is semi-positive definite, let $\boldsymbol{Z} = \boldsymbol{\Lambda}^{\frac{1}{2}}\boldsymbol{U}^T$, and then $\boldsymbol{K} = \boldsymbol{Z}^T\boldsymbol{Z}$ holds. Hence, we can view the i-th column of $\boldsymbol{Z}$ as $\boldsymbol{z}_i - \overline{\boldsymbol{z}}$. Translate the origin to the centroid, keep up-to the p-th principal components, we obtain the following.

$$\tilde{\boldsymbol{z}}_i = (\sqrt{\lambda_1}\boldsymbol{e}^1(i), \sqrt{\lambda_2}\boldsymbol{e}^2(i), \ldots, \sqrt{\lambda_p}\boldsymbol{e}^p(i))^T \qquad (1 \le i \le n) \tag{6}$$

Note that the above (6) and (2) are identical up to a constant ($\sqrt{n}$).

Unfortunately, DTW distances do not satisfy the triangle inequality, and the matrix $\boldsymbol{K}$ computed from (4) is not necessarily semi-positive definite. Nevertheless, the first method embeds data in the Euclidean space, simply by neglecting negative eigenvalues / vectors.

4.2 Second method: embedding in a pseudo Euclidean space

Like the first method, $\boldsymbol{K}$ is computed from DTW distances using (4), and is eigen-decomposed as in (5). While the first method embeds data by viewing $\boldsymbol{K}$ as a matrix of inner products in a Euclidean space, the second method embeds data by viewing $\boldsymbol{K}$ as a matrix of symmetric bilinear forms in $\Re^{(n^+,n^-)}$, a pseudo Euclidean space [3], without neglecting negative eigenvalues / vectors [3–5].

The concrete embedding procedure is as follows. Take the absolute value $\bar{\boldsymbol{\Lambda}}$ of the eigenvalue matrix $\boldsymbol{\Lambda}$ in (5): $\bar{\boldsymbol{\Lambda}} = \mathrm{diag}(|\lambda_1|, |\lambda_2|, \ldots, |\lambda_n|)$. Let $\boldsymbol{Z}$ be $\bar{\boldsymbol{\Lambda}}^{\frac{1}{2}}\boldsymbol{U}^T$, choose p eigenvalues / vectors whose absolute values are the largest. The coordinates in the embedded space will be as follows.

$$\tilde{\boldsymbol{z}}_i = (\sqrt{|\lambda_{m_1}|}\boldsymbol{e}^{m_1}(i), \ldots, \sqrt{|\lambda_{m_p}|}\boldsymbol{e}^{m_p}(i))^T \qquad (1 \le i \le n) \qquad (7)$$

We explain briefly about a pseudo Euclidean space [4]. A pseudo Euclidean space $\Re^{(n^+,n^-)}$ is a vector space with the following bilinear form : $\langle\cdot,\cdot\rangle_{\boldsymbol{M}}$, which corresponds to the inner product in a Euclidean space.

$$\langle \boldsymbol{z}, \boldsymbol{z}' \rangle_{\boldsymbol{M}} = \boldsymbol{z}^T \boldsymbol{M} \boldsymbol{z}'$$

$$\boldsymbol{M} = \begin{pmatrix} \boldsymbol{I}_{n^+} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{I}_{n^-} \end{pmatrix}_{n \times n}$$

M is called as the signature matrix of a pseudo Euclidean space. For a pseudo Euclidean space, one can define, from its bilinear form $\langle\cdot,\cdot\rangle_{\boldsymbol{M}}$, a pseudo metric (distance): $\|\boldsymbol{z}-\boldsymbol{z}'\|^2_{\boldsymbol{M}} = (\boldsymbol{z}-\boldsymbol{z}')^T\boldsymbol{M}(\boldsymbol{z}-\boldsymbol{z}')$. The second method seeks a mapping $\boldsymbol{\Phi}_2 : \mathcal{X} \to \Re^{(n^+,n^-)}$ which satisfies

$$\|\boldsymbol{\Phi}_2(X_i) - \boldsymbol{\Phi}_2(X_j)\|^2_{\boldsymbol{M}} = d^2(X_i, X_j) \qquad (1 \le i, j \le n) \qquad (8)$$

4.3 Third method: the Laplacian eigenmap

The third method embeds time series data by employing the Laplacian Eigenmap technique [6, 7]. The procedure is as follows.

[3] n^+, n^- in $\Re^{(n^+,n^-)}$ stands for the number of positive and negative eigenvalues of $\boldsymbol{K}$, respectively.

[4] In the literature on pattern recognition, a feature space generally means a Hilbert space. The second method, however, considers a pseudo Euclidean space also as a feature space

1. Compute the similarity matrix $\boldsymbol{W}$ from DTW distances.
$$\boldsymbol{W}_{ij} = \begin{cases} e^{-d^2(X_i,X_j)/t} & i \neq j \wedge d(X_i, X_j) < \epsilon \\ 0 & \text{if otherwise} \end{cases} \tag{9}$$
where $t(> 0)$ is a hyper parameter.
2. Compute the Laplacian Matrix $\boldsymbol{L}$.
$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$,$\boldsymbol{D}$ is a diagonal matrix such that $\boldsymbol{D}_{ii} = \sum_{j=1}^{n} \boldsymbol{W}_{ij}$.
3. Solve a generalized eigenvalue problem: $\boldsymbol{Le} = \lambda \boldsymbol{De}$, and compute p smallest eigenvectors $\boldsymbol{e}^1, \ldots, \boldsymbol{e}^p$ $(\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_p)$ [5] .
4. Compute the coordinates in the embedded space.
Let $\tilde{\boldsymbol{U}} = \left[\boldsymbol{e}^1, \boldsymbol{e}^2, \ldots, \boldsymbol{e}^p\right]$, and $\tilde{\boldsymbol{Z}} = [\tilde{\boldsymbol{z}}_1, \ldots, \tilde{\boldsymbol{z}}_n] = \boldsymbol{U}^T$, that is to say
$$\tilde{\boldsymbol{z}}_i = (\boldsymbol{e}^1(i), \boldsymbol{e}^2(i), \ldots, \boldsymbol{e}^p(i))^T \qquad (1 \leq i \leq n) \tag{10}$$

5 Distances before and after the embedding

We investigate how the distances between time series data change when they are embedded in vector spaces.

As we can see clearly from Eq. (3) and (8), the first and the second methods perform, so to speak, a *global* embedding which maintains both short DTW distances and long DTW distances equally.

Next, let us consider the meaning of the solution for the generalized eigenvalue problem: $\boldsymbol{Le} = \lambda \boldsymbol{De}$ in the third method [6]. To begin with, seek a mapping from time series data to p dimensional space, $\Psi : \mathcal{X} \rightarrow \mathcal{R}^p$ $(X_i \mapsto \tilde{\boldsymbol{z}}_i)$, such that $\sum_i \sum_j \|\tilde{\boldsymbol{z}}_i - \tilde{\boldsymbol{z}}_j\|^2 \boldsymbol{W}_{ij}$ will be minimized. In other words, we try to map those time series which are close to each other (in DTW distances) to nearby points in $\mathcal{R}^p$. Let $n \times p$ matrix $\tilde{\boldsymbol{U}}$ be such that $\tilde{\boldsymbol{U}}^T = [\tilde{\boldsymbol{z}}_1 \tilde{\boldsymbol{z}}_2 \ldots \tilde{\boldsymbol{z}}_n]$, then it can be shown that the following holds [6].
$$\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \|\tilde{\boldsymbol{z}}_i - \tilde{\boldsymbol{z}}_j\|^2 \boldsymbol{W}_{ij} = \mathrm{tr}(\tilde{\boldsymbol{U}}^T \boldsymbol{L} \tilde{\boldsymbol{U}}) \tag{11}$$
Hence, we are left with the following minimization problem.
$$\tilde{\boldsymbol{U}}^* = \mathrm{argmin}_{\tilde{\boldsymbol{U}}^T \boldsymbol{D} \tilde{\boldsymbol{U}} = I} \mathrm{tr}(\tilde{\boldsymbol{U}}^T \boldsymbol{L} \tilde{\boldsymbol{U}}) \tag{12}$$

It is well known that the above minimization problem is reduced to finding the p smallest eigenvalues / vectors for a generalized eigenvalue problem: $\boldsymbol{Le} = \lambda \boldsymbol{De}$. Since the distances after the embedding are weighted according to the similarity $\boldsymbol{W}_{ij}$ in (11), the third method performs, so to speak, a *local* embedding which maintains short DTW distances, neglecting long distances.

[5] It has recently been pointed out [11] that $\boldsymbol{K}_L = \boldsymbol{L}^\dagger$, i.e., the pseudo inverse of the Laplacian matrix $\boldsymbol{L}$ is the kernel matrix for the Laplacian eigenmap technique. Therefore, embedding using the eigenvectors with the smallest eigenvalues of $\boldsymbol{L}$ is equivalent to kernel PCA for the kernel matrix: $\boldsymbol{K}_L$.

[6] From (11), we can see that $\boldsymbol{L}$ is semi-positive definite.

6 Classification

6.1 Classifier training

We consider here linear classifiers only for simplicity.

$$f(X) = \langle \boldsymbol{w}, \tilde{\boldsymbol{\Phi}}(X) \rangle + b = \langle \boldsymbol{w}, \tilde{\boldsymbol{z}} \rangle + b \tag{13}$$

The term $\langle \boldsymbol{w}, \tilde{\boldsymbol{\Phi}}(X) \rangle$ in the above should be $\langle \boldsymbol{w}, \tilde{\boldsymbol{\Phi}}(X) \rangle_{\boldsymbol{M}}$ for the second method. However, by setting $\boldsymbol{w}' = \boldsymbol{M}\boldsymbol{w}$, and considering $\langle \boldsymbol{w}, \tilde{\boldsymbol{z}} \rangle_{\boldsymbol{M}} = \langle \boldsymbol{w}', \tilde{\boldsymbol{z}} \rangle$, pseudo Euclidean coordinates will be treated as Euclidean coordinates from now on [4].

Least Mean Square Error Here, we seek a linear classifier which minimizes the mean square error for the labeled data: $\{(\tilde{\boldsymbol{\Phi}}(X_i), y_i) \mid 1 \leq i \leq s\}$. In other words, the following error function in terms of $\boldsymbol{w}, b$ should be minimized.

$$\mathrm{Err}(\boldsymbol{w}, b) = \sum_{i=1}^{s} \{y_i - (\langle \boldsymbol{w}, \tilde{\boldsymbol{z}}_i \rangle + b)\}^2 \tag{14}$$

Maximal Margin A hyperplane which has the maximal margin will be obtained by minimizing $\|\boldsymbol{w}\|^2$ under the following constraint [12]

$$y_i(\langle \boldsymbol{w}, \tilde{\boldsymbol{z}}_i \rangle + b) \geq 1, \quad i = 1, \ldots, s \tag{15}$$

6.2 Classifying unlabeled data

Let $f(X) = \langle \boldsymbol{w}^*, \tilde{\boldsymbol{z}} \rangle + b^*$, and for $X_i (i > s)$,

$$y_i = \begin{cases} 1, & \text{if } f(X_i) \geq 0 \\ -1, & \text{if } f(X_i) < 0 \end{cases} \tag{16}$$

7 Experiments

7.1 Experiment 1

We compare the three embedding methods using Australian sign language (ASL) data [13]. The ASL data consists of 95 signs obtained from 5 subjects, each of which is a sequence of 9 dimensional feature vectors. We picked up two pairs, "sad" and "what", "go" and "please" among the 95 signs, and classified each pairs into two classes. Each sign class has 70 samples.

As to the third method, the similarity matrix was computed by using 8 nearest neighbors instead of ϵ distance neighbors. The value for t, $t = 10000$, was determined experimentally.

We varied the total number of training data. The rest were used as test data. The dimensionality of the embedded space, p, was set to 20% of the number of

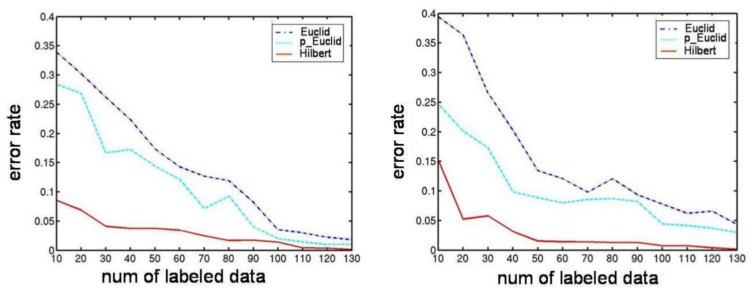

Fig. 1. Experiment 1 "sad" vs "what" (left) and "go" vs "please" (right). Average error rate from 30 trials are plotted. Solid lines (Hilbert) are for the third method.

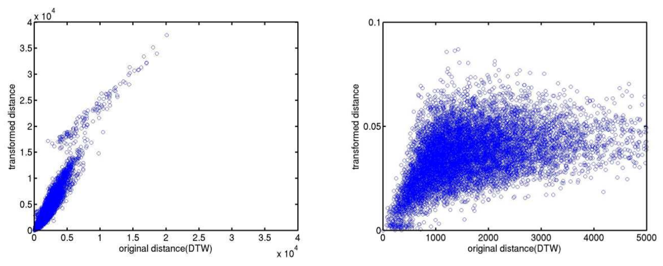

Fig. 2. Distances before and after the embedding: 1st Method (left) and 3rd Method (right) for 140 data of "go" and "please" signs

the training data. We used the maximal margin linear classifier for the first and second method, and the least mean squared error classifier for the third method, about which good results have been reported [4, 5, 7].

Fig. 1 shows that the third method has the best accuracy. Let us consider the reason. Since DTW distances are originally pattern matching scores, short distances which show a good match tend to be reliable, but long distances tend to be unreliable. While the first and the second method maintain both short and long distances equally alike in the embeddings, the third method tries to maintain only short distances by putting more weights on short distances. We thus conjecture that the third method has yielded the best result because it is compatible with the above nature of DTW distances (See also Fig. 2, which supports the analysis in Sec. 5 on distances.)

7.2 Experiment 2

In the second experiment, we compare the third method: Laplacian eigenmap embedding, which had the highest accuracy in the first experiment, with k near-

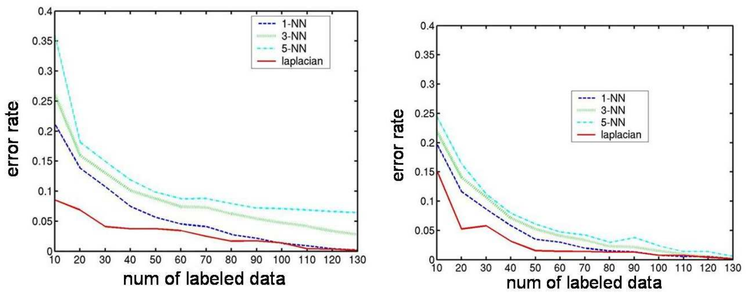

Fig. 3. Experiment 2: "sad" vs "what" (left) and "go" vs "please" (right). The average error rate is plotted for the same ASL data used in the first experiment.

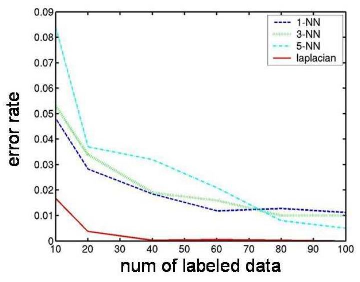

Fig. 4. Experiment 2: "upward shift" vs "increasing trend" in Control Chart Time Series [13].

est neighbors ($k = 1, 3, 5$). K nearest neighbors also use DTW distances, but do not embed data.

The results in Fig. 3 and 4 show that the third method has the best classification accuracy. We conjecture the reasons as follows. Firstly, the third method uses a linear classifier, which is expected to be more robust to noise than k nearest neighbors. Secondly, the third method seems to use unlabeled data effectively [14], because the coordinates of test data in the embedded space are determined by the DTW distances not only to the labeled data (the training data), but also to the unlabeled data (the test data), as far as they are within 8 nearest neighbors.

8 Conclusion

We have proposed an approach to embed time series data in a vector space based on the distances obtained by Dynamic Time Warping, and to classify them in the embedded space. Under the problem setting in which both labeled data and unlabeled data are given beforehand, we have considered three embeddings, embedding in a Euclidean space by MDS, embedding in a Pseudo-Euclidean space, and embedding in a Euclidean space by the Laplacian eigenmap technique. We have found that embedding by the Laplacian eigenmap technique leads to the best classification result. Furthermore, the proposed approach with Laplacian eigenmap embedding shows better performance than k-nearest neighbors.

References

1. L. Rabiner and B. Juang, Fundamentals of Speech Recognition, Prentice Hall, 1993.
2. W.S. Torgerson, Theory and methods of scaling, J.Wiley & Sons, 1958.
3. L. Goldfarb, "A new approach to pattern recognition", Progress in Pattern Recognition,Elsevier Science Publishers B.V. vol.2, pp.241-402,1985.
4. T. Graepel, R. Herbrich, P. Bollmann-Sdorra, and K. Obermayer, "Classification on pairwise proximity data", In Advances in Neural Information Processing 11, pp.438-444, 1999.
5. E. Pekalska, P. Paclik, and R.P.W Duin, "A generalized kernel approach to dissimilarity-based classification", Journal of Machine Learning Research, Special Issue on Kernel Methods, vol.2, no.2, pp.175-211, 2002.
6. M. Belkin and P. Niyogi, "Laplacian eigenmaps and spectral techniques for embedding and clustering", Advances in Neural Information Processing Systems 14, pp.585-591, 2002.
7. M. Belkin and P. Niyogi, "Using manifold structure for partially labeled classification", Advances in Neural Information Processing Systems 15, pp.929-936, 2003.
8. B. Schölkopf, A.J. Smola, K.R. Müller, "Nonlinear component analysis as a kernel eigenvalue problem", Neural Computation, 10, pp.1299-1319, 1998.
9. H. Shimodaira, K. Noma, M. Nakai, S. Sagayama, "Dynamic time-alignment kernel in support vector machine", Neural Information Processing Systems 14, pp.921-928, 2002.
10. C. Bahlmann, B. Haasdonk, and Hans Burkhardt, "On-line handwriting recognition with support vector machines-a kernel approach", Proc. 8th Int. W/S on Frontiers in Handwriting Recognition, pp.49-54, 2002.
11. J. Ham, D.D. Lee, S. Mika, and B. Schölkopf, "A kernel view of the dimensionality reduction of manifolds", TR-110, Max-Planck-Institut fur biologische Kybernetik, Tubingen, 2003.
12. C. Corres and V. Vapnik, "Support vector networks", Machine Learning, vol.20, pp.273-297, 1995.
13. S. Hettich abd S.D. Bay, UCI Repository of KDD Databases, http://kdd.ics.uci.edu/ 1999.
14. M. Seeger, "Learning with labeled and unlabeled data", Technical report, Institute for Adaptive and Neural Computation, University of Edinburgh, 2001.