Dimensionality reduction for pattern recognition based on difference of distribution among classes

Masaomi Nishimura[1] Kazuyuki Hiraoka[2] and Taketoshi Mishima[2]
[1]Graduate School of Science and Engineering, Saitama University
[2]Department of Information and Computer Sciences, Saitama University
255 Shimo-Okubo, Saitama-city, Saitama, 338-8570, Japan.
Tel/Fax: +81-48-858-3723
E-mail: nishi@me.ics.saitama-u.ac.jp

Abstract: For pattern recognition on high-dimensional data, such as images, the dimensionality reduction as a preprocessing is effective. By dimensionality reduction, we can (1) reduce storage capacity or amount of calculation, and (2) avoid "the curse of dimensionality" and improve classification performance. Popular tools for dimensionality reduction are Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Independent Component Analysis (ICA) recently. Among them, only LDA takes the class labels into consideration. Nevertheless, it has been reported that the classification performance with ICA is better than that with LDA because LDA has restriction on the number of dimensions after reduction. To overcome this dilemma, we propose a new dimensionality reduction technique based on an information theoretic measure for difference of distribution. It takes the class labels into consideration and still it does not have restriction on number of dimensions after reduction. Improvement of classification performance has been confirmed experimentally.

1. Introduction

When performing a pattern classification, first, we train the classifier by sample data together with their class labels, and then, classify the real-data using that trained classifier. If the dimension of data is large, taking raw data into classifier is not useful because amount of calculation or storage capacity will be too large and it brings classification performance bad influence of the curse of dimensionality. Therefore, taking a dimensionality reduction into raw data before pattern classification will be the better way.

Popular dimensionality reduction tools used for such cases are Principal Component Analysis (PCA), Linear Discriminant Analysis (LDA), and Independent Component Analysis (ICA). In these, PCA and ICA does not use class labels which sample data have. Only LDA takes class information into dimensionality reduction process. However, it has been reported that classifying with ICA gives the best classification performance among them[1]. One reason for this is that LDA has a restriction that the number of reduced dimensions is limited up to the number of classes, and it makes shortage of information for pattern classification.

In the present study, the authors propose a dimensionality reduction method which has the following desired properties at the same time:

- efficient use of class labels in sample data
- no restriction on the number of dimensions after the dimensionality reduction

This is based on Kullback-Leibler (KL) divergence, which is used as a measure of "difference" between classes after the dimensionality reduction. We explain the algorisms of the dimensionality reduction method in section 2. And the effectiveness of the method is confirmed experimentally in section 3.

2. Proposed Method

2.1 Setting of problem

In the present paper, we study problem of two classes. High-dimensional data $\boldsymbol{x}(t) \in R^N, (t = 1, \ldots, T)$ and whose class labels $c(t) = \{1, 2\}$ are given. And we hope to determine a 'nice' reduction matrix A, where A is an $N \times L$ matrix. Then we apply a certain classification method to the low-dimensional reduced data $\boldsymbol{y}(t) = A^T \boldsymbol{x}(t)$.

2.2 Kullback-Leibler divergence

The authors consider that A is 'nice' when the distributions of $\boldsymbol{y}$ are clearly different between class 1 and 2. In order to measure difference between distributions, we use Kullback-Leibler (KL) divergence

$$D(p \parallel q) = E_p\left[\log \frac{p}{q}\right] = \int p(\boldsymbol{x}) \log \frac{p(\boldsymbol{x})}{q(\boldsymbol{x})} d\boldsymbol{x}. \quad (1)$$

Here q and p are probability density functions of class 1 and 2.
In particular, KL divergence between two normal distributions

$$\begin{cases} \tilde{q}(\cdot) \sim N(\mu_q, \sigma_q^2) \\ \tilde{p}(\cdot) \sim N(\mu_p, \sigma_p^2) \end{cases} \quad (2)$$

is

$$\begin{aligned} D(\tilde{p} \parallel \tilde{q}) &= \int \tilde{p}(y) \log \frac{\tilde{p}(y)}{\tilde{q}(y)} dy \\ &= \frac{1}{2}\left(-\log \frac{\sigma_p^2}{\sigma_q^2} + \frac{\sigma_p^2}{\sigma_q^2} + \frac{1}{\sigma_q^2}\Delta\mu^2 - 1\right), \quad (3) \\ \Delta\mu &\equiv \mu_p - \mu_q. \end{aligned}$$

2.3 Reduction to one dimension

When $L = 1$, A is a vector and $\boldsymbol{y}$ is a scalar, so we denote A as $\boldsymbol{a}$ and $\boldsymbol{y}$ as y. Let $q_0(\boldsymbol{x})$ and $p_0(\boldsymbol{x})$ be the empirical distributions of classes $c = 1$ and $c = 2$, respectively. We approximate q_0 and p_0 with normal distributions q and p which have same mean vector and covariance matrix as q_0 and p_0,

$$\begin{cases} q(\cdot) \sim N(\boldsymbol{\mu}_q, V_q), \\ p(\cdot) \sim N(\boldsymbol{\mu}_p, V_p) \end{cases} . \tag{4}$$

With a linear transformation

$$\boldsymbol{x}' = \sqrt{V_q^{-1}}(\boldsymbol{x} - \boldsymbol{\mu}_q), \tag{5}$$

we can assume that q is standard normal distribution:

$$\begin{cases} q(\cdot) \sim N(0, I) & \Delta\boldsymbol{\mu} = \boldsymbol{\mu}_p - \boldsymbol{\mu}_q, \\ p(\cdot) \sim N(\Delta\boldsymbol{\mu}, V), & V = \sqrt{v_q^{-1}} V_p \sqrt{V_q^{-1}}, \end{cases} \tag{6}$$

where I is the identity matrix. In the above setting, KL divergence after reduction is

$$\varphi(\boldsymbol{a}) = \frac{1}{2}\left(\boldsymbol{a}^T V \boldsymbol{a} - \log \boldsymbol{a}^T V \boldsymbol{a} + (\Delta\boldsymbol{\mu}^T \boldsymbol{a})^2\right). \tag{7}$$

We hope to find $\boldsymbol{a}$ which maximizes $\varphi(\boldsymbol{a})$. Since φ satisfies

$$\varphi(\alpha\boldsymbol{a}) = \varphi(\boldsymbol{a}) \quad (\alpha \neq 0, \alpha \in R), \tag{8}$$

we restrict

$$||\boldsymbol{a}|| = 1 \tag{9}$$

without loss of generality. In summary, the optimization problem

$$\varphi(\boldsymbol{a}) \to \max, \quad ||\boldsymbol{a}|| = 1 \tag{10}$$

is led by the above discussion. This optimization problem is solved by steepest descent method:

$$\boldsymbol{F}(\boldsymbol{a}) \equiv \frac{\partial\varphi}{\partial\boldsymbol{a}} = \left\{\left(1 - \frac{1}{\boldsymbol{a}^T V \boldsymbol{a}}\right) V + \Delta\boldsymbol{\mu}\Delta\boldsymbol{\mu}^T\right\} \boldsymbol{a}, \tag{11}$$

$$\tilde{\boldsymbol{a}} \leftarrow \boldsymbol{a} + \eta \boldsymbol{F}(\boldsymbol{a}), \tag{12}$$

$$\boldsymbol{a} \leftarrow \tilde{\boldsymbol{a}}/||\tilde{\boldsymbol{a}}||. \tag{13}$$

Here η is a positive, small, constant number. Procedures (12) and (13) are repeated until they converge. In addition, in order to avoid convergence to local maximum, we try the above optimization procedures several times from different initial values of $\boldsymbol{a}$. Then we adopt the trial which achieves the largest $\varphi(\boldsymbol{a})$.

2.4 reduction to L dimensions

A direct extension of the above procedure to L dimension case is optimization of reduction matrix A with the target function $D(p \parallel q)$. However, we take a greedy method instead so that we can save calculations and write its program easily. First, we compute $\boldsymbol{a}_1$ by the above method:

$$\boldsymbol{a}_1 = \arg\max_{||\boldsymbol{a}||=1} \varphi(\boldsymbol{a}). \tag{14}$$

Second, we introduce a restriction that $y_2 = \boldsymbol{a}_2^T \boldsymbol{x}$ does not have any correlation with the previously extracted component $y_1 = \boldsymbol{a}_1^T \boldsymbol{x}$. We perform the optimization of $\boldsymbol{a}_2$ under this restriction. It makes information overlap smaller, so that the classification performance will be better. Rest vectors $\boldsymbol{a}_3, \ldots, \boldsymbol{a}_L$ are sequentially determined in the same way. Namely, we introduce restrictions that $y_k = \boldsymbol{a}_k^T \boldsymbol{x}, (k \geq 3)$ does not have any correlation with $y_1, \ldots, y_{k-1}$, and we optimize $\boldsymbol{a}_3, \ldots, \boldsymbol{a}_L$ under these restrictions. Note that these restrictions are equivalent to

$$\boldsymbol{a}_i^T \mathrm{Var}[\boldsymbol{x}] \boldsymbol{a}_k = 0, \quad (i = 1, 2, \ldots, k-1). \tag{15}$$

From this, we obtain the optimization problem

$$\boldsymbol{a}_k = \arg\max_{||\boldsymbol{a}||=1} \varphi(\boldsymbol{a}), \tag{16}$$

$$\boldsymbol{a}_k^T \mathrm{Var}[\boldsymbol{x}] \boldsymbol{a}_j = 0, (j = 1, \ldots, k-1). \tag{17}$$

And furthermore, to make computation easier, we replaces (17) with

$$\boldsymbol{a}_i^T \boldsymbol{a}_j = 0, \quad (j = 1, \ldots, k-1). \tag{18}$$

To this end, the updating rule for the optimization problem is as follows:

$$\tilde{\boldsymbol{a}}_k' \leftarrow \boldsymbol{a}_k + \eta \boldsymbol{F}(\boldsymbol{a}_k), \tag{19}$$

$$\tilde{\boldsymbol{a}}_k \leftarrow \boldsymbol{a} - H^T A^T \tilde{\boldsymbol{a}}_k', \tag{20}$$

$$\boldsymbol{a} \leftarrow \tilde{\boldsymbol{a}}_k/||\tilde{\boldsymbol{a}}_k||, \tag{21}$$

where $H = (A^T A)^{-1} A^T$ is a generalized inverse of A.

3. experiments

We have experimentally compared classification performance after dimensionality reduction by PCA, LDA, and the proposed method. As the classifier in these experiments, we use Support Vector Machine (SVM) which is popular in recent years. In order to measure generalization ability, data set is divided into two sets: training set and test set. The procedure of experiments is as follows: (1) Determine the reduction matrix A by PCA, LDA, and the proposed method based on the training set $\{\boldsymbol{x}\}$. (2) Apply A to the training set and obtain the reduced training set $\{\boldsymbol{y} = A^T \boldsymbol{x}\}$. (3) Train SVM for the reduced training set. (4) Classify the test set via the above trained SVM and measure its performance[1]. Radial Basis Function (RBF) kernel is used for all experiments. For each experiment, RBF parameter γ has decided to the value which gave the best performance.

3.1 Experiment with synthetic data

First, we illustrate the effect of our criterion (7) with a synthetic data.

We have educed a two dimensional synthetic data into one dimension. The data have plural clusters (Fig.1). Fig.2 is the plot of reduced data (upper is by PCA,

[1]In this experiment, we used 'MATLAB Support Vector Machine Toolbox' provided by Dr. Gavin Cawley[2].

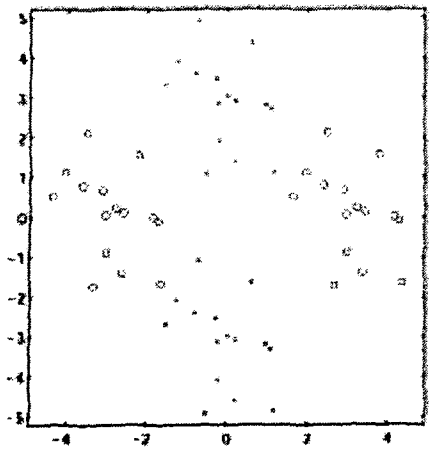


Figure 1. two dimensional synthetic data (o is class $c = 1$ and x is $c = 2$)

middle is by LDA, and lower is by our method). These reduced data have been input to SVM[2] and classification performance for each case has been measured (Table 1). The classification with PCA is poor because the variance of whole data is almost same for any directions. The performance by LDA is also low because the within-class mean of each class is almost same. In contrast, our method could extract the most effective component for classification.

Figure 2. dimension reduced data (upper : PCA, middle : LDA, lower : our's)

PCA	LDA	our's
46.7	43.3	96.7

Table 1. rate of correct classification [%] by SVM

3.2 Experiment with real-world data

3.2.1 Experiment with WPBC dataset

Second, we experimented classifying a real-world data "Wisconsin Prognostic Breast Cancer (WPBC)"[3]. It is 32-dimensional data of various follow-ups on breast cancer patients. Class $c = 1$ (nonrecur) includes 151 records and $c = 2$ (recur) has 47 records. We reduced these data to L dimensions ($L = 1, \ldots, 31$) by PCA and by our method. By LDA, only reduction to one dimension is possible since the number of classes is two: this is an intrinsic restriction of LDA. After dimensionality reduction, we have tried classification of each reduced data by SVM. The horizontal axis of Fig. 3 is dimensions of reduced data and the vertical axis is correct classification rate of each case. Performances of our method are higher than that of PCA and LDA in any cases.

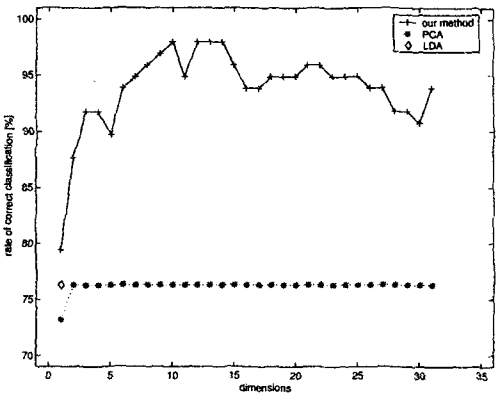


Figure 3. rate of correct classification for dimension-reduced WPBC data. RBF parameter $\gamma = 0.01$ for our method, and $\gamma = 0.05$ for PCA and LDA.

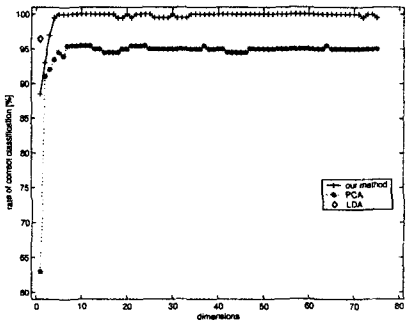
3.2.2 Experiment with the multi-feature digit dataset

Third, we experimented with the data of the multi-feature digit dataset[4]. Among them, we used Fourier coefficients of the character shapes. It is a dataset of handwritten numerals, 76-dimensional and 10 classes (numeral '0' to '9') data. Each class includes 200 records. We have tried classification of two digits, for example, '1' and '4'. It is observed that most combinations of two digits are too easy for benchmark of classification. To get difficult combination, we classified all combinations using LDA, and we picked six combinations which results worst performance. For these difficult combinations, the following experiments are performed. Reduce the data of combination to L dimensions ($L = 1, \ldots, 75$) and classify each reduced data by SVM. The results of classifications are shown in Fig. 4. Fig. 4(a) is the result of classification between class '1' and '2', Fig. 4(b) is between class '1' and '3', Fig. 4(c) is between class '1' and '4', Fig. 4(d) is between class '1' and '7', Fig. 4(e) is between class '4' and '7', and Fig. 4(f) is between class '6' and '9'. In each result, classification performance with LDA is best when the dimension of reduced data is 1. However, when the data are reduced to the optimal dimensions, our method gives the best classification performance.

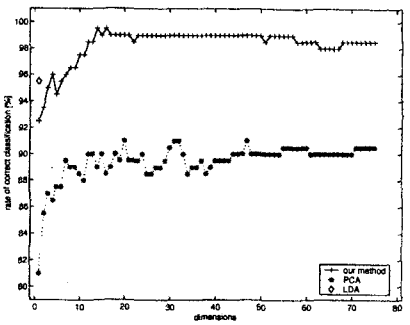
[2]RBF parameter $\gamma = 0.5$ and regularization parameter $C = 100$.

[3]It is provided by W. Nick Street, Computer Sciences Dept., University of Wisconsin, 1210 West Dayton St., Madison, WI 53706i. ftp://ftp.cs.wisc.edu/math-prog/cpo-dataset/machine-learn/WPBC/

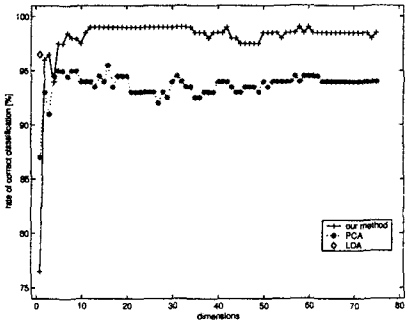
[4]It is provided by Robert P.W. Duin, Department of Applied Physics, Delft University of Technology, P.O. Box 5046, 2600 GA Delft, The Netherlands. ftp://ftp.ics.uci.edu/pub/machine-learning-databases/mfeat/



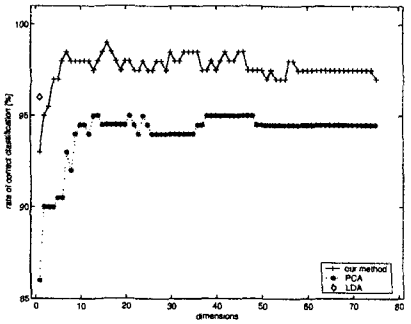
(a) classification of '1' and '2'



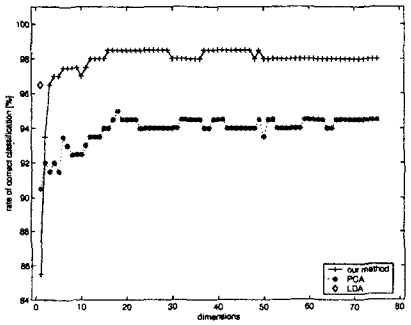
(b) classification of '4' and '9'



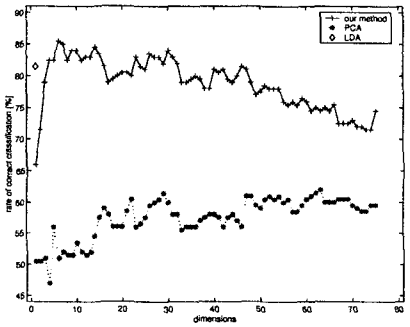
(c) classification of '1' and '3'



(d) classification of '1' and '7'



(e) classification of '4' and '9'



(f) classification of '4' and '9'

Figure 4. rate of correct classification for dimension-reduced multi-feature digit dataset. RBF parameter $\gamma = 0.01$ for our method, and $\gamma = 0.05$ with PCA and LDA.

4. Conclusion

We have proposed a new method of linear dimensionality reduction based on KL divergence, and experimentally demonstrated improvement of classification performance compared with PCA and LDA. Now we are trying to extend this method to data which have three or more classes. We are also planning more experiments, including comparison with ICA and application to data which have hundreds or thousands of dimensions.

References

[1] G. Donato, M.S. Bartlett, J.C. Hager, P. Ekman, and T.J. Sejnowski, "Classifying facial actions," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 20, no. 10, pp. 974– 989, 1999.

[2] Cawley, G. C., "MATLAB Support Vector Machine Toolbox (v0.50β) [`http://theoval.sys.uea.ac.uk/~gcc/svm/toolbox`] University of East Anglia, School of Information Systems, Norwich, Norfolk, U.K. NR4 7TJ, 2000

This work has been partly supported by JSPS (Japan Society for the Promotion of Science), 14580405.