Robust Speaker Identification Using Multimodal Discriminant Analysis with Kernels

Min-Seok Kim, IL-Ho Yang and Ha-Jin Yu*
School of Computer Science, University of Seoul, Korea
E-mail: {ms, heisco, hjyu}@uos.ac.kr

Abstract

In this paper, we propose kernel multimodal Fisher discriminant analysis (kernel MFDA), a new non-linear feature transformation method, which can be applied to large-scale problems such as speaker recognition tasks. Our proposed method has characteristics of kernel Fisher discriminant analysis (kernel FDA) as well as kernel principal component analysis (kernel PCA). The memory requirement of our proposed method is much lower than the other kernel methods. In the experiments, we apply our proposed method to a speaker identification task, and then we compare the accuracy of this method with kernel FDA and kernel PCA in clean and noisy environments. As the results, our proposed method outperforms kernel PCA.

1. Introduction

Speaker recognition systems have been widely used in ubiquitous era. These systems should be robust in various environments to satisfy users' demand. Recently, there are a lot of researches to build the robust speaker recognition systems. One of them is the robust feature extraction method using principal component analysis (PCA) [4][6].

PCA is a dimensionality reduction method to drop needless information, and the feature vectors extracted by PCA have good information for classification. Furthermore, these features are denoised. However, PCA may be ineffective when the feature vectors have nonlinear structure. Kernel principal component analysis [6] (kernel PCA) is the nonlinear version of PCA. It is more appropriate than PCA when the feature vectors have nonlinear structure. In researches with toy examples or with a small number of training set, kernel PCA outperforms PCA. However, the memory requirement and the computational complexity increase quadratically with the number of training data. For this reason, it is difficult to apply kernel PCA-based feature extraction method to large-scale problems such as speaker or speech recognition systems. Some of the researches derive kernel PCA by the random sampled training data [1][2]. However, it does not guarantee that the random sampled data represent all of the training set.

For the large-scale problems, several approaches are proposed [8]. Greedy filtering [8] and its application that greedy kernel PCA are applied successfully to speaker recognition [12]. In this paper, we propose kernel multimodal Fisher discriminant analysis (kernel MFDA), a new non-linear feature transformation method, which

Table 1. Notations used in this paper

$\phi(\cdot)$	Nonlinear mapping function, from *input space* to *feature space*, $\phi(\cdot):\Re^d \rightarrow F$
$(\cdot)'$	Transpose
$\mathbf{X}$	Training set, $\mathbf{X}=[\mathbf{x}_1,\ldots,\mathbf{x}_l]'$, $\phi(\mathbf{x}_i)\in F$
$\mathbf{X}^F$	$\mathbf{X}^F=[\phi(\mathbf{x}_1),\ldots,\phi(\mathbf{x}_l)]'$
$k(\cdot,\cdot)$	Kernel function such as Gaussian kernel [4]
$\mathbf{K}_{\cdot}$	Kernel matrix, for instance, $\mathbf{K}_{\mathbf{X}}$ is the kernel matrix of $\mathbf{X}$ and $\mathbf{K}_{\mathbf{RX}}$ is the kernel matrix of $\mathbf{R}$ and $\mathbf{X}$.
$\mathbf{1}_{i\times j}$	$i\times j$ matrix whose elements are 1
$\mathbf{R}$	Reduced set $\mathbf{R}=[\mathbf{r}_1,\ldots,\mathbf{r}_m]'$, $\mathbf{R}\subset\mathbf{X}$ by greedy filtering [5][8]
$\mathbf{R}^F$	$\mathbf{R}^F=[\phi(\mathbf{r}_1),\ldots\phi(\mathbf{r}_m)]'$
d	Dimension of vector $\mathbf{x}_i$
l	Total number of the training data,
m	Number of the reduced data by greedy filtering [5][8]
$\tilde{d}$	Number of the selected eigenvectors. It is to be new dimension of output data
t	Total number of the test data
$\mathbf{S}_i$	Training set of the ith class, $\mathbf{S}_i\subset\mathbf{X}$, $\bigcup_{i=1}^{s}\mathbf{S}_i=\mathbf{X}$
s	Number of classes
k	Number of the selected centers per class which is a parameter of the modified k-means algorithm
$\mathbf{M}_i$	Centers of $\mathbf{S}_i$, $\mathbf{M}_i=[\mathbf{m}_{i,1},\ldots,\mathbf{m}_{i,k}]'$. $\mathbf{m}_{i,j}$ is the center of the jth component of the ith class in *input space.*
$\mathbf{M}_i^F$	Centers of $\mathbf{S}_i^{\phi}$, $\mathbf{M}_i^F=[\mathbf{m}_{i,1}^F,\ldots,\mathbf{m}_{i,k}^F]'$. $\mathbf{m}_{i,j}^F$ is the center of the jth component of the ith class in the *feature space*.
$n(i,j)$	Number of training data which belongs to the jth component of the ith class
$\boldsymbol{\mu}^F$	Center of $\mathbf{X}^F$
$kdist(\cdot,\cdot)$	Distance in *feature space*

* Corresponding author

1082-3409/09 $26.00 © 2009 IEEE
DOI 10.1109/ICTAI.2009.122

has characteristics of both kernel Fisher discriminant analysis (kernel FDA) [7] and kernel PCA. It requires much smaller memory space than the other kernel methods including greedy kernel PCA, kernel PCA, and kernel FDA.

This paper is organized as follows. In the next section, we describe our proposed method. In section 3, we show the experimental results of a speaker identification task. Finally, section 4 presents the conclusions.

2. Proposed Method

2.1. Main idea

Linear transformation methods such as PCA [6] are not appropriate for the situations that the training set has the nonlinear structure. For this reason, kernel PCA [4][6] had been proposed. Multimodal Fisher discriminant analysis (MFDA) [3] is another solution for the nonlinear structure. Figure 1 shows an example of multimodal distribution of two speakers A and B. In this case, FDA does not work but FDA of their components works well (it means that each of components is regarded as one class). In the MFDA, the new scatters [3] are used instead of the conventional within-class scatter and between-class scatter.

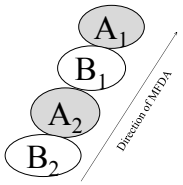

Figure 1. An example showing the multimodal distributions of two speakers A and B

Our proposed method is inspired by MFDA method. However, it is difficult to derive the kernelized MFDA directly, because it is complicated to compute the centers of the components in *feature space* ($\phi(\mathbf{m}_{i,j})$ is not $\mathbf{m}^F_{i,j}$). Although kernel k-means method exists [4], it is not considered here because we make efforts to reduce the computational complexity. To compute $\mathbf{m}^F_{i,j}$, we use the modified kernel *k*-means method using the distance between the vectors $\phi(\mathbf{x}_i)$ and $\phi(\mathbf{x}_j)$ [4] in the *feature space*:

$$kdist(\mathbf{x}_i,\mathbf{x}_j)=k(\mathbf{x}_i,\mathbf{x}_i)-2k(\mathbf{x}_i,\mathbf{x}_j)+k(\mathbf{x}_j,\mathbf{x}_j). \quad (1)$$

If we use the Gaussian kernel, equation 1 can be reduced to

$$kdist(\mathbf{x}_i,\mathbf{x}_j)\equiv 2-2k(\mathbf{x}_i,\mathbf{x}_j), \quad (2)$$

where the Gaussian kernel is defined as

$$k(\mathbf{x}_i,\mathbf{x}_j)=\exp(-\|\mathbf{x}_i-\mathbf{x}_j\|^2/(2\sigma^2)) . \quad (3)$$

2.2. The Algorithm and the Complexity

Table 2 shows the algorithm of the modified *k*-means using $kdist(\cdot,\cdot)$. There are two key ideas. The first one is in (b-2). Originally, the real center $\hat{\mathbf{m}}^F_{i,j}$ of the group $\mathbf{G}^F_{i,j}$ in the *feature space* should be represented by a linear combination of $\mathbf{G}^F_{i,j}$. But this algorithm uses the closest vector from the real center $\hat{\mathbf{m}}^F_{i,j}$. By this point, the computational complexity of the kernel *k*-means is reduced. The second one is using the Gaussian kernel. By Gaussian kernel, the result of $k(\mathbf{x}_i,\mathbf{x}_i)$ is always 1.

Table 2. Modified kernel *k*-means algorithm using $kdist(\cdot,\cdot)$

1. Select *k* initial centers randomly: $\mathbf{M}^F_i=[\mathbf{m}^F_{i,1},\ldots,\mathbf{m}^F_{i,k}]'$
2. Iterate δ times:
   (a) E-step:
   Compute $kdist(\cdot,\cdot)$ between $\mathbf{m}^F_{i,j}$ and all of the training data in $\mathbf{S}^F_i$, and assign their closest centers.
   (b) M-step:
   for *j*=1 to *k*:
   (b-1) Set $\mathbf{G}^F_{i,j}$ to be the group of $n(i,j)$ data whose closest center is $\mathbf{m}^F_{i,j}$ (Now the real center $\hat{\mathbf{m}}^F_{i,j}$ of $\mathbf{G}^F_{i,j}$ is $\mathbf{G}_{i,j}\mathbf{1}_{n(i,j)\times 1}/n(i,j)$)
   (b-2) Set $\mathbf{m}^{\phi}_{i,1}$ to the closest vector from the real center $\hat{\mathbf{m}}^{\phi}_{i,1}$

Using the algorithm in table 2, we can derive $\mathbf{M}^F_i$ for *i* =1,…,*s*. Kernel MFDA is to kenerlize the between-class scatter of MFDA[3] by using the centers computed by the modified kernel *k*-means. The new between-class scatter of our proposed method is defined as:

$$\tilde{\mathbf{S}}^F_B=\sum_{i=1}^{c}\sum_{j=1}^{k}(\mathbf{m}^F_{i,j}-\boldsymbol{\mu}^F)(\mathbf{m}^F_{i,j}-\boldsymbol{\mu}^F)' . \quad (4)$$

Let $\mathbf{M}^F=[\mathbf{m}^F_{1,1},\ldots,\mathbf{m}^F_{1,k},\ldots,\mathbf{m}^F_{s,k}]'$ denote the total set of the centers. Then equation 4 can be represented by the matrix notation

$$\tilde{\mathbf{S}}^F_B=(\mathbf{M}^F-\boldsymbol{\mu}^F\mathbf{1}_{1\times sk})(\mathbf{M}^F-\boldsymbol{\mu}^F\mathbf{1}_{1\times sk})', \quad (5)$$

where $\boldsymbol{\mu}^F$ is computed as

$$\boldsymbol{\mu}^F=\frac{1}{sk}\mathbf{M}^F\mathbf{1}_{sk\times 1}, \quad (6)$$

and pluging it into equation 5, we can derive the following equations

$$\tilde{\mathbf{S}}^F_B=(\mathbf{M}^F-\frac{1}{sk}\mathbf{M}^F\mathbf{1}_{sk\times sk})(\mathbf{M}^F-\frac{1}{sk}\mathbf{M}^F\mathbf{1}_{sk\times sk})' \quad (7)$$

$\tilde{\mathbf{S}}^F_B=\overline{\mathbf{M}}^F\,\overline{\mathbf{M}}^{F\prime}$ where

$$\overline{\mathbf{M}}^F = (\mathbf{M}^F - \frac{1}{sk}\mathbf{M}^F \mathbf{1}_{sk\times sk}).$$

To maximize $\widetilde{\mathbf{S}}_B^F$, the eigenvalue decomposition can be applied:

$$\lambda_i \mathbf{u}_i = \widetilde{\mathbf{S}}_B^F \mathbf{u}_i . \quad (8)$$

In our method, the following equation is used.

$$\mathbf{u}_i = \overline{\mathbf{R}}^F \boldsymbol{\beta}_i \text{ for } i=1,\ldots \widetilde{d} , \quad (9)$$

where $\overline{\mathbf{R}}^F$ is defined to be the centered $\mathbf{R}^F$ by $\boldsymbol{\mu}^F$ ($\mathbf{R}^F$ is the reduced set by greedy filtering [5][8]) and $\boldsymbol{\beta}_i$ is the coefficient of the linear combination of $\overline{\mathbf{R}}^F$. Like kernel PCA [6], by multiplying $\overline{\mathbf{R}}^{F}{}'$ to equation 8 and pluging-in equation 7 and 9, the following equation is derived:

$$\lambda_i \overline{\mathbf{R}}^{F}{}' \overline{\mathbf{R}}^F \boldsymbol{\beta}_i = \overline{\mathbf{R}}^{F}{}' \overline{\mathbf{M}}^F \overline{\mathbf{M}}^{F}{}' \overline{\mathbf{R}}^F \boldsymbol{\beta}_i . \quad (10)$$

We define $\overline{\mathbf{K}}_{\mathbf{R}} = \overline{\mathbf{R}}^{F}{}' \overline{\mathbf{R}}^F$ and $\overline{\mathbf{K}}_{\mathbf{RM}} = \overline{\mathbf{R}}^{F}{}' \overline{\mathbf{M}}^F$. Then $\overline{\mathbf{M}}^{F}{}' \overline{\mathbf{R}}^F$ is $\overline{\mathbf{K}}_{\mathbf{RM}}{}'$, and equation 10 can be re-written as

$$\lambda_i \boldsymbol{\beta}_i = \overline{\mathbf{K}}_{\mathbf{R}}{}^{-1} \overline{\mathbf{K}}_{\mathbf{RM}} \overline{\mathbf{K}}_{\mathbf{RM}}{}' \boldsymbol{\beta}_i \quad (11)$$

where

$$\overline{\mathbf{K}}_{\mathbf{R}} = \mathbf{K}_{\mathbf{R}} - \frac{1}{sk}\mathbf{K}_{\mathbf{RM}}\mathbf{1}_{sk\times m} - \frac{1}{sk}\mathbf{1}_{m\times sk}\mathbf{K}_{\mathbf{MR}} + \frac{1}{(sk)^2}\mathbf{1}_{m\times sk}\mathbf{K}_{\mathbf{M}}\mathbf{1}_{sk\times m} ,$$

$$\overline{\mathbf{K}}_{\mathbf{RM}} = \mathbf{K}_{\mathbf{RM}} - \frac{1}{sk}\mathbf{K}_{\mathbf{RM}}\mathbf{1}_{sk\times sk} - \frac{1}{sk}\mathbf{1}_{m\times sk}\mathbf{K}_{\mathbf{M}} + \frac{1}{(sk)^2}\mathbf{1}_{m\times sk}\mathbf{K}_{\mathbf{M}}\mathbf{1}_{sk\times m} . \quad (12)$$

Now $\boldsymbol{\beta}_i$ is computed by the eigenvectors of $\hat{\mathbf{K}}$. Then the eigenvectors corresponding to the $\widetilde{d}$ largest eigenvalues are selected:

$$\mathbf{B} = [\boldsymbol{\beta}_1, \ldots \boldsymbol{\beta}_{\widetilde{d}}]'. \quad (13)$$

Finally, given the test data $\mathbf{Y} = [\mathbf{y}_1, \ldots \mathbf{y}_t]'$, the transformed data onto kernel $\mathbf{Y}_{output}$ is computed using the kernel matrix $\mathbf{K}_{\mathbf{YR}}$ of $\mathbf{Y}$ and $\mathbf{R}$.

$$\overline{\mathbf{K}}_{\mathbf{YR}} = \mathbf{K}_{\mathbf{YR}} - \frac{1}{sk}\mathbf{K}_{\mathbf{YM}}\mathbf{1}_{sk\times m} - \frac{1}{sk}\mathbf{1}_{t\times sk}\mathbf{K}_{\mathbf{MR}} + \frac{1}{(sk)^2}\mathbf{1}_{t\times sk}\mathbf{K}_{\mathbf{M}}\mathbf{1}_{sk\times m},$$

$$\mathbf{Y}_{output} = \mathbf{B}' \overline{\mathbf{K}}_{\mathbf{YR}} \quad (14)$$

In summary, the characteristics of our proposed method, kernel MFDA, are:

1. It maximizes the between-class scatters in the *feature space*.
2. As the pervious applications [1][2], it can be regarded as the kernel PCA of the randomly sampled data (if we think the centers as the randomly sampled data).
3. Its memory requirement is very low. So we can apply it to large-scale problems efficiently.

Table 3 shows the memory requirements of the various feature extraction method. In our experiment, the number of training data used in our experiments is 159,329, and the number of classes (speakers) is 50. Also, the number of reduced data m is 300, and the number of components in class k is 4, 8, and 16. Thus kernel PCA and kernel FDA is infeasible in our experiments. Our proposed method needs more computational time to compute the centers in the *feature space*. But the memory requirement is much lower than greedy kernel PCA ($O(l/s)/O(lm) \approx 1/sm$).

Table 3. Memory requirements

Method	Memory requirements
Kernel PCA	$O(l^2)$
Greedy filtering	$O(dm)$
Kernel PCA by greedy filtering (greedy kernel PCA)	$O(lm)$
Modified *k*-means	$O(l/s)$
Kernel MFDA	$O(m^2)$

4. Experimental Results

For speaker identification experiments, we used YOHO database which consists of 138 speakers prompted to read combination lock phrases, for example, "67 34 85." The YOHO database has 'enroll' and 'verify' mode. The 'enroll' consists of 4 sessions with 24 utterances in each session, and the 'verify' consists of 10 sessions with 4 utterances in each session. We experimented for 50 speakers labeled from 101 to 154. We use only the first session in 'enroll' for training and all sessions in 'verify' for testing. In the verification data, we added noises ('babble', 'restaurant' in Aurora2 database [9]) with SNR 20 dB artificially using FaNT [10].

Each frame was represented by 20 mel-frequency cepstral coefficients (MFCCs) and the log energy (21-dimensional feature vector) with 30ms window and 20ms shift from the audio recording. The frames of data corresponding to silence were removed from the utterances automatically using energy threshold, and we performed cepstral mean subtraction (CMS) to each utterance. To build the speaker models, we use Gaussian mixture models (GMMs) [11]. We used Gaussian kernel as kernel function $k(\cdot,\cdot)$ with various σ^2 : 6872, 29128, and 75458 determined by percentile as in table 4. The percentile is regarded as the upper and lower bound value of each dimension. Therefore $\sigma^2 \equiv \sum_{i=1}^{d} (v_i^u - v_i^l)^2$ means the norm of its subtraction. Then the term in exp() is normalized by this σ^2.

We compared kernel MFDA with PCA, MFDA, kernel PCA, kernel FDA, and plane MFCCs. Table 4 shows the error rate of the overall experiments in three noise environments (clean, babble and restaurant). The average

error rate of our proposed method is 5.08% when k=4, σ^2=75458, and the number of mixture 64.

Table 4. Algorithm to determine σ^2

1. Compute the normal distribution $N_i(\mathbf{x};\boldsymbol{\mu}_i,\boldsymbol{\sigma}^2)$ of each dimension of $\mathbf{X}$ for i =1, …, d.
2. Compute the ($-z_{0.4}$, $z_{0.4}$), ($-z_{0.3}$, $z_{0.3}$) and ($-z_{0.2}$, $z_{0.2}$) percentiles each dimension. Let (v_i^l, v_i^u) denote them respectively for i =1, …, d.
3. $\sigma^2 \equiv \sum_{i=1}^{d}(v_i^u - v_i^l)^2$

Table 5. The error rate of the overall experiments (%)

Typeof feature	GMMs with 32 mixtures				GMMs with 64 mixtures			
	C.	B.	R.	Avg.	C.	B.	R.	Avg.
MFCCs (baseline)	7.50	14.85	14.00	12.12	6.65	13.25	11.00	10.30
PCA	4.30	10.4	7.85	7.52	3.95	10.2	7.65	7.27
Kernel PCA† (σ^2=29128)	4.30	9.45	7.35	7.03	4.25	8.70	6.80	6.58
Proposed method† (k=4, σ^2=75458)	3.00	7.85	6.10	5.65	2.70	7.10	5.45	5.08

C.: Clean, B.: Babble noise, R.: Restaurant noise,

Avg.: the average accuracy of clean, babble, and restaurant environment

†: Greedy filtering is applied

Our proposed method shows better accuracy in the overall environments over the other methods. The reduced errors are 2.19 % and 1.50% over PCA and kernel PCA respectively where the number of mixture is 64.

5. Conclusions

We have proposed kernel multimodal discriminant analysis for speaker recognition which is one of large-scale problems. This method has characteristics of both kernel Fisher discriminant analysis (kernel FDA) and kernel principal component analysis (kernel PCA). The characteristics are summarized as follows: Firstly, it maximizes the between-class scatters in feature space. Secondly, it can be regarded as kernel PCA of randomly sampled data. Finally, the memory requirement of our proposed method is much lower than the other kernel methods. So we can apply it to large-scale problems efficiently. In the experiments we have evaluated this method on speaker identification task and showed that the accuracy of our proposed method outperforms the other methods including kernel PCA.

6. References

[1] A. Lima, H. Zen, Y. Nankaku, C. Miyajima, K. Tokuda, and T. Kitamura, "On the Use of Kernel PCA for Feature Extraction in Speech Recognition", In *proc. of eurospeech 2003*, 2003, pp. 2625-2628.

[2] T. Takiguchi and Y. Ariki, "Robust Feature Extraction using Kernel PCA", In *int. Conf. on Acoustics, Speech and Signal Processing*, 2006, pp. 509-512.

[3] M.A. El-Gamal, M.F. Abu El-Yazeed, and M.M.H. El Ayadi, "Dimensionality reduction for text-independent speaker identification using Gaussian mixture model", In *Proceedings of MWSCAS 2003* Vol. 2, 30-30 Dec. 2003, pp. 625- 628.

[4] Shaw-Taylor, J. and Cristianini, N., *Kernel Methods for Pattern Analysis*, Cambridge University Press, New York, 2004

[5] Franc, V. *Optimization Algorithms for Kernel Methods*, PhD thesis, Centre for Machine Perception, Czech Technical University, 2005.

[6] B. Schölkopf, A. Smola, and K.Müller, "Kernel Principal Component Analysis", In *Int. Conf. on Artificial Neural Networks*, 1997, pp. 583–588.

[7] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller, "Fisher discriminant analysis with kernels", in *Neural Networks for Signal Processing IX*, Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, Eds. Piscataway, NJ: IEEE, 1999, pp. 41–48.

[8] V. Franc and V. Hlavac, "Greedy Algorithm for a Training Set Reduction in Kernel Methods", *Lecture Notes in Computer Science*, Vol. 2756, 2003, pp. 426-433.

[9] H.-G. Hirsch and D. Pearce, "The aurora experimental framework for the performance evaluation of speech recognition systems under noisy conditions", in *ASR 2000*, 2000, pp. 181–188.

[10] H.-G. Hirsch, "Fant-fltering and noise adding tool", http://dnt.kr.hs-niederrhein.de/download.html.

[11] D. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models", *Speech Communication*, vol. 17, 1995, pp. 91–108.

[12] M.-S. Kim, I.-H. Yang and H.-J. Yu, "Robust Speaker Identification Using Greedy Kernel PCA", In *Proc. of 20th IEEE International Conference on Tools with Artificial Intelligence*, 2008, pp. 143-146.