Kernel Oriented Discriminant Analysis for Speaker-Independent Phoneme Spaces

Heeyoul Choi[†], Ricardo Gutierrez-Osuna[†], Seungjin Choi[‡] and Yoonsuck Choe[†]
[†]*Dept. of Computer Science, Texas A&M University*
3112 TAMU, College Station, TX 77843-3112, USA
[‡]*Dept. of Computer Science, Pohang University of Science and Technology*
San 31, Hyoja-dong, Pohang, 790-784, Korea
[†]{*hchoi, rgutier, choe*}*@cs.tamu.edu,* [‡]*seungjin@postech.ac.kr*

Abstract

Speaker independent feature extraction is a critical problem in speech recognition. Oriented principal component analysis (OPCA) is a potential solution that can find a subspace robust against noise of the data set. The objective of this paper is to find a speaker-independent subspace by generalizing OPCA in two steps: First, we find a nonlinear subspace with the help of a kernel trick, which we refer to as kernel OPCA. *Second, we generalize OPCA to problems with more than two phonemes, which leads to oriented discriminant analysis (ODA). In addition, we equip ODA with the kernel trick again, which we refer to as* kernel ODA. *The models are tested on the CMU ARCTIC speech database. Our results indicate that our proposed kernel methods can outperform linear OPCA and linear ODA at finding a speaker-independent phoneme space.*

1 Introduction

Speech utterances contain information about the linguistic content of the message as well as the identity of the speaker. For the purposes of speech recognition, it is the linguistic content that is important; identity information can be regarded as noise. Therefore, in an optimal feature space, information about linguistic content should be maximized while speaker dependent content minimized.

Oriented principal component analysis (OPCA) [2] was proposed by Malayath et al. [5] as a potential method to find such speaker-independent phoneme space. OPCA is an extension of principal component analysis (PCA). Like PCA, OPCA maximizes variance in directions defined as informative, but in addition also minimizes variance in directions considered to be noisy. In the original formulation of Malayath et al. [5], OPCA was used to separate two phonemes and two speakers. Recently, various nonlinear mapping methods such as Isomap have been developed [8, 9]. However, these manifold learning methods assume that data points lie on one connected manifold, which is not the case when the data set consists of several disconnected classes or clusters.

In this paper, we extend OPCA to the non-linear case by means of the kernel trick used in kernel PCA and kernel fisher discriminant (KFD) [7, 6]. Our method, referred to as *kernel OPCA*, employs a geodesic-distance-based kernel similar to our previous work [1], but the technique can be easily extended to other kernel functions (e.g. polynomial, exponential, hyperbolic tangent functions), as long as they satisfy the Mercer kernel condition, i.e., positive semi-definiteness of the kernel matrix. The rationale behind a geodesic-distance kernel is two-fold. First, the geodesic distance has been demonstrated to find nonlinear structures of data sets [9, 1]. Second, the parameter tuning is not sensitive to the performance as long as the neighborhood size is within a proper range.

As a second step, we propose a generalization of this algorithm, kernel OPCA, to multi-class discrimination problems, and demonstrate its effectiveness on multiple phonemes. For this purpose, we adapted the classical linear discriminant analysis (LDA) solution to the oriented discriminant analysis (ODA) formulation, and applied the kernel trick to obtain a *kernel ODA* method. Experimental results with the CMU ARCTIC speech database [4] show that our methods find more informative low-dimensional space from a nonlinearly structured data set.

978-1-4244-2175-6/08/$25.00 ©2008 IEEE

2 Kernel ODA

Kernel ODA can be considered as a generalized version of kernel OPCA for multi-class problems. Here, to avoid clutter, we show the derivation of kernel OPCA in detail. Since the derivation of kernel ODA is very similar to that of kernel OPCA, we give just the objective function of kernel ODA.

Following Malayath et al. [5], we assume a problem with two speakers and two phonemes. We define a difference vector d_l to capture differences between two phonemes for the same speaker, and a difference vector d_s to capture differences between two speakers for the same phoneme. From these difference vectors, we can then estimate a covariance matrix[1] for each of the two sources of information in the data (i.e., speaker-specific and linguistic) as:

$$\begin{aligned} \boldsymbol{R}_l &= E\left[(\boldsymbol{d}_l - \overline{\boldsymbol{d}_l})(\boldsymbol{d}_l - \overline{\boldsymbol{d}_l})^T\right], \\ \boldsymbol{R}_s &= E\left[(\boldsymbol{d}_s - \overline{\boldsymbol{d}_s})(\boldsymbol{d}_s - \overline{\boldsymbol{d}_s})^T\right], \end{aligned} \tag{1}$$

where $\overline{\boldsymbol{d}_l}$ and $\overline{\boldsymbol{d}_s}$ are the mean difference between phonemes and speakers, respectively, and $E[\cdot]$ is the expectation operator. Then, the objective function $\boldsymbol{J}_{OPCA}(\boldsymbol{w})$ to be maximized can be written as follows:

$$\boldsymbol{J}_{OPCA}(\boldsymbol{w}) = \frac{\text{Signal}}{\text{Noise}} = \frac{\boldsymbol{w}^T \boldsymbol{R}_l \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{R}_s \boldsymbol{w}}, \tag{2}$$

where $\boldsymbol{w}$ are the basis vectors of the projected space. Note that, by maximizing $\boldsymbol{J}_{OPCA}$, we also maximize the signal-to-noise ratio, i.e., the variance due to phonetic content relative to the variance due to speaker information. This equation is similar to the objective function in LDA [6], except that $\boldsymbol{R}_l$ and $\boldsymbol{R}_s$ are covariance matrices of 'phoneme difference' and 'speaker difference', instead of 'between-class scatter' and 'within-class scatter'.

In order to "kernelize" OPCA, the objective function in Eq. (2) is implemented as an inner product matrix. Let $\boldsymbol{X} = [\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots, \boldsymbol{x}_{4N}]$ and $\boldsymbol{x}_i \in \mathcal{R}^{d\times 1}$, where

$$\boldsymbol{x}_i = \begin{cases} \text{Speaker A, Phoneme X} & \text{if } 1 \le i \le N \\ \text{Speaker B, Phoneme X} & \text{if } N+1 \le i \le 2N \\ \text{Speaker A, Phoneme Y} & \text{if } 2N+1 \le i \le 3N \\ \text{Speaker B, Phoneme Y} & \text{if } 3N+1 \le i \le 4N \end{cases}.$$

As in KFD [6], we transform $\boldsymbol{w}^T \boldsymbol{R}_l \boldsymbol{w}$ and $\boldsymbol{w}^T \boldsymbol{R}_s \boldsymbol{w}$ into $\boldsymbol{\alpha}^T \boldsymbol{M} \boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^T \boldsymbol{N} \boldsymbol{\alpha}$, respectively using the kernel trick as follows.

$$\boldsymbol{\alpha}^T \boldsymbol{M} \boldsymbol{\alpha} = \boldsymbol{w}^T \boldsymbol{R}_l^\Phi \boldsymbol{w}, \tag{3}$$

$$\boldsymbol{\alpha}^T \boldsymbol{N} \boldsymbol{\alpha} = \boldsymbol{w}^T \boldsymbol{R}_s^\Phi \boldsymbol{w}, \tag{4}$$

where $\boldsymbol{R}_l^\Phi$ and $\boldsymbol{R}_s^\Phi$ correspond to $\boldsymbol{R}_l$ and $\boldsymbol{R}_s$ in nonlinear feature space obtained by applying the nonlinear mapping Φ to the original data points $\boldsymbol{X}$. Once these $\boldsymbol{M}$ and $\boldsymbol{N}$ matrices are defined, the remaining part of the method is the same as in KFD [6]. We describe the details of the case $\boldsymbol{R}_l$ below ($\boldsymbol{R}_s$ follows a similar derivation).

First, we express the objective function in terms of the input data $\boldsymbol{X}$ instead of the difference vectors $\boldsymbol{d}_l$ and $\boldsymbol{d}_s$ using $\boldsymbol{d}_l = [\boldsymbol{x}_{2N+1} - \boldsymbol{x}_1, \cdots, \boldsymbol{x}_{4N} - \boldsymbol{x}_{2N}]$, which represents the difference between phonemes. Then

$$\begin{aligned} \boldsymbol{R}_l = \frac{1}{2N} \sum_i^{2N} &(\boldsymbol{x}_{2N+i} - \boldsymbol{x}_i - \overline{\boldsymbol{x}_{2N+i} - \boldsymbol{x}_i}) \\ &\cdot (\boldsymbol{x}_{2N+i} - \boldsymbol{x}_i - \overline{\boldsymbol{x}_{2N+i} - \boldsymbol{x}_i})^T, \end{aligned} \tag{5}$$

and

$$\boldsymbol{w} = \sum_i^{4N} \alpha_i \boldsymbol{x}_i. \tag{6}$$

Let $\boldsymbol{H}^{ik} = \boldsymbol{x}_i^T (\boldsymbol{d}_l^k - \overline{\boldsymbol{d}_l^k})$. Now, with $\boldsymbol{K}_{i,j} = \boldsymbol{x}_i{}^T \boldsymbol{x}_j$, $\boldsymbol{H}^{ik} = \boldsymbol{K}_{i,2N+k} - \boldsymbol{K}_{i,k} - \frac{1}{2N}\sum_m^{2N} (\boldsymbol{K}_{i,2N+m} - \boldsymbol{K}_{i,m})$. Finally, $\boldsymbol{w}^T \boldsymbol{R}_l \boldsymbol{w}$ is given by

$$\boldsymbol{w}^T \boldsymbol{R}_l \boldsymbol{w} = \frac{1}{2N} \boldsymbol{\alpha}^T \boldsymbol{H} \boldsymbol{H}^T \boldsymbol{\alpha}. \tag{7}$$

Therefore, we obtain $\boldsymbol{M}$ by

$$\boldsymbol{M} = \frac{1}{2N} \boldsymbol{H} \boldsymbol{H}^T. \tag{8}$$

Likewise, in case of the denominator $\boldsymbol{w}^T \boldsymbol{R}_s \boldsymbol{w}$ in Eq. (2), with $\boldsymbol{d}_s = [\boldsymbol{x}_{N+1} - \boldsymbol{x}_1, \cdots, \boldsymbol{x}_{2N} - \boldsymbol{x}_N, \boldsymbol{x}_{3N+1} - \boldsymbol{x}_{2N+1}, \cdots, \boldsymbol{x}_{4N} - \boldsymbol{x}_{3N}]$, we can derive $\boldsymbol{N}$ as

$$\boldsymbol{N} = \frac{1}{2N} \boldsymbol{G} \boldsymbol{G}^T + \mu \boldsymbol{I}, \tag{9}$$

where $\boldsymbol{G} = [\boldsymbol{G}_1 \boldsymbol{G}_2]$, and $\boldsymbol{G}_{1,ik} = \boldsymbol{K}_{i,N+k} - \boldsymbol{K}_{i,k} - \frac{1}{2N}\sum_{m=1}^{N} (\boldsymbol{K}_{i,N+m} + \boldsymbol{K}_{i,3N+m} - \boldsymbol{K}_{i,m} - \boldsymbol{K}_{i,2N+m})$, and $\boldsymbol{G}_{2,ik} = \boldsymbol{K}_{i,3N+k} - \boldsymbol{K}_{i,2N+k} - \frac{1}{2N}\sum_{m=1}^{N} (\boldsymbol{K}_{i,N+m} + \boldsymbol{K}_{i,3N+m} - \boldsymbol{K}_{i,m} - \boldsymbol{K}_{i,2N+m})$. Note that, for regularization purposes, we add a multiple of the identity matrix, $\mu \boldsymbol{I}$ to $\boldsymbol{N}$ in Eq. (9), where μ is a small number to make $\boldsymbol{N}$ positive definite [6]. In our experience, this regularization term makes the algorithm more stable.

Given a novel test data $\boldsymbol{x}_t$, the projected point $\boldsymbol{y}_t$ can be calculated as

$$\boldsymbol{y}_t = \boldsymbol{w}^T \boldsymbol{x}_t = \sum_i^{4N} \alpha_i \boldsymbol{K}_{T,it}. \tag{10}$$

[1] Since Eq. (1) is the covariance of the differences, we can define the correlation. Consequently, $\boldsymbol{M}$ and $\boldsymbol{N}$ change slightly with respect to Eqs. (8) and (9), respectively. In our experiments, the use of this correlation showed a slightly better performance than the covariance-based algorithm.

Several Mercer kernel functions (i.e. polynomial, exponential, or hyperbolic tangent function) can be used for the kernel matrices $\boldsymbol{K}$ and $\boldsymbol{K}_T$. Here, we use a kernel matrix based on the geodesic distance. As discussed in previous work [1], this approach has the advantage that it can find a nonlinear structure of data set without critical parameters affecting the performance.

In kernel OPCA, we assumed a problem with two phonemes and two speakers. We extend the solution to problems with more than two phonemes. This extension leads to a new method, which we term kernel ODA. We first define ODA in terms of the covariance matrix, then derive the kernel ODA solution. With only two speakers, the correlation matrix $\boldsymbol{R}_s$ remains the same as before. Only the correlation matrix $\boldsymbol{R}_l$ must be adjusted to handle more than two phonemes. Our solution is to use the 'between-class scatter matrix' $\boldsymbol{R}_L$ in the LDA solution. With this minor adjustment, the final objective function of ODA becomes

$$\boldsymbol{J}_{ODA}(\boldsymbol{w}) = \frac{\boldsymbol{w}^T \boldsymbol{R}_L \boldsymbol{w}}{\boldsymbol{w}^T \boldsymbol{R}_s \boldsymbol{w}}, \tag{11}$$

where $\boldsymbol{R}_s$ is given in Eq. (1) and $\boldsymbol{R}_L$ is given by

$$\boldsymbol{R}_L = \sum_{i}^{C} n_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T, \tag{12}$$

where C is the number of classes (phonemes), n_i is the number of phonemes in the ith class, $\boldsymbol{\mu}_i$ is the average of the ith class, and $\boldsymbol{\mu}$ is the average of all data. To kernelize it as in Eqs. (3) and (4), $\boldsymbol{M}$ is obtained from the KFD solution [6] instead of Eq. (8), and $\boldsymbol{N}$ is obtained from the kernel OPCA solution in Eq. (9). Projections of novel test data are obtained with Eq. (10). To calculate $\boldsymbol{M}$ and $\boldsymbol{N}$, we again use the kernel matrices $\widetilde{\boldsymbol{K}}$ and $\widetilde{\boldsymbol{K}_T}$ from kernel Isomap [1].

3 Experiments

We validated the proposed methods through a series of experiments using the CMU ARCTIC speech database [4]. As performance measures, we applied quadratic classifiers to the projection of the test data and measured the Bhattacharyya distance of the class-conditional distributions [3].

3.1 Two speakers and Two phonemes

The CMU ARCTIC database is a phonetically balanced corpus from US speakers, which was designed for unit selection speech synthesis research. The database includes US English male ('bdl', 'rms') and female ('slt', 'clb') speakers. For a representative example, we extracted two phonemes ('AH' and 'IH') for each speaker, and used Mel frequency cepstral coefficients (MFCCs) as the feature vectors. We used two speakers ('bdl', 'slt') for training and two speakers ('rms', 'clb') for testing. We used 300 samples per phoneme of each speaker for training, and 400 samples for testing, for a total of 1,200 training samples and 1,600 test samples. Note that since the phonemes were extracted from real sentences, two samples from the same speaker and the same phoneme class 'AH' may have significantly different MFCCs as a result of coarticulatory effects.

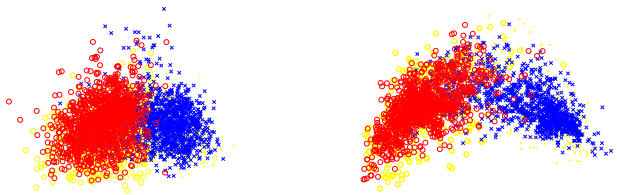

Figure 1. Subspaces for two speakers and two phonemes with (a) linear OPCA and (b) kernel OPCA. Blue crosses correspond to the phoneme 'AH,' whereas red circles correspond to the phoneme 'IH'. Yellow circles denote the training data.

Fig. 1 shows the subspaces for linear OPCA and kernel OPCA. Even though both scatterplots show speaker-independent subspaces, the kernel OPCA solution appears to provide increased separability. Indeed, we measured the Bhattacharyya distance [3] between the two clusters of phonemes and compared the classification rate based on quadratic classifier. The results, summarized in Table 1, indicate that kernel OPCA provides better phoneme discrimination than linear OPCA using either measure. Paired T-test indicates that the difference in classification performance between both methods is statistically significant (p=0.0406; n=24).

3.2 Two speakers and Multiple phonemes

The CMU ARCTIC database is also used for these experiments. In this case, we extracted three phonemes ('AH', 'IH' and 'OW') for each speaker and used MFCCs as the encoding vector. As in the previous experiment, we used two speakers ('bdl', 'slt') for training and two speakers ('rms', 'clb') for testing, resulting in 300 samples per phoneme of each speaker for training, and 300 samples for testing, for a total of 1,800 training samples and 1,800 test samples.

Table 1. Measurement of the performance for two speakers and two phonemes (MFCCs) on OPCA and KOPCA.

Methods	B-dist Train	B-dist Test	Hit Rate on test data
OPCA + Cov	14.71	14.70	85.88%
OPCA + Cor	14.52	14.53	87.58%
KOPCA + Cov	24.11	23.80	88.13%
KOPCA + Cor	24.33	23.91	89.54%

Fig. 2 shows the resulting subspaces for linear ODA and kernel ODA. Kernel ODA provides more scattered clusters than linear ODA (both within and between classes), in agreement with kernel OPCA in the previous experiments. Classification rates using a quadratic classifier were 72.93% (linear ODA) and 78.17% (kernel ODA). When KFD is optimized with some kernel functions (here RBF was the best) and parameters, it has 74.86% hit rate with the same classifier as before, which means just phoneme information is not enough to find a speaker-independent space.

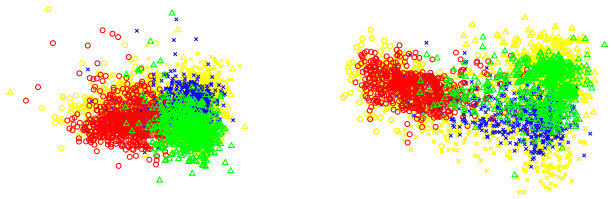

Figure 2. Subspaces for two speakers and three phonemes with (Left) linear ODA and (Right) kernel ODA. The blue crosses are 'AH', the red circles are 'IH', and the green triangles are 'OW'.

4 Conclusion

In this article, we proposed a two-pronged generalization of oriented PCA. First, we found a nonlinear subspace by means of the kernel trick, which led to kernel OPCA. Second, we extended kernel OPCA to problems with more than two classes, which led to linear ODA and kernel ODA. Experimental results on the CMU ARCTIC corpus showed that our proposed methods, kernel OPCA and kernel ODA, provide better separability than their linear counterparts (OPCA and linear ODA) in finding a speaker-independent phoneme space, as measured by classification rates and the Bhattacharyya distance.

These algorithms can be viewed as nonlinear manifold-learning strategies for problems where data points exist on several clustered manifolds corresponding to their classes. These algorithms were tested with relatively small data sets in the speech domain. Additional work is required to determine the extent to which these results will hold when applied to a larger data set of speakers and the entire phonetic space.

Acknowledgments

The authors would like to thank Jobany Rodriguez and Daniel Felps for their comments and help on speech processing. One of the authors was supported by StarVision Technologies's student sponsorship program and part of this work was supported by Korea Ministry of Knowledge Economy under the ITRC support program supervised by the IITA (IITA-2008-C1090-0801-0045).

References

[1] H. Choi and S. Choi. Robust Kernel Isomap. *Pattern Recognition*, 40(3):853–862, March 2007.

[2] K. I. Diamantaras and S. Y. Kung. *Principal Component Neural Networks: Theory and Applications*. John Wiely & Sons, INC, 1996.

[3] K. Fukunaga. *An Introduction to Statistical Pattern Recognition*. Academic Press, New York, NY, 1990.

[4] J. Kominek and A. W. Black. CMU ARCTIC databases for speech synthesis, 2003. URL `http://festvox.org/cmu_arctic/`.

[5] N. Malayath, H. Hermansky, and A. Kain. Towards decomposing the sources of variability in speech. In *Proc. EUROSPEECH*, pages 497–500, Rhodes, Greece, 1997.

[6] S. Mika, G. Ratsch, J. Weston, B. Schölkopf, and K. Müller. Fisher discriminant analysis with kernels. In *Proceedings of IEEE Neural Networks for Signal Processing Workshop*, pages 41–48, 1999.

[7] B. Schölkopf, A. J. Smola, and K. R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5):1299–1319, 1998.

[8] H. S. Seung and D. D. Lee. The manifold ways of perception. *Science*, 290:2268–2269, 2000.

[9] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290:2319–2323, 2000.