SUBSPACE KERNEL DISCRIMINANT ANALYSIS FOR SPEECH RECOGNITION

Hakan Erdoğan

Faculty of Engineering and Natural Sciences
Sabanci University
Orhanli Tuzla 34956 Istanbul Turkey
haerdogan@sabanciuniv.edu

ABSTRACT

Kernel Discriminant Analysis (KDA) has been successfully applied to many pattern recognition problems. KDA transforms the original problem into a space of dimension N where N is the number of training vectors. For speech recognition, N is usually prohibitively high increasing computational requirements beyond current computational capabilities. In this paper, we provide a formulation of a subspace version of KDA that enables its application to speech recognition, thus conveniently enabling nonlinear feature space transformations that result in discriminatory lower dimensional features.

1. INTRODUCTION

Speech recognition can be cast as a pattern classification problem where we would like to classify an input acoustic signal into one of all possible sentences. However, the number of classes or the number of all possible sentences are extremely high, so that it is unreasonable to solve the problem as a regular classification problem. Thus, we use dynamic generative sub-unit models (HMMs) and language modeling to model the probability of sentences and solve a huge search problem instead. Still, inherent to speech recognition is a desire to be able to accurately classify acoustic frames into subunits of sentences, typically words, syllables, word states, phones or phone states depending on the modeling of the problem. The idea is that, if we are able to classify acoustics into these subunits well, the sentence recognition is going to be more accurate as well.

For this reason, it is reasonable to borrow pattern classification techniques to help in speech recognition. Linear Discriminant Analysis (LDA) is one such technique that can be used to extract relevant features from speech signals that discriminate well between sub-unit classes. The problem is usually cast as a dimension reduction problem where many candidate features are pulled together and a rectangular linear transformation is found where the resultant feature vector has "reasonable" or "small" dimensions, yet it carries the most discriminative information to separate classes well.

LDA has been used in speech recognition extensively. LDA is limited to "linear" projections of original data space. Linear projections are limited in their power to discriminate between classes that are not linearly separable in the original feature space. It is plausable that classes are not linearly separable, but nonlinearly separable in the original space. In this case, it is possible to transform original data space to a even higher dimensional space $\mathcal{F}$ (possibly infinite dimensional) where the classes are linearly separable. Kernel versions of LDA and principal component analysis (PCA), called KDA and KPCA for short, enables this transformation without having too much extra computation.

Kernel-based machine learning and pattern classification techniques have achieved considerable success recently. Among those are support vector machines (SVM) and kernel versions of PCA and LDA [1]. Already, kernel versions of LDA and PCA are very popular among pattern classification community. These methods enable to extract features or decision rules that have nonlinear boundaries. The methods achieve this with a slight increase in computation by using the kernel trick. That is, if the algorithm uses only scalar products in the transformed space $\mathcal{F}$, that scalar product can be computed using kernels that use only original features and do not require actual computation of high dimensional features themselves in space $\mathcal{F}$.

In this paper, we apply a subspace version of KDA to speech recognition. One trouble for KDA for speech recognition is that the original derivation requires computation of features in an N-dimensional space where N is the number of training samples. This value is typically very high for speech recognition; e.g. 6000 samples per minute of audio. In speech recognition, it is possible to use more than 200 hours of audio to train a system. Thus, it becomes impractical to use KDA directly. So, we propose a subspace version of KDA and evaluate its performance in this paper.

First, we introduce LDA in section 2. We describe the subspace version of KDA (SKDA) in section 3. Application of SKDA to speech recognition problem and possible subspaces are analyzed in section 4. In section 5, we present results of phone recognition experiments on the

TIMIT database. We conclude in section 6 with a summary and suggestions for future work.

2. LINEAR DISCRIMINANT ANALYSIS

Consider a classification problem with multiple classes and multiple features. While Principal Component Analysis (PCA) finds projections of features that are efficient for representation of data, linear discriminant analysis (LDA) or Fisher Discriminant (FD) seeks projections of data that are efficient for discrimination between classes.

Let $x \in \mathbb{R}^n$ be a feature vector in the original space. We would like to find a transformation $y = \boldsymbol{\theta} x$, $\boldsymbol{\theta} : \mathbb{R}^n \rightarrow \mathbb{R}^p$ with $p < n$. We seek to choose new features y such that most of the class-discriminating information in x is retained in y. This dimension reduction will also help us battle "the curse of dimensionality" and train statistical models more efficiently due to dimension reduction.

Let $\{x_i \in \mathbb{R}^n\}_{1 \leq i \leq N}$ denote N training samples each labeled with a class label $l_i \in \{1 \ldots K\}$. Let $N_k = \sum_{l_i=k} 1$ be the number of training vectors in class k. Then, $\sum_{k=1}^{K} N_k = N$ is the total number of training samples. We define the following entities:

$$\begin{aligned} \mu &= \frac{1}{N} \sum_{i=1}^{N} x_i, \quad \mu_k = \sum_{l_i=k} x_i, \\ \Sigma_k &= \frac{1}{N_k} \sum_{l_i=k} x_i x_i^T - \mu_k \mu_k^T, \end{aligned}$$

where μ is the overall mean, μ_k is the sample mean for class k and Σ_k is the covariance matrix for class k.

In LDA, we define between and within class scatter matrices B and W respectively as follows:

$$\begin{aligned} B &= \sum_{i=1}^{K} N_k (\mu_k - \mu)(\mu_k - \mu)^T, \\ W &= \sum_{i=1}^{K} N_k \Sigma_k. \end{aligned} \tag{1}$$

Here, the determinant of B indicates how much class-means are separated from each other in the feature space and the determinant of W indicates the average variation within each class. If the features are transformed using a linear transformation $\boldsymbol{\theta}$, then the between and within class covariance matrices in the transformed feature space become $\boldsymbol{\theta} B \boldsymbol{\theta}^T$ and $\boldsymbol{\theta} W \boldsymbol{\theta}^T$ respectively.

LDA minimizes the ratio of determinants of between class and within class scatters after transformation. We choose the transformation $\hat{\boldsymbol{\theta}}$ that minimizes the objective function

$$\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \frac{|\boldsymbol{\theta} B \boldsymbol{\theta}^T|}{|\boldsymbol{\theta} W \boldsymbol{\theta}^T|}.$$

There is a closed form solution to the above optimization problem. Simply, the columns of the optimum transform $\boldsymbol{\theta}$ are given by the eigenvectors corresponding to the largest p eigenvalues of the generalized eigenvalue problem $Bv = \lambda W v$ [2].

Thus, finding the LDA transform amounts to gathering the statistics B and W from training data and solving the generalized eigenvalue problem. One problem with LDA is that the transformed space might not be well modeled with diagonal Gaussians (as is typical in speech recognition), thus we usually follow LDA transform with an MLLT transform [3] such that the transformed featues are well represented with diagonal Gaussians.

LDA has some limitations. LDA is Bayes optimal when the classes have identical covariances. Heteroscedastic LDA removes this assumption allowing each class to have different covariances and optimizes an appropriate maximum likelihood objective function [4]. Another way to improve LDA is by using the Kernel trick which we review next.

3. SUBSPACE KERNEL DISCRIMINANT ANALYSIS

Kernel-based techniques have been very popular recently in machine learning and pattern classification literature. Support vector machines (SVM), kernel LDA and kernel PCA are among those methods. The idea of KDA (or KFD for Kernel Fisher Discriminant) is to solve the LDA problem in a so-called kernel feature space $\mathcal{F}$.

In kernel-based techniques, features $x \in \mathbb{R}^n$ are mapped nonlinearly through mapping $\Phi(\cdot)$ to $\Phi(x) \in \mathcal{F}$, where $\mathcal{F}$ is potentially a much higher dimensional feature space. For a given classification or learning problem, one considers the same algorithm in $\mathcal{F}$ instead of $\mathbb{R}^n$. The advantage of this nonlinear mapping is that even if the classes are not linearly separated in the original space, one hopes that they are linearly separated in the higher dimensional space $\mathcal{F}$. A linear discriminant in $\mathcal{F}$ yields a nonlinear discriminant in the original space.

The disadvantage of going into a high dimensional space could be computatinal requirements in that space. However, for certain feature spaces $\mathcal{F}$, there is a highly effective trick for computing scalar products using kernel functions, *i.e.* :

$$(\Phi(x) \cdot \Phi(y)) = K(x, y).$$

These spaces are characterized by their kernel function and we need not transform original features to $\mathcal{F}$ if our algorithm can be implemented by just using scalar products. Some common kernel functions are:

$$\begin{aligned} K(x, y) &= \exp\left(-||x - y||^2 / c\right), \\ K(x, y) &= ((x \cdot y) + c)^d, \end{aligned}$$

called Gaussian RBF and polynomial kernels, where c, d are parameters.

Kernel Fisher Discriminant were introduced by Mika *et al.*[5, 6] for a two class problem. It can be generalized to a multiclass problem in a straightforward manner [7]. We need to transform the LDA problem formulation in $\mathcal{F}$ space to use scalar products only. To this end, first consider the case $p = 1$, *i.e.* $\boldsymbol{\theta} = v^T$ is a row vector, $v \in \mathcal{F}$. Objective function is the Rayleigh coefficient:

$$f(\boldsymbol{v}) = \frac{\boldsymbol{v}^T B \boldsymbol{v}}{\boldsymbol{v}^T W \boldsymbol{v}},$$

where B and W are theoretically computed in $\mathcal{F}$ space using $\Phi(x)$ features.

Assume we can express $\boldsymbol{v}$ in terms of a linear combination of mapped vectors:

$$\boldsymbol{v} = \sum_{j=1}^{m} \tilde{v}_j \Phi(p_j), \tag{2}$$

where $\tilde{v}_j \in \mathbb{R}$ and we call $(p_j \in \mathbb{R}^n)_{j=1}^m$ *pivot* vectors. Define a new feature vector $\tilde{x} \in \mathbb{R}^m$ obtained from original feature vector $x \in \mathbb{R}^n$ by its j^{th} component as:

$$\tilde{x}_j = (\Phi(p_j) \cdot \Phi(x)). \tag{3}$$

Note that, with this definition, we can replace the transformation scalar product $\boldsymbol{\theta} x = \boldsymbol{v}^T x$ in the high dimensional kernel space $\mathcal{F}$ with a scalar product of the coefficient vector $\tilde{\boldsymbol{v}}$ and the tranformed features $\tilde{x}$ in $\mathbb{R}^m$,

$$\boldsymbol{v}^T \Phi(x) = \sum_{j=1}^{m} \tilde{v}_j \Phi(p_j)^T \Phi(x) = \tilde{\boldsymbol{v}}^T \tilde{x}.$$

Then, we can also express the Rayleigh coefficient numerator and denominator by the following:

$$\begin{aligned} \boldsymbol{v}^T B \boldsymbol{v} &= \tilde{\boldsymbol{v}}^T \tilde{B} \tilde{\boldsymbol{v}}, \\ \boldsymbol{v}^T W \boldsymbol{v} &= \tilde{\boldsymbol{v}}^T \tilde{W} \tilde{\boldsymbol{v}}, \end{aligned}$$

where $\tilde{B}$ and $\tilde{W}$ are obtained using the feature vectors $(\tilde{x}_i)_{i=1}^N$ instead of the original features $(x_i)_{i=1}^N$ as in (1).

So, this results in a simple formula. Transform original features nonlinearly to $\tilde{x}$ domain (using kernel scalar products) of dimension M. Compute statistics $\tilde{B}$ and $\tilde{W}$ using the new features and obtain $\tilde{\boldsymbol{v}}$ vector using regular LDA method. Note, $\tilde{\boldsymbol{v}}$ will be the eigenvector corresponding to the largest eigenvalue of the generalized eigenvalue problem involving $\tilde{B}$ and $\tilde{W}$. When it comes to applying the kernel LDA to a new feature vector x, we first transform it into $\tilde{x}$ as in equation (3) and then find $\tilde{\boldsymbol{v}}^T \tilde{x}$ to be the one dimensional feature to be used.

Generalizing the above discussion to $p > 1$ is trivial. In this case, we assume each row $\boldsymbol{v}_i^T$ of matrix $\boldsymbol{\theta} = [\boldsymbol{v}_1 \boldsymbol{v}_2 \cdots \boldsymbol{v}_p]^T$ is in the span of the mapped pivot vectors and the discussion follows through and boils down to finding the coefficients $\tilde{\boldsymbol{v}}_i$ for each row of the transform. We can call the resulting transform matrix as $\tilde{\boldsymbol{\theta}}$.

In the above discussion in equation (2), we assumed that the transform row vector $\boldsymbol{v}$ is in the span of the images of pivot vectors p_j. KDA literature claims that the unconstrained vector $\boldsymbol{v}$ that solves the LDA problem in space $\mathcal{F}$ should be in the span of transformed training vectors $(\Phi(x_i))_{i=1}^N$ [5]. This result is said to follow from the theory of reproducing kernels [6].

Our discussion above which assumes $\boldsymbol{v}$ is in the span of M pivot vectors, is not a limitation for the original KDA. By choosing pivot vectors as the training data, we can obtain unconstrained KDA solution. However, training set size is prohibitively high in speech recognition. Usually, 6000 features per minute are extracted from speech data and we could use hundreds of hours of data to train our systems which results in the value of N being in the order of hundreds of millions. So, using training data directly as pivots is simply impractical. That is the reason why we choose M pivot vectors instead of all the training data to compute a solution which is not the optimal KDA solution, but a subspace solution which nevertheless could be powerful. We call our method "subspace KDA" to emphasize that we operate in a smaller subspace of the high dimensional feature space $\mathcal{F}$ instead of the (at most) N dimensional subspace as in regular KDA. How well the subspace KDA performs depends on how well we can approximate the subspace with a much smaller dimensional subspace (of at most M dimensions) by using appropriate pivot vectors.

It is interesting to note that, subspace KDA amounts to choosing a new set of features by nonlinearly transforming the original ones and working with the new features for the rest of the problem as we explore in the next section.

4. CHOOSING PIVOT VECTORS

Pivot vectors are an essential part of the subspace KDA transform. Intuitively, we need to choose them appropriately for better performance. If we increase the number of pivots, it becomes harder to compute the statistics and $\tilde{B}$ and $\tilde{W}$ matrices may become more rank-deficient.

If we choose the pivot vectors to be the basic unit vectors, that is $p_j = e_j$ for $j = 1, \ldots, n$, where e_j is the basic unit vector with value 1 at index j and 0 elsewhere. In this case, for a polynomial kernel with parameter d and c:

$$\tilde{x}_j = (x_j + c)^d,$$

that is each feature vector is elementwise transformed nonlinearly as shown above to $\tilde{x}$ and the vectors $\tilde{x}$ act as our new features.

A more intelligent choice for pivot vectors would be to choose them among training vectors. One approach is to choose them randomly [8]. A resonable choice is to use class means as pivot vectors. The class means can be considered to "summarize" the training data, hence the resulting $\boldsymbol{\theta}$ matrix can be assumed to approximate the true KDA transform better by choosing pivot vectors that summarize the training data. Looking from another angle, we would like to get mapped pivots $\Phi(p_j)$ that are most independent from each other, yet still have a span that is close to the span of the original mapped training vectors. This is our intuition, but it has to be studied mathematically in rigor. For the purposes of this paper, we leave the discussion here and continue to our results.

5. RESULTS

We applied the discussed technique to the phone recognition problem in TIMIT database [9]. We mapped 64 phones in TIMIT transcriptions down to 48 as in [9] for obtaining monophone models. During performance calculations, we further mapped 48 phones down to 39 as is typical[9].

We built triphone models using different features with 39 dimensions each. MFCC features are standard 12 cepstra + energy and Δ and $\Delta\Delta$ dynamic features. LDA features are obtained by transforming 91 dimensional spliced static MFCC features from 7 neighboring frames to 39 dimensions and applying MLLT transform afterwards. SKDA features are obtained by the procedure defined in section 3 using 48 monophone spliced means as pivot vectors ($M = 48$). We used Gaussian RBF as the kernel operator when obtaining 48-dimensional $\tilde{x}$ features with parameter c being 3 times the L_2 norm squared of the average training vector (for numerical stability). Then, the dimension is reduced from 48 to 39 using LDA-type processing as defined above. MLLT is applied to SKDA features as well. We report here phone recognition results for an initial study of the effectiveness of the subspace KDA technique without using any language model on phone sequences. The results are presented in Table 1. Correct detection results ignore insertion errors as in [9]. As it can be observed in the table, we could not achieve increase in accuracy by using SKDA features instead of MFCC or regular LDA features in this work. The reason for worse results for SKDA could be due to using a small subspace of the mapped training data instead of the whole space. We will explore different pivot vector options in the future to obtain better performance.

Features	MFCC	LDA	SKDA
Accuracy	62.92%	70.59 %	54.97%
Correct detection	80.34%	82.34 %	71.53%

Table 1. Phone recognition accuracy on TIMIT test set with different features of dimension 39.

6. CONCLUSION

Unlike many pattern classification problems, speech recognition involves many instances of training data due to relative ease of obtaining speech data. This fact is useful in one sense since we can use huge amounts of data to train complex statistical models for speech recognition. However, for the formulation of the Kernel Fisher's Discriminant, this is a disadvantage since the problem is transformed into a problem in an N dimensional space where N is the amount of training data.

We propose a subspace version of kernel discriminant analysis that can be suitable for speech recognition and evaluate its performance in a phone recognition task. Our initial experiments did not yield an increase in performance for SKDA over MFCC or LDA features. This could be due to the suboptimal subspace dimension and the choice of pivot vectors in this study. We will explore different configurations in our future work.

SKDA needs to be tested for larger databases and large vocabulary speech recognition tasks. However, it is safe to say that SKDA has great potential for improving speech recognititon performance since it enables "nonlinear" transformation of features to a more discriminative space with minimal increase in computation time.

7. REFERENCES

[1] K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Tr. Neural Net.*, vol. 12, no. 2, pp. 181–202, March 2001.

[2] R. O. Duda and P. E. Hart, *Pattern classification and scene analysis*, John Wiley & Sons, New York, 1973.

[3] R. A. Gopinath, "Maximum likelihood modeling with Gaussian distributions for classification," in *Proc. IEEE Conf. Acoust. Speech Sig. Proc.*, volume 2, pp. 661–4, 1998.

[4] N. Kumar and A. G. Andreou, "Heteroscedastic discriminant analysis and reduced rank HMMs for improved speech recognition," *Speech Communication*, vol. 26, pp. 283–97, 1998.

[5] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K.-R. Muller, "Fisher discriminant analysis with kernels," in *Neural Networks for Signal Processing*, pp. 41–48, 1999.

[6] S. Mika, *Kernel Fisher Discriminants*, PhD thesis, University of Technology Berlin, October 2002.

[7] G. Baudat and F. Anouar, "Generalized disriminant analysis using a kernel approach," *Neural Computation*, vol. 12, no. 10, pp. 2385–2404, 2000.

[8] 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 *Eurospeech*, 2003.

[9] K.-F. Lee and H.-W. Hon, "Speaker-independent phone recognition using hidden Markov models," *IEEE Tr. Acoust. Sp. Sig. Proc.*, vol. 37, no. 11, pp. 1641–1648, November 1989.