On Kernel Discriminant Analyses Applied to Phoneme Classification

András Kocsor

Research Group on Artificial Intelligence
of the Hungarian Academy of Sciences and University of Szeged
H-6720 Szeged, Aradi vértanúk tere 1., Hungary
kocsor@inf.u-szeged.hu

Abstract. In this paper we recall two kernel methods for discriminant analysis. The first one is the kernel counterpart of the ubiquitous Linear Discriminant Analysis (Kernel-LDA), while the second one is a method we named Kernel Springy Discriminant Analysis (Kernel-SDA). It seeks to separate classes just as Kernel-LDA does, but by means of defining attractive and repulsive forces. First we give technical details about these methods and then we employ them on phoneme classification tasks. We demonstrate that the application of kernel functions significantly improves the recognition accuracy.

1 Motivation

In the last two decades the dominant method for speech recognition has been the hidden Markov modeling approach [12]. In the meantime, the theory of machine learning has developed considerably and now has a wide variety of classification algorithms for pattern recognition problems [4]. One such development is the "kernel-idea", which has become a key notion in machine learning [2], [5], [13].

The primary goal of this paper is to show alternative methods for phoneme classification using state of the art kernel discriminant analyses. We describe here both the well-know kernel version of Linear Discriminant Analysis [1], [7], [9] and the Kernel Springy Discriminant Analysis, which we first proposed in [8].

2 Kernel Discriminant Analyses

Without loss of generality we shall assume that as a realization of multivariate random variables, there are n-dimensional real attribute vectors in a compact set $\mathcal{X}$ over $\mathbb{R}^n$ describing objects in a certain domain, and that we have a finite $n \times k$ sample matrix $X = [\mathbf{x}_1, \ldots, \mathbf{x}_k]$ containing k random observations. Let us assume as well that we have r classes and an indicator function $\mathcal{L} : \{1, \ldots, k\} \to \{1, \ldots, r\}$, where $\mathcal{L}(i)$ gives the class label of the sample $\mathbf{x}_i$. Let k_j further denote the number of vectors associated with label j in the sample data. Now let the dot product be implicitly defined by the kernel function $\kappa : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ in some finite or infinite dimensional dot product space $\mathcal{F}$ with associated mapping $\phi : \mathcal{X} \to \mathcal{F}$ such that

$$\forall \mathbf{x}, \mathbf{z} \in \mathcal{X} \quad \kappa(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{z}). \tag{1}$$

Usually ϕ is the feature map and $\mathcal{F}$ is the kernel feature space. This space and dot product calculations over it are defined only implicitly via the kernel function itself. The space $\mathcal{F}$ and map ϕ may not be explicitly known. We need only define the kernel function, which then ensures an implicit evaluation over $\mathcal{F}$. The construction of kernels, when such a mapping ϕ exists, is a non-trivial problem. Based on Mercer's theorem we can use continuous, symmetric and positive definite functions as kernels [2], [13].

The goal of discriminant analyses is to find a mapping $h : \mathcal{X} \to \mathcal{Y}$ which leads to a new set of features that are optimal according to a given class separation criterion. In the case of kernel discriminant analysis the mapping is nonlinear and has the following form: $\mathbf{z} \to AF^T\phi(\mathbf{z})$, $(\mathbf{z} \in \mathcal{X})$, where A is a method dependent, real $m \times k$ matrix and $F = [\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_k)]$ is called the image matrix of the sample. We should note here that $K = F^T F$ is the so-called kernel matrix and, $F^T\phi(\mathbf{z})$ can be calculated implicitly via the kernel function, i.e. $F^T\phi(\mathbf{z}) = [\kappa(\mathbf{x}_1, \mathbf{z}), \ldots, \kappa(\mathbf{x}_k, \mathbf{z})]^\top$.

2.1 Kernel-LDA

The 'kernelized‘ counterpart of Linear Discriminant Analysis, the Kernel-LDA defines the row vectors of matrix A by the stationary points of the following Rayleigh-quotient[7]:

$$\tau(\boldsymbol{a}) = \frac{(F\boldsymbol{a})^\top \mathcal{B}(F\boldsymbol{a})}{(F\boldsymbol{a})^\top \mathcal{W}(F\boldsymbol{a})}, \qquad F\boldsymbol{a} \in \mathcal{F}, \tag{2}$$

where $\mathcal{B}$ is the *Between-class*, while $\mathcal{W}$ is the *Within-class Scatter Matrix* of the images of the sample over ϕ. Here the *Between-class Scatter Matrix* $\mathcal{B}$ shows the scatter of the class mean vectors $\boldsymbol{\mu}_j$ around the overall mean vector $\boldsymbol{\mu}$:

$$\begin{aligned} \mathcal{B} &= \textstyle\sum_{j=1}^{r} \frac{k_j}{k}(\boldsymbol{\mu}_j - \boldsymbol{\mu})(\boldsymbol{\mu}_j - \boldsymbol{\mu})^\top \\ \boldsymbol{\mu} &= \frac{1}{k}\textstyle\sum_{i=1}^{k} \phi(\mathbf{x}_i) \\ \boldsymbol{\mu}_j &= \frac{1}{k_j}\textstyle\sum_{\mathcal{L}(i)=j} \phi(\mathbf{x}_i) \end{aligned} \tag{3}$$

The *Within-class Scatter Matrix* $\mathcal{W}$ represents the weighted average scatter of the covariance matrices $\mathcal{C}_j$ of the vectors with the class label j:

$$\begin{aligned} \mathcal{W} &= \textstyle\sum_{j=1}^{r} \frac{k_j}{k}\mathcal{C}_j \\ \mathcal{C}_j &= \frac{1}{k_j}\textstyle\sum_{\mathcal{L}(i)=j}(\phi(\mathbf{x}_i) - \boldsymbol{\mu}_j)(\phi(\mathbf{x}_i) - \boldsymbol{\mu}_j)^\top \end{aligned} \tag{4}$$

After some algebraic rearrangement, Eq. (2) takes the following form:

$$\tau(\boldsymbol{a}) = \frac{\boldsymbol{a}^\top K(R - \hat{I})K\boldsymbol{a}}{\boldsymbol{a}^\top K(I - R)K\boldsymbol{a}}, \qquad F\boldsymbol{a} \in \mathcal{F}, \tag{5}$$

where K is the kernel matrix, $[\hat{I}]_{ij} = 1/k$ and

$$[R]_{ij} = \begin{cases} \frac{1}{k_t} & \text{if } \ t = \mathcal{L}(i) = \mathcal{L}(j) \\ 0 & \textit{otherwise.} \end{cases} \tag{6}$$

This means that Eq. (2) can be expressed in terms of dot products of $\phi(\mathbf{x}_1), \ldots,$ $\phi(\mathbf{x}_k)$ and that the stationary points of this quotient can be computed by solving the generalized eigenvector problem $K(R - \hat{I})K\boldsymbol{a} = \lambda(K(I - R)K)\boldsymbol{a}$. To define the transformation matrix A of Kernel-LDA we use only those eigenvectors which correspond to the m dominant real eigenvalues.

2.2 Kernel-SDA

Kernel Springy Discriminant Analysis (Kernel-SDA) [8] was invented with goals very similar to those of Kernel-LDA. The name Kernel Springy Discriminant Analysis stems from the utilization of a spring & antispring model, which involves searching for directions with optimal potential energy using attractive and repulsive forces. In our case sample pairs in each class are connected by springs, while those of different classes are connected by antisprings. New features can be easily extracted by taking the projection of a new point in those directions having a small spread in each class, while different classes are spaced out as much as possible.

Now let the dot product again be implicitly defined by the kernel function κ in some finite or infinite dimensional feature space $\mathcal{F}$ with associated transformation ϕ such that $\kappa(\mathbf{x}, \mathbf{z}) = \phi(\mathbf{x}) \cdot \phi(\mathbf{z})$ for all $\mathbf{x}, \mathbf{z}$. Further, let $\delta(F\boldsymbol{a})$ the potential of the spring model along the direction $F\boldsymbol{a}$ in $\mathcal{F}$, be defined by

$$\delta(F\boldsymbol{a}) = \sum_{i,j=1}^{k} ((\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j))^\top F\boldsymbol{a})^2 [M]_{ij}, \quad F\boldsymbol{a} \in \mathcal{F} \tag{7}$$

where

$$[M]_{ij} = \begin{cases} -1, \text{ if } \mathcal{L}(i) = \mathcal{L}(j) \\ \ \ 1, \text{ otherwise} \end{cases} \qquad i, j = 1, \ldots, k. \tag{8}$$

Naturally, the elements of matrix M may be initialized with values different from ±1 as well. The elements can be considered as a kind of force constant and any pair of data points can have different force constant values. Similar to Kernel-LDA, Kernel-SDA defines the row vectors of matrix A by the stationary points of a Rayleigh-quotient, which in this case has the following form:

$$\tau(\boldsymbol{a}) = \frac{\delta(F\boldsymbol{a})}{(F\boldsymbol{a})^\top (F\boldsymbol{a})} = \frac{(F\boldsymbol{a})^\top \mathcal{D} (F\boldsymbol{a})}{(F\boldsymbol{a})^\top (F\boldsymbol{a})}, \tag{9}$$

where

$$\mathcal{D} = \sum_{i,j=1}^{k} \left(\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\right) \left(\phi(\mathbf{x}_i) - \phi(\mathbf{x}_j)\right)^\top [M]_{ij}. \tag{10}$$

Technically speaking, with the above τ definition Kernel-SDA searches for those directions $F\boldsymbol{a} \in \mathcal{F}$ along which a large potential is obtained. It is straightforward to see that the Rayleigh quotient for Kernel-SDA has the form:

$$\tau(\boldsymbol{a}) = 2\frac{\boldsymbol{a}^\top K(\tilde{M} - M)K^\top \boldsymbol{a}}{\boldsymbol{a}^\top K \boldsymbol{a}}, \tag{11}$$

where K is the kernel matrix and $\tilde{M}$ is a diagonal matrix with the sum of each row of M in the diagonal. Eq. (11) means that Eq. (9) can be expressed as a function of dot products of $\phi(\mathbf{x}_1), \ldots, \phi(\mathbf{x}_k)$. Now the stationary points of $\tau(\boldsymbol{a})$ can be obtained via an eigenanalysis of the following generalized eigenproblem: $(K(\tilde{M} - M)K^\top)\boldsymbol{a} = \lambda K\boldsymbol{a}$. To define the transformation matrix A we use the dominant m eigenvectors.

3 Phoneme Classification Results

Now we proceed with a description of the experiments. In this section we investigate the effect of the previous methods applied prior to classification in the phoneme classification task.

Evaluation Domain. The classification techniques combined with discriminant analyses as feature space transformation methods were compared using a corpus which consists of several speakers pronouncing Hungarian numbers. 77% of the speakers were used for training and 23% for testing. The ratio of male and female talkers was 50%-50% in both the training and testing sets. The recordings were made using a commercial microphone in a reasonably quiet environment, at a sample rate of 22050Hz. The whole corpus was manually segmented and labeled. Since some of these labels represented only allophonic variations of the same phoneme, some labels were fused, and so we actually worked with a set of 28 labels. We made tests as well with two other groupings where the labels were grouped into 11 and 5 classes, based on phonetic similarity. Hence we had three phonetic groupings, which henceforth will be denoted by *grp1*, *grp2* and *grp3*.

Initial Features. Before feature extraction the energy of each word was normalized. After this the signals were processed in 512-point frames (23.2 ms), where the frames overlapped by a factor of 3/4. A Fast Fourier Transform was applied on each frame. After that 24 critical band energies and 16 mel-frequency cepstral coefficients were calculated. Besides the above ones we also wanted to do experiments with some more phonetically based features like formants. We used gravity centers in 4 frequency bands as a crude approximation for formants. Doing this we got $24 + 16 + 4$ features altogether for each frame. Afterwards, for each feature we took the average of the frame-based measurements for the first quarter, the central part and the last quarter of the phoneme, which led to $44 \times 3 = 132$ features for each phoneme. By adding the duration of each phoneme to this set we finally got a feature set consisting of 133 elements.

Feature Space Transformation. After initial feature extraction we applied feature space transformation methods, hoping for a better classification. Besides LDA and SDA we also employed Principal Component Analysis (PCA), which served as a baseline method for comparison. In the case of PCA, SDA, Kernel-SDA the original feature space was reduced to 32 dimensions, while in the case of LDA and Kernel-LDA the number of features kept was always the number of classes minus one.

Learning Methods. We employed five well-known classification algorithms during the tests. TiMBL [3] is a Memory Based Learner which means a new example is evaluated based on consuming the previous examples stored in the memory. C4.5 [11] is based on the well-known ID3 tree learning algorithm. The OC1 (Oblique Classifier 1) algorithm [10] learns by creating *oblique* decision trees. The fourth, Artificial Neural Networks [4], is a conventional pattern recognition tool. In the experiments we employed the most common feed-forward multilayer perceptron network with the backpropagation learning rule. *Gaussian Mixture Model* (GMM) [4] is a well-known discriminative learning method. It assumes that the class-conditional probability distribution can be well approximated by a convex sum of multidimensional normal distributions.

Experimental Results and Evaluation. The same experiments were carried out on the three phoneme groupings *grp1*, *grp2*, *grp3*, all the learning methods being tested not just on each set but with each transformation technique. Table 1 depicts the recognition accuracies for *grp1*, *grp2* and *grp3*, respectively. The columns show the five feature transformation methods and the rows correspond to the classification algorithms applied. The maximum is shown in bold.

On examining the classifiers the first thing we notice is that the general preference of the methods on the phoneme classification task is the following: C4.5$\prec$OC1$\prec$GMM$\prec$TiMBL$\prec$ANN. As regards the feature space transformation methods we realized that the base-line PCA method was outperformed by LDA (cf. [6]) and SDA, which was in turn surpassed by their kernel versions (Kernel-LDA and Kernel-SDA). Based on these observations we summarize the efficiency relations of the methods as follows: PCA$\prec$(LDA$\approx$SDA)$\prec$(Kernel-LDA$\approx$Kernel-SDA). Another thing we realized was that the efficiency of Kernel-SDA improved when the number of classes decreased. We also noticed that Kernel-SDA considerably helps the efficiency of C4.5, which for instance resulted in the best accuracy value for the *grp*3 recognition problem.

4 Conclusions

This paper sought to study the effects of some kernel discriminant analyses on phoneme classification, a basic task of speech recognition. After inspecting the test results we can confidently say that it is worth experimenting with these methods in order to obtain better classification results. The use of non-linearity brought further improvements on the classification accuracy. Still, we should note that the goals of feature space transformation and learning are practically the same. That is, if we have a very efficient learner then there is almost no need for a feature space transformation. Put the other way round, a proper transformation of the feature space may make the data so easily separable that quite simple learners will suffice. These are, of course, extreme examples.

Table 1. Recognition accuracies for the classifier-transformation combinations

phoneme groupings	classifier	PCA	LDA	SDA	K-LDA	K-SDA
Grp1 (28 classes)	TIMBL	75.23	83.33	80.49	89.32	88.07
	C4.5	56.20	67.80	66.90	83.80	78.90
	OC1	60.17	70.86	68.91	75.55	78.56
	ANN	84.46	86.94	83.22	**90.86**	87.76
	GMM	74.82	86.23	79.85	89.20	82.94
Grp2 (11 classes)	TIMBL	82.74	86.11	84.21	93.96	93.96
	C4.5	69.30	83.00	79.60	88.58	92.50
	OC1	74.41	85.22	81.56	91.54	91.54
	ANN	90.60	89.78	89.18	92.84	**94.73**
	GMM	80.91	87.12	84.04	91.89	92.13
Grp3 (5 classes)	TIMBL	88.17	90.95	90.36	95.62	95.62
	C4.5	79.10	92.50	90.90	94.30	**96.60**
	OC1	86.88	92.67	88.83	95.39	92.84
	ANN	93.09	93.09	92.61	94.68	95.80
	GMM	90.13	92.26	89.24	93.61	89.06

References

1. G. Baudat and F. Anouar: Generalized discriminant analysis using a kernel approach, Neural Comput., Vol. 12 (2000) 2385-2404
2. N. Cristianini, J. Shawe-Taylor: An Introduction to Support Vector Machines and other kernel-based learning methods, Cambridge University Press (2000)
3. W. Daelemans, J. Zavrel, K. Sloot, A. Bosch: "TiMBL: Tilburg Memory Based Learner version 2.0 Reference Guide", ILK Technical Report - ILK 99-01, Computational Linguistics, Tilburg University, The Netherlands (1999)
4. R.O. Duda, P.E. Hart, D.G. Stork: Pattern Classification, Wiley & Sons, NY (2001)
5. Kernel Machines Web site: http://www.kernel-machines.org
6. A. Kocsor et al.: A Comparative Study of Several Feature Transformation and Learning Methods for Phoneme Classification, Int. Journal of Speech Technology, Vol. 3., No. 3/4 (2000) 263-276
7. A. Kocsor, L. Tóth, D. Paczolay: A Nonlinearized Discriminant Analysis and its Application to Speech Impediment Therapy, in V. Matousek et al. (eds.): Proc. of TSD 2001, Springer Verlag LNAI Series, Vol. 2166 (2001) 249-257
8. A. Kocsor, K. Kovács: Kernel Springy Discriminant Analysis and Its Application to a Phonological Awareness teaching System, in P. Sojka et al. (eds.): Proc. of TSD 2002, Springer Verlag LNAI Series, Vol. 2448 (2002) 325-328
9. S. Mika, G. Rätsch, J. Weston, B. Schölkopf, K.-R. Müller: Fisher discriminant analysis with kernels, in: Y.-H. Hu et al. (eds.): Neural Networks for Signal Processing IX, IEEE (1999) 41-48
10. S. K. Murthy, S. Kasif, S. Salzberg: A System for Induction of Oblique Decision Trees, Journal of Artificial Intelligence Research, Vol. 2 (1994) 1-32
11. J. R. Quinlan: C4.5: *Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, California (1993)
12. L. Rabiner, B.H. Juang: Fundamentals of Speech Recognition, Prent. Hall (1993)
13. V. N. Vapnik: Statistical Learning Theory, John Wiley & Sons Inc (1998)