Open access • Journal Article • DOI:10.1162/089976600300014980

Generalized Discriminant Analysis Using a Kernel Approach

G. Baudat, F. Anouar

Institutions: Institut national de la recherche agronomique

Published on: 01 Oct 2000 - Neural Computation (MIT Press)

Topics: Kernel Fisher discriminant analysis, Linear discriminant analysis, Optimal discriminant analysis, Multiple discriminant analysis and Kernel (statistics)

Generalized Discriminant Analysis Using a Kernel Approach

BAUDAT G. [(1)], ANOUAR F. [(2)]

(1) MEI, Mars Electronics International, Chemin Pont-du Centenaire 109,
Plan-les-Ouates, BP 2650, CH- 1211 Genève 2, Suisse
Email: gaston.baudat@eu.effem.com
(2) INRA-SNES, Institut National de Recherche en Agronomie,
Rue Georges Morel, 49071 Beaucouzé, France
E-mail: fatiha.anouar@geves.fr

Abstract

We present a new method that we call Generalized Discriminant Analysis (GDA) to deal with nonlinear discriminant analysis using kernel function operator. The underlying theory is close to the Support Vector Machines (SVM) insofar as the GDA method provides a mapping of the input vectors into high dimensional feature space. In the transformed space, linear properties make it easy to extend and generalize the classical Linear Discriminant Analysis (LDA) to non linear discriminant analysis. The formulation is expressed as an eigenvalue problem resolution. Using a different kernel, one can cover a wide class of nonlinearities. For both simulated data and alternate kernels, we give classification results as well as the shape of the separating function. The results are confirmed using a real data to perform seed classification.

1. Introduction

Linear discriminant analysis (LDA) is a traditional statistical method which has proven successful on classification problems [Fukunaga, 1990]. The procedure is based on an eigenvalue resolution and gives an exact solution of the maximum of the inertia. But this method fails for a nonlinear problem. In this paper, we generalize LDA to nonlinear problems and develop a Generalized Discriminant Analysis (GDA) by mapping the input space into a high dimensional feature space with linear properties. In the new space, one can solve the problem in a classical way such as the LDA method. The main idea is to map the input space into a convenient feature space in which variables are nonlinearly related to the input space. This fact has been used in some algorithms such as unsupervised learning algorithms [Kohonen, 1994] [Anouar, Badran, Thiria, 1998] and in support vector machine (SVM) [Vapnik, 1995] [Schölkopf, 1997]. In our approach, the mapping is close to the mapping used for support vector method which is a universal tool to solve pattern recognition problems. In the feature space, the SVM method selects a subset of the training data and defines a decision function that is a linear expansion on a basis whose elements are nonlinear functions parameterized by the support vectors. SVM was extended to different domains such as regression and estimation [Vapnik, Golowich, Smola,1997]. The basic ideas behind SVM have been explored by Schölkopf et al. to extend principal component analysis (PCA) to nonlinear kernel PCA for extracting structure from high dimensional data set [Schölkopf, Smola, Müller, 1996] [Schölkopf, Smola, Müller, 1998]. The authors also propose nonlinear variant of other algorithms such that Independent Component Analysis (ICA) or kernel-k-means. They mention that it would be desirable to develop nonlinear form of discriminant analysis based on kernel method. A related approach using an explicit map into a higher dimensional space instead of kernel method was proposed by [Hastie, Tibshirani, Buja, 1994]. The foundations for the kernel developments described here can be connected to kernel PCA. Drawn from these works, we show how to express the GDA method as a linear algebraic formula in the transformed space using kernel operators.
In the next section we introduce the notations used for this purpose. Then we review the standard LDA method. The formulation of the GDA using dot product and matrix form is explained in the third section. Afterwards, we develop the eigenvalue resolution. The last section is devoted to the experiments on simulated and real data.

2. Notations

Let x be a vector of the input set X with M elements. X_l designs subsets of X, thus: $X = \bigcup_{l=1}^{N} X_l$. N is the number of the classes. x^t represents the transpose of the vector x. The cardinality of the subsets X_l is denoted by n_l thus $\sum_{l=1}^{N} n_l = M$. C is the covariance matrix:

$$C = \frac{1}{M} \sum_{j=1}^{M} x_j x_j^t \tag{1}$$

Suppose that the space X is mapped into a Hilbert space F through a nonlinear mapping function ϕ :

$$\begin{aligned} \phi : X &\to F \\ x &\to \phi(x) \end{aligned} \tag{2}$$

The covariance matrix in the feature space F is:

$$V = \frac{1}{M} \sum_{j=1}^{M} \phi(x_j) \phi^t(x_j) \tag{3}$$

We assume that the observations are centered in F [Schölkopf, Smola, Müller, 1998]. Nevertheless, the way to center data in the feature space is given in appendix C.

By B we denote the covariance matrix of the class centers. B represents the inter-classes inertia in the space F:

$$B = \frac{1}{M} \sum_{l=1}^{N} n_l \bar{\phi}_l \bar{\phi}_l^{\,t} \tag{4}$$

Where $\bar{\phi}_l$ is the mean value of the class l:

$$\bar{\phi}_l = \frac{1}{n_l} \sum_{k=1}^{n_l} \phi(x_{lk}) \tag{5}$$

Where x_{lk} is the element k of the class l.

In the same manner the covariance matrix (3) of F elements can be rewritten using the classe indexes :

$$V = \frac{1}{M} \sum_{l=1}^{N} \sum_{k=1}^{n_l} \phi(x_{lk}) \phi^t(x_{lk}) \tag{6}$$

V represents the total inertia of the data into F.

In order to simplify, when there is no ambiguity in index of x_{lj} , the class index l is omitted.

In order to be able to generalize LDA to nonlinear case we formulate it in a way which uses exclusively dot product. Therefore, we consider an expression of dot product on the Hilbert space F [Aizerman, Braverman, Rozonoér, 1964] [Boser, Guyon, Vapnik, 1992] given by the following kernel function:

$$k(x_i, x_j) = k_{ij} = \phi^t(x_i) \phi(x_j)$$

For a given classes p and q, we express this kernel function by:

$$(k_{ij})_{pq} = \phi^t(x_{pi}) \phi(x_{qj}) \tag{7}$$

Let K be a *(MxM)* matrix defined on the class elements by $(K_{pq})_{\substack{p=1,\dots,N \\ q=1,\dots,N}}$, where (K_{pq}) is a matrix composed of dot product in the feature space F:

$$K = (K_{pq})_{\substack{p=1,\dots,N \\ q=1,\dots,N}} \quad \textit{where} \quad K_{pq} = (k_{ij})_{\substack{i=1,\dots,n_p \\ j=1,\dots,n_q}} \tag{8}$$

K_{pq} is a $(n_p x n_q)$ matrix and K is a (MxM) symmetric matrix such that ${K_{pq}}^t = K_{pq}$.

We also introduce the matrix:

$$W = (W_l)_{l=1,\dots,N} \tag{9}$$

Where W_l is a $(n_l x n_l)$ matrix with terms all equal to: $\frac{1}{n_l}$. W is a (MxM) block diagonal matrix.

In the next section, we will formulate the generalized discriminant analysis method in the feature space *F* using the definition of the covariance matrix *V* (6), the classes covariance matrix *B* (4), the matrices *K* (8) and *W* (9).

3. GDA Formulation in feature space

LDA is a standard tool for classification. It is based on a transformation of the input space into a new one. The data are described as a linear combination of the new coordinate values which are called principal components and represent the discriminant axis. For the common LDA [Fukunaga,1990], the classical criteria for class separability is defined by the quotient between the inter-classes inertia and the intra-classes inertia. This criteria should be larger when the inter-classes inertia is larger and the intra-classes inertia is smaller. It was shown that this maximization is equivalent to eigenvalue resolution [Fukunaga,1990] (see appendix A). Assuming that the classes have a multivariate Gaussian distribution, each observation can be assigned to the class having the maximum posterior probability using the Mahalanobis distance.

Using kernel functions, we generalize LDA to the case where in the transformed space the principal component are nonlineraly related to the input variables. The kernel operator *K* allows the construction of nonlinear separating function in the input space that is equivalent to linear separating function in the feature space *F*. As such for the LDA, the purpose of the GDA method is to maximize the inter-classes inertia and minimize the intra-classes inertia. This maximization is equivalent to the following eigenvalue resolution : we have to find eigenvalues λ and eigenvectors υ, solutions of the equation:

$$\lambda V\upsilon = B\upsilon \tag{10}$$

The largest eigenvalue of (10) gives the maximum of the following quotient of the inertia (Appendix A) :

$$\lambda = \frac{\upsilon^t B\upsilon}{\upsilon^t V\upsilon} \tag{11}$$

As the eigenvectors are linear combinations of *F* elements, there exist coefficients $\alpha_{pq}\,(p=1,\cdots,N;q=1,\cdots,n_p)$ such that:

$$\upsilon = \sum_{p=1}^{N}\sum_{q=1}^{n_p}\alpha_{pq}\phi(x_{pq}) \tag{12}$$

All solutions υ lie in the span of $\phi(x_{ij})$.

Let us consider the coefficient vector $\alpha = (\alpha_{pq})_{p=1,\cdots,N;q=1,\cdots,n_p}$; it can be written in a condensed way as $\alpha = (\alpha_p)_{p=1,\cdots,N}$, where $\alpha_p = (\alpha_{pq})_{q=1,\cdots,n_p}$, α_p is the coefficient of the vector υ in the class *p*.

We show in the appendix B, that (11) is equivalent to the following quotient:

$$\lambda = \frac{\alpha^t KWK\alpha}{\alpha^t KK\alpha} \tag{13}$$

This equation developed in appendix B is obtained by multiplying (10) by $\phi^t(x_{ij})$ which makes it easy to rewrite in a matrix form. (10) has the same eigenvector as [Schölkopf, Smola, Müller, 1998] :

$$\lambda\phi^t(x_{ij})V\upsilon = \phi^t(x_{ij})B\upsilon \tag{14}$$

We then express (14) by using the powerful idea of dot product [Aizerman, Braverman, Rozonoér, 1964] [Boser, Guyon, Vapnik, 1992] between the mapped pattern defined by the matrices *K* in *F* without having to carry out the map ϕ. We rewrite the two terms of the equality (14) in a matrix form using the matrices *K* and *W* which gives (13) (see appendix B).

The purpose of the next section is to resolve the eigenvector system (13), which requires an algebraic decomposition of the matrix *K*.

4. Eigenvalue resolution

Let us use the eigenvectors decomposition of the matrix *K*,

$$K = P\Gamma P^t$$

Here, we consider Γ the diagonal matrix of non-zero eigenvalues and P the matrix of normalized eigenvectors associated to Γ. Thus Γ^{-1} exists. *P* is an orthonormal matrix that is:

$P^t P = I$, where *I* is the identity matrix.

Substituting *K* in (13), we get:

$$\lambda = \frac{(\Gamma P^t \alpha)^t P^t W P (\Gamma P^t \alpha)}{(\Gamma P^t \alpha)^t P^t P (\Gamma P^t \alpha)}$$

Let us proceed to variable modification using β such that:

$$\beta = \Gamma P^t \alpha \tag{15}$$

Substituting in the latter formula we get (16):

$$\lambda = \frac{\beta^t P^t W P \beta}{\beta^t P^t P \beta} \tag{16}$$

Therefore we obtain:

$$\lambda P^t P \beta = P^t W P \beta$$

As P is orthonormal, the latter equation can be simplified and gives (17), for which solutions are to be found by maximizing λ:

$$\lambda \beta = P^t W P \beta \tag{17}$$

For a given β there exists at least one α satisfying (15) in the form: $\alpha = P\Gamma^{-1}\beta$. α is not unique.

Thus the first step of the system resolution consists in finding β according to the equation (17), which corresponds to a classical eigenvector system resolution. Once β are calculated, we compute α.
Note that one can achieve this resolution by using other decomposition of *K* or other diagonalization method. We refer to the QR decomposition of *K* [Wilkinson, Reinsch, 1971] which allows working in a subspace which simplifies the resolution.

The coefficients α are normalized by requiring that the corresponding vectors υ be normalized in *F*:

$$\upsilon^t \upsilon = 1$$

Using (11):

$$\upsilon^t \upsilon = \sum_{p=1}^{N} \sum_{q=1}^{n_p} \sum_{l=1}^{N} \sum_{h=1}^{n_l} \alpha_{pq} \alpha_{lh} \phi^t(x_{pq}) \phi(x_{lh}) = 1$$

$$\upsilon^t \upsilon = \sum_{p=1}^{N} \sum_{l=1}^{N} \alpha_p{}^t K_{pl} \alpha_l = 1$$

$$\upsilon^t \upsilon = \alpha^t K \alpha = 1 \tag{18}$$

The coefficients α are divided by $\sqrt{\alpha^t K \alpha}$ in order to get normalized vectors υ.
Knowing the normalized vectors υ, we then compute projections of a test point z by:

$$\upsilon^t \phi(z) = \sum_{p=1}^{N} \sum_{q=1}^{n_p} \alpha_{pq} k(x_{pq}, z) \tag{19}$$

GDA procedure is summarized in the following steps:

1. Compute the matrices *K* (7) (8) and *W* (9),
2. Decompose *K* using eigenvectors decomposition,
3. Compute eigenvectors β and eigenvalues of the system (17),
4. Compute eigenvectors υ using α (12) and normalize them (18),
5. Compute projections of test points onto the eigenvectors υ (19).

5. Experiments

In this section, two sets of simulated data are first studied, then the results are confirmed on Fisher's iris data [Fisher,1936] and on the seed classification. The type of simulated data is chosen in order to emphasize the influence of the kernel function. We have used a polynomial kernel of degree *d* and a gaussian kernel to solve the corresponding classification problem. Other kernel forms can be used, provided that they fulfil the Mercer's theorem [Vapnik,1995], [Schölkopf, Smola, Müller, 1998]. Polynomial and gaussian kernels are among the classical kernels used in the literature.

Polynomial kernel using a dot product: $k(x, y) = (x.y)^d$, where d is the polynomial degree.

Gaussian kernel: $k(x, y) = \exp(\frac{-\|x - y\|^2}{\sigma})$, where the parameter σ has to be chosen.

These kernel functions are used to compute *K* matrix elements: $k_{ij} = k(x_i, x_j)$.

5.1. Simulated Data

Example 1: separable data

Without loss of generality, two 2-d classes are generated and studied in the feature space obtained with different type of kernel function. This example aims to illustrate the behavior of the GDA algorithm according to the choice of the kernel function.

For the first class (class1), a set of 200 points (x, y) is generated as in the following:

X is a normal variable with a mean equal to 0 and a standard variation equal to $\sqrt{2}$: $X \sim N(0, \sqrt{2})$,

Y is generated according to the following variable: $Y_i = X_i * X_i + N(0, 0.1)$.

The second class (class2) corresponds to 200 points (x, y) , where

X is a variable such that: $X \sim N(0, 0.001)$,

Y is a variable such that: $Y \sim N(2, 0.001)$.

Note that the variables X and Y are independent here. 20 examples by class are randomly chosen to compute the separating function. Both data and the separating function are visualized on the figure 1.

We construct a decision function corresponding to a polynomial of degree two. Suppose that the input vector $x = (x_1, x_2, \cdots, x_t)$ has *t* components, where *t* is termed the dimensionality of the input space. The feature space *F* has $\frac{t(t+1)}{2}$ coordinates of the form $\phi(x) = (x_1^2, \ldots, x_t^2, x_1x_2, \ldots, x_ix_j, \ldots)$ [Poggio, 1975] [Vapnik, 1995]. The separating hyperplane in *F* space is a second degree polynomial in the input space. The separating function is computed on the training set by finding a threshold such that the projection curves (figure 1.b) are well separated. Here, the chosen threshold corresponds to the value 0.7. The polynomial separation is represented for the whole data on the figure 1.a):

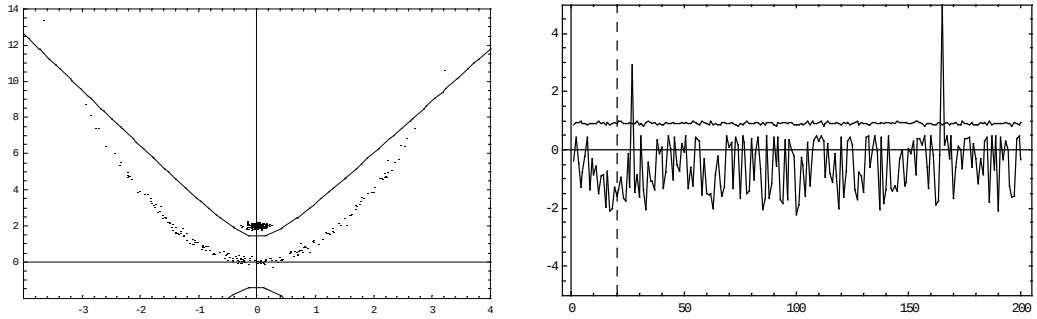


Figure 1: a) Represents the separating function for the two classes using the first discriminant axis. In the input space the separating function is computed using a polynomial kernel type with d=2.
b) Projections of all examples on the first axis with an eigenvalue λ equal to 0.765.
Dotted line separates the training examples from the others.

Notice that the nonlinear GDA produces a separating function which reflects the structure of the data. As for the LDA method, the maximal number of principal components with non-zero eigenvalues is equal to the number of classes minus one [Fukunaga, 1990]. For this example, the first axis is sufficient to separate the two classes of the learning set.

It can be seen from the figure 1.b) that the two classes can clearly be separated using one axis except for two examples where the curves overlap. The two misclassified examples do not belong to the training set. The vertical dotted line indicates the 20 examples of the training set of each class. We can observe that the examples of the class 2 are almost all projected on one point.

In the following, we give the results using a gaussian kernel. As previously, the separating function is computed on the training set and represented for the whole data on the figure 2.a).

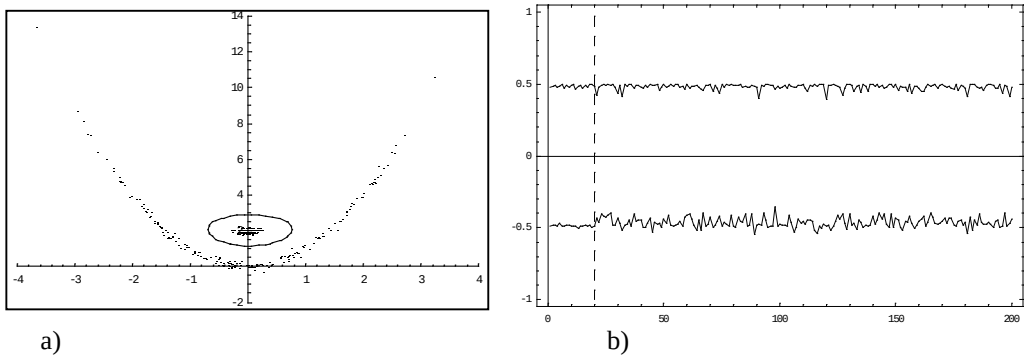


Figure 2: a) Represents the separating function on the whole data using a gaussian kernel with σ=1,
b) Represents the projection of all examples on the first discriminant axis with an eigenvalue λ equal to 0.999.

In this case, all examples are well separated. When projecting the examples on the first axis, we obtain the curves given on the figure 2.b), which are well-separated lines. The positive line corresponds to the class 2 and the negative one corresponds to the class 1. The line of the threshold zero separates all the training examples as well as all the testing examples. The corresponding separation in the input space is an ellipsoid (figure 2.a).

Example 2: non-separable data

We consider two overlapped clusters in two dimensions. Each cluster contains 200 samples. For the first cluster, samples are uniformly located upon a circle of radius of 3. A normal noise with a variance of 0.05 is added to the X and Y coordinates. For the second cluster, the X and Y coordinates follow a normal

distribution with a mean vector $\begin{pmatrix}0\\0\end{pmatrix}$ and a covariance matrix $\begin{pmatrix}5.0 & 4.9\\4.9 & 5.0\end{pmatrix}$. This example will illustrate the behavior of the algorithm on non-separated data and the classification results will be compared to SVM results. Therefore, 200 samples of each cluster are used for the learning step and 200 for the testing step. The GDA is performed using a Gaussian kernel operator with a sigma equal to 0.5. The separating function and the whole data are represented on the figure 3.

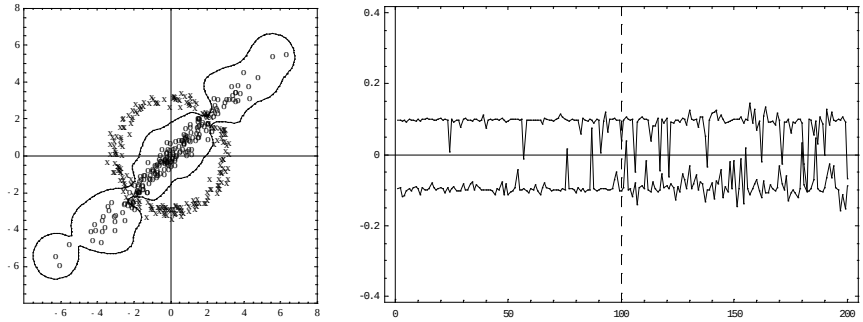


Figure 3: a) 200 samples of the first cluster are represented by cross and 200 samples of the second cluster by circles. The separating function is computed on the learning set using a Gaussian kernel with sigma = 0.5.
b) Projections of all samples on the first axis with an eigenvalue equal to 0.875. The dotted vertical line separates the learning sample from the testing samples.

To evaluate the classification performance we use the Mahalanobis distance to assign samples to the classes. The percentage of correct classification for the learning set is 98% and for the testing set it is equal to 93.5%. The SVM classifier of a free Matlab software [Gunn, 1997] has been used to classify these data with the same kernel operator and the same value of sigma (sigma = 0.5 and C=∞). The percentage of correct classification for the learning set is 99% and for testing set it is equal to 83%. By performing the parameter C of the SVM classifier with a Gaussian kernel, the best results obtained (with sigma = 1 and C=1) are 99% on the learning set and 88% on the testing set.

5.2. Fisher's Iris data

The iris flower data were originally published by Fisher [Fisher, 1936], for examples in discriminant analysis and cluster analysis. Four parameters, including sepal length, sepal width, petal length, and petal width, were measured in millimeters on fifty iris specimens from each of three species, Iris *setosa*, Iris *versicolor* and Iris *virginica*. So the set of data contains 150 examples with 4 dimensions and 3 classes. One class is linearly separable from the two other; the latter are not linearly separable from each other. For the following tests all iris examples are centered. Figure 4 shows the projection of the examples on the first two discriminant axes using LDA method, which is a particular case of GDA when the kernel is a polynomial with degree one.

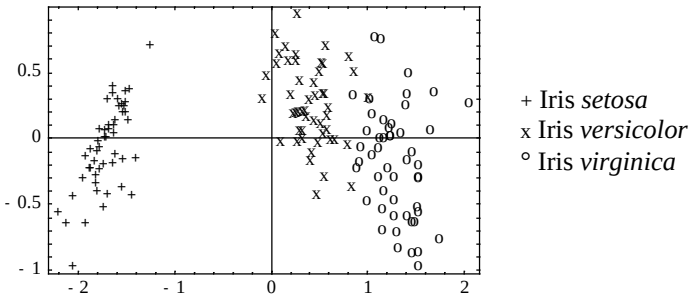


Figure 4: Represents the projection of Iris data on the first two axes using a linear discrimination LDA method. LDA is derived from GDA associated to a polynomial kernel with degree one (d=1).

Relation to kernel PCA

Kernel PCA proposed by Schölkopf [Schölkopf, Smola, Müller, 1998] is designed to capture the structure of the data. The method reduces the sample dimension in a nonlinear way for the best representation in lower dimensions keeping the maximum of inertia. However, the best axis for the representation is not necessarily the best axis for the discrimination. After Kernel PCA, the number of features is selected according to the percentage of initial inertia to keep for the classification process. The authors propose different classification methods to achieve this task. Kernel PCA is a useful tool for unsupervised and nonlinear problem for feature extraction. In the same manner GDA can be used for supervised and nonlinear problem for feature extraction and for classification. Using GDA, one can find a reduced number of discriminant coordinate that are optimal for separating the groups. With two such coordinates one can visualise a classification map that partitions the reduced space into regions.

Kernel PCA and GDA can produce a very different representation which highly dependent on the structure of the data. Figure 5 shows the results of applying both kernel PCA and GDA to the iris classification problem using the same gaussian kernel with σ=0.7. The projection on the first two axes seems to be insufficient for kernel PCA to separate the classes, more than two axes will certainly improve the separation. With two axes, GDA algorithm produces better separation of this data because of the use of the inter-classes inertia.

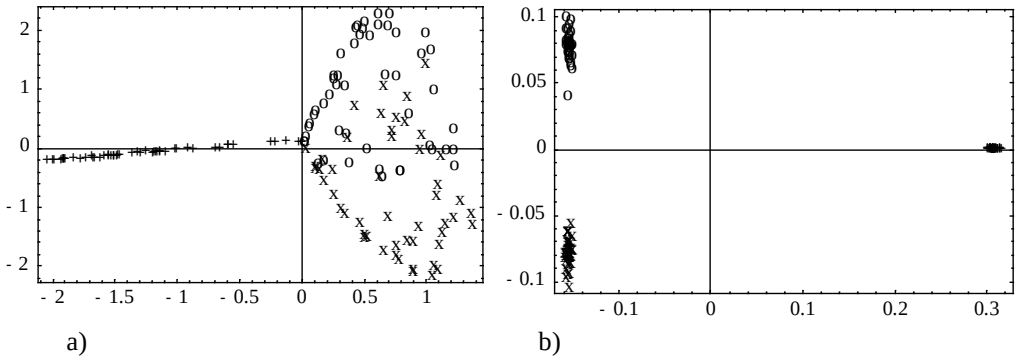


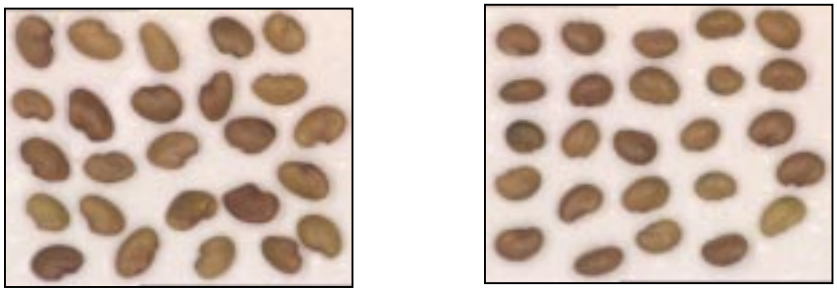
Figure 5: a) Gives the projection of the whole examples on the first two axes using nonlinear kernel PCA with a gaussian kernel and σ=0.7 .
b) Gives the projection of the examples on the first two axes using the GDA method with a gaussian kernel σ=0.7.

As can be seen from the figure 5.b) the three classes are well separated: each class is nearly projected on one point, which is the center of gravity. Note that the first two eigenvalues are equal to 0.999 and 0.985.

In addition, we assigned the test examples to the nearest class according to the Mahalanobis distance and using the prior probability of each class. We apply the assignment procedure with the "leave one out " test method. We measured the percentage of correct classification. For GDA the result is equal to 95.33% of correct classification. This percentage can be compared to those of Radial Basis Function network (RBF network) 96.7% and MultiLayer Perceptron network (MLP) 96.7% [Gabrijel, Dobnikar, 1997].

5.3. Seed classification

Seed samples were supplied by the National Seed Testing Station of France. The aim is to perform seed classification methods in order to help analysts for the successful operation of national seed certification. Seed characteristics are extracted using a vision system and image analysis software. Three seed species are studied: *Medicago sativa* L. (lucerne), *Melilotus sp* and *Medicago lupulina* L.. These species present the same appearance and are difficult for analysts to identify (figure 6).



a) *Medicago sativa* L. b) *Medicago lupulina* L.

Figure 6: Examples of images seeds a) *Medicago sativa* L. (lucerne) b) *Medicago lupulina* L.

224 training seeds and 150 testing seeds were placed in random positions and orientations in the field of the camera. Each seed was represented by five variables extracted from image seeds and describing the morphology and the texture of seeds. Different classification methods were compared in term of percentage of correct classification. The results are summarized on the table 1.

	Percentage of correct classification	
Method	Training	Test
k-nearest neighbors	81.7	81.1
Linear dicriminant analysis (LDA)	72.8	67.3
Probabilistic neural network (PNN)	100	85.6
Generalized dicriminant analysis (GDA) Gaussian kernel (sigma = 0.5)	100	85.1
Nonlinear Support vectors machines Gaussian kernel (sigma = 0.5, C=∞)	99	82.5

Table 1: Comparison of GDA and other classification methods for the discrimination of three seed species: *Medicago sativa* L. (lucerne), *Melilotus sp* and *Medicago lupulina* L.

k-nearest neighbors classification was performed with 15 neighbors and gives better results than LDA method. SVM classifier was tested with different kernel and for several values of the upper bound parameter C to relax the constraints. In this case the best results are 99% on the learning set and 85.2% on the testing set for C=1000. The classification results obtained by GDA method with a Gaussian kernel, probabilistic neural network (PNN) [Specht, Kalantri, Ahmed, Chan, 1990] [Musavi, 1993] and SVM classifier are nearly the same. However, the advantage of GDA method is that it based on a formula calculation and not on an optimization approximation such as for PNN classifier or SVM for which the user have to chose and adjust some parameters. Moreover, the SVM classifier is initially developed for two classes problems and its adaptation to multi-classes problems is time costing.

6. Discussion and future work

The dimensionality of the feature space is huge and depends on the size of the input space. A function which successfully separates the training data may not generalize well. One has to find a compromise between the training and the generalization performances. It was shown for SVM that, the test error depends only on the expectation of the number of support vectors and the number of training examples [Vapnik,1995], and not on the dimensionality of the feature space. Our current investigation is to establish the relationship between GDA resolution and SVM resolution. Therefore, we can improve, performance of generalization, accuracy and speed using the wide studies of SVM technique [Burges, Schölkopf, 1997] [Schölkopf, Smola, Müller, 1998]. Nevertheless, the fascinating idea of using a kernel approach is that we can construct an optimal separating hyperplane in the feature space without considering this space in an explicit form. We only have to calculate the dot product. But the choice of the kernel type remains an open problem. However, the possibility to use any desired kernels allows generalizing classical algorithms. For instance, there are similarities between GDA with a gaussian kernel

and probabilistic neural networks (PNN). Like the GDA method, PNN can be viewed as a mapping operator built on a set of input-output observations, but the procedure to define decision boundaries is different for the two algorithms. In GDA the eigenvalue resolution is used to find a set of vectors which define an hyperplane separation and give a global minimum according to the inertia criterion. In PNN the separation is found by trial-and-error measurement on the training set. The PNN and more general neural networks always find a local minimum.

7. Conclusion

We have developed a generalization of discriminant analysis as nonlinear discrimination. We described the algebra formulation and the eigenvalue resolution. The motivation for exploring the algebraic approach is to develop an exact solution and not an approximate optimization. The GDA method gives an exact solution even if some points require further investigation, such as the choice of the kernel function. In terms of classification performance, for the small databases studied here, the GDA method competes with support vector machines and probabilistic neural network classifier.

Appendix A

Given two symmetric matrices A and B with the same size. B is supposed inversible. It shown that [Saporta, 1990] :

The quotient $\frac{v^t A v}{v^t B v}$ is maximal for *v* eigenvector of $B^{-1}A$ associated to the large eigenvalue λ.

Maximizing the quotient requires that the derivative with respect to *v* vanish :

$$\frac{(v^t Bv)(2Av)-(v^t Av)(2Bv)}{(v^t Bv)^2}=0$$

Which implies :

$$B^{-1}Av=(\frac{v^t Av}{v^t Bv})v$$

v is then an eigenvector of $B^{-1}A$ associated to the eigenvalue $\frac{v^t Av}{v^t Bv}$. The maximum is reached for the largest eigenvalue.

Appendix B

In this appendix we rewrite formula (14) in a matrix form in order to obtain the formula (13):

$$\lambda\phi^t(x_{ij})V\upsilon=\phi^t(x_{ij})B\upsilon \qquad (14) \qquad\qquad \lambda=\frac{\alpha^t KWK\alpha}{\alpha^t KK\alpha} \qquad (13)$$

We develop each term of the equality (14) according to the matrices *K* and *W*., using (6) and (11), the left term of (14) gives:

$$V\upsilon=\frac{1}{M}\sum_{l=1}^{N}\sum_{k=1}^{n_l}\phi(x_{lk})\phi^t(x_{lk})\sum_{p=1}^{N}\sum_{q=1}^{n_p}\alpha_{pq}\phi(x_{pq})$$
$$=\frac{1}{M}\sum_{p=1}^{N}\sum_{q=1}^{n_p}\alpha_{pq}\sum_{l=1}^{N}\sum_{k=1}^{n_l}\phi(x_{lk})\left[\phi^t(x_{lk})\phi(x_{pq})\right]$$

$$\lambda\phi^t(x_{ij})V\upsilon = \frac{\lambda}{M}\sum_{p=1}^{N}\sum_{q=1}^{n_p}\alpha_{pq}\phi^t(x_{ij})\sum_{l=1}^{N}\sum_{k=1}^{n_l}\phi(x_{lk})\left[\phi^t(x_{lk})\phi(x_{pq})\right]$$

$$= \frac{\lambda}{M}\sum_{p=1}^{N}\sum_{q=1}^{n_p}\alpha_{pq}\sum_{l=1}^{N}\sum_{k=1}^{n_l}\left[\phi^t(x_{ij})\phi(x_{lk})\right]\left[\phi^t(x_{lk})\phi(x_{pq})\right]$$

Using this formula for all class i and for all its element j we obtain:

$$\lambda(\phi^t(x_{11}),\ldots,\phi^t(x_{1n_1}),\ldots,\phi^t(x_{ij}),\ldots,\phi^t(x_{N1}),\cdots,\phi^t(x_{Nn_N}))V\upsilon = \frac{\lambda}{M}KK\alpha \qquad (20)$$

According to (4), (5) and (12), the right term of (14) gives:

$$B\upsilon = \frac{1}{M}\sum_{p=1}^{N}\sum_{q=1}^{n_p}\alpha_{pq}\phi(x_{pq})\sum_{l=1}^{N}n_l\left[\frac{1}{n_l}\sum_{k=1}^{n_l}\phi(x_{lk})\right]\left[\frac{1}{n_l}\sum_{k=1}^{n_l}\phi(x_{lk})\right]^t$$

$$= \frac{1}{M}\sum_{p=1}^{N}\sum_{q=1}^{n_p}\alpha_{pq}\sum_{l=1}^{N}\left[\sum_{k=1}^{n_l}\phi(x_{lk})\right]\left[\frac{1}{n_l}\right]\left[\sum_{k=1}^{n_l}\phi^t(x_{lk})\phi(x_{pq})\right]$$

$$\phi^t(x_{ij})B\upsilon = \frac{1}{M}\sum_{p=1}^{N}\sum_{q=1}^{n_p}\alpha_{pq}\sum_{l=1}^{N}\left[\sum_{k=1}^{n_l}\phi^t(x_{ij})\phi(x_{lk})\right]\left[\frac{1}{n_l}\right]\left[\sum_{k=1}^{n_l}\phi^t(x_{lk})\phi(x_{pq})\right]$$

For all class i and for all its elements j we obtain:

$$(\phi^t(x_{11}),\ldots,\phi^t(x_{1n_1}),\ldots,\phi^t(x_{ij}),\ldots,\phi^t(x_{N1}),\cdots,\phi^t(x_{Nn_N}))B\upsilon = \frac{1}{M}KWK\alpha \qquad (21)$$

Combining (20) and (21) we obtain:

$\lambda KK\alpha = KWK\alpha$, which is multiplied by α^t to obtain (13).

Appendix C

In this appendix, we show how to center the element of *K* in the feature space *F*.
For a given x_i, the image $\phi(x_i)$ is centered according to:

$$\tilde{\phi}(x_i) = \phi(x_i) - \frac{1}{M}\sum_{k=1}^{M}\phi(x_k)$$

Thus we define the centered kernel function :

$$\tilde{k}(x_i, x_j) = \tilde{k}_{ij} = \tilde{\phi}^t(x_i)\tilde{\phi}(x_j)$$

If we introduce the class index, for a given observation x_{pi}, element i of the class p, the image $\phi(x_{pi})$ is centered according to:

$$\tilde{\phi}(x_{pi}) = \phi(x_{pi}) - \frac{1}{M}\sum_{l=1}^{N}\sum_{k=1}^{n_l}\phi(x_{lk}) \qquad (22)$$

We have then to define the covariance matrix *K* with centered points:

$(\tilde{k}_{ij})_{pq} = \tilde{\phi}^t(x_{pi})\tilde{\phi}(x_{qj})$ for a given class p and q.

$$(\tilde{k}_{ij})_{pq} = \left[\phi(x_{pi}) - \frac{1}{M}\sum_{l=1}^{N}\sum_{k=1}^{n_l}\phi(x_{lk})\right]^t\left[\phi(x_{qj}) - \frac{1}{M}\sum_{h=1}^{N}\sum_{m=1}^{n_m}\phi(x_{hm})\right]$$

$$(\tilde{k}_{ij})_{pq} = (k_{ij})_{pq} - \frac{1}{M}\sum_{l=1}^{N}\sum_{k=1}^{n_l}(1_{ik})_{pl}(k_{kj})_{lq} - \frac{1}{M}\sum_{h=1}^{N}\sum_{m=1}^{n_m}(k_{im})_{ph}(1_{mj})_{hq} + \frac{1}{M^2}\sum_{l=1}^{N}\sum_{k=1}^{n_l}\sum_{h=1}^{N}\sum_{m=1}^{n_m}(1_{ik})_{pl}(k_{km})_{lh}(1_{mj})_{hq}$$

$$\tilde{K}_{pq} = K_{pq} - \frac{1}{M}\sum_{l=1}^{N}1_{pl}K_{lq} - \sum_{h=1}^{N}K_{ph}1_{hq} + \frac{1}{M^2}\sum_{l=1}^{N}\sum_{h=1}^{N}1_{pl}K_{lh}1_{hq}$$

$$\tilde{K} = K - \frac{1}{M} 1_N K - \frac{1}{M} K 1_N + \frac{1}{M^2} 1_N K 1_N$$

Where we had introduced the following matrix:

$1_{pl} = (1_{ik})_{i=1,..,n_p;k=1,...,n_l}$, $(n_p x n_l)$ matrix which all elements are equal to 1.

$1_N = (1_{pl})_{p=1,..,N;l=1,..,N}$, *(MxM)* matrix.

We thus replace K by $\tilde{K}$, then solve the eigenvalue problem and normalize the corresponding vectors. Afterwards the test patterns z are projected onto the eigenvectors (19) expressed with $\tilde{K}$:

$$v^t \tilde{\phi}(z) = \sum_{p=1}^{N} \sum_{q=1}^{n_p} \alpha_{pq} \tilde{k}(x_{pq}, z)$$

Acknowledgements

The authors are grateful to Scott Barnes (Engineer at MEI, USA), Philippe Jard (Applied Research Manager at MEI, USA) and Ian Howgrave-Graham (R&D Manager at Landis & Gyr, Switzerland) for their comments about this manuscript. We also thank Rodrigo Fernandez (Research Associate at the university of Paris Nord) for accepting to compare results using his own SVM classifier software.

References

Anouar F., Badran F., Thiria S., "Probabilistic Self Organizing Map and Radial Basis Function", Journal Neurocomputing 20, 83-96, 1998.

Aizerman M. A., Braverman E. M., Rozonoér L. I., "Theoretical foundations of the potential function method in pattern recognition learning", Automation and Remote Control, 25:821-837, 1964.

Bishop C.M., "Neural Network for Pattern Recognition", Clarendon Press, Oxford, 1995.

Boser B. E., Guyon I. M., Vapnik V. N., "A training algorithm for optimal margin classifiers". In D. Haussler , editor, 5th Annual ACM Workshop on COLT, pages 144-152, Pittsburgh, PA, 1992. ACM Press.

Burges C. J.C., Schölkopf B., "Improving the Accuracy and Speed of Support Vector Machines", Neural Information Processing Systems, vol 9. MIT Press, Cambridge, MA, 1997.

Burges C. J.C., "Simplified support vector decision rules", In L. Saitta (Ed.) Proc. 13th Intl. Conf on Machine Learning. San Mteo, CA: Morgan Kaufmann. 1996.

Burges C. J.C., "A Tutorial on Support Vector machine for Pattern Recognition ", support vector web page, http://svm.first.gmd.de, 1998.

Fernandez R., Viennet E., "Face identification with support vector machines", Proceedings ESANN, 1999.

Fernandez R., "Machines a vecteurs de support pour la reconnaissance des formes: proprietes et applications", Thesis of University of Paris Nord, 1999.

Fisher R.A., "The use of multiple measurements in taxonomic problems", Annual Eugenics, 7, Part II, 179-188, 1936.

Fukunaga K., "Introduction to Statistical Pattern Recognition", Academic Press, INC, 2nd ed, 1990.

Gabrijel I., Dobnikar A., "Adaptive RBF Neural Network", Proceeding of SOCO'97 conference, Nîmes, pp. 164-170, France, 1997.

Gunn S. R., "Support vectors machines for classification and regression", Technical report, Image Speech and Intelligent Systems Research Group, University of Southampton, http://www.isis.ecs.soton.ac.uk/resource/svminfo/, 1997.

Harville D. A., "Matrix algebra from a statistician's perspective", Springer Verlag, New York, Inc.

James R. Bunch, Linda Kaufman, "Some stable methods for calculating inertia and solving symmetric linear systems", Mathematics of computation, 31(137):163-179, 1977.

Kohonen T., "Self-Organizing Maps", Springer. 1994.

Hastie T., Tibshirani R., Buja A., "Flexible discriminant analysis", JASA, 89:1255-1270, 1994.

Musavi M. T., Kalantri K., Ahmed W., Chan K. H., "A minimum error neural network (MNN)", Neural Networks, vol 6, pp.397-407, 1993.

Poggio T., "On optimal nonlinear associative recall", Biological Cybernetics, 19:201-209, 1975.

Saporta G., "Probabilites, analyse des donnees et statistique", Editions Technip, 1990.

Schölkopf B., Smola A., Müller K. R., "Nonlinear component analysis as a kernel eigenvalue problem", Technical report 44, MPI fur biologische kybernetik, 1996.

Schölkopf B., Smola A., Müller K. R., "Nonlinear Component Analysis as A Kernel Eigenvalue Problem", Neural Computation 10, 1299-1319, 1998.

Schölkopf B., "Support Vector Learning", R. Oldenbourg Verlag, Munich, 1997.

Specht D.F. "Probabilistic Neural Networks", Neural Networks, 3(1), 109-118, 1990.

Vapnik V., "The Nature of Statistical Learning Theory", Springer Verlag N.Y., 189p, 1995.

Vapnik V., Golowich S. E., Smola A., "Support Vector Method for Function Approximation, Regression Estimation, and Signal Processing" Neural Information Processing Systems, vol 9. MIT Press, Cambridge, MA, 1997.

Wilkinson J.H., Reinsch C., "Linear Algebra", vol.II of Handbook for Automatic Computation, New York: Springer-Verlag, 1971.

Recent References since the paper was submitted :

Jaakkola T.S., Haussler D., "Exploiting Generative Models in Discriminative Classifiers", To appear in M.S. Kearns, S.A. Solla and D.A. Cohn, editors, Advances in Neural Information Processing Systems 11, MIT Press, Cambridge, MA., 1999.

Mika S., Rätsch G., Weston J., Schölkopf B., Müller K. R., "Fisher Discriminant Analysis with Kernels", Proc. IEEE Neural Networks for Signal Processing Workshop, NNSP, 1999.