Margin Maximizing Discriminant Analysis

András Kocsor[1], Kornél Kovács[1], and Csaba Szepesvári[2]

[1] Research Group on Artificial Intelligence of the Hungarian Academy of Sciences, University of Szeged, Aradi vértanúk tere 1., 6720 Szeged, Hungary
{kocsor, kkornel}@inf.u-szeged.hu
[2] Computer and Automation Research Institute of the Hungarian Academy of Sciences, Kende u. 13-17, 1111 Budapest, Hungary
szcsaba@sztaki.hu

Abstract. We propose a new feature extraction method called Margin Maximizing Discriminant Analysis (MMDA) which seeks to extract features suitable for classification tasks. MMDA is based on the principle that an ideal feature should convey the maximum information about the class labels and it should depend only on the geometry of the optimal decision boundary and not on those parts of the distribution of the input data that do not participate in shaping this boundary. Further, distinct feature components should convey unrelated information about the data. Two feature extraction methods are proposed for calculating the parameters of such a projection that are shown to yield equivalent results. The kernel mapping idea is used to derive non-linear versions. Experiments with several real-world, publicly available data sets demonstrate that the new method yields competitive results.

1 Introduction

In this paper we consider feature extraction in a classification context. Feature extraction can be used for data visualization, e.g. plotting data in the coordinate system defined by the principal components of the data covariance matrix. Visualization may help us to find outliers, or meaningful clusters. Another good use of feature extraction is noise reduction. In classification the goal is to suppress irrelevant information in order to make the classification task using the transformed data easier and simpler.

Feature extraction is the process of transforming the input patterns either by means of a linear or a non-linear transformation. Linear transformations are more amenable to mathematical analysis, while non-linear transformations are more powerful. When linear methods are applied to non-linearly transformed data, the full method becomes non-linear. One important case is when the linear method uses only dot-products of the data. In this case the kernel mapping idea [1, 15, 19] can be used to obtain an efficient implementation whose run time does not depend on the dimensionality of the non-linear map's image space. This 'kernel mapping' idea applies to many well-known feature extraction methods like principal component analysis and linear discriminant analysis. In classification, the best known example utilizing this idea is the support vector machine (SVM) [19].

Principal component analysis (PCA) [9] is one of the most widely-known linear feature extraction methods used. It is an unsupervised method that seeks to represent the input patterns in a lower dimensional subspace such that the expected squared reconstruction error is minimized. By its very nature PCA is not meant for classification tasks. So, in the worst case, the Bayes error rate may become arbitrarily bad after the data is projected onto the first few principal components even if the untransformed data was perfectly classifiable. We shall call this phenomenon a *filtering disaster*. PCA can still be very useful e.g. for suppressing "small noise" which corrupts the input patterns regardless of the class labels. PCA has been generalized to KPCA [18] by using the kernel mapping idea.

Classical linear discriminant analysis (LDA) [7] searches for directions that allow optimal discrimination between the classes provided that the input patterns are normally distributed for all classes $j = 1, \ldots, m$ and share the same covariance matrix. If these assumptions are violated LDA becomes suboptimal and a filtering disaster may occur. Recently, LDA has been generalized using the kernel mapping technique [2, 14, 17] as well.

Discriminant analysis as a broader subject addresses the problem of finding a transformation of the input patterns such that classification using the transformed data set becomes easier (e.g. by suppressing irrelevant components, or noise). More recent methods in discriminant analysis include the "Springy Discriminant Analysis" (SDA) (and its non-linear kernelized counterpart, KSDA), which was derived using a mechanical analogy [10, 11] or, in a special case, as a method for maximizing the between-class average margin itself averaged for all pairs of distinct classes [12]. The goal of the algorithm proposed in [6] is to find a linear transformation of the input patterns such that the statistical relationship between the input and output variables is preserved. The authors of this article use reproducing kernel Hilbert spaces (RKHS) to derive an appropriate contrast function. One distinctive feature of their approach is that the method is completely distribution free. There are many other methods available but we will not discuss them here due to lack of space.

In this paper we propose a new linear feature extraction method that we will call Margin Maximizing Discriminant Analysis (MMDA). MMDA projects input patterns onto the subspace spanned by the normals of a set of pairwise orthogonal margin maximizing hyperplanes. The method can be regarded as a non-parametric extension of LDA which makes no normality assumptions on the data but, instead, uses the principle that the separating hyperplane employed should depend on the decision boundary only. A deflation technique is proposed to complement this principle to extract a sequence of orthogonal projection directions. A corresponding non-linear feature extraction method is derived using the kernel mapping technique. The performance of the proposed methods is examined on several real-world datasets. Our findings show that the new method performs quite well and, depending on the dataset may sometimes perform better than any of the other methods tested, resulting in an increase in classification accuracy.

2 Principles of MMDA

MMDA makes use of the principal idea underlying LDA: projecting the input data onto the normal of a given hyperplane which separates the two classes best and provides all the information a decision maker needs to classify the input patterns. However, at this point LDA places normality assumptions on the data, whereas we make no such assumptions, but propose to employ margin maximizing hyperplanes instead.

This choice was motivated by the following desirable properties of such hyperplanes [5, 3, 8]: (i) without any additional information they are likely to provide good generalization on future data; (ii) these hyperplanes are insensitive to small perturbations of correctly classified patterns lying further away from the separating hyperplane; moreover, (iii) they are insensitive to small variations in their parameters. In addition to these properties, margin maximizing hyperplanes are insensitive to the actual *probability distribution* of patterns lying further away from the decision boundary. Hence when a large mass of the data lies far away from the ideal decision boundary we can expect the new method to win against those methods that minimize some form of average loss/cost since those methods necessarily take into account the full distribution of the input patterns. An example of such a situation is depicted in Figure 1. Note that such situations are expected to be quite common in practical applications like character recognition and text categorization. Actually, the original motivation of MMDA stems from a character recognition problem. Suppose that there are two character classes. Suppose also that the input space is the space of character images. Then, let us concentrate only on two pixels. Specifically, let us assume that pixel 1 is such that for characters in class 1, it can be 'on' or 'off', but in the majority of cases it is 'on'. Further, let us assume that pixel 1 is never 'on' for characters in class 2. Suppose too that pixel 2 is such that it is always 'off' for characters in class 1 and it is always 'on' for characters in class 2. Admittedly, these are strong simplifying assumptions, but similar cases do occur in real-world character recognition tasks. In this simplified case, the ideal feature extractor should actually work like a feature selection method: since the two classes are well separated by using pixel 2, it should project the 2D space of the two pixels onto the second coordinate. Now notice that this is the situation depicted in Figure 1. LDA and PCA fail to find such a projection, but MMDA succeeds in doing so.

We supplement the idea of projecting onto the space spanned by the normal of a margin maximizing hyperplane by a deflation technique which guarantees that all subsequent hyperplanes (and all subsequent normals) are orthogonal to each other. As a result each successive feature extraction step extracts "new" information unrelated to information extracted in the previous steps.

Deflation can be incorporated as a step to transform the data covariance matrix. However, we can also incorporate a suitable orthogonality criterion in the equations defining the margin maximizing hyperplane. We will show the equivalence of these two approaches in the next section and discuss their relative merits.

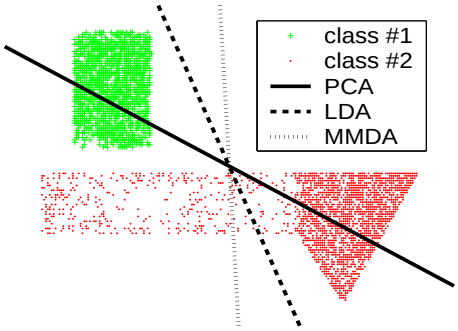

Fig. 1. An illustration of the behavior of PCA, LDA and MMDA for a binary classification problem. The figure shows the one-dimensional subspace represented by a hyperplane that PCA, LDA and MMDA project the data onto. Although the data is linearly separable, PCA and LDA fail to find a subspace such that the data when projected onto the subspace remains linearly separable. MMDA avoids this problem by projecting onto the normal of a separating hyperplane when such a hyperplane exists.

3 Linear Feature Extraction

3.1 The Deflation Approach

Let X, y be the training data, where $X = (x_1, \ldots, x_n)$ are the input patterns ($x_j \in \mathbb{R}^d$) and $y \in \{-1, +1\}^n$ are the corresponding target labels. We will assume that (x_i, y_i), $i = 1, \ldots, n$ are independent, identically distributed random variables.

Assuming that the data (X, y) is separable, the maximum margin separating hyperplane can be found as a solution of a quadratic programming problem [5]. When the data is not separable the maximum margin separation problem is modified to simultaneously maximize the margin and minimize the error [19]. This still results in a quadratic programming problem. In order to introduce the corresponding equations formally, let us fix a positive real number C that we will use to weight the misclassification cost. Then the maximum margin separation (MMS) problem is defined as follows: Given (X, y, C) find $w \in \mathbb{R}^d$, $b \in \mathbb{R}$ and $\xi = (\xi_1, \ldots, \xi_n)^T \in \mathbb{R}^n$ such that[3]

$$\begin{aligned} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n}\xi_i \rightarrow \min \text{ s.t.} \\ y_i(w^T x_i + b) \geq 1 - \xi_i, \\ \xi_i \geq 0, \quad i = 1, \ldots, n. \end{aligned} \tag{1}$$

MMDA now proceeds as follows: Given (X, y, C), find the solution of the MMS problem (X, y, C). Let this solution be (w_1, b_1). The first extracted feature component is $f_1(x) = w_1^T x$. Now transform the data by projecting it onto a space

[3] Here $\|w\|_2$ denotes the ℓ^2 norm of w.

orthogonal to w_1. For simplicity, assume that w_1 is normalized so $\|w_1\|_2 = 1$. Then the projected data is given by

$$x'_i = x_i - (w_1^T x_i) w_1. \tag{2}$$

Let X' denote the matrix $(x'_1, \ldots, x'_n)$ and let (w_2, b_2) be the solution of the MMS problem (X', y, C). Then the second extracted feature component is $f_2(x) = w_2^T x'$, where $x' = x - (w_1^T x) w_1$. This procedure can be repeated as many times as desired. The following proposition shows that w_1 and w_2 are orthogonal.

Proposition 1 *Let (w_1, b_1) be the solution of the MMS problem (X, y, C) and (w_2, b_2) be the solution of the MMS problem (X', y, C), where $X' = (x'_1, \ldots, x'_n)$ with x'_i defined by (2). Then the vectors w_1 and w_2 are orthogonal*[4] *.*

A corollary of this proposition is that $f_2(x) = w_2^T(x - (w_1^T x) w_1) = w_2^T x$. Similarly, if $w_3, \ldots, w_r$ $(r \leq d)$ are the normals extracted up to step r then $w_1, \ldots, w_r$ are pairwise orthogonal and the ith feature value $f_i(x)$ can be computed via:

$$f_i(x) = w_i^T x. \tag{3}$$

In order to derive our first practical algorithm let us note that the solution of the MMS problem is typically obtained via the Langrangian dual of (1):

$$\begin{gathered} -\frac{1}{2}\alpha^T R \alpha + \alpha^T 1 \to \max \\ \text{such that} \quad y^T \alpha = 0, \quad 0 \leq \alpha \leq C1, \end{gathered} \tag{4}$$

where the matrix R is defined by $R = YX^T XY$ and $Y = \operatorname{diag}(y_1, \ldots, y_n)$ and $\alpha \in \mathbb{R}^n$ [19]. Here $C1 = (C, \ldots, C)^\top \in \mathbb{R}^d$ and the comparison of vectors is made one component at a time. Given α, the solution of (4), the solution of the MMS problem (X, y, C) is recovered through $w = X\alpha$ and $b = 1^T \alpha$. We shall call (4) the dual MMS problem parameterized by (R, y, C).

Let X' be defined as before. Notice that (4) depends on the data vector X only through the matrix R. Hence, the Langrangian dual defined for the transformed data X' takes the form in (4), but R needs to be recalculated. The next proposition shows how to do this in the general case when the data is projected onto a subspace spanned by an orthonormal system:

Proposition 2 *Let X' be the data X projected onto a space orthogonal to the orthonormal system $W = (w_1, \ldots, w_r)$. Then*

$$R' = Y(X')^T X' Y = Y\left(X^T X - V^T V\right) Y, \tag{5}$$

where we define V by $V = W^T X$. In particular, if $W = XA$ for some matrix A then R' can be calculated by $R' = Y\left(K - (KA)(KA)^T\right) Y$, where $K = X^T X$.

The significance of this result is that it shows it is possible to use existing SVM code to extract a sequence of orthogonal margin maximizing hyperplanes just by transforming the matrix R. This proposition is given extra weights as it shows that it is possible to apply the kernel mapping idea to MMDA. This will be considered in more detail in Section 4.

[4] We omit the proofs where needed throughout the paper due to a lack of space.

3.2 The Direct Method

The deflation approach requires $O(n^2)$ calculations when calculating the transformed matrix R'. The method we consider in this section avoids this at the price of slightly increasing the dimensionality of the quadratic programming problem.

Let us define the maximum margin separation problem with orthogonality constraint (MMSO problem) as follows: Let u be a d-dimensional vector: $u \in \mathbb{R}^d$. The MMSO problem parameterized by (X, y, C, u) is to find $w \in \mathbb{R}^d$, $b \in \mathbb{R}$ and $\xi = (\xi_1, \ldots, \xi_n) \in \mathbb{R}^n$ such that

$$\begin{aligned} \frac{1}{2}\|w\|_2^2 + C\sum_{i=1}^{n}\xi_i \rightarrow \min \text{ s.t.} \\ y_i(w^T x_i + b) \geq 1 - \xi_i, \\ \xi_i \geq 0, \quad i = 1, \ldots, n, \\ u^T w = 0. \end{aligned} \tag{6}$$

Let $H = H_{(u,0)}$ be a hyperplane with normal $u \in \mathbb{R}^d$ (we assume $\|u\|_2 = 1$ as before) and bias 0. Let $X' = (x'_1, \ldots, x'_n)$ be the matrix whose columns are composed of the x_i vectors projected onto H: $x'_i = x_i - (u^T x_i)u$ as before. The following proposition shows the equivalence of the solutions of MMSO problem and the solutions obtained using the deflation approach:

Proposition 3 *Let C have a fixed positive value. Given the data (X, y) and the hyperplane H with normal u satisfying $\|u\|_2 = 1$ and bias 0, the following holds: Let X' denote the data projected onto the hyperplane H. Then the solutions of the MMS problem (X', y, C) and the MMSO problem (X, y, C, u) coincide.*

According to this last proposition, we obtain equivalent solutions to those gotten using the deflation approach when orthogonality constraints are added to the MMS problem. It is readily seen that the proposition remains true when the number of orthogonality constraints – r, say – is bigger than one. The corresponding MMSO problem will be denoted by (X, y, C, U), where $U = (u_1, \ldots, u_r)$ is the matrix of vectors that are used to define the orthogonality constraints.

It is not difficult to prove that the solution of an MMSO problem (X, y, C, U) may be obtained by solving the following dual quadratic programming problem:

$$\begin{aligned} -\frac{1}{2}\left(\alpha^\top Y K Y \alpha + \gamma^\top U^T U \gamma\right) + \alpha^\top 1 + \gamma^\top U^T X Y \alpha \rightarrow \max \\ \text{such that} \quad y^T \alpha = 0, \quad 0 \leq \alpha \leq C1. \end{aligned} \tag{7}$$

Since the number of columns of U is r, the dimensionality of γ will also be r, and hence the number of variables in the above quadratic programming problem will be $n + r$.

The direct method works as follows: Given the data (X, y, C), let (w_1, b_1) be the solution of the MMS problem (X, y, C). Assuming that the solution vectors $(w_1, b_1), \ldots, (w_{r-1}, b_{r-1})$ have already been computed, (w_r, b_r) is obtained as the solution of the MMSO problem (X, y, C, W_{r-1}), where $W_{r-1} = (w_1, \ldots, w_{r-1})$.

Now we will show (i) that the dual MMSO optimization problem (X, y, C, W_r) can be put into a form where the dependence on X is only through the dot product matrix $K = X^T X$ and (ii) that the matrices involved in the dual MMSO optimization problem can be computed in an incremental manner in time $O(mn)$, where m is the number of non-zero elements of $\alpha^{(r)}$. We know that the vectors in W_r lie in the span of X: $w_i = X\alpha^{(i)}$. Therefore $W_r = XA_r$ where $A_r = (\alpha^{(1)}, \ldots, \alpha^{(r)})$. Hence, $W_r^T W_r = A_r^T K A_r$ and $W_r^T XY = A_r^T KY$. So $W_r^T W_r = [A_{r-1},\, \alpha^{(r)}]^T K [A_{r-1},\, \alpha^{(r)}]$, where the subblocks can be computed by $A_{r-1}^T K A_{r-1}$, $A_{r-1}^T \alpha^{(r)}$, $(\alpha^{(r)})^T A_{r-1}$ and $(\alpha^{(r)})^T K \alpha^{(r)}$, respectively. Further, $W_r^T XY = A_r^T KY$ and hence $(W_r^T XY)^T = (YKA_{r-1},\, YK\alpha^{(r)})$. Thus the direct method may be computationally cheaper than the deflation approach when the value of m (the number of support vectors) obtained in step r is much smaller than the number of data points.

4 Non-linear Feature Extraction

It is often the case that the problem of extracting relevant features can be made substantially easier when the data is mapped into an appropriate high dimensional space by some non-linear mapping ϕ and linear methods are applied to the transformed data. If the algorithm is expressible in terms of dot products and if the non-linear mapping $\phi : \mathbb{R}^d \to \mathcal{H}$ is such that the dot products of the images of any two points x and y under ϕ can be computed as a function of x and y only and in poly(d)-time without explicitly calculating $\phi(x)$ or $\phi(y)$ then the algorithm remains tractable, regardless of the dimensionality of $\mathcal{H}$. This allows us to consider very high or even infinite dimensional image spaces $\mathcal{H}$. We may as well start by choosing a symmetric positive definite function $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$, called the kernel function (see e.g. [5]). Then the closure of the linear span of the set $\{\, k(x, \cdot) \,|\, x \in \mathbb{R}^d \,\}$ gives rise to a Hilbert space $\mathcal{H}$ where the inner product is defined such that it satisfies $\langle k(x_1, \cdot), k(x_2, \cdot) \rangle = k(x_1, x_2)$ for all points $x_1, x_2 \in \mathbb{R}^d$ [13]. The choice of k automatically gives rise to the mapping $\phi : \mathbb{R}^d \to \mathcal{H}$ defined by $\phi(x) = k(x, \cdot)$. This is called the kernel mapping idea [1, 15, 19].

It is clear that the kernel mapping idea can be used to obtain an efficient non-linear version of MMDA too: Firstly, the MMS problem at the heart of MMDA is actually the problem solved by SVMs, which itself builds on the kernel mapping idea [5]. It is well known that the MMS problem can be efficiently solved in the $\mathcal{H}$ feature space. However, for the sake of completeness, we shall briefly describe how to 'kernelize' the MMS problem. The input patterns X appear in Equation (4) only through the dot product matrix $X^T X$. Hence, defining the matrix K by

$$K_{ij} = k(x_i, x_j), \quad 1 \le i, j \le n \tag{8}$$

and replacing $X^T X$ in Equation (4) by K, we obtain a quadratic programming problem such that (if α denotes its solution) $w(\cdot) = \sum_{i=1}^n \alpha_i k(x_i, \cdot)$ and $b = \alpha^T 1$ is the solution of the MMS problem (Φ, y, C), where $\Phi = (\phi(x_1), \ldots, \phi(x_n))$. Now assuming that r directions $W = (w_1, \ldots, w_r)$ have already been determined, the $(r+1)$th direction can be computed as the solution of the dual MMS problem

Table 1. The characteristics of datasets used in the experiments. In the cases marked by * 10-fold class-balanced cross-validation was used to measure performances.

Dataset	# classes	# attribs	# train	# test
Bupa	2	7	699	*
Pima	2	8	768	*
Iono	2	34	351	*
Heart	2	13	303	*
DNA	3	181	2000	1186
Satimage	6	36	4435	2000
Optdigits	10	64	3823	1797

with R replaced by R', where R' is defined in Proposition 2. Since $W = \Phi A$ for an appropriate matrix A (the jth column of A is the solution of the jth dual subproblem) and $\Phi = (\phi(x_1), \ldots, \phi(x_n))$, R' can be computed by Proposition 2, where K is now defined by (8). It was also found that the dual of the MMSO problem can be expressed in terms of $X^T X$ when W lies in the span of X. Hence the dual of the MMSO problem can also be expressed using K only, and thus it can be solved efficiently, regardless of the dimensionality of $\mathcal{H}$.

Finally, rewriting (3) in terms of the kernel function we find that the ith component of the feature extraction mapping can be evaluated using

$$f_i(x) = \sum_{j=1}^{n} \alpha_j^{(i)} k(x_i, x), \tag{9}$$

where $\alpha^{(i)}$ is the solution of the ith dual subproblem. Eq. (9) follows directly from $w_i(\cdot) = \sum_{j=1}^{n} \alpha_j^{(i)} k(x_i, \cdot)$ and the fact that $f_i(x) = \langle w_i, \phi(x) \rangle = \sum_{j=1}^{n} \alpha_j^{(i)} \langle \phi(x_i), \phi(x) \rangle$.

5 Experimental Results

In the first experiment we sought to demonstrate the visualization capability of MMDA. We used the *Wine* dataset from the UCI machine learning repository [4] which has 13 continuous attributes, 3 classes and 178 instances. We applied PCA, LDA and MMDA to these data sets. Two dimensional projections of the data are shown in Figure 2. In the case of PCA and LDA, the data is projected onto the eigenvectors corresponding to the two largest eigenvalues. Since MMDA is defined for binary classification problems, with multi-class problems we need to group certain classes together. In this example the first direction is obtained by grouping classes 2 and 3 together into a single class, while the second direction is obtained by grouping classes 1 and 3 together. It can be seen that for both LDA and MMDA the data became separable in the projection space. It was also noticed that the margin of separation is larger for the case of MMDA, as expected. Note that the size of the margin can be controlled to some extent by the parameter C. For this figure we used $C = 1$ and the data was centered and

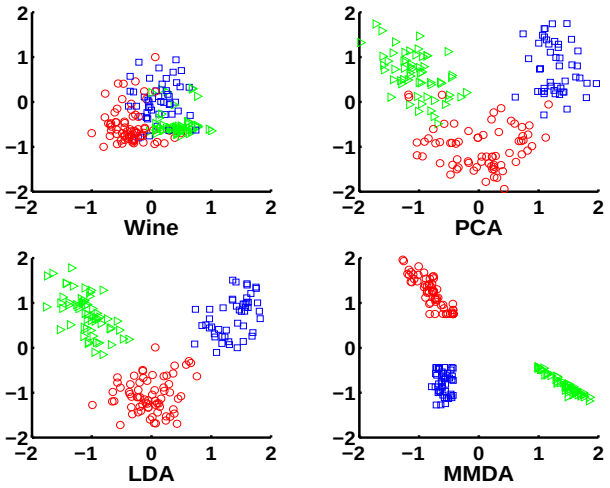

Fig. 2. Scatter plot of wine data projected onto a two-dimensional subspace. The upper left subfigure shows the projection onto the first two attributes, while the other three show the results of a PCA, LDA and MMDA transformation, respectively.

scaled to have unit variance (this transformation was applied in all of our other experiments as well). Actually, the data is not linearly separable in the case of the PCA projection.

Next we investigate whether MMDA can estimate useful subspaces that preserve information necessary for the classification task. For this we ran MMDA on a number of binary classification problems. We changed the number of dimensions of the estimated subspace and measured the classification accuracy that could be achieved by projecting data on the extracted subspace. This experiment was run with both the linear and kernelized versions of MMDA. Since there is obviously no optimal classifier we decided to estimate the quality of the extracted subspace by training an artificial neural network (ANN) classifier on the projected data. The ANN was trained for a fixed number of iterations using batch gradient descent with a constant learning rate. There is one hidden layer and the number of hidden nodes is three times the number of inputs. Our experiments showed that, on the datasets used, this method is competitive with the results of SVMs. We chose to combine ANNs with linear feature extraction since (i) we wanted to keep the algorithms simple and since (ii) the test speed of the resulting composite classifier was then usually very high. High classification speed is important for some applications like OCR. SVMs need special postprocessing to achieve comparably high speeds, therefore we decided to use ANNs.

The characteristics of the datasets used in this experiment are shown in Table 4, while the results are presented in Figure 3. The results labeled 'original' were obtained using an ANN trained on the original, untransformed data. It may be seen that for a number of datasets very good classification rates are achieved

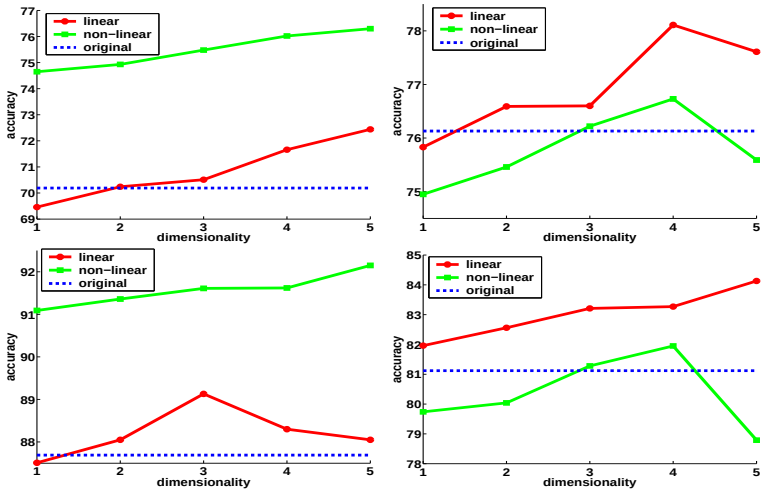

Fig. 3. Accuracies achieved by training a neural network on the subspace extracted by (K-)MMDA shown for 4 dataset. The figures in the top-to-down and left-to-right order are results obtained for the datasets called Bupa, Pima, Ionosphere and Heart Disease.

with only a few features. Also, in certain cases performance drops when the dimensionality of the subspace is increased.

Next, we tested the performance of the method on a number of larger multi-class problems. For multi-class problems we used the "one vs. all" approach: basically when the number of classes wa m we ran (K-)MMDA m times with one class against all the others. We chose this approach for its simplicity. This is probably a suboptimal approach, though our initial experiments with output-coding suggests that accuracies obtained this way are quite good.[5] In this case we tested the interaction of MMDA with several classifiers. These were the ANN introduced earlier, support vector machines with the linear kernel and C4.5 [16]. The results for the three datasets are shown in Figure 4. For comparision we also included the results obtained with 'no feature extraction', PCA and LDA. For the datasets DNA and Optdigits we got competitive results, but for Satimage the result for the tested cases were worse than those obtained with the other methods tested. In particular, in the case of Satimage all feature extractors yielded worse results than those using *no* feature extractor. We conjecture that the optimal subspace for Satimage might be just the (untransformed) space of input patterns.[6]

[5] Note that we lose pairwise orthogonality (directions extracted for different subproblems are not necessarily orthogonal). In the future we plan to investigate the case when pairwise orthogonality is enforced. Note here that as a kernel for K-MMDA fourth order cosine kernels were used.

[6] In these experiments K-MMDA was implemented using a Gauss-Seidel iteration (or the Adatron) and (without loss of generality) we set $b = 0$.

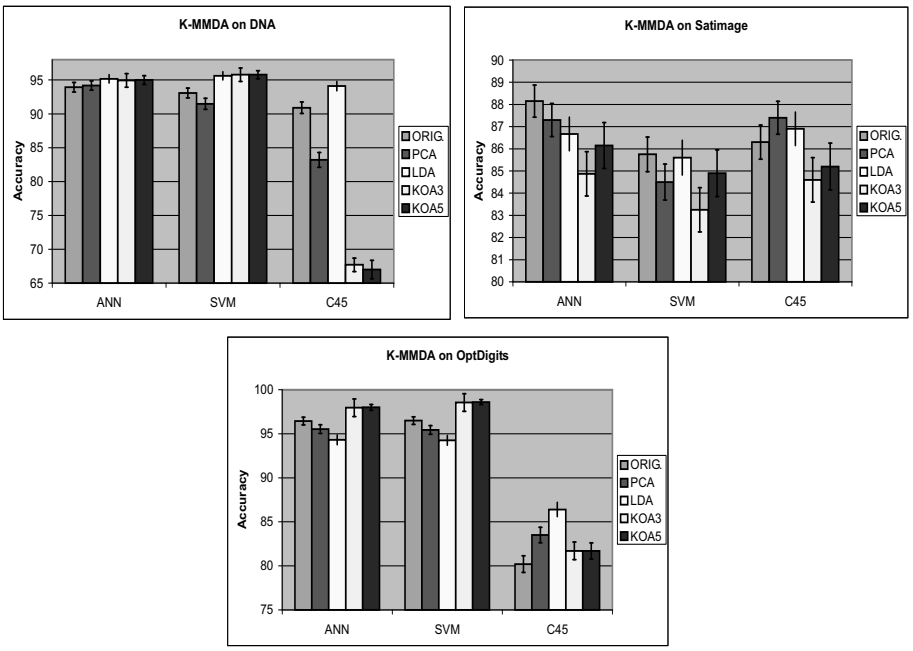

Fig. 4. The interaction of classifiers and feature extractors. The results for K-MMDA are shown. The label KOA$< i >$ means that K-MMDA was run in a one-vs-all manner, and for each subproblem where i is the number of directions extracted per subproblem.

6 Discussion and Conclusions

One common feature of PCA, LDA and SDA is that they require finding a number of principal eigenvectors of a matrix whose dimension scales with the dimensionality of the input space in the linear case, and scales with the number of input patterns in the non-linear case. Our method requires the solution of constrained quadratic optimization problems. As a result, in the case of non-linear feature extraction our method extracts sparse solutions, whilst the kernelized versions of PCA, LDA and SDA extract dense solutions (when no additional 'tricks' are used). The maximum number of features derived using LDA is actually the minimum of the dimensionality of the space and the number of classes minus one. For high dimensional spaces with a few classes this limits the use of LDA [12]. Unlike the standard LDA with (linear) MMDA we can extract as many features as the dimensionality of the pattern space (feature space) allows.

Non-linear MMDA should benefit more from the margin maximizing idea than the linear version as the non-linear version typically works in very high dimensional (sometimes infinite dimensional) feature spaces. This was partially confirmed by our experiments where, for certain datasets, K-MMDA was shown to give excellent results.

In summary, our experiments so far have shown that MMDA can indeed compete with other alternative feature extraction methods. One nice aspect of MMDA is that it can be implemented on top of existing SVM software. Therefore we believe that the proposed method will be a useful tool for researchers using machine learning. In the future we plan to investigate the properties of MMDA more thoroughly (e.g. in multi-class problems). Extensions that make use of different norms penalizing w may also be of interest.

References

1. M. Aizerman, E. Braverman, and L. Rozonoer. Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control*, 25:821–837, 1964.
2. G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12(10):2385–2404, 2000.
3. K. P. Bennett and C. Campbell. Support vector machines: Hype or hallelujah? *SIGKDD Explorations*, 2(2):1–13, 2000.
4. C.L. Blake and C.J. Merz. UCI repository of machine learning databases, 1998.
5. B.E. Boser, I. Guyon, and V. Vapnik. A training algorithm for optimal margin classifiers. In *Computational Learing Theory*, pp. 144–152, 1992.
6. K. Fukumizu, F.R. Bach, and M.I. Jordan. Dimensionality reduction for supervised learning with reproducing kernel Hilbert spaces. *Journal of Machine Learning Research*, 5:73–99, 2004.
7. K. Fukunaga *Statistical Pattern Recognition.* Acad. Press, NY, 1989.
8. R. Herbrich and T. Graepel. A PAC-bayesian margin bound for linear classifiers: Why SVMs work. In *Advances in Neural Information Processing Systems 13*, pp. 224–230, Cambridge, MA, 2000. MIT Press.
9. I.J. Jolliffe. *Principal Component Analysis.* Springer-Verlag, NY, 1986.
10. A. Kocsor, K. Kovács. Kernel Springy Discriminant Analysis and Its Application to a Phonological Awareness Teaching System, in: P. Sojka, I. Kopecek, K. Pala (Eds.): *Proc. of TSD 2002*, LNAI 2448, pp. 325-328, Springer Verlag, 2002.
11. A. Kocsor, L. Tóth. Kernel-Based Feature Extraction with a Speech Technology Application. *IEEE Trans. Signal Processing.* 52(8), 2004.
12. H. Li, T. Jiang, and K. Zhang. Efficient and robust feature extraction by maximum margin criterion. In *Advances in Neural Information Processing Systems 16*, pp. 97 - 104, Vancouver, Canada, 2003.
13. J. Mercer. Functions of positive and negative type and their connection with the theory of integral equations. *Phil. Trans. Roy. Soc. London, A*, 209:415–446, 1909.
14. S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Müller. Fisher discriminant analysis with kernels. In Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, editors, *Neural Networks for Signal Processing IX*, pp. 41–48. IEEE, 1999.
15. T. Poggio. On optimal nonlinear associative recall. *Biol. Cyber.*, 19:201–209, 1975.
16. J.R. Quinlan. C4.5: Programs for Machine Learning, Morgan Kaufmann, San Mateo, CA, 1993.
17. V. Roth and V. Steinhage. Nonlinear discriminant analysis using kernel functions. In *Adv. in Neural Information Processing Systems NIPS 12*, pp. 568–574, 1999.
18. B. Schölkopf, A. Smola, and K. Müller. Kernel principal component analysis. In *Adv. in Kernel Methods - SV Learning*, pp. 327–352. MIT Press, Cambr., MA, 1999.
19. V. Vapnik. *The Nature of Statistical Learning Theory.* Spr.-Verl., NY, USA, 1995.