Acta Cybernetica **16** (2003) 271-278.

# Various Hyperplane Classifiers Using Kernel Feature Spaces*

Kornél Kovács† and András Kocsor‡

**Abstract**

In this paper we introduce a new family of hyperplane classifiers. But, in contrast to Support Vector Machines (SVM) - where a constrained quadratic optimization is used - some of the proposed methods lead to the unconstrained minimization of convex functions while others merely require solving a linear system of equations. So that the efficiency of these methods could be checked, classification tests were conducted on standard databases. In our evaluation, classification results of SVM were of course used as a general point of reference, which we found were outperformed in many cases.

## 1 Introduction

Numerous scientific areas such as bioinformatics, pharmacology and artificial intelligence depend on classification and regression methods which may be linear or non-linear, but it now seems that by using the so-called kernel idea, linear methods can be readily generalized to nonlinear ones. The key idea was originally presented in Aizermann's paper [1] and it was successfully applied in the context of the ubiquitous Support Vector Machines [10]. The roots of SV methods can be traced back to the need for the determination of the optimal parameters of a separating hyperplane, which can be formulated both in input space or in kernel induced feature spaces. However, optimality can vary from method to method and SVM is just one of several possible approaches.

Without loss of generality we shall assume that, as a realization of multivariate random variables, there are $m$-dimensional real attribute vectors in a compact set $\mathcal{X}$ over $\mathbb{R}^m$ describing objects in a certain domain, and that we have a finite $n \times m$ sample matrix $X = [\mathbf{x}_1, \ldots \mathbf{x}_n]^T$ containing $n$ random observations. Let us assume as well that we have an indicator function $\mathcal{L} : \mathbb{R}^m \to L \subseteq \mathbb{R}$, where $\mathcal{L}(\mathbf{x}_i)$ gives the label of the sample $\mathbf{x}_i$, and let us denote the vector $[\mathcal{L}(\mathbf{x}_1), \ldots, \mathcal{L}(\mathbf{x}_n)]^T$ by $\mathcal{L}(X)$.

*This work was supported under the contract IKTA No. 2001/055 from the Hungarian Ministry of Education.

†Department of Informatics, University of Szeged H-6720 Szeged, Arpád tér 2., Hungary, e-mail: kkornel@inf.u-szeged.hu

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

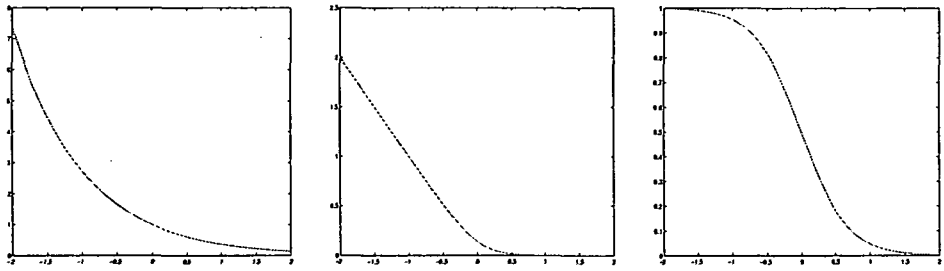


Figure 1: Three possible loss functions

Here, a finite set $L$ means a classification task. Should L be an infinite set, the task will be a regression problem.

In this paper we will restrict our investigations only to that of binary classification ($L = \{-1, +1\}$), as multiclass problems can be dealt with by applying binary classifiers [3]. But regression problems will not be entirely excluded here, since binary classifiers will be derived from regression formulae.

## 2 Linear classifiers with various loss-functions

Linear classification attempts to separate the sample points with different labels via a hyperplane. A hyperplane is a set of point $\mathbf{z}$:

$$\begin{pmatrix}\mathbf{z}^T & 1\end{pmatrix}\mathbf{a} = 0 \qquad \mathbf{z} \in \mathbb{R}^m, \quad \mathbf{a} \in \mathbb{R}^{m+1}. \tag{1}$$

For a point $\mathbf{z}$ the left-hand side of Eq. (1) is a signed expression with absolute value proportional to the distance from the hyperplane. In addition, the sign of this expression corresponds to the sign of the half-space the point lies in.

A point $\mathbf{x}_i$ is well-separated by a hyperplane with parameter $\mathbf{a}$ if and only if:

$$\mathcal{L}(\mathbf{x}_i) \cdot \begin{pmatrix}\mathbf{x}_i^T & 1\end{pmatrix}\mathbf{a} > 0 \qquad i \in \{1, \ldots, n\}.$$

Based on these products a target function - whose lower value indicates a better separation - can be defined:

$$\tau(\mathbf{a}) = \sum_{i=1}^{n} g\left(\mathcal{L}(\mathbf{x}_i) \cdot \begin{pmatrix}\mathbf{x}_i^T & 1\end{pmatrix}\mathbf{a}\right), \tag{2}$$

where $g : \mathbb{R} \to \mathbb{R}$ is a strictly monotonic decreasing function, called a loss function. Of the many possibilities [6], three candidates are shown in Fig. 1. We should note here that using a signum-function approximating loss function, the measure estimates the number of poorly separated points when $\alpha \to \infty$.

Minimizing $\tau(\mathbf{a})$ we get an unconstrained minimization of a strictly convex function, which is in marked contrast to the quadratic optimization with constraints

in SVM. With a suitably smooth loss function, the gradient vector of $\tau(\mathbf{a})$ will be smooth as well, hence one can apply quasi-Newton methods or even the Newton iteration method.

After obtaining the optimal parameter of the separating hyperplane the binary classification of an arbitrary point $\mathbf{z}$ can be carried out by:

$$sign\left(\begin{pmatrix}\mathbf{z}^T & 1\end{pmatrix}\mathbf{a}\right).$$

# 3 Linear regression in classification

Linear regression attempts to optimally fit a hyperplane onto the indicator function $\mathcal{L}$. The indicator function has values $\mathcal{L}(\mathbf{x}_1),\ldots,\mathcal{L}(\mathbf{x}_n)$ at the sample points $\mathbf{x}_1,\ldots,\mathbf{x}_n$ while the regression hyperplane has function values $f(\mathbf{x}_1),\ldots,f(\mathbf{x}_n)$, where

$$f(\mathbf{z}) = \begin{pmatrix}\mathbf{z}^T & 1\end{pmatrix}\mathbf{a} \qquad \mathbf{z}\in\mathbb{R}^m, \quad \mathbf{a}\in\mathbb{R}^{m+1}.$$

Thus the error of the sample point $\mathbf{x}_i$ can be expressed by

$$\epsilon_i = \mathcal{L}(\mathbf{x}_i) - f(\mathbf{x}_i) = \mathcal{L}(\mathbf{x}_i) - \begin{pmatrix}\mathbf{x}_i^T & 1\end{pmatrix}\mathbf{a}.$$

The optimal parameter of the regression hyperplane can be obtained by minimizing the following sum:

$$\min_{\mathbf{a}}\sum_{i=1}^{n}\epsilon_i^2 = \min_{\mathbf{a}}\|\mathcal{L}(X) - X_1\mathbf{a}\|_2^2 \qquad X_1 = \begin{pmatrix}\mathbf{x}_1^T & 1\\ \vdots & \vdots\\ \mathbf{x}_n^T & 1\end{pmatrix},$$

whose well-known solution is given by

$$\mathbf{a} = (X_1^T X_1)^+ X_1^T \mathcal{L}(X), \tag{3}$$

where $^+$ denotes the Moore-Penrose pseudo-inverse.

Though the regression makes use of the hyperplane in a different sense from that in the classification problem, the regression-based binary classification of an arbitrary point $\mathbf{z}$ can still be performed in the same way as that for a linear classifier:

$$sign\left(\begin{pmatrix}\mathbf{z}^T & 1\end{pmatrix}\mathbf{a}\right).$$

# 4 Minor Component Classifier

Let us take the sample points $X$ with the corresponding labels $\mathcal{L}(X)$, and represent $(\mathbf{x}_1^T,\mathcal{L}(\mathbf{x}_1))^T,\ldots,(\mathbf{x}_n^T,\mathcal{L}(\mathbf{x}_n))^T$ as vectors in $\mathbb{R}^{n+1}$. In this extended space a hyperplane with parameter $\bar{\mathbf{a}}$ contains points $\mathbf{z}$ where

$$\begin{pmatrix}\mathbf{z}^T & \mathcal{L}(\mathbf{z}) & 1\end{pmatrix}\bar{\mathbf{a}} = 0, \qquad \mathbf{z}\in\mathbb{R}^m, \quad \bar{\mathbf{a}}\in\mathbb{R}^{m+2}.$$

The distance of $(\mathbf{x}_i\ \mathcal{L}(\mathbf{x}_i))$ from the hyperplane is

$$\delta(\mathbf{x}_i, \mathcal{L}(\mathbf{x}_i)) = \frac{\begin{pmatrix}\mathbf{x}_i^T & \mathcal{L}(\mathbf{x}_i) & 1\end{pmatrix}\bar{\mathbf{a}}}{\|\mathbf{a}\|_2},$$

so there exists an optimal hyperplane fitting on the extended sample points with least error:

$$\min_{\bar{\mathbf{a}}} \sum_{i=1}^{n} \delta(\mathbf{x}_i, \mathcal{L}(\mathbf{x}_i))^2 = \min_{\bar{\mathbf{a}}} \frac{\bar{\mathbf{a}}^T X_2^T X_2 \bar{\mathbf{a}}}{\bar{\mathbf{a}}^T \bar{\mathbf{a}}} \qquad X_2 = \begin{pmatrix} \mathbf{x}_1^T & \mathcal{L}(\mathbf{x}_1) & 1 \\ \vdots & \vdots & \vdots \\ \mathbf{x}_n^T & \mathcal{L}(\mathbf{x}_n) & 1 \end{pmatrix}. \quad (4)$$

It can be proved that eigenvectors of $X_2^T X_2$ are the stationary points of the above functional with the corresponding eigenvalues as function values. Thus the solution of the minimization problem can be readily obtained by finding the eigenvector of $X_2^T X_2$ which has the smallest eigenvalue [4].

We should note that the better the fit of a hyperplane onto the points, the lower the deviation of the sample points projections onto the normal vector of the hyperplane. Finding the best hyperplane means performing a Minor Component Analysis (MCA) [5] in the extended space, as MCA searches for directions with a small deviation of the sample points projections.

The binary classification of a point $\mathbf{u}$ in the original space can be performed by computing the absolute distances in the extended space for both labels $\{-1, 1\}$ and probabilities can be assigned to the labels via normalization:

$$P(\mathcal{L}(\mathbf{u}) = 1) = \frac{|\delta(\mathbf{u}, -1)|}{|\delta(\mathbf{u}, 1)| + |\delta(\mathbf{u}, -1)|}$$

$$P(\mathcal{L}(\mathbf{u}) = -1) = \frac{|\delta(\mathbf{u}, 1)|}{|\delta(\mathbf{u}, 1)| + |\delta(\mathbf{u}, -1)|}$$

# 5 Kernel-based nonlinearization

The proposed methods, linear classifiers, linear regression and minor component classifier performs linear separation in the original sample space. Making the separation nonlinear with kernels it must be shown that the methods optimal solutions are in the linear subspace of the appropriate extended points:

$$\mathbf{a} = X_1 \alpha \quad \alpha \in \mathbb{R}^n,$$

and

$$\bar{\mathbf{a}} = X_2 \beta \quad \beta \in \mathbb{R}^n.$$

Regarding a linear classifier the parameter vector $\mathbf{a}$ can be decomposed into two perpendicular components $\mathbf{a}_1$ and $\mathbf{a}_2$, where the first component lies in the subspace of the extended sample points $X_1$:

$$\mathbf{a} = \mathbf{a}_1 + \mathbf{a}_2 \qquad \mathbf{a}_1 = X_1 \alpha, \quad \alpha \in \mathbb{R}^n, \quad \mathbf{a}_1 \perp \mathbf{a}_2.$$

The form of the measure $\tau$ then becomes

$$\begin{aligned}\tau(\mathbf{a}) &= \sum_{i=1}^{n} g\left(\mathcal{L}(\mathbf{x}_i)\cdot\begin{pmatrix}\mathbf{x}_i^T & 1\end{pmatrix}(\mathbf{a}_1+\mathbf{a}_2)\right) = \\ &= \sum_{i=1}^{n} g\left(\mathcal{L}(\mathbf{x}_i)\cdot\begin{pmatrix}\mathbf{x}_i^T & 1\end{pmatrix}\mathbf{a}_1 + \mathcal{L}(\mathbf{x}_i)\cdot\begin{pmatrix}\mathbf{x}_i^T & 1\end{pmatrix}\mathbf{a}_2\right) = \\ &= \sum_{i=1}^{n} g\left(\mathcal{L}(\mathbf{x}_i)\cdot\begin{pmatrix}\mathbf{x}_i^T & 1\end{pmatrix}\mathbf{a}_1\right),\end{aligned}$$

because $\mathbf{a}_2$ is orthogonal to all the extended sample points $\begin{pmatrix}\mathbf{x}_i^T & 1\end{pmatrix}$.

Because the measure depends only on $\mathbf{a}_1$, thus the minimization in fact can be performed in the linear subspace of the extended sample points $X_1$. Actually, this result holds true far the other methods as well.

Utilizing the introduced formulas the solutions of the proposed methods can be found by optimizing $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ respectively:

$$\min_{\boldsymbol{\alpha}} \sum_{i=1}^{n} g(\mathcal{L}(\mathbf{x}_i)\cdot\begin{pmatrix}\mathbf{x}_i^T & 1\end{pmatrix} X_1^T\boldsymbol{\alpha}), \tag{5}$$

$$\boldsymbol{\alpha} = (X_1^T X_1 X_1 X_1^T)^{+} X_1^T X_1 \mathcal{L}(X), \tag{6}$$

$$\min_{\boldsymbol{\beta}} \frac{\boldsymbol{\beta}^T X_2 X_2^T X_2 X_2^T \boldsymbol{\beta}}{\boldsymbol{\beta}^T X_2 X_2^T \boldsymbol{\beta}}. \tag{7}$$

Supposing that the pairwise dot products of the extended sample points are known the above optimizations have some polynomial time complexity that depends on the sample points number. Since the time complexity of these methods is not a function of dimension, the original vectors can be transformed to a new space $\mathcal{F}$ with $\phi : \mathcal{X} \to \mathcal{F}$ (see Fig. 2) where the separation can be achieved perhaps more effectively. Now let the dot product be implicitly defined by the kernel function $\kappa$ in this finite or infinite dimensional feature space $\mathcal{F}$ with the associated transformation $\phi$:

$$\kappa(\mathbf{x},\mathbf{y}) = \phi(\mathbf{x})\cdot\phi(\mathbf{y})$$

Algorithms using only the dot product can be executed in the kernel feature space by kernel function evaluations alone. Moreover, since $\phi$ is generally nonlinear the resultant method is nonlinear in the original sample space. Knowing $\phi$ explicitly - and, consequently, knowing $\mathcal{F}$ - is not necessary. We need only define the kernel function, which then ensures an implicit evaluation. The construction of an appropriate kernel function (i.e. when such a function $\phi$ exists) is a non-trivial problem, but there are many good suggestions about the sorts of kernel functions [2, 7, 10] which might be adopted along with some background theory. Among the functions available, the two most popular kernels are:

Polynomial kernel: $\kappa(\mathbf{x},\mathbf{y}) = \left(\mathbf{x}^T\mathbf{y}+1\right)^d, \qquad d\in\mathbb{N}$

Gaussian RBF kernel: $\kappa(\mathbf{x},\mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{r}}, \qquad r\in\mathbb{R}^+$

For a given kernel function the dimension of the feature space $\mathcal{F}$ is not always unique as in the case of a polynomial kernel, where it is at least $\binom{m+d-1}{d}$, while with the Gaussian RBF kernel we get an infinite dimension feature space.

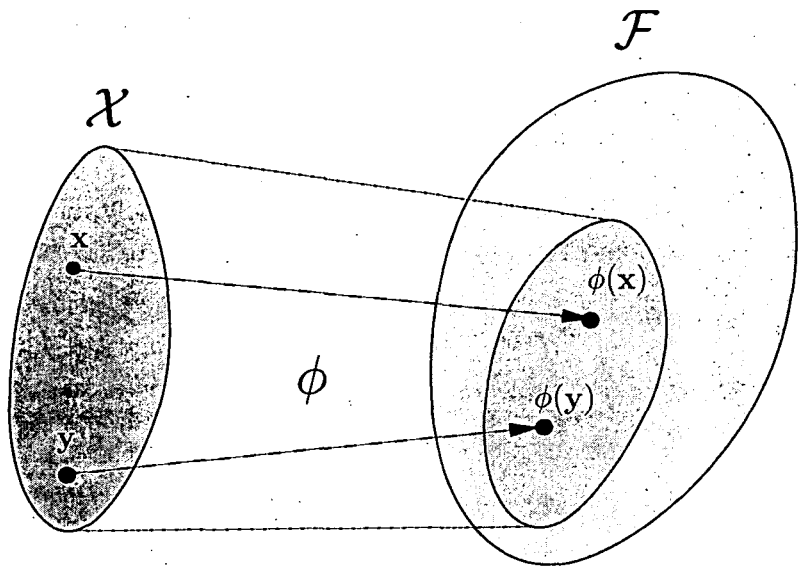


Figure 2: The "kernel-idea". $\mathcal{F}$ is the closure of the linear span of the mapped data. The dot product in the kernel feature space $\mathcal{F}$ is defined implicitly. The dot product of $\phi(\mathbf{x})$ and $\phi(\mathbf{y})$ is $\kappa(\mathbf{x}, \mathbf{y})$.

Employing the kernel-idea to make the proposed methods (5), (6) and (7) non-linear, we obtain the following three expressions:

$$\min_{\boldsymbol{\alpha}} \sum_{i=1}^{n} g\left(\mathcal{L}(\mathbf{x}_i) \cdot \sum_{j=1}^{n} \alpha_i \kappa\left(\left(\mathbf{x}_i^T \quad 1\right), \left(\mathbf{x}_j^T \quad 1\right)\right)\right), \tag{8}$$

$$\boldsymbol{\alpha} = (K^T K)^+ K^T \mathcal{L}(X), \tag{9}$$

$$\min_{\boldsymbol{\beta}} \frac{\boldsymbol{\beta}^T \bar{K} \bar{K} \boldsymbol{\beta}}{\boldsymbol{\beta}^T \bar{K} \boldsymbol{\beta}}, \tag{10}$$

where the matrices $K$ and $\bar{K}$ contain the pairwise dot products of transformed points:

$$\begin{aligned} K_{ij} &= \kappa\left(\left(\mathbf{x}_i^T \quad 1\right), \left(\mathbf{x}_j^T \quad 1\right)\right) \\ \bar{K}_{ij} &= \kappa\left(\left(\mathbf{x}_i^T \quad \mathcal{L}(\mathbf{x}_i) \quad 1\right), \left(\mathbf{x}_j^T \quad \mathcal{L}(\mathbf{x}_j) \quad 1\right)\right). \end{aligned}$$

The solution of (10) can be obtained by finding the eigenvector corresponding to the smallest nontrivial eigenvalue of the generalized eigenproblem $\bar{K}\bar{K}\boldsymbol{\beta} = \lambda \bar{K}\boldsymbol{\beta}$.

Table 1: The best training and testing results using tenfold cross validations. A set of kernel functions with different parameters were used during the tests, but only the best results are summarized here.

| | linear classifier | linear regression | MCC | SVM |
|---|---|---|---|---|
| BUPA | 72.29 | 71.70 | **73.10** | 72.40 |
| | **65.98** | 65.40 | 62.24 | 65.60 |
| chess | **100.0** | 97.42 | 95.98 | **100.0** |
| | **98.08** | 90.73 | 88.49 | **98.08** |
| echo | **100.0** | 92.35 | 91.57 | **100.0** |
| | 89.54 | 89.57 | **90.32** | 90.10 |
| hheart | 86.64 | 85.96 | 85.27 | **87.10** |
| | 80.08 | 79.73 | **80.40** | **80.40** |
| monks | **100.0** | 93.35 | 93.35 | **100.0** |
| | 87.88 | 88.81 | **89.60** | 89.10 |
| spiral | **100.0** | **100.0** | **100.0** | **100.0** |
| | 88.48 | 87.23 | **90.80** | 89.20 |

Note here that if the transformed sample points lies entirely on a hyperplane in the space $\mathcal{F}$ then the normal vector of the hyperplane is not in the subspace of the transformed sample points. Thus perfect fitting of the hyperplane is never realized in regression methods nonlinearized with kernels.

# 6 Experimental Results and Evaluation

When evaluating the efficiency of a new algorithm the usual method is to assess its performance by making use of standard databases. To this end we selected a set of databases from the UCI Repository [9]. Namely, we carried out tests using the BUPA liver, chess, echo, Hungarian heart, monks and spiral databases. All sets were normalized so that each feature had a zero mean and unit deviation and we applied a tenfold cross-validation on all the sets. Since a recent study [3] compared five different Support Vector algorithms using the UCI Repository and concluded that the methods have no significant difference in efficiency, we will employ [8] as the SVM classifier. The numerical results of tenfold cross-validations are shown in Table 1, where the best result is emphasized in bold. It confirms that regression based classification methods are indeed just as effective as the original separation algorithms. Moreover, making use of the labels in the regression task with the Minor Component Classifier the usual classification methods were surpassed in many cases so MCC can now be considered as a rival classification method.

# References

[1] M. AIZERMANN, 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] CRISTIANINI, N. AND SHAWE-TAYLOR, J. *An Introduction to Support Vector Machines and other kernel-based learning methods,* Cambridge University Press, 2000.

[3] HSU, C.-W. AND LIN, C.-J. *A comparison of methods for multi-class support vector machines,* IEEE Transactions on Neural Networks, Vol. 13, pp. 415-425, 2002.

[4] FUHRMANN, D. R. AND LIU, B. *An iterative algorithm for locating the minimal eigenvector of a symmetric matrix,* Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing, pp. 45.8.1-45.8.4, 1984.

[5] LUO, F., UNBEHAUEN, R., CICHOCKI, A. *A minor component analysis algorithm,* Neural Networks, Vol. 10/2, pp. 291-297, 1997.

[6] EVGENIOU, T., PONTIL, M., POGGIO, T. *Regularization Networks and Support Vector Machines,* Advances in Computational Mathematics, Vol. 13/1, pp. 1-50, 2000.

[7] SMOLA, A., BARTLETT, P., SCHÖLKOPF, B., SCHUURMANS, D. *Advances in Large Margin Classifiers,* MIT Press, Cambridge, 2000.

[8] COLLOBERT, R. AND BENGIO, S. *SVMTorch: Support Vector Machines for Large-Scale Regression Problems,* Journal of Machine Learning Research, vol 1, pages 143-160, 2001.

[9] BLAKE, C. L. AND MERZ, C. J. *UCI repository of machine learning databases,* http://www.ics.uci.edu/ mlearn/MLRepository.html, 1998.

[10] VAPNIK, V. N. *Statistical Learning Theory,* John Wiley & Sons Inc., 1998.