An Improved Training Algorithm for Nonlinear Kernel Discriminants

Fahed Abdallah, Cédric Richard, *Member, IEEE*, and Régis Lengellé

***Abstract*—A simple method to derive nonlinear discriminants is to map the samples into a high-dimensional feature space $\mathcal{F}$ using a nonlinear function and then to perform a linear discriminant analysis in $\mathcal{F}$. Clearly, if $\mathcal{F}$ is a very high, or even infinitely, dimensional space, designing such a receiver may be a computationally intractable problem. However, using Mercer kernels, this problem can be solved without explicitly mapping the data to $\mathcal{F}$. Recently, a powerful method of obtaining nonlinear kernel Fisher discriminants (KFDs) has been proposed, and very promising results were reported when compared with the other state-of-the-art classification techniques. In this paper, we present an extension of the KFD method that is also based on Mercer kernels. Our approach, which is called the nonlinear kernel second-order discriminant (KSOD), consists of determining a nonlinear receiver via optimization of a general form of second-order measures of performance. We also propose a complexity control procedure in order to improve the performance of these classifiers when few training data are available. Finally, simulations compare our approach with the KFD method.**

***Index Terms*—Kernel Fisher discriminant, learning machine, second-order criteria, support vector machines.**

I. Introduction

DATA-DRIVEN design of linear receivers of the form $S(X) = W^T X - \nu$ consists of finding optimum W and ν in the sense of a preselected criterion, e.g., a second-order criterion such as the Fisher criterion or the generalized signal-to-noise ratio (SNR), from a training set $\mathcal{A}_n = \{(X_i, Y_i)\}_{i=1,\ldots,n}$. Here, the X_i's are training samples, and the Y_i's indicate either class $\mathcal{C}_0$ or $\mathcal{C}_1$. Since linear discriminant analysis is generally not complex enough for most real-world data, it is important to deal with nonlinear discriminant analysis methods. In recent years, a great interest has been shown in kernel-based algorithms to develop a nonlinear generalization of linear receivers; see [1] and references therein. Kernel-based classification algorithms were primarily used in support vector machines (SVMs) [2], [3]. By mapping the samples into a high-dimensional feature space and reformulating the problem into dot product form in order to use Mercer kernels, an effective solution for nonlinear discriminant analysis has been obtained [4, ch. 5]. This exploits the notion that performing a nonlinear data transformation into some high-dimensional feature space increases the probability of having linearly separable classes in the transformed space. In [5], a nonlinear classification technique based on Fisher discriminants has been proposed. It also uses the Mercer kernel trick and allows the efficient computation of linear Fisher discriminants in feature space. Very promising results had been reported using this approach, called the kernel Fisher discriminant method (KFD), when compared with other state of the art classification techniques. In this paper, we present an extension of the KFD method that also deals with nonlinear discriminant analysis using kernel functions and second-order measures of performance. It consists of determining a kernel-based receiver via optimization of a general form of second-order measures of performance.

The performance of a receiver depends on several factors such as its structure, the number of training data, and the dimension of the space they span [4, ch. 3]. A common way of training a classifier is to adjust its free parameters to minimize an empirical risk, or training error, which can be estimated as the frequency of errors on the training set. However, receivers that minimize empirical risk do not necessarily minimize the generalization error, i.e., the error over the full distribution of possible inputs and their corresponding outputs. The key issue is to tune the complexity of the classifier to the amount of training data in order to get the best generalization performance [4, ch. 4]. One technique to reach this tradeoff is to minimize a cost function composed of two terms: the ordinary training error plus some measure of the receiver complexity. Several schemes have been proposed in statistical inference literature [4], [6]. Here, we propose a technique for controlling the complexity of KSOD receivers that consists of selectively pruning their components in a dual space, which is reminiscent of the optimal brain damage (OBD) method [7].

The paper is organized as follows. In Section II, we start with a brief review of second-order criteria, showing how they can be used for designing linear detectors from training data. We present our nonlinear approach in Section III, and we use kernel functions in order to obtain a very simple algorithm. In Section IV, we propose a complexity control procedure for improving classification performance. Finally, some concluding remarks are presented in Section V.

II. Second-Order Measures of Quality for Signal Classification

A. Definition

Let $S(X) : \mathbb{R}^d \longrightarrow \mathbb{R}$ be an arbitrary measurable function. Second-order measures of performance $\Psi(m_0, m_1, \sigma_0^2, \sigma_1^2)$ are

Manuscript received June 24, 2003; revised October 5, 2003. The associate editor coordinating the review of this paper and approving it for publication was Dr. Kenneth E. Barner.

The authors are with the Laboratoire de Modélization et Sûreté des Systèmes (LM2S), Université de Technologie de Troyes (UTT), 10010 Troyes cedex, France.

Digital Object Identifier 10.1109/TSP.2004.834346

defined in terms of first- and second-order moments of $S(X)$, namely

$$m_i \triangleq \mathrm{E}\{S(X)|Y=i\}, \quad \sigma_i^2 \triangleq \mathrm{Var}\{S(X)|Y=i\} \tag{1}$$

with $i \in \{0,1\}$. There have been many articles exploring, individually, the relevancy of well-known second-order criteria such as Fisher, mean-square error, and generalized SNR [8]. In [9], the aim of the authors is to unify these results by showing that there exists a broad class of second-order criteria that lead to Bayes-optimal solutions for general nonlinear detectors design.

Here, we use such criteria for designing linear decision statistics $S(X)$. If d-dimensional observation X is normally distributed, $S(X)$ is also normally distributed. Therefore, the error in the projected one-dimensional (1-D) space is determined by m_i and σ_i^2. Even if X is not normal, $S(X)$ can be close to normal for large d since it is the summation of d terms, and the central limit theorem may come into effect [10, ch. 4]. Then, second-order criteria appear as reasonable measures of separability in the projected space.

B. Designing Linear Discriminants Using Second-Order Criteria

Linear classifiers are the simplest one as far as implementation is concerned and are directly related to many known techniques such as correlations and Euclidean distances [10, ch. 4]. A well-known strategy for deriving linear detectors consists of using Fisher criterion [8]. However, it is stated in [11, ch. 4] that linear discriminants derived via maximization of Fisher criterion can be arbitrarily inappropriate: There are distributions where even though the two classes are linearly separable, the Fisher linear discriminant has an error probability close to 1. We will now show how a general form of second-order criteria can be used to design optimum linear detectors.

The conditional expected values and variances of the statistic $S(X) = W^T X - \nu$ are given by

$$m_i = \mathrm{E}\{S(X)|Y=i\} = W^T M_i - \nu \tag{2}$$
$$\sigma_i^2 = \mathrm{Var}\{S(X)|Y=i\} = W^T \mathbf{\Sigma}_i W \tag{3}$$

where M_i and $\mathbf{\Sigma}_i$ are the conditional expected vectors and covariance matrices of X. Let Ψ be any second-order criterion. The optimal statistic $S(X)$ is obtained by equating to zero the partial derivatives of Ψ with respect to W and ν:

$$\begin{cases} \frac{\partial \Psi}{\partial W} = \frac{\partial \Psi}{\partial \sigma_0^2}\frac{\partial \sigma_0^2}{\partial W} + \frac{\partial \Psi}{\partial \sigma_1^2}\frac{\partial \sigma_1^2}{\partial W} + \frac{\partial \Psi}{\partial m_0}\frac{\partial m_0}{\partial W} + \frac{\partial \Psi}{\partial m_1}\frac{\partial m_1}{\partial W} = 0 \\ \frac{\partial \Psi}{\partial \nu} = \frac{\partial \Psi}{\partial \sigma_0^2}\frac{\partial \sigma_0^2}{\partial \nu} + \frac{\partial \Psi}{\partial \sigma_1^2}\frac{\partial \sigma_1^2}{\partial \nu} + \frac{\partial \Psi}{\partial m_0}\frac{\partial m_0}{\partial \nu} + \frac{\partial \Psi}{\partial m_1}\frac{\partial m_1}{\partial \nu} = 0. \end{cases} \tag{4}$$

The partial derivatives of m_i and σ_i^2 with respect to W and ν are given by

$$\frac{\partial \sigma_i^2}{\partial W} = 2\mathbf{\Sigma}_i W, \quad \frac{\partial m_i}{\partial W} = M_i, \quad \frac{\partial \sigma_i^2}{\partial \nu} = 0, \quad \frac{\partial m_i}{\partial \nu} = -1. \tag{5}$$

Then, (4) can be reformulated as

$$\begin{cases} 2\left[\frac{\partial \Psi}{\partial \sigma_0^2}\mathbf{\Sigma}_0 + \frac{\partial \Psi}{\partial \sigma_1^2}\mathbf{\Sigma}_1\right] W = -\left[\frac{\partial \Psi}{\partial \eta_0}M_0 + \frac{\partial \Psi}{\partial \eta_1}M_1\right] \\ \frac{\partial \Psi}{\partial \eta_0} + \frac{\partial \Psi}{\partial \eta_1} = 0. \end{cases} \tag{6}$$

Substituting the second equation into the first leads to

$$2\left[\frac{\partial \Psi}{\partial \sigma_0^2}\mathbf{\Sigma}_0 + \frac{\partial \Psi}{\partial \sigma_1^2}\mathbf{\Sigma}_1\right] W = \frac{\partial \Psi}{\partial \eta_0}[M_1 - M_0]. \tag{7}$$

Dividing both sides by $((\partial\Psi/\partial\sigma_0^2) + (\partial\Psi/\partial\sigma_1^2))$ and eliminating any scale factor multiplying to W and $(M_1 - M_0)$, we finally obtain the optimum projection direction under which the optimum value of any given second-order criteria Ψ is reached. This satisfies the following relation [10, ch. 4], [12], [13]:

$$[\rho\mathbf{\Sigma}_0 + (1-\rho)\mathbf{\Sigma}_1]\, W = [M_1 - M_0] \tag{8}$$

where the parameter ρ depends on Ψ according to

$$\rho = \frac{\frac{\partial \Psi}{\partial \sigma_0^2}}{\frac{\partial \Psi}{\partial \sigma_0^2} + \frac{\partial \Psi}{\partial \sigma_1^2}}. \tag{9}$$

Once the functional form of Ψ is selected, the optimum bias ν can be obtained from the second equation in (6). Note that $\rho \in [0,1]$ if, and only if, $\partial\Psi/\partial\sigma_0^2$ and $\partial\Psi/\partial\sigma_1^2$ are of the same sign. This means that Ψ varies in the same way with σ_0^2 and σ_1^2, which is a desirable but nonmandatory requirement for design criteria [14]. In addition to Fisher criterion

$$\Psi_{\mathrm{Fisher}}\left(m_0, m_1, \sigma_0^2, \sigma_1^2\right) = \frac{(m_1 - m_0)^2}{P(X \in \mathcal{C}_0)\sigma_0^2 + P(X \in \mathcal{C}_1)\sigma_1^2} \tag{10}$$

this leads to $\rho = P(X \in \mathcal{C}_0)$, and several well-known second-order criteria such as the SNR and mean square error satisfy this property.

Relation (8) shows that the optimum direction W depends on the criterion Ψ via a single parameter $\rho \in \mathbb{R}$. Rather than arbitrarily selecting Ψ, we suggest the use of a complementary performance measure such as error probability to adjust ρ. Let ρ^* denote the optimum, i.e., $\rho^* = \arg\min_{\rho\in[0,1]} P_e(S_\rho)$, where $S_\rho(X) = W_\rho^T X - \nu_\rho$ with W_ρ satisfying the linear system (8). In the sense of the error probability, the structure S_{ρ^*} obviously performs better than or equal to receivers maximizing Fisher criterion, SNR, or mean square error. We will now discuss the problems that arise in implementing this scheme when the designer does not know the distribution of the data.

C. Data-Driven Approach

In the previous section, we have assumed that the conditional expected vectors and conditional covariance matrices of the observation X are given. However, if only a set of samples $\mathcal{A}_n = \{(X_i, Y_i)\}_{i=1,\ldots,n}$ is available without any prior knowledge of the density function $p(X,Y)$, M_i and $\mathbf{\Sigma}_i$ must be estimated. The error probability $P_e(S)$, here used as a complementary performance measure to select Ψ via ρ, cannot be computed either. A

convenient approach consists of estimating $P_e(S)$ by its discrete approximation computed with the training examples

$$P_{\text{emp}}(S, \mathcal{A}_n) = \sum_{i=1}^{n} \mathbf{1}_{\{D(X_i) \neq Y_i\}} \tag{11}$$

where $D(X) = 1$ if $S(X) > 0$, and 0 otherwise. This strategy, which is called the *resubstitution method*, results in an optimistically biased estimate because the same data are used for both designing and testing the detector. In order to avoid this bias, one can estimate $P_e(S)$ on a separate set in order to assure independence between design and test samples. A serious problem concerning the applicability of this approach, which is called the *holdout method*, is that an additional sample is rarely available in practice. In such cases, there exist strategies that rely on the training data only, e.g., the *jacknife* and the *bootstrap* methods [19]; see also [11, ch. 8] and references therein.

Minimization of empirical risk uses extensive computation because it is generally not unimodal. In addition, it is difficult to optimize by standard techniques such as gradient descent because the gradients are zero almost everywhere [11, ch. 4]. However, the advantage of the formulation (8) is that there is only one parameter $\rho \in [0, 1]$ to tune for both selecting a second-order criterion Ψ and designing the detector. This makes the training stage very simple to implement, particularly with the following iterative procedure.

1) Estimate the conditional expected vectors $\hat{M}_i$ and covariance matrices $\hat{\mathbf{\Sigma}}_i$ of X.
2) Set ρ to zero.
3) While $(\rho \leq 1)$ repeat
   - Solve (8) to get the vector W, and save the result as W_ρ.
   - Find the threshold ν_ρ that minimizes an estimate of the generalization error.
   - Update $\rho : \rho \leftarrow \rho + \Delta\rho$, where $\Delta\rho$ is a selected step.
4) Select the best detector characterized by $(\alpha_{\rho^*}, \nu_{\rho^*})$.

As the parameter ρ varies from 0 to 1, this algorithm only explores the subset of second-order criteria Ψ whose derivatives with respect to σ_0^2 and σ_1^2 have the same sign; see (9). This desirable but nonmandatory requirement for design criteria makes the process much simpler than adjusting ρ in $\mathbb{R}$. In the next section, we will use the same approach to design nonlinear kernel-based detection structures.

III. Nonlinear Discriminant Analysis

A simple method of obtaining a nonlinear discriminant is to map the samples into a high-dimensional feature space $\mathcal{F}$ using a nonlinear function

$$\begin{aligned} \phi : \mathbb{R}^d &\longrightarrow \mathcal{F} \\ X &\longmapsto \phi(X) \end{aligned}$$

and then to perform a linear discriminant analysis in $\mathcal{F}$ with the set $\{(\phi(X_i), Y_i)\}_{i=1,\ldots,n}$. This reflects the notion that performing a nonlinear data transformation into some specific high-dimensional feature spaces increases the probability of having linearly separable classes within the transformed space.

Clearly, if $\mathcal{F}$ is a very high, or even infinitely, dimensional space, deriving $S(X)$ may be a computationally intractable problem. However, by using the theory of reproducing kernels [2], such a problem can be solved without explicitly mapping the data to the feature space $\mathcal{F}$. Recently, a method of obtaining nonlinear kernel Fisher discriminant, called the KFD method, has been proposed [5] and widely studied [15], [16]. A closed-form solution to this problem has also been obtained in [17]. Here, we propose an extension of the KFD approach called nonlinear kernel second-order discriminant (KSOD) method, that it is also based on Mercer kernels and second-order criteria.

A. Nonlinear Kernels

The idea of constructing KSODs comes from considering a general expression for the inner product in Hilbert space [4, ch. 5]

$$\phi(U) \cdot \phi(V) = \kappa(U, V) \tag{12}$$

where κ is called *kernel*. Let $\kappa(U, V) \in \mathcal{L}_2$ be any symmetric function. According to the Hilbert–Schmidt Theory [18], it can be expanded as

$$\kappa(U, V) = \sum_{i=1}^{\infty} \lambda_i \phi_i(U) \phi_i(V) \tag{13}$$

where λ_i and ϕ_i are eigenvalues and eigenfunctions given by

$$\int \kappa(U, V) \phi_i(U) dU = \lambda_i \phi_i(V). \tag{14}$$

A sufficient condition to ensure that $\kappa(U, V)$ defines an inner product in a feature space is that all the λ_i's in (13) are positive. According to Mercer's theorem, this condition is achieved if, and only if

$$\iint g(U) \kappa(U, V) g(V) dU dV > 0 \tag{15}$$

for all g such that

$$\int g^2(U) dU < \infty. \tag{16}$$

An example of kernel satisfying Mercer's theorem is the polynomial kernel defined as

$$\kappa(U, V) = (1 + U^T V)^q \tag{17}$$

with $U, V \in \mathbb{R}^d$, and $q \in \mathbb{N}^*$. The mapping ϕ associated with this kernel can be easily determined from (13). As an example, for $d = 2$ and $q = 2$, we directly obtain

$$\phi(U) = \left(1, \sqrt{2}u_1, \sqrt{2}u_2, u_1^2, u_2^2, \sqrt{2}u_1 u_2\right) \tag{18}$$

where $U = (u_1, u_2)$. Using different Mercer kernels, one can design learning machines with different types of nonlinear decision surfaces in input space. Classical radial basis functions (RBFs) have received significant attention, most commonly

with a Gaussian of the form $\kappa(U,V) = \exp(-\|U-V\|^2/\beta^2)$. The exponential radial basis function (ERBF) given by $\kappa(U,V) = \exp(-\|U-V\|/\beta^2)$, which produces a piecewise linear separating surface, is also a typical Mercer kernel. Other examples may be found in [4, ch. 5].

B. KSOD Method

Let Ψ be any second-order criterion. From (8), it directly follows that the discriminant function $S^\phi(X) = \phi(X)^T W^\phi$ operating in $\mathcal{F}$ is optimum in the sense of Ψ if it satisfies

$$\left[\rho\boldsymbol{\Sigma}_0^\phi + (1-\rho)\boldsymbol{\Sigma}_1^\phi\right] W^\phi = \left[M_1^\phi - M_0^\phi\right] \tag{19}$$

where M_i^ϕ and $\boldsymbol{\Sigma}_i^\phi$ denote the conditional expected vectors and covariance matrices of $\phi(X)$, respectively. They can be estimated from training data as follows:

$$\hat{M}_i^\phi = \frac{1}{n_i}\sum_{X\in\mathcal{C}_i}\phi(X) \tag{20}$$

$$\hat{\boldsymbol{\Sigma}}_i^\phi = \frac{1}{n_i}\sum_{X\in\mathcal{C}_i}\phi(X)\phi(X)^T - \left(\hat{M}_i^\phi\right)\left(\hat{M}_i^\phi\right)^T \tag{21}$$

where n_i is the number of samples from class $\mathcal{C}_i$ in the training set. Equation (19) may be difficult to solve when $\mathcal{F}$ is a very high-dimensional space. However, one can get around this by using Mercer kernels. They satisfy

$$\kappa(X_i, X_j) = \phi(X_i)^T\phi(X_j) \tag{22}$$

which is the inner product of the X_i's in the feature space $\mathcal{F}$. We will now show how this property can be used to solve the problem (19) without explicitly mapping the data in the feature space $\mathcal{F}$.

Following [5], W^ϕ must lie in the span of all training samples in $\mathcal{F}$ according to the theory of reproducing kernels. This means that W^ϕ has a dual kernel representation and can be expressed as

$$W^\phi = \sum_{i=1}^{n}\alpha(i)\phi(X_i) = \mathbf{Q}\alpha. \tag{23}$$

Here, $\mathbf{Q}$ denotes the matrix $[\phi(X_1)\cdots\phi(X_n)]$, and the $\alpha(i)$'s are the dual parameters. Multiplying (19) by $\mathbf{Q}^T$ and using (23) yields

$$\mathbf{N}_\rho\alpha = M \tag{24}$$

with

$$\mathbf{N}_\rho = \left[\rho\mathbf{Q}^T\hat{\boldsymbol{\Sigma}}_0^\phi\mathbf{Q} + (1-\rho)\mathbf{Q}^T\hat{\boldsymbol{\Sigma}}_1^\phi\mathbf{Q}\right] \tag{25}$$

and

$$M = \mathbf{Q}^T\left[\hat{M}_1^\phi - \hat{M}_0^\phi\right]. \tag{26}$$

Using (22), the n by n matrix $\mathbf{N}_\rho$ can be reformulated as follows:[1]

$$\mathbf{N}_\rho = \left[\frac{\rho}{n_0}\mathbf{K}_0(\mathbf{I}-\mathbf{1}_{n_0})\mathbf{K}_0^T + \frac{1-\rho}{n_1}\mathbf{K}_1(\mathbf{I}-\mathbf{1}_{n_1})\mathbf{K}_1^T\right]. \tag{27}$$

In the above equation, $\mathbf{K}_i$ is a n by n_i matrix whose components are given by

$$\mathbf{K}_i(p,q) = \kappa(X_p, X_q) \tag{28}$$

for all $X_p \in (\mathcal{C}_0\cup\mathcal{C}_1)$, $X_q \in \mathcal{C}_i$. $\mathbf{I}$ is the identity matrix, and $\mathbf{1}_{n_i}$ is the matrix with all elements set to $(1/n_i)$. Each component of M in (24) is defined as

$$M(j) = \frac{1}{n_1}\sum_{X\in\mathcal{C}_1}\kappa(X,X_j) - \frac{1}{n_0}\sum_{X\in\mathcal{C}_0}\kappa(X,X_j). \tag{29}$$

From (23), the projection of any new sample X onto W^ϕ is finally given by the kernel expression

$$\phi(X)^T W^\phi = \sum_{i=1}^{n}\alpha(i)\kappa(X_i, X) \tag{30}$$

where the n-dimensional vector α has to be determined from (24). Note that the functional form of $\phi(X)$ does not need to be known. It is implicity defined by the choice of the kernel function κ. Testing different kernel functions such as the polynomial kernel, RBF, and ERBF, one can cover a wide class of nonlinearities and get a powerful nonlinear decision function in input space.

C. Algorithm

With the aim of exploring the set of second-order criteria to design kernel-based detection structures, one can use the iterative procedure described below.

1) Given any Mercer kernel κ, compute the kernel matrices $\mathbf{K}_0$ and $\mathbf{K}_1$ from (28).
2) Compute M from (29).
3) Set ρ to zero.
4) While $(\rho \le 1)$ repeat
   - Compute $\mathbf{N}_\rho$ from (27).
   - Solve (24) to get the vector α, and save the result as α_ρ.
   - Find the threshold ν_ρ that minimizes an estimate of the generalization error.[2]
   - Update $\rho : \rho \leftarrow \rho + \Delta\rho$, where $\Delta\rho$ is a selected step.
5) Select the best detector characterized by $(\alpha_{\rho^*}, \nu_{\rho^*})$.

As the parameter ρ varies from 0 to 1, this algorithm only explores second-order criteria Ψ whose derivatives with respect to σ_0^2 and σ_1^2 have the same sign; see (9). This nonmandatory requirement for design criteria makes the procedure much simpler than adjusting ρ in $\mathbb{R}$. Note that the numerical problems caused by inverting the ill-conditioned matrix $\mathbf{N}_\rho$ in (24) can be avoided by using, e.g., Tikhonov regularization [27] or truncated

[1]Note that (27) includes data centering in feature space.

[2]See Section II-C for details and references on how to estimate the generalization error.

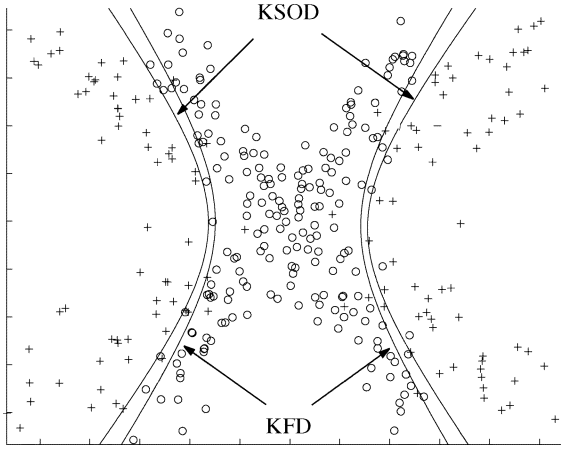

Fig. 1. Separating surfaces in the input space obtained with KFD and KSOD methods. The samples from the two classes are represented by crosses and circles.

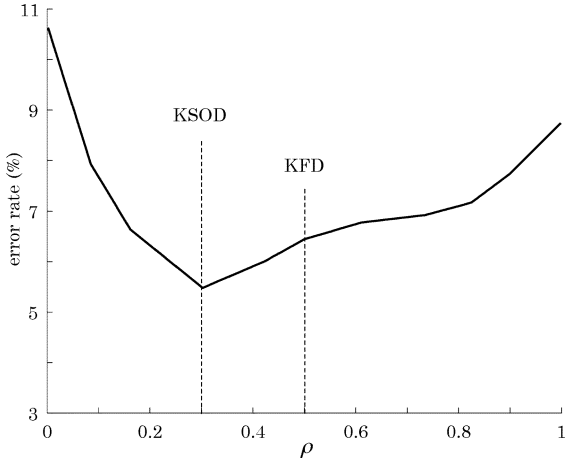

Fig. 2. Selection of ρ based on an estimate of the generalization error. Here, KSOD and KFD receivers are associated, respectively, with $\rho = 0.3$ and $\rho \equiv P(X \in \mathcal{C}_0) = 1/2$.

singular value decomposition [29, ch. 2 and 3]. The standard approach used in the KFD algorithm [5], [15], [17] is to replace $\mathbf{N}_\rho$ with $\mathbf{N}_\rho + \eta\mathbf{I}$, where $\eta > 0$ controls the smoothness of the solution. Here, $\mathbf{I}$ is the identity matrix. As for the Tikhonov approach, it is not a trivial matter to choose an appropriate value for the regularization parameter η. Various algorithms are discussed in [29]. Cross-validation will be used in this paper.

We will now illustrate the algorithm described above with a toy data set, which consists of two noise parabolic shapes mirrored at the x and y axis, as shown in Fig. 1. At first, a 400-sample training set was generated and partitioned into two 200-sample competing classes $\mathcal{C}_0$ and $\mathcal{C}_1$. These data were used to design a KSOD receiver with the polynomial kernel κ of degree 2, 60% of them being dedicated to the determination of α_ρ and 40% to the selection of ν_ρ and ρ based on an estimate of the generalization error over this subset. Fig. 2 gives this error as a function of ρ. It also indicates KSOD and KFD receivers, which correspond, respectively, to $\rho = 0.3$ and $\rho \equiv P(X \in \mathcal{C}_0) = 1/2$. Fig. 1 shows their separating surfaces in the input space. In this experiment, note that the regularization parameter η was set to 10^{-5} based on results of several preliminary runs using the same experimental setup as above.

D. Comparison to KFD

To compare KSOD and KFD approaches, nine experiments were conducted as in [5] on artificial and real-world data from the UCI, DELVE, and STATLOG benchmark repositories.[3] The iterative procedure described in Section III-C was carried out with the RBF kernel function and $\Delta\rho = 0.05$. For each of the nine problems, results were averaged over 40 runs, which were conducted individually as follows. A 8500-sample set $\mathcal{A}_{8500}$ was generated for each run. This set was randomly split into a 400-sample training set $\mathcal{A}_{\text{train}}$, a 100-sample holdout cross-validation set $\mathcal{A}_{\text{cross}}$, and a 8000-sample test set $\mathcal{A}_{\text{test}}$. Each model (α_ρ, ν_ρ) was trained on $\mathcal{A}_{\text{train}}$, and its performance was evaluated on $\mathcal{A}_{\text{cross}}$. The model with the lowest estimated generalization error on $\mathcal{A}_{\text{cross}}$ was selected as the KSOD receiver $(\alpha_{\rho^*}, \nu_{\rho^*})$, whereas the KFD receiver was picked by directly setting $\rho \equiv P(X \in \mathcal{C}_0) = 1/2$. The generalization performance of both receivers were estimated based on $\mathcal{A}_{\text{test}}$. The mean error rates over the 40 runs presented in Table I show that the KSOD method is more efficient than KFD in most cases. Note that these performance levels are different from those reported in [5]. However, the experimental setups are different, e.g., only 200 training data were used by Mika *et al.* To test for significant differences between these two approaches, we used the Wilcoxon Rank Test. The probability values p, given in Table I, indicate if the differences between error scores are statistically significant or not over the 40 runs: p is close to 0 if the differences between the KSOD and the KFD results are statistically significant; otherwise, p is close to 1. One can observe that the KFD is outperformed by the KSOD method on the Cancer, Thyroid, Solar, Diabetes, and the German data. This illustrates the ability of our approach to provide a statistically significant increase in performance of classifiers over the original Fisher algorithm. The KSOD is marginally superior on the Titanic and Heart data, and both competing approaches provide equivalent solutions for the Waveform and Ringnorm data. Interestingly, similar performances were observed with SVMs, as shown in Table II. It would seem that classifiers obtained with KSOD, KFD, and SVM algorithms in these cases are close to the optimum solution in the sense of classical detection theories.

TABLE I
COMPARISON OF KFD AND KSOD (BEST METHOD IN BOLDFACE)

Data set	KFD error rate	KSOD error rate	Wilcoxon p
Thyroid	0.40	**0.25**	0.05
B. cancer	9.76	**7.31**	0.01
Diabetes	19.91	**19.31**	0.09
German	21.40	**20.99**	0.12
Heart	4.46	**4.42**	0.62
Solar	34.66	**32.24**	0.05
Waveform	**11.14**	**11.14**	1
Ringnorm	1.54	**1.53**	0.73
Titanic	26.80	**26.34**	0.59

[3]The data were downloaded from http://www.first.gmd.de/~raetsch/.

TABLE II
COMPARISON OF KSOD-OBD, SVM AND RVM (BEST METHOD IN BOLDFACE)

Data set	KSOD-OBD	SVM	Wilcoxon p_1	RVM	Wilcoxon p_2
Thyroid	**0.22** (75)	0.33 (156)	0.03	0.6 (16)	0.01
B. cancer	**6.65** (70)	7.10 (364)	0.14	-	-
Diabetes	**17.23** (140)	17.68 (308)	0.10	21.34(82)	0
German	**20.94** (190)	21.06 (400)	0.27	25.75 (200)	0
Heart	**4.28** (95)	4.52 (388)	0.31	10.88 (98)	0
Solar	**31.33** (45)	32.73 (364)	0.05	36.82(59)	0
Waveform	11.14 (400)	**11.07** (400)	0.68	-	-
Ringnorm	**1.50** (20)	1.52 (396)	0.45	2.3(11)	0
Titanic	**25.87** (15)	28.88 (400)	0.09	28.58 (53)	0.03

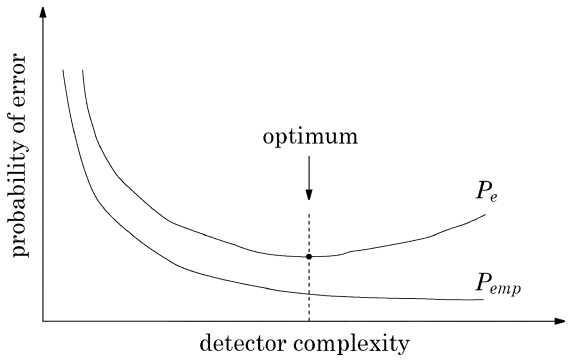

Fig. 3. Schematic illustration of the behavior of generalization error P_e and empirical error P_{emp} during a typical training stage, as a function of the detector complexity. Note that P_{emp}, which is an estimate of P_e based on training data, is also called training error.

IV. COMPLEXITY CONTROL OF KSOD

Achieving good generalization performance with a receiver requires matching its complexity to the amount of available training data [4, ch. 2]. As illustrated in Fig. 3, if the detector is too complex, it is likely to learn the training data, but it will probably not generalize properly. In contrast, if it is not complex enough, it might not be able to extract all the discriminant information available in the training set. This experimental evidence, which is known as the curse of dimensionality, has been studied theoretically by Vapnik and Chervonenkis [21]. In particular, these authors have formally defined the complexity of a detector, which is called the dimension of Vapnik–Chervonenkis, or VC-dimension. This parameter, hereafter denoted V_c, can be used to compute a confidence interval for the error probability of any decision structure S designed from training data. The following inequality holds with a probability of $(1-\epsilon)$:

$$|P_e(S) - P_{\text{emp}}(S, \mathcal{A}_n)| \leq E(n, V_c, \epsilon) \tag{31}$$

where

$$E(n, V_c, \epsilon) = \sqrt{\frac{V_c}{n}\left(1 + \log\frac{2n}{V_c}\right) - \frac{1}{n}\log\frac{\epsilon}{4}}. \tag{32}$$

Here, $\mathcal{A}_n$ denotes a n-sample training set, $P_e(S)$ is the probability of error of the decision structure S, and $P_{\text{emp}}(S, \mathcal{A}_n)$ represents an estimate of this probability based on $\mathcal{A}_n$.

The cardinality n of the training set is generally fixed so that one needs to carefully control V_c in order to reach a low $P_e(S)$. Several strategies such as structural risk minimization (SRM) [21] and minimum description length (MDL) [6] have been proposed to achieve this task. Note that for generalized linear detectors, V_c equals the number of free parameters [4, ch. 3]. The generalization ability of these structures can then be controlled directly by pruning some of their free parameters. A technique proposed by LeCun *et al.*, which is called optimal brain damage (OBD), has been widely used to reduce the size of neural networks by selectively deleting weights. In order to adjust the VC-dimension, and therefore improve the performance of the classification structure given by (30), we will now propose an efficient method for pruning components of the dual vector α provided by (24), which is reminiscent of the OBD method.

A. OBD Method Applied to KSOD Learning Algorithm

We define the best candidate for pruning as the component of α involving the smallest variations of the squared error $\mathcal{E}_\rho$ defined from (24) as

$$\mathcal{E}_\rho = \|\mathbf{N}_\rho \alpha - M\|^2 \tag{33}$$

where $\mathbf{N}_\rho$ and M are given by (27) and (29), respectively. A perturbation $\delta\alpha$ modifies the objective function $\mathcal{E}_\rho$ by the quantity

$$\delta\mathcal{E}_\rho = \sum_i \frac{\partial \mathcal{E}_\rho}{\partial \alpha(i)}\delta\alpha(i) + \frac{1}{2}\sum_i \frac{\partial^2 \mathcal{E}_\rho}{\partial \alpha(i)^2}\delta\alpha(i)^2 + \frac{1}{2}\sum_{i\neq j} \frac{\partial^2 \mathcal{E}_\rho}{\partial \alpha(i)\partial\alpha(j)}\delta\alpha(i)\delta\alpha(j) + O\left(\|\alpha\|^2\right) \tag{34}$$

where $\alpha(i)$ denotes the i^{th} component of α. If α satisfies (24), i.e., $\alpha = \alpha_\rho$, the first term in (34) is zero. To facilitate the decision about the component $\alpha(i)$, which will be set to zero, the pruning process is performed on a basis of eigenvectors of $\mathbf{N}_\rho$. On such a basis, the squared error function can be written as

$$\mathcal{E}_\rho = \sum_i \left[\lambda_\rho(i)\tilde{\alpha}(i) - \tilde{M}(i)\right]^2 \tag{35}$$

with $\tilde{\alpha} = \mathbf{P}_\rho^T \alpha$ and $\tilde{M} = \mathbf{P}_\rho^T M$. The ith column of the matrix $\mathbf{P}_\rho$ is the eigenvector corresponding to the i^{th} eigenvalue $\lambda_\rho(i)$ of $\mathbf{N}_\rho$. If $\tilde{\alpha} = \tilde{\alpha}_\rho \triangleq \mathbf{P}_\rho^T \alpha_\rho$, where α_ρ satisfies (24), replacing $\mathcal{E}_\rho$ in (34) yields

$$\delta\mathcal{E}_\rho = \sum_i \left[\lambda_\rho(i)\delta\tilde{\alpha}(i)\right]^2. \tag{36}$$

Pruning the i^{th} component of $\tilde{\alpha}_\rho$ then increases $\mathcal{E}_\rho$ by

$$\delta\mathcal{E}_\rho(i) = \left[\lambda_\rho(i)\tilde{\alpha}_\rho(i)\right]^2 \tag{37}$$

since $\delta\tilde{\alpha}(i) = \tilde{\alpha}_\rho(i)$ when $\tilde{\alpha}_\rho(i)$ is set to zero. Therefore, the components of $\tilde{\alpha}_\rho$ associated with the smallest variations of $\mathcal{E}_\rho$ given by (37) are good candidates for pruning. The pruning process is continued as long as it improves performance, which must be simultaneously estimated. Several measures of performance are mentioned in the next subsection.

Executing the OBD-based method described above as a computer program is time consuming if it must be repeated

for each ρ. We will show that, in fact, $\delta\mathcal{E}_\rho(i)$ does not depend on ρ. Suppose $\mathbf{A}$ and $\mathbf{B}$ are symmetric non-negative matrices. If $\ker(\mathbf{A}) = \ker(\mathbf{B})$, then there exists a nonsingular matrix $\mathbf{P}$ such that both $\mathbf{P}^T\mathbf{A}\mathbf{P}$ and $\mathbf{P}^T\mathbf{B}\mathbf{P}$ are diagonal.[4] This result can be applied to $\mathbf{Q}^T\hat{\mathbf{\Sigma}}_0^\phi\mathbf{Q}$ and $\mathbf{Q}^T\hat{\mathbf{\Sigma}}_1^\phi\mathbf{Q}$, which represent the standard data-based estimates of the conditional covariance matrices of the $\mathbf{Q}^T\phi(X_i)$'s. Indeed, the condition $\ker(\mathbf{Q}^T\hat{\mathbf{\Sigma}}_0^\phi\mathbf{Q}) = \ker(\mathbf{Q}^T\hat{\mathbf{\Sigma}}_1^\phi\mathbf{Q})$ means that the competing classes $\mathcal{C}_0$ and $\mathcal{C}_1$ span the same space. If not, the detection problem would be trivial. Then, there exists a matrix $\mathbf{P}$, independent of ρ, such that $\mathbf{Q}^T\hat{\mathbf{\Sigma}}_0^\phi\mathbf{Q}$ and $\mathbf{Q}^T\hat{\mathbf{\Sigma}}_1^\phi\mathbf{Q}$ are both diagonal. On the basis of the column vectors of $\mathbf{P}$, the matrix $\mathbf{N}_\rho$ is then transformed into a diagonal matrix $\tilde{\mathbf{N}}_\rho$ with general diagonal element $\lambda_\rho(i)$. Let $\tilde{M} = \mathbf{P}^T M$. Since α_ρ satisfies (24), we have $\lambda_\rho(i)\tilde{\alpha}(i) = \tilde{M}(i)$ if $\lambda_\rho(i) \neq 0$. From (37), it directly follows that

$$\delta\mathcal{E}_\rho(i) = \tilde{M}(i)^2 \tag{38}$$

if $\lambda_\rho(i) \neq 0$. This shows that the variations of $\mathcal{E}_\rho$ do not depend on ρ since $\tilde{M}$ does not depend on ρ. In addition, note from (37) that $\delta\mathcal{E}_\rho(i) = 0$ if $\lambda_\rho(i) = 0$, which means that the corresponding component $\tilde{\alpha}_\rho(i)$ can be directly set to zero.

B. Algorithm

The OBD pruning process can be implemented in two different but equivalent ways. The first one is to change coordinates to a principal axis representation using $\mathbf{P}$, i.e., compute $\tilde{\mathbf{N}}_\rho$ and $\tilde{M}$, then solve the learning (24) and prune the weights $\tilde{\alpha}_\rho(i)$ corresponding to small increase $\delta\mathcal{E}_\rho(i)$ given by (38). Unfortunately, this procedure must be repeated for each ρ since it involves a posterior modification of the structure of receivers. This drawback can be overcome by directly adjusting the dimension of the feature space and thereby reducing the number of necessary free weights $\tilde{\alpha}_\rho(i)$. The following iterative procedure implements this strategy.

1) Change data coordinates to a principal axis representation, i.e., compute $\tilde{\mathbf{N}}_\rho$ and $\tilde{M}$.
2) While measured performance improves, do the following.
   - Prune the ith component of data corresponding to the smallest value $\tilde{M}(i)^2$, i.e., delete the ith column and row of $\tilde{\mathbf{N}}_\rho$ and the ith element of $\tilde{M}$.
   - Apply the KSOD procedure.

As indicated above, the pruning process continues as long as estimated performance improves. Vapnik and Chervonenkis have proposed the guaranteed risk as a criterion [23]. It is defined as the upper bound of $P_e(S)$ in (31), i.e., $P_{\text{emp}}(S, \mathcal{A}_n) + E(n, V_c, \epsilon)$, where $E(n, V_c, \epsilon)$ may be interpreted as a penalty for complexity. This approach is called *structural risk minimization*. Provided that enough training data is available, a simpler strategy is to stop pruning when the error on a separate validation set reaches a minimum. Other ways of predicting generalization performance include *jacknife* and *bootstrap* procedures; see, for example, [24, ch. 9].

[4]A more general assertion is proved in [22, Th. 8.7.1].

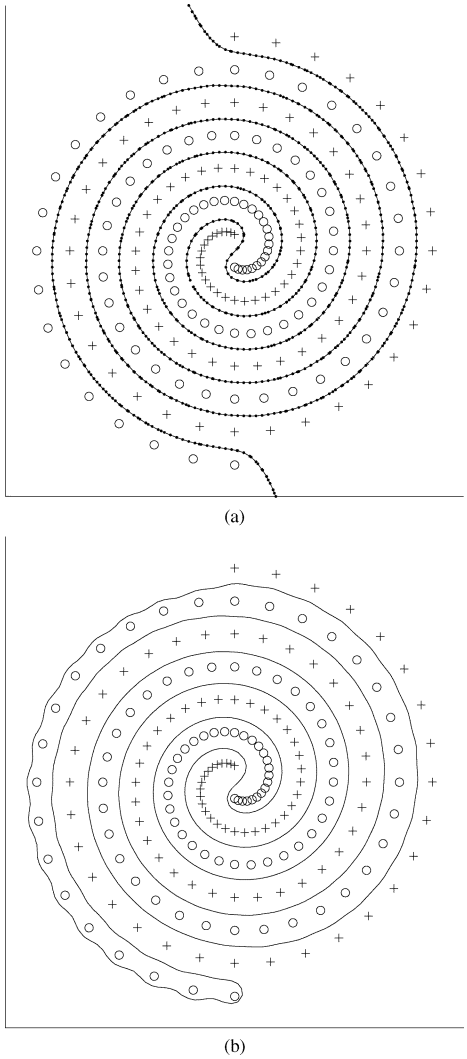

Fig. 4. Solution of the two spirals classification problem. In (a), the KSOD-OBD decision function (solid line) was obtained with only four components of $\bar{\alpha}_\rho$ out of a total of 194, whereas the SVM (dotted line) needed 194 support vectors. In (b), only two components of the vector $\bar{\alpha}_\rho$ out of 194 were used to design the KSOD-OBD decision function (solid line).

Let us now concentrate on a computer simulation to illustrate the iterative procedure presented in this subsection. It is concerned with the two-spirals problem [26], which is a toy classification problem. The task is to discriminate between two sets of 97 training points that lie on two distinct spirals in the x–y plane. These spirals coil three times around the origin and around one another. Fig. 4 proposes interesting results obtained with an ERBF kernel having a width β equal to 1. As shown in Fig. 4(a), one can prune 190 components of $\tilde{\alpha}$ out of a total of 194 without affecting the margins between the decision function (solid line) and the training samples. Such a very economical solution generally has a good generalization property. A similar decision function was obtained (dotted line) with the SVM method[5] that selected 100% of the 194 training data as support vectors. Pruning 192 out of a total of 194 components of $\tilde{\alpha}$ leads to a degraded classification function, as can be seen in Fig. 4(b).

[5]The Matlab code used to optimize SVM in this paper was downloaded from http://www.isis.ecs.soton.ac.uk/resources/svminfo.

C. Experimental Comparisons

Experiments were conducted on the nine benchmark problems considered in Section III, with the same experimental setup. The OBD process was stopped when the error rate on the holdout cross-validation set $\mathcal{A}_{\text{cross}}$ reached a minimum. Next, the performance of these structures were estimated with the test set $\mathcal{A}_{\text{test}}$. Comparing Tables I and II, it can be seen that the OBD algorithm improves the performance of KSOD receivers. In order to illustrate the competitiveness of OBD-based KSOD detectors, they were compared to other state-of-the-art sparse kernel machines: SVM and RVM [28].[6] Obviously, all were trained and tested strictly following the same experimental conditions, in particular, by using the same training sets $\mathcal{A}_{\text{train}}$ to design detection structures and the same holdout cross-validation sets $\mathcal{A}_{\text{cross}}$ to adjust parameters such as regularization constants. Table II clearly shows that the KSOD-OBD method is more efficient than SVM and RVM in most cases. This result is confirmed by the Wilcoxon Rank Tests p_1 between KSOD-OBD and SVM and p_2 between KSOD-OBD and RVM, except for the Waveform and Ringnorm data. In these two cases, it would seem that classifiers are close to the optimum solution in the sense of classical detection theories since they were obtained with training algorithms of different types. Let us analyze now the sparsity of solutions. One of the most important properties of SVMs is that solutions are generally sparse [1]. However, as indicated in Table II between parentheses, the median number of support vectors required by SVMs to solve each benchmark problem was very large in comparison with the median number of components $\tilde{\alpha}(i)$ selected by the OBD method. Such very economical solutions generally have good generalization performance. The RVM method required fewer kernel functions in most cases. Unfortunately, the resulting performances were generally unsatisfactory. Another disadvantage of RVMs is in the complexity of the training phase, as it is necessary to repeatedly compute and invert the Hessian matrix [28]. The problem of the ill-conditioned Hessian matrix prevented us from training RVMs for the Cancer and the Waveform data.

V. Conclusion

In this paper, we have proposed a method of obtaining kernel-based decision structures. As in the well-known KFD approach, it uses the Mercer trick to compute linear discriminants in feature space that correspond to powerful nonlinear decision functions in input space. The training algorithm consists of optimizing a general form of second-order criteria. One of its main advantages lies in its simplicity: There is only one parameter to tune for both exploring the whole family of second-order criteria and designing the detector. It is well known in pattern recognition that the generalization error of classifiers depends on their learning capacity and the number of available training data. The OBD-based procedure developed here is a powerful tool for tuning the complexity of generalized linear receivers of various kinds and improving their performance. We have successfully experimented OBD-based KSOD receivers on simulated and real data. In particular, experiments on benchmark data have shown the ability of our approach to design KSOD detectors with improved generalization performance. This methodology may offer a helpful support for designing efficient classifiers in many applications of current interest.

[6]The Matlab code used to design RVM in these experiments was downloaded from http://www.research.microsoft.com/mlp/rvm.

References

[1] K. R. Müller, S. Mika, G. Rätsch, K. Tsuda, and B. Schölkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Networks*, vol. 12, pp. 181–201, May 2001.

[2] B. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. Fifth Annual Workshop Computational Learning Theory*, Pittsburgh, PA, 1992, pp. 144–152.

[3] C. Cortes and V. Vapnik, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.

[4] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer Verlag, 1995.

[5] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K. R. Müller, "Fisher discriminant analysis with kernels," *Adv. Neural Networks Signal Process.*, pp. 41–48, 1999.

[6] J. Rissanen, "Stochastic complexity in statistical inquiry," *Ann. Statist.*, vol. 14, no. 3, pp. 1080–1100, 1986.

[7] Y. LeCun, J. S. Denker, and S. A. Solla, "Optimal brain damage," in *Proc. Advances Neural Inform. Process. Syst.*, vol. 2, 1990, pp. 598–605.

[8] W. A. Gardner, "A unifying view of second-order measures of quality for signal classification," *IEEE Trans. Commun.*, vol. COM-28, pp. 807–816, June 1980.

[9] C. Richard, R. Lengellé, and F. Abdallah, "Bayes-optimal detectors design using relevant second-order criteria," *IEEE Signal Processing Lett.*, vol. 9, Jan. 2002.

[10] K. Fukunaga, *Statistical Pattern Recognition*. San Diego, CA: Academic, 1990.

[11] L. Devroye, L. Gyorfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. New York: Springer-Verlag, 1996.

[12] D. W. Patterson and R. L. Mattson, "A method of finding linear discriminant functions for a class of performance criteria," *IEEE Trans. Inform. Theory*, vol. IT-12, pp. 380–387, June 1966.

[13] C. Richard, "Une méthodologie pour la détection à structure imposée," Ph.D. dissertation, Compiègne Univ. Technol., Compiègne, France, 1998.

[14] F. Abdallah, C. Richard, and R. Lengellé, "On equivalence between detectors obtained from second-order measures of performance," in *Proc. XI European Signal Process. Conf.*, Toulouse, France, Sept. 3–6, 2002.

[15] S. Mika, G. Rätsch, and K. R. Müller, "A mathematical programming approach to the kernel Fisher algorithm," in *Proc. Advances Neural Inform. Process. Syst.*, T. K. Leen, T. G. Dietterich, and V. Tresp, Eds. Cambridge, MA: MIT Press, 2001, pp. 591–597.

[16] S. Mika, A. J. Smola, and B. Schölkopf, "An improved training algorithm for kernel Fisher discriminants," in *Proc. AISTATS*, T. Jaakkola and T. Richardson, Eds. San Francisco, CA: Morgan Kaufmann, 2001, pp. 98–104.

[17] S. A. Billings and K. L. Lee, "Nonlinear Fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm," *Neural Networks*, vol. 15, pp. 263–270, 2002.

[18] R. Courant and D. Hilbert, *Methods of Mathematical Physics*. New York: Wiley, 1953.

[19] B. Efron, "Bootstrap methods: another look at jacknife," *Ann. Statist.*, vol. 7, pp. 1–26, 1979.

[20] J. Friedman, "Regularized discriminant analysis," *J. Amer. Statist. Assoc.*, vol. 84, no. 405, pp. 165–175, 1989.

[21] V. Vapnik and A. Y. Chervonenkis, "On the uniform convergence of relative frequencies of events to their probabilities," *Theory Probability Applicat.*, vol. 16, pp. 264–280, 1971.

[22] G. H. Golub and C. F. Van Loan, *Matrix Computations*. Baltimore, MD: Johns Hopkins Univ. Press, 1993.

[23] V. Vapnik and A. Y. Chervonenkis, *Theory of Pattern Recognition* (in Russian). Moscow, Russia: Nauka, 1974.

[24] R. O. Duda, P. E. Hart, and D. C. Stork, *Pattern Classification*. New York: Wiley, 2001.

[25] P. Lachenbruch and M. Mickey, "Estimation of error rates in discriminant analysis," *Technometr.*, vol. 10, pp. 1–11, 1968.

[26] K. J. Lang and M. J. Witbrock, "Learning to tell two spirals apart," in *Proc. Connectionnist Summer Schools*. San Francisco, CA: Morgan Kaufmann, 1988.
[27] A. N. Tikhonov and V. Y. Arsenin, *Solutions of Ill-Posed Problems*. New York: Wiley, 1977.
[28] M. Tipping, "The relevance vector machine," *Adv. Neural Inform. Process. Syst.*, vol. 12, pp. 652–658, 2000.
[29] C. Hansen, *Rank-Deficient and Ill-Posed Problems: Numerical Aspects of Linear Inversion*, PA: SIAM, 1998.

Fahed Abdallah received the Dipl.-Ing. degree in electrical engineering in 1999 and the M.S. degree in industrial control in 2000, both from the Faculty of Engineering, the University of Lebanon, Beirut. He is pursuing the Ph.D. degree at Troyes University of Technology, Troyes, France.

His scientific interests are in the fields of machine leaning and kernel methods.

Cédric Richard (M'01) was born in Sarrebourg, France, on January 24, 1970. He received the Dipl.-Ing. and the M.S. degrees in 1994 and the Ph.D. degree in 1998 from Compiègne University of Technology, Compiègne, France, all in electrical and computer engineering.

From 1999 to 2003, he was an Associate Professor at Troyes University of Technology, Troyes, France. Since 2003, he has been a Professor with the Systems Modeling and Dependability Laboratory, Troyes University of Technology. His current research interests involve time-frequency analysis, statistical estimation and decision theories, and pattern recognition.

Régis Lengellé was born on March 30, 1958. He received the Dipl.-Ing, M.S., and Ph.D. degrees from the Compiègne University of Technology, Compiègne, France, in 1980, 1981, and 1983, respectively, and the Habilitation à Diriger des Recherches from the University Henri Poincaré Nancy I, Nancy, France, in 1994.

From 1985 to 1993, he was an Associate Professor at Compiègne University of Technology. Since 1994, he has been a Professor at Troyes University of Technology, Troyes, France, with research interests focused in signal processing, statistical decision theory, and pattern recognition, with applications to systems monitoring.