RKHS Bayes Discriminant: A Subspace Constrained Nonlinear Feature Projection for Signal Detection

Umut Ozertem, *Member, IEEE*, and Deniz Erdogmus, *Senior Member, IEEE*

***Abstract*—Given the knowledge of class probability densities, *a priori* probabilities, and relative risk levels, Bayes classifier provides the optimal minimum-risk decision rule. Specifically, focusing on the two-class (detection) scenario, under certain symmetry assumptions, matched filters provide optimal results for the detection problem. Noticing that the Bayes classifier is in fact a nonlinear projection of the feature vector to a single-dimensional statistic, in this paper, we develop a smooth nonlinear projection filter constrained to the estimated span of class conditional distributions as does the Bayes classifier. The nonlinear projection filter is designed in a reproducing kernel Hilbert space leading to an analytical solution both for the filter and the optimal threshold. The proposed approach is tested on typical detection problems, such as neural spike detection or automatic target detection in synthetic aperture radar (SAR) imagery. Results are compared with linear and kernel discriminant analysis, as well as classification algorithms such as support vector machine, AdaBoost and LogitBoost.**

***Index Terms*—Classification, nonlinear detection filter, nonlinear dimensionality reduction, nonlinear matched filter.**

I. Introduction

DETECTION of a target signal buried in noise is fundamental to a variety of signal processing applications including communications and biomedical engineering. Under additive white Gaussian noise and linear channel assumptions, the second-order statistics are sufficient to perfectly describe the signal characteristics, and the optimal solution under some symmetry conditions is achieved by the traditional *matched filter*, which is a widely used simple and efficient method. However, these assumptions are quite restrictive to model real life scenarios, and the limitations of the matched filter are already defined by the assumptions under which its statistical optimality can be proven. Specifically, since it relies on correlation, the matched filter becomes suboptimal in signal detection performance, if the noise distribution is skewed, non-Gaussian, or the waveform suffers a nonlinear channel distortion. Besides, using a template for the signal to be detected, the matched-filter method, by definition, assumes that the exact form of signal that is to be detected is known and time invariant. The matched filter has been the workhorse for the hypothesis testing problems due to its simplicity. On the other hand, if the optimality constraints are relaxed, such as the presence of non-Gaussian noise, it is a tedious task to derive the optimal detectors, and suboptimal detectors enter the scene. These easy-to-compute approximations typically consist of a nonlinear preprocessor coupled with the usual linear matched filter [1], [2]. Adding a whitening step to the preprocessing stage is a commonly used method that helps to relax the white noise assumption. As opposed to the traditional second-order-statistical methods, contemporary techniques emphasize exploiting higher order statistics; nonlinear kernel matched filters (nonlinear spectral matched filters and kernel linear discriminant analysis) have been studied, where the input data is first mapped into a potentially infinite-dimensional space called kernel induced feature space by a nonlinear function [3], [4].

The link between signal detection problem and the two-class classification problem is clear if one considers the delayed signal samples as the feature space of the classification problem. In this scenario, the instances of the signal to be detected and noise portions can be used for training the classifier. Furthermore, in some applications, the signal detection algorithm operates over a set of features derived from the signal rather than the raw signal itself [for example, the target detection application using synthetic aperture radar (SAR) images], which makes classification algorithms directly applicable to the signal detection problem. For this reason, we will present comparisons to some commonly used classification techniques in the experimental results section along with a brief discussion.

In this paper, we are motivated by the strength of the kernel methods. The connections between kernel methods and kernel density estimation as well as geometric insights on Bayes discriminants are exploited to design an analytically solvable optimal kernel nonlinear detection filter in the corresponding reproducing kernel Hilbert space (RKHS). Optimization involves determining the eigenvectors of the typical kernel matrix, which is of complexity $O(N^3)$ for N training samples and forward testing, as usual, has complexity $O(N)$. Training and testing complexity could be reduced by using a few largest eigenvectors and fast-Gauss transform-type approximations; however, these computational simplification issues remain outside the scope of this paper and will not be addressed.

II. Subspace Constrained Nonlinear Detector Design in the RKHS

The RKHS formalism is briefly described before the design procedure is explained. The RKHS theory states that the eigenfunctions $\{\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x}), \ldots\}$ of a kernel function $K : \Re^n \times$

Manuscript received December 27, 2007; revised January 08, 2009; accepted January 19, 2009. This work was supported by the National Science Foundation (NSF) under Grants ECS-0524835 and ECS-0622239.

U. Ozertem is with Yahoo! Labs, Santa Clara, CA 95054 USA (e-mail: umut@yahoo-inc.com).

D. Erdogmus is with the Electrical and Computer Engineering Department, Northeastern University, Boston, MA 02115 USA (e-mail: erdogmus@ece.neu.edu).

Digital Object Identifier 10.1109/TNN.2009.2021473

$\Re^n \rightarrow \Re$ that preferably (but not necessarily for all applications) satisfies the Mercer conditions for positive definiteness [5] form a basis for the Hilbert space of square integrable nonlinear functions under the norm induced by the kernel function in accordance with Green's equation[1] [6], [7]. Functions in this space can be represented as linear combinations of kernels as in the following form:

$$f(\cdot) = \sum_{i=1}^{N} \alpha_i K(\cdot, \mathbf{x}_i) \tag{1}$$

where N is the number of samples in the feature space. Let $g(.)$ be another function in this space with $g(\cdot) = \sum_{j=1}^{N'} \beta_j K(\cdot, \mathbf{x'}_j)$. The inner product of these functions is defined as

$$\langle f, g \rangle_K = \sum_{i=1}^{N} \sum_{j=1}^{N'} \alpha_i \beta_j K(\mathbf{x}_i, \mathbf{x'}_j). \tag{2}$$

Note that the inner product of the kernel function with an arbitrary function in RKHS gives that function itself, which is known as the reproducing property (note that the reproducing property replaces the sifting property of the Dirac delta function in the Euclidean space of square integrable functions according to the Euclidean norm). This can be written as

$$\langle K(\cdot, \mathbf{x}), f(\cdot) \rangle_K = \sum_{i=1}^{N} \alpha_i K(\mathbf{x}_i, \mathbf{x}) = f(\mathbf{x}). \tag{3}$$

A kernel is positive definite iff for any sample set $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ the kernel matrix $\mathbf{K}_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$ is positive definite; then one can write its eigendecomposition with all positive eigenvalues as

$$K(\mathbf{x}, \mathbf{x}') = \sum_{k=1}^{\infty} \lambda_k \varphi_k(\mathbf{x}) \varphi_k(\mathbf{x}') \tag{4}$$

where $\varphi_k(\mathbf{x})$ are the eigenfunctions and λ_k are the eigenvalues (ordered from largest to smallest by convention) of the kernel function. Hence, RKHS is potentially an infinite-dimensional space. In practice, it is approximated by its subspace with dimensionality equal to or less than the number of training samples, but this approximation is accurate because the eigenvalues of most kernels decay to zero fast, diminishing the significance of smaller eigenfunctions. This is due to the equivalent formulation of the RKHS that arises from (4). Functions such as those in (1) can also be represented by linear combinations of the eigenfunctions of the kernel as follows (letting $f_k = \langle f(.), \varphi_k(.) \rangle_K$):

$$f(\cdot) = \sum_{k=1}^{\infty} \lambda_k^{1/2} f_k \varphi_k(\cdot) \tag{5}$$

with the smoothness condition that $\sum_k f_k^2 < \infty$. The inner product of two functions, with g expanded similar to (5), becomes

$$\langle f, g \rangle_K = \sum_{k=1}^{\infty} f_k g_k. \tag{6}$$

[1]For a translation invariant kernel, find a differential operator that satisfies $D_K K(\mathbf{x} - \mathbf{y}) = \delta(\mathbf{x} - \mathbf{y})$; then $\|f\|_K^2 = \int \|D_K f(\mathbf{x})\|^2 d\mathbf{x} < \infty$.

Note that in the RKHS with reproducing kernel function $K(.,.)$, an arbitrary function $f(.)$ can be expressed both as a linear combination of eigenfunctions of this kernel as in (5) or as a linear combination of shifted versions of the kernel as in (1). In the representation presented in (5), we prefer to explicitly represent the eigenvalues of the kernel λ_k and the inner product coefficients f_k. For *low-pass* kernels (such as the typical Gaussian), higher indexed eigenfunctions have higher frequency content, thus exponentially decaying eigenvalues impose the desired smoothness and approximation conditions on the solutions obtained.

A. Bayesian Minimum Risk Detector in RKHS

In the two-class (detection) problem, given relative positive-valued risks r_i, *a priori* probabilities p_i, and class conditional probability densities $q_i(\mathbf{x})$, the total data probability distribution (mixture of two classes) is $q(\mathbf{x}) = p_1 q_1(\mathbf{x}) + p_2 q_2(\mathbf{x})$ and the optimal Bayes discriminant is

$$y_{\mathbf{r}} = r_1 p_1 q_1(\mathbf{x}) - r_2 p_2 q_2(\mathbf{x}) \tag{7}$$

with a decision threshold of zero. We can also define a risk-weighted data distribution (does not integrate to one anymore unless risks are selected such that $r_1 p_1 + r_2 p_2 = 1$) as follows: $q_{\mathbf{r}}(\mathbf{x}) = r_1 p_1 q_1(\mathbf{x}) + r_2 p_2 q_2(\mathbf{x})$; with this notation, we now have $q(\mathbf{x})$ denoted also by $q_{\mathbf{1}}(\mathbf{x})$, where $\mathbf{1}$ denotes a vector of ones with appropriate dimensionality throughout the paper. Note that the Bayes discriminant function is a nonlinear projection of the original feature vector $\mathbf{x}$ to a single-dimensional statistic.[2] With this projection, the output dimensionality may increase as the number of classes increases, however, specifically for the detection problem, the output dimensionality is one. A more important observation here is that the Bayes discriminant function as well as the total data probability distribution and the risk-weighted data distribution are all linear combinations of class conditional probability densities, where the coefficients are given in terms of relative risks and class priors. Hence, rather than the whole space, one can seek the Bayes discriminant function in the span of class conditional probability densities (in other words, one's discriminant function should be regularized or constructed to satisfy this geometric property for the respective class conditional distribution assumptions implicitly or explicitly assumed by the method).

In the RKHS, similar to (5) which is for an arbitrary function, the true class conditional densities are written explicitly in terms of the eigenvalues and eigenfunctions as

$$q_c(\mathbf{x}) = \sum_{k=1}^{\infty} \lambda_k^{1/2} \mu_k^c \varphi_k(\mathbf{x}) \tag{8}$$

where $\boldsymbol{\mu}^c$ is the expansion coefficient vector. Here we assume that the class conditional distributions are in the RKHS of the selected kernel function $K(.,.)$, hence they satisfy $\int \|D_K q_c(\mathbf{x})\|^2 d\mathbf{x} = \|q_c\|_K^2 = \iint q_c(\mathbf{x}) K(\mathbf{x}, \mathbf{x}') q_c(\mathbf{x}') d\mathbf{x} d\mathbf{x}' < \infty$. Note that these coefficients are the expectations of the eigenfunctions according to the class distribution, i.e.,

[2]One can show that optimal Bayes discriminant for a C-class problem projects the features to exactly $C - 1$ dimensions.

$\mu_k^c = \lambda_k^{-1/2} \int q_c(\mathbf{x})\varphi_k(\mathbf{x})d\mathbf{x} = \lambda_k^{-1/2} E_{q_c}[\varphi_k(\mathbf{x})]$. From this expansion, one easily obtains

$$\begin{aligned} y_{\mathbf{r}} &= r_1 p_1 \sum_{k=1}^{\infty} \lambda_k^{1/2} \mu_k^1 \varphi_k(\mathbf{x}) - r_2 p_2 \sum_{k=1}^{\infty} \lambda_k^{1/2} \mu_k^2 \varphi_k(\mathbf{x}) \\ &= \sum_{k=1}^{\infty} \left(r_1 p_1 \mu_k^1 - r_2 p_2 \mu_k^2 \right) \lambda_k^{1/2} \varphi_k(\mathbf{x}) \\ &= \sum_{k=1}^{\infty} \lambda_k^{1/2} v_{\mathbf{r}k} \varphi_k(\mathbf{x}) \end{aligned} \tag{9}$$

where $\mathbf{v_r}$ is the expansion coefficient vector for Bayes minimum risk discriminant for the risk vector $\mathbf{r} = [r_1, r_2]^T$. In matrix vector form, this is

$$\mathbf{v_r} = \mathbf{M}\mathbf{P}\mathbf{r}' \tag{10}$$

where $\mathbf{M}$ is the matrix of class coefficients, $\mathbf{P}$ is the matrix of class *a priori* probabilities, and $\mathbf{r}'$ is the discriminant risk vector

$$\mathbf{M} = [\boldsymbol{\mu}^1 \ \boldsymbol{\mu}^2] \quad \mathbf{P} = \begin{bmatrix} p_1 & 0 \\ 0 & p_2 \end{bmatrix} \quad \mathbf{r}' = \begin{bmatrix} r_1 \\ -r_2 \end{bmatrix}. \tag{11}$$

Similarly, one can show that the risk-weighted data distribution $q_{\mathbf{r}}(\mathbf{x})$ has the coefficient vector $\boldsymbol{\mu}_{\mathbf{r}} = \mathbf{M}\mathbf{P}\mathbf{r}$. The Bayes detector always has an optimal threshold of zero, however, since there is a one-to-one relationship between every threshold and risk ratio of the classes, in practice, one can select a projection assuming $\mathbf{r} = \mathbf{1}$ (minimum probability of error) and selecting nonzero thresholds to account for various risk levels. In general, (10) and the last argument show that one needs to search for optimal nonlinear projections for detection only in the linear span of the columns of $\mathbf{M}$ and orthogonal to $\boldsymbol{\mu}_{\mathbf{r}}$, possibly for $\mathbf{r} = \mathbf{1}$.

Note that the expression in (10) is an exact representation of the class conditional distribution for a given class label c. While theoretically infinite terms could be linearly combined to evaluate this quantity, in practice, an approximation involving finite number of terms must be employed. Next section describes a commonly employed approximation approach to this end.

B. Nystrom Sample Estimator for Eigenfunctions and Optimal Detector

In practice, analytical expressions of eigenfunctions for arbitrary kernel selections are not easily obtained (though not impossible for some). The Nystrom approximation [18] utilizes the available training data $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ from both classes in the detection problem in order to obtain a weighted approximation of the eigenfunctions in the RKHS in accordance with (1). Specifically, if we let $\varphi(\mathbf{x}) = [\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x}), \ldots, \varphi_N(\mathbf{x})]^T$, then

$$\boldsymbol{\varphi}(\mathbf{x}) \approx N^{1/2} \boldsymbol{\Lambda}^{-1} \boldsymbol{\Phi} \mathbf{k}(\mathbf{x}). \tag{12}$$

In (10), $\mathbf{k}(\mathbf{x}) = [K(\mathbf{x}, \mathbf{x}_1), \ldots, K(\mathbf{x}, \mathbf{x}_N)]^T$ and $\mathbf{K} = \boldsymbol{\Phi}^T \boldsymbol{\Lambda} \boldsymbol{\Phi}$ is the spectral decomposition of the Gram matrix with orthonormal eigenvector matrix (in its columns) $\boldsymbol{\Phi}^T$ and diagonal eigenvalue matrix $\boldsymbol{\Lambda}$.[3] The Gram matrix is composed of entries $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$. Note that this approximation corresponds to employing the following finite-rank approximation of the kernel function: $K(\mathbf{x}, \mathbf{x}') \approx \mathbf{k}(\mathbf{x})^T \mathbf{K}^{-1} \mathbf{k}(\mathbf{x}')$. Similarly, one can approximate the class distribution coefficients using $\mathbf{M} \approx N^{1/2} \boldsymbol{\Phi} [\mathbf{m}_1 \ \ \mathbf{m}_2] \operatorname{diag}(N_1^{-1}, N_2^{-1})$ where $\mathbf{m}_c$ is a membership vector, whose values are defined such that $\mathbf{m}_{ck} = 1$ if $c_k = c$, 0 otherwise. Here, N_c is the number of samples in class c (note that $N_c = \mathbf{1}^T \mathbf{m}_c$). From this point on, we use λ for the eigenvalues of $\mathbf{K}$. Substituting (12) in (8), we obtain

$$q_c(\mathbf{x}) \approx \sum_{i=1}^{N} \left[N^{1/2} \sum_{j=1}^{N} \lambda_j^{-1/2} \mu_j^c \Phi_{ji} \right] K(\mathbf{x}, \mathbf{x}_i). \tag{13}$$

Collecting the terms in brackets in a vector and denoting it by $\boldsymbol{\gamma}^c$, we can see that the coefficients for $q_1(\mathbf{x})$ in its approximation of the form in (13) will be $p_1\boldsymbol{\gamma}^1 + p_2\boldsymbol{\gamma}^2$. Letting $\boldsymbol{\Gamma} = [\boldsymbol{\gamma}^1 \ \ \boldsymbol{\gamma}^2]$, we also see that the coefficients for the Bayes discriminant $y_{\mathbf{r}}(\mathbf{x})$ will have coefficients $\boldsymbol{\gamma}_r = \boldsymbol{\Gamma}\mathbf{P}\mathbf{r}'$. Substituting the approximation for $\mathbf{M}$, this becomes $\boldsymbol{\gamma}_{\mathbf{r}} = \mathbf{K}^{-1}[\mathbf{m}_1 \ \ \mathbf{m}_2]\mathbf{r}'$.

Consider a projection of the form $y(\mathbf{x}) = N^{1/2} \boldsymbol{\Phi}^T \boldsymbol{\Lambda}^{-1} \mathbf{v}$. In accordance with the last argument of Section II-A, we propose selecting this projection, which is in the linear span of $\mathbf{M}$ and orthogonal to $q_{\mathbf{r}}(\mathbf{x})$. This selection corresponds to $\mathbf{v} = \mathbf{M}\mathbf{P}[\sqrt{p_2/p_1}, -\sqrt{p_1/p_2}]^T$. Substituting the approximation for $\mathbf{M}$, $y(\mathbf{x}) = \boldsymbol{\beta}^T \mathbf{k}(\mathbf{x})$, where

$$\boldsymbol{\beta} = \mathbf{K}^{-1} [\mathbf{m}_1 \quad \mathbf{m}_2] \begin{bmatrix} \sqrt{\dfrac{p_2}{p_1}} \\ -\sqrt{\dfrac{p_1}{p_2}} \end{bmatrix}. \tag{14}$$

Hence, we conclude that the proposed projection in (14), referred to as the RKHS Bayes discriminant (RKHS-BD), although selected to be orthogonal to $q_1(\mathbf{x})$, in fact, corresponds to selecting the risk vector as $\mathbf{r} = [\sqrt{p_2/p_1}, \sqrt{p_1/p_2}]^T$. This modification in the assumed risk is due to the approximation of $\mathbf{M}$ interacting with the approximation of the eigenfunctions in (12). In summary, given labeled training data $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$, the proposed subspace constrained nonlinear detector (SCND) has the form $y(\mathbf{x}) = \boldsymbol{\beta}^T \mathbf{k}(\mathbf{x})$, where $\boldsymbol{\beta} = \mathbf{K}^{-1}[\mathbf{m}_1 \ \ \mathbf{m}_2]\mathbf{r}'$ for a desired risk vector $\mathbf{r}$ (which must be selected taking the approximation into account).

C. Selecting the Kernel Function

An important practical consideration is the selection of a suitable kernel.[4] This is a common issue in all kernel-based methods, since the quality of the results depends on a suitable kernel size selection. Typically, the problem of finding the optimal kernel is simplified by constraining the kernel function into a parametric family, and trying to optimize the parameters of the function by the quality of the solutions obtained. This method adds a significant amount of unnecessary computational load, and the kernel functions optimized by this approach also depend on the quality measure used here while evaluating the results. Often the kernel size is varied and the one that gives the best cross-validation solution is selected. This is a computationally expensive procedure.

[3] Note that $\boldsymbol{\Lambda}_{ii} = N\lambda_i$.

[4] Note that one could also explicitly design the kernel by selecting a desired set of eigenvalue–function pairs.

An alternative approach to select the suitable kernel function is to exploit the connection to kernel density estimation. Nonparametric density estimation is a well-researched field and there is a wide literature about the selection of suitable kernel function, including a wide range of methods that range from heuristics to principled approaches such as maximum likelihood [8]. A straightforward method is to use a circular Gaussian kernel, with a width parameter (variance) determined utilizing the Silverman's rule of thumb given in [9]

$$\sigma^2 = \frac{1}{n}\,\mathrm{tr}(\mathbf{\Sigma_x})\left(\frac{4}{(2n+1)N}\right)^{2/(n+4)} \tag{15}$$

where n is the dimensionality of the data $\mathbf{x}$, N is the number of samples, and $\mathbf{\Sigma_x}$ is the sample covariance of the training set. Further improvements can be achieved by utilizing a variable kernel size for each class or even each data point itself, as well as using anisotropic kernels or employing kernel optimization methods. Variable size kernel density estimation is known to have a better outlier performance, where the kernels are selected as a function of the likelihood of particular samples being outliers. For example, one can use the median of K nearest neighbor distances of each sample with spherical Gaussian kernel for variable size kernel density estimate (KDE) and scale the nearest neighbor distances with a global scaling factor optimized using maximum likelihood to obtain kernel widths. Using the sample covariance of K nearest neighbor instead of their distances leads to an anisotropic Gaussian kernel function for each data sample. For the variable size KDEs, the required computational cost increases as the quality of the density estimate improves.

In our illustrative experiments, we used the Silverman's rule given in (15) as a simple and computationally inexpensive selection. The results we present could be improved by incorporating computationally more expensive kernel optimization techniques or using variable size kernel estimates. However, for a fair comparison, we will keep using the fixed size Gaussian kernel function here to be able to keep the computational requirement of our algorithm at the same level with the ones it is being compared with.

D. Summary of Implementation

Table I summarizes the implementation of the proposed detector design in RKHS. A MATLAB implementation of the algorithm will also be made available at the authors' web pages [19].

TABLE I
SUMMARY OF THE RKHS-BD IMPLEMENTATION

Given training data $\mathbf{x}_i \in \Re^n$, with corresponding class labels $c_i \in \{1,2\}$, $i=1,\ldots,N$
Training Phase:
- Select the kernel bandwidth (for example as in (15)). The kernel may also be selected as a data-dependent manner.
- Construct the kernel matrix $\mathbf{K}$, where $\mathbf{K}_{ij} = K(\mathbf{x}_i - \mathbf{x}_j)$ and decompose into eigenvalues-vectors such that $\mathbf{K} = \mathbf{\Phi}_\mathbf{x}^T \mathbf{\Lambda} \mathbf{\Phi}_\mathbf{x}$.
- Build the membership vectors such that $\mathbf{m}_{ck}=1$ if $c_i=c$, 0 otherwise.
- Evaluate $\boldsymbol{\beta}=\mathbf{K}^{-1}[\mathbf{m}_1\ \mathbf{m}_2]\mathbf{r}$. If the relative risk values are not provided, use $\mathbf{r}' = [\sqrt{p_2/p_1}, -\sqrt{p_1/p_2}]^T$ to achieve minimum error, which assumes equal risk values for misdetection and false alarm.
Testing (Operational) Phase:
- For each test sample, evaluate $y(\mathbf{x})=\boldsymbol{\beta}^T\mathbf{k}(\mathbf{x})$.
- Decide 1 if $y(\mathbf{x})>0$, decide 2 otherwise.

III. Experimental Results

In this section, we will provide experiments to evaluate the performance of the proposed RKHS-BD and provide comparisons with other methods. To relax the white noise assumption, the linear matched filter is usually coupled with a whitening step at the preprocessing stage. Since the utilization of this prewhitening is a commonly used technique, which increases the matched-filter performance by enforcing the covariance circular symmetry condition on noise, we will couple the matched filter with a prewhitening step in our comparisons.

We start with a comparison of RKHS-BD with the KDE-based Bayes classifier, and illustrate the underlying class conditional densities for these two methods. The detection problem can also be regarded as a binary classification problem, and there are many linear and nonlinear topologies to solve these classification problems in the literature. Therefore, we also provide comparisons of the RKHS-BD with (Fisher) linear discriminant analysis (LDA) [15], which is a widely used linear classifier, and its nonlinear extension kernel (Fisher) LDA [14], [16], as well as the KDE-based approximate Bayes classifier, support vector machine (SVM), AdaBoost, and Logitboost.

A. RKHS-BD Versus KDE-Based Bayes Classifier

We compare RKHS-BD with the KDE-based Bayes detector on a simple illustrative example. For illustrative purposes, we employ a 2-D toy data set for this example, and present the underlying density estimates for each of these methods. The density estimate for the usual KDE-based Bayes classifier is shown in Fig. 1(a). The underlying density estimate that RKHS-BD utilizes is a weighted KDE, where the selection of the weights is inherently determined by the method such that the class conditional probability densities become approximately uniform in the support of the data, and they sharply decay to zero outside the support of the data.

This behavior can be interpreted as increasing the significance of the samples that are close to the boundary while building the classifier—similar to what largest margin classifiers do. The underlying class conditional probability densities for this data set are presented in Fig. 1. Since the classes are selected to be well separated for better illustration, for this trivial data set, we do not present the classifier performances. Performance comparisons for KDE Bayes classifier will be included in the following experiments performed on real data.

B. Neural Spike Detection

Automatic detection of neural spikes is an important first step in the study of biological neural systems and their engineering applications, such as brain machine interfaces [10]. Currently, all cortical brain machine interface applications rely on an

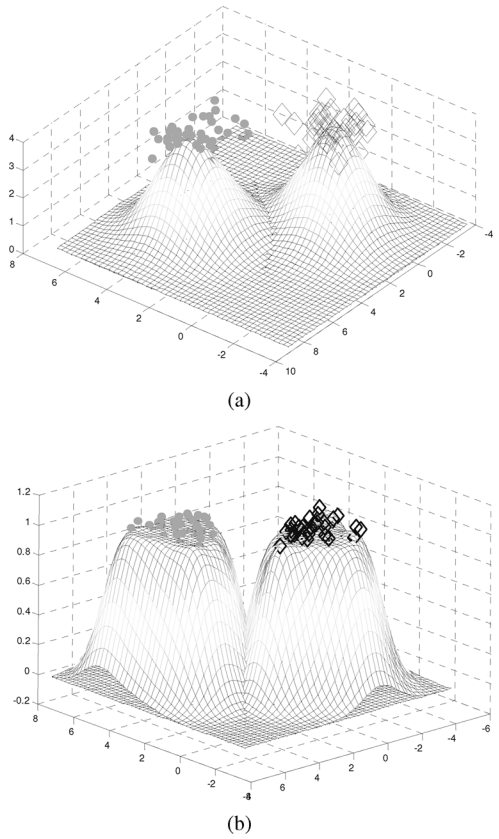

Fig. 1. Class conditional densities are shown for (a) KDE-based Bayes classifier and (b) RKHS-BD, along with the corresponding data for each class, denoted with • and ◇ signs.

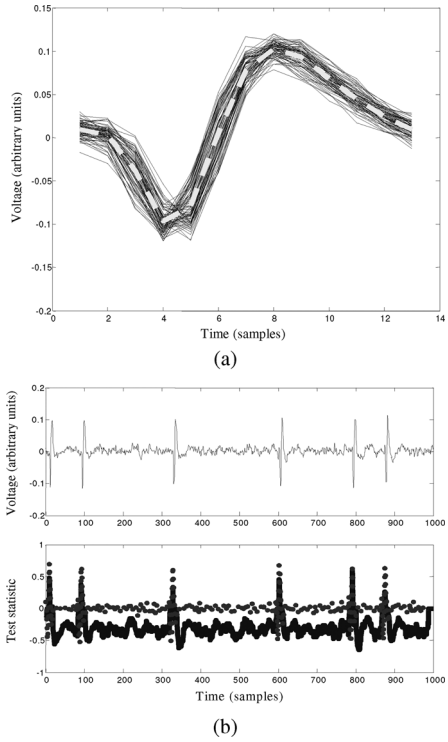

Fig. 2. (a) Spike template obtained by averaging according to common convention (thick–dashed line) with actual sample spikes (thin–solid line). (b) Sample microelectrode signal (top), the matched filter output (bottom–dashed), and RKHS-BD output (bottom–solid).

accurate detection of neural spike timing as neural activity is characterized by various statistics of the spike distributions. The majority of the spike detection methods that have been currently used can be categorized in three main groups: simple thresholding, energy-based methods, and template matching. Although they are computationally effective, the first two methods that are based on thresholding either the integrated magnitude square of the signal or the raw signal itself are very primitive and only applicable in high signal-to-noise ratio (SNR) cases. Template matching, on the other hand, provides a more powerful tool for low SNR scenarios by considering linear correlations between the data and the predefined signal template. These low SNR neural signals are often collected using microelectrode arrays in neural tissue.

In this experiment, the RKHS-BD is compared with the conventional correlation-based matched-filter detector coupled with a prewhitening step. Outputs of the proposed system and the linear matched filter are presented in Fig. 2 for online mode of operation. For the given microelectrode recording and the time stamps of the spike times, neural spikes with a length of 13 samples have been collected from the data. First, a spike template of 13 samples is generated for the matched filter. Using uniform sampling, 200 of these spikes have collected and averaged over their aligned time index to obtain the spike template for the matched filter. The same set of neural spikes is used for identifying the weights of the proposed system in conjunction with a set of equal number of nonspike samples. This corresponds to assuming that the existence and nonexistence of a spike are equal at 0.5, which is surely a suboptimal assumption for a typical spike train signal. The relative importance of missing and false alarms could also have been introduced manually by assigning appropriate risk values in place of class priors. For this reason, rather than providing the detection and false alarm rates for specific risk values we present the performance as a receiver operating characteristic (ROC) curve. Along the ROC curve, r_1/r_2 ratio varies from zero to infinity, which shows the results for all possible risk function pairs. For the testing phase, both for the matched filter and the RKHS-BD, the time series signal is transformed into a 13-dimensional input stream using a tapped delay line, and detection results for each 13-dimensional input of both systems are associated with the time index of the centering sample, accordingly.

The resulting spike template is presented in Fig. 2(a) (the dashed line) along with the individual spikes that have been used here to generate the template. In some cases, neural spikes may show significant differences in time due to the nature they have been created, which makes the template matching methods vulnerable. However, observing Fig. 2(a), one can conclude that the neural spike pattern used in this experiment has a stationary nature. Fig. 2(b) shows a comparison for the output of the matched filter and the RKHS-BD along with the original input signal. For the outputs in this figure, both algorithms have neither a miss nor a false alarm; but, still the RKHS-BD generates a relatively better separation.

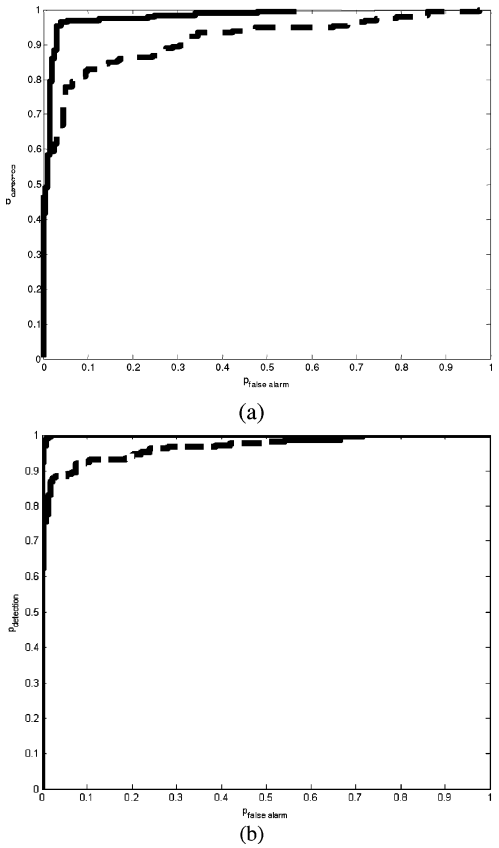

Fig. 3. ROC curves for the linear matched filter (dashed) and SCND (solid): (a) without the prewhitening step, and (b) after a prewhitening step.

The batch mode results for the proposed nonlinear matched filter are presented in Fig. 3. Fig. 3(a) shows the RKHS-BD results along with the corresponding results obtained for the matched filter. Provided the prewhitening step, both the matched filter and the RKHS-BD perform better, which is presented in Fig. 3(b). Providing higher detection rate and lower false alarm rates, RKHS-BD demonstrates superior performance, either with or without the prewhitening step.

At this point, one may argue that the prewhitening step should not change the results of RKHS-BD, because it inherently should be able to capture the linear and nonlinear characteristics of the data regardless of any scaling and rotation with respect to the eigenvectors. In general, this argument is true. However, we used spherically symmetrical kernels in our experiments as given in (15) to have a lower computational cost, and the prewhitening helps to normalize the data along its eigenvectors, which actually makes the results using a spherical kernel more accurate. Possibly, better results could still be achieved using anisotropic kernels that will inherently handle the whitening, or data-dependent variable anisotropic kernels that exploit an estimate of the local eigenspread of the data. Comparing the results in Fig. 3(a) and (b), one can also see that RKHS-BD without prewhitening [Fig. 3(a), solid line] performs better than linear matched filter cascaded with a prewhitening step [Fig. 3(b), dashed line]. Since RKHS-BD and kernel LDA produced very similar results for this data set, we do not include kernel LDA in this comparison. We will present comparisons with this method in Section III-C.

C. Automatic Target Detection Using Satellite Images

Automatic target detection results for SAR images from MSTAR imagery [11] will be presented in this section. For the experiments, BMP-2 sn-9596 targets with 15° depression angle have been used, and the background samples have been obtained from the available clutter image samples with the same depression angle. Fig. 4 shows some examples of the targets to be detected and the background clutter images used in this experiment. The size of the images of BMP-2 targets is 138×139. Therefore, the size background samples obtained from the public clutter data have been arranged to the same size as that of the target images (through random sampling). To demonstrate the relative size of the target images, two 138×139 rectangles are marked on the background image in Fig. 3(b), and two examples of background clutter are shown.

Features used for classification here are simple Gabor coefficients. A total of 24 Gabor filters have been used for this purpose; six equally spaced orientations in the spatial domain and four wavelet scales. After obtaining the 24-dimensional feature vectors for each sample target or background image, the data set has been partitioned into training and testing sets. Here, we used 240 target and 240 background images in total, and the resulting data after the Gabor filter feature extraction step consists of 480 24-dimensional samples. For all the methods compared in this section, training set consists of 80 samples from each class, and the testing set contains 160 samples from each class.

Associated results for the testing set are presented in Fig. 4(c) with ROC curves. In this experiment, we compare four methods: RKHS-BD, LDA, kernel linear discriminant analysis, and KDE-based Bayes classifier. RKHS-BD demonstrates superior performance in terms of probability of detection and probability of false alarm for a wide range of risk function ratios. To assure a fair comparison, the same kernel function is used for RKHS-BD, KLDA, and KDE-Bayes classifiers.[5]

D. Sonar Mine Detection

The sonar signal data set consists of sonar signals bounced off a metal cylinder and those bounced off a roughly cylindrical rock. Each sonar reflection is represented by a 60-dimensional vector, and each dimension represents the energy that falls within a particular energy band, integrated over a certain period of time. There are 208 60-dimensional sonar signals in this data set; 111 of them belong to mines and 97 of them are obtained by bouncing sonar signals from rocks under similar conditions. The sonar signals are collected from a variety of different aspect angles. This data set was originally used by Gorman and Sejnowski in their study of sonar signal classification [17].

As in the previous experiment, in this experiment, we compare four different methods: RKHS-BD, LDA, KLDA, and

[5]Note that the aim of this experiment is not to compare results with earlier work on SAR-ATR literature, since more informative and specialized features or using more training samples will certainly improve the final results. This experimental setup aims to demonstrate the favorable performance of the proposed method over alternatives for a given set of feature vectors.

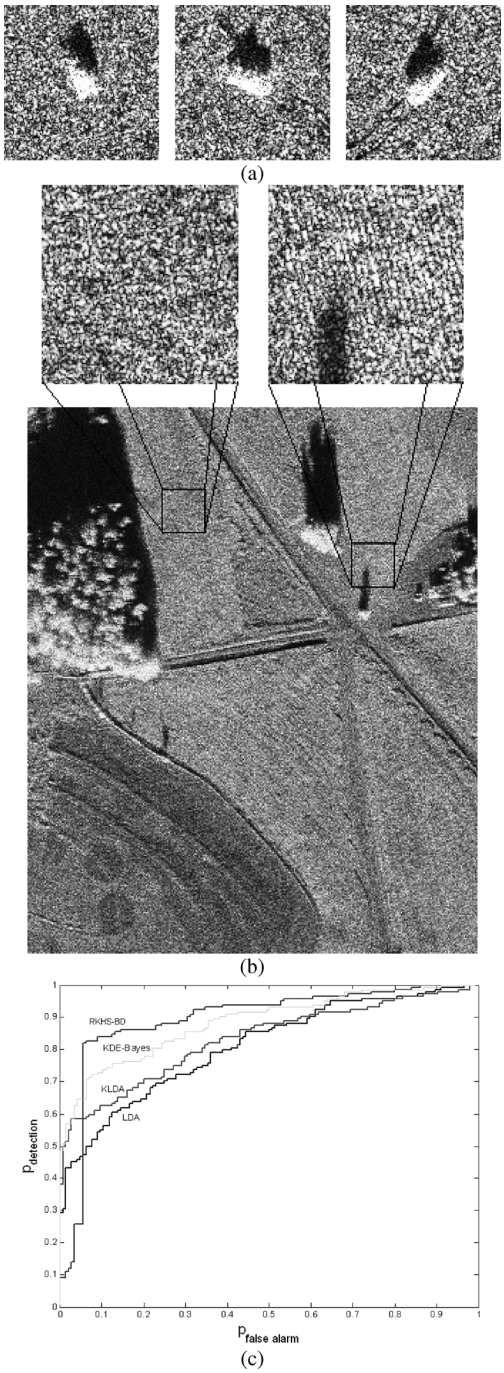

Fig. 4. (a) BMP-2 sn-9596 examples. (b) Sample background image selected from the public clutter data. (c) ROC curves for RKHS-BD, KLDA, LDA, and KDE-Bayes classifiers.

KDE-based Bayes classifier. For all these methods, the same training and testing data is used, and the training set contains 69 independently selected sonar signals; 37 mines and 32 rocks. The remaining 139 sonar signals are used in the training phase. As in the previous experiment, RKHS-BD, KLDA, and KDE-Bayes classifiers all use the same kernel function. The performance is presented as ROC curves and RKHS-BD shows superior performance.

All performance comparisons presented so far include only dimensionality-reduction-based classification algorithms. Similar to our approach, for dimensionality-reduction-based approaches, it is very easy to adjust the tradeoff between the misdetection and false alarm by shifting the decision boundary (that is the threshold for 1-D subspace projections) towards either class. Therefore, all performance comparisons with dimensionality-reduction-based methods have been presented using ROC curves that summarize the performance for all possible relative risk levels. On the other hand, in general, this may not be the case for a classification algorithm, and one may have to rerun the algorithm many times to obtain results for different risk ratios.

Aside from comparisons with dimensionality-reduction-based classification methods, we will also provide comparisons with SVM [20], AdaBoost [21], and LogitBoost [22]. SVM is based on the idea of constructing a separating hyperplane in the feature space of data samples such that the margin between two data sets is maximized. Although the original proposition of Vapnik was a linear classifier, later the idea of using a nonlinear kernel function to transform the problem into RKHS is used to build a nonlinear classifier. Hence, SVM that is built in the same RKHS is directly comparable to our approach. Another common approach in classification is boosting [23]. Boosting is based on the idea of building a strong learner (for example, a classifier) from a set of weak learners. The main variation between boosting algorithms in the literature is their method of weighting training data points and hypotheses, and two most commonly used boosting approaches are AdaBoost and LogitBoost.

To compare SVM, LogitBoost, and AdaBoost to the proposed approach, we use the sonar mine detection data set. One third of the data set has been independently selected for training and Table II shows the average classification error for 50 Monte Carlo simulations along with average computation times. For AdaBoost and LogitBoost, we will present results with different number of iterations to show the tradeoff between performance and computation time. For RKHS-BD, we used Silverman's rule (15) to select the kernel bandwidth. With the same kernel bandwidth, SVM tends to assign most of the test samples into one class, which might be due to the small sample size. Therefore, we experimented with different kernel sizes in SVM and presented the best result as well. In the comparisons, we use a publicly available MATLAB SVM toolbox [24].

IV. Conclusion

If the class conditional probability densities and the associated risks are known, the optimal results are given by the Bayes classifier. Specifically, for a two-class scenario, linear matched filter provides the optimal results under certain symmetry conditions; however, in the presence of non-Gaussian noise or nonlinear channel distortion, traditional matched filter and other linear methods easily lose optimality. For these cases, coupling the matched filter with a nonlinear preprocessor is a computationally efficient method to obtain suboptimal processors that will outperform the linear matched filter. However, in most of the cases, these methods are not reliable at approximating the results obtained for an optimal classifier.

We approach the problem in a different way. Regarding the Bayes discriminant function as a nonlinear mapping from the

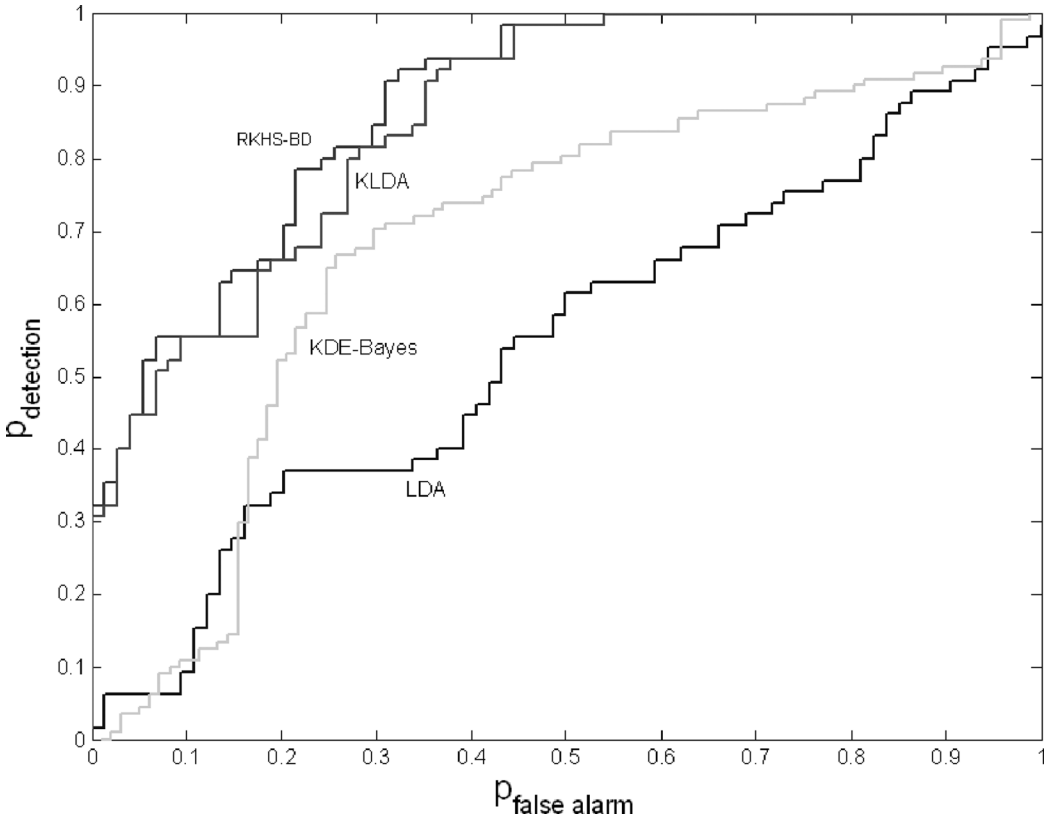

Fig. 5. ROC curves for RKHS-BD, KLDA, LDA, and KDE-Bayes classifiers on the sonar mine detection data.

TABLE II
AVERAGE ERROR AND COMPUTATION TIMES

	Average Error (±std)	Time
RKHS-BD ($\sigma = 0.14$)	**0.2173±0.0323**	**0.0935 s**
AdaBoost (100 iter)	0.2378±0.0355	6.0455 s
AdaBoost (20 iter)	0.2871±0.0357	1.4392 s
LogitBoost (100 iter)	0.2307±0.0353	6.0699 s
LogitBoost (20 iter)	0.2528±0.0331	1.4467 s
SVM ($\sigma = 0.14$)	0.4620±0.0132	1.4576 s
SVM ($\sigma = 0.65$)	0.2112±0.0382	1.3534 s

original feature space to one dimension, we seek for the solution in the subspace spanned by the class conditional probability densities, and we achieve an analytical solution for the approximate Bayes decision boundary, and computationally expensive optimization procedures are avoided. We tested our approach on typical detection problems on real data. Even for a computationally inexpensive suboptimal kernel selection, RKHS-BD provided superior performance as opposed to linear matched filter, LDA, and KLDA.

RKHS-BD produced only slightly better results as compared to KLDA. However, note that KLDA requires an optimization step and is known to be numerically unstable if there are not enough training samples. On the other hand, the proposed method achieves the similar performance level with an analytical solution—both for the class discriminant functions and the optimal threshold. Similar conclusions can be made for the comparisons with SVM, AdaBoost, and LogitBoost. SVM and, given enough number of iterations, boosting algorithms we used in the comparisons produced very similar results with RKHS-BD. However, note that the analytical solution achieved by our proposition not only eliminates the numerical stability issues, but also yields faster computation times, since an iterative optimization scheme is not required.

Constraining the search space for nonlinear designs has potential uses in regularizing classifiers, and classification problems in very high-dimensional spaces with a few number of training samples. Defining the subspace that the Bayes discriminant function lies in, one can see that the behavior of the projection function that lies in the orthogonal space of class conditional probability densities does not affect the performance. Hence, the required regularization term can be sought in the subspace of class conditional densities. In the case of many dimensions with too few examples, which is typical in biomedical problems, the generalization performance in the high-dimensional space is an important problem, and constraining the space as we propose above, one can achieve a better generalization performance.

Computational complexity of the proposed algorithm is $O(N^3)$ if all eigenvectors of the kernel are utilized, where N is the number of training samples, in its raw form, since inversion of the kernel matrix is required. In practice, a few significant eigenvectors of the kernel matrix could be selected and extracted sequentially. An efficient subspace approximation to the inverse kernel matrix would reduce training complexity easily. Furthermore, in testing, N kernel evaluations are utilized for each novel test data. Techniques based on truncated polynomial expansions such as the fast Gauss transform and its variants (for the particular kernel) could be implemented to

realize this conceptually trivial computational simplification methodology. In this paper, we did not consider these issues, since the corresponding solutions are well known in the kernel machine literature and they are directly applicable.

ACKNOWLEDGMENT

The authors would like to thank J. C. Sanchez for providing neural spike data set.

REFERENCES

[1] F. Chapeau-Blondeau, "Nonlinear test statistic to improve signal detection in non-Gaussian noise," *IEEE Signal Process. Lett.*, vol. 7, no. 7, pp. 205–207, Jul. 2000.
[2] S. Kay, *Fundamentals of Statistical Signal Processing: Detection Theory*. Englewood Cliffs, NJ: Prentice-Hall, 1998.
[3] D. Erdogmus, R. Agrawal, and J. C. Principe, "A mutual information extension to the matched filter," *Signal Process.*, vol. 85, no. 5, pp. 927–935, 2005.
[4] H. Kwon and N. M. Nasrabadi, "Hyperspectral target detection using kernel spectral matched filter," in *Proc. Comput. Vis. Pattern Recognit.*, 2004, vol. 8, p. 127.
[5] J. Mercer, "Functions of positive and negative type, and their connection with the theory of integral equations," *Trans. London Philosoph. Soc. A*, vol. 209, pp. 415–446, 1909.
[6] G. Wahba, *Spline Models for Observational Data*. Philadelphia, PA: SIAM, 1990.
[7] H. Weinert, Ed., *Reproducing Kernel Hilbert Spaces: Applications in Statistical Signal Processing*. Stroudsburg, PA: Hutchinson Ross, 1982.
[8] R. P. W. Duin, "On the choice of smoothing parameters for Parzen estimators of probability density functions," *IEEE Trans. Comput.*, vol. C-25, no. 11, pp. 1175–1179, Nov. 1976.
[9] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*. London, U.K.: Chapman & Hall, 1986.
[10] I. Obeid and P. Wolf, "Evaluation of spike-detection algorithms for a brain-machine interface application," *IEEE Trans. Biomed. Eng.*, vol. 51, no. 6, pp. 905–911, Jun. 2004.
[11] Air Force Research Laboratory, MSTAR Data, [Online]. Available: http://www.mbvlab.wpafb.af.mil/public/sdms/datasets/mstar/overview.htm
[12] C. Fowlkes, S. Belongie, F. Chung, and J. Malik, "Spectral grouping using the Nystrom method," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 298–305, Feb. 2004.
[13] G. H. Golub and C. F. van Loan, *Matrix Computations*, 3rd ed. Baltimore, MD: The Johns Hopkins Univ. Press, 1996.
[14] B. Scholkopf and A. J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. Cambridge, MA: MIT Press, 2002.
[15] K. Fukunaga, *Introduction to Statistical Pattern Recognition*. New York: Academic, 1990.
[16] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Comput.*, vol. 12, pp. 2385–2404, 2000.
[17] R. P. Gorman and T. J. Sejnowski, "Analysis of hidden units in a layered network trained to classify sonar targets," *Neural Netw.*, vol. 1, pp. 75–89, 1988.
[18] J. Suykens, J. de Brabanter, T. van Gestel, and J. Vandewalle, *Least Squares Support Vector Machines*. London, U.K.: World Scientific, 2002.
[19] A Matlab Implementation Will be Made Available at the Author's Web Pages, [Online]. Available: http://www.csee.ogi.edu/~ozertemu and http://www.csee.ogi.edu/~deniz
[20] V. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.
[21] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," in *Proc. Int. Conf. Mach. Learn.*, 1996, pp. 148–156.
[22] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting," *Ann. Statist.*, vol. 28, no. 2, pp. 337–407, 2000.
[23] M. Kearns, "Thoughts on hypothesis boosting," 1988, unpublished.
[24] G. C. Cawley, MATLAB Support Vector Machines Toolbox, [Online]. Available: http://theoval.sys.uea.ac.uk/~gcc/svm/toolbox

Umut Ozertem (S'06–M'09) received the B.S. degree in electrical engineering from the Middle East Technical University, Ankara, Turkey, in 2003 and the M.S. and Ph.D. degrees in electrical engineering from the Oregon Health and Science University, Portland, in 2006 and 2008, respectively.

Prior to joining Yahoo! Labs, he worked at Intel Corporation between January and July 2007. His research interests include nonparametric machine learning, adaptive and statistical signal processing, information theory and its applications to signal processing, and adaptive learning algorithms.

Dr. Ozertem serves as a Reviewer for Elsevier's *Signal Processing*, *Neurocomputing*, *Pattern Recognition Letters*, the IEEE TRANSACTIONS ON IMAGE PROCESSING, the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS, the IEEE SIGNAL PROCESSING LETTERS, the IEEE TRANSACTIONS ON BIOMEDICAL ENGINEERING, and numerous conferences.

Deniz Erdogmus (S'95–M'02–SM'07) received the B.S. degrees in electrical engineering and in mathematics and the M.S. degree in electrical engineering from the Middle East Technical University, Ankara, Turkey, in 1997 and 1999, respectively, and the Ph.D. degree in electrical and computer engineering from the University of Florida, Gainesville, in 2002.

He was a Postdoctoral Research Associate at the University of Florida until 2004. Prior to joining the Northeastern University faculty in 2008, where he is currently an Assistant Professor of Electrical and Computer Engineering, he held an Assistant Professor position jointly at the Computer Science and Electrical Engineering (CSEE) and Biomedical Engineering (BME) Departments of the Oregon Health and Science University. His expertise is in information theoretic and nonparametric machine learning and adaptive signal processing, specifically focusing on cognitive signal processing including brain interfaces and technologies that collaboratively improve human performance in a variety of tasks.

Dr. Erdogmus has been serving as an Associate Editor of the IEEE TRANSACTIONS ON SIGNAL PROCESSING, the IEEE TRANSACTIONS ON NEURAL NETWORKS, the IEEE SIGNAL PROCESSING LETTERS, Elsevier's *Neurocomputing*, *Neural Processing Letters*, and Hindawi's *Computational Intelligence and Neuroscience*. He is a member of the IEEE Signal Processing Society (IEEE-SPS) Machine Learning for Signal Processing Technical Committee. He is a member of Tau Beta Pi (TBP) and Eta Kappa Nu (HKN).