

Nonlinear Multiclass Discriminant Analysis

Junshui Ma, José L. Sancho-Gómez, and Stanley C. Ahalt, *Member, IEEE*

Abstract—An alternative nonlinear multiclass discriminant algorithm is presented. This algorithm is based on the use of kernel functions and is designed to optimize a general linear discriminant analysis criterion based on scatter matrices. By reformulating these matrices in a specific form, a straightforward derivation allows the kernel function to be introduced in a simple and direct way. Moreover, we propose a method to determine the value of the regularization parameter τ , based on this derivation.

Index Terms—Discriminant analysis, feature extraction, kernel method.

I. INTRODUCTION

MANY discriminant analysis methods have been developed over the last four decades, and perhaps the most well established are those generally referred to as the Fisher's discriminant analysis (FDA), and its extension, the linear discriminant analysis (LDA) [1]–[3]. However, because FDA and LDA are intrinsically linear, their performance is limited when applied to problems with nonlinear classification boundaries.

In order to overcome the limitation associated with conventional FDA and LDA, several nonlinear discriminant algorithms have been proposed in recent years. Most of these algorithms have employed a technique referred to as the *kernel trick* [4]–[6]. An efficient two-class nonlinear classification technique based on FDA was first derived by Mika *et al.* [4] and was then extended to multiclass problems [6]. Subsequently, another form of a multiclass nonlinear discriminant algorithm based on a variant of the general LDA criteria was developed [5]. However, the techniques employed to derive these nonlinear algorithms are not straightforward, and the forms of their resultant algorithms are relatively complex.

In this letter, we present an alternative algorithm for nonlinear, multiclass discriminant analysis. This letter has two primary contributions with respect to previous results. First, our algorithm is a simple and direct extension of a general LDA criterion, and its derivation is straightforward. This simplicity allows us to easily extend the formulation in this letter to new algorithms [11]. Second, we provide a simple method that can be used to determine the value of a critical parameter τ , which

Manuscript received July 30, 2001; revised October 9, 2002. This work was supported in part by the HPCMO PET program and by the Spanish Ministry of Science and Technology. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Elias S. Manolakos.

J. Ma was with the Department of Electrical Engineering, The Ohio State University, Columbus, OH 43210 USA. He is now with Aureon Biosciences Corporation, Yonkers, NY 10701 USA.

J. L. Sancho-Gómez is with the Department of Signal Theory and Communications, Universidad Carlos III de Madrid, 28911 Leganés-Madrid, Spain.

S. C. Ahalt is with the Department of Electrical Engineering, The Ohio State University, Columbus, OH 43210 USA.

Digital Object Identifier 10.1109/LSP.2003.813680

TABLE I
NOMENCLATURE USED IN THIS PAPER

\mathcal{I} :	d -dimensional original space
\mathcal{L} :	ℓ -dimensional space after a linear transformation from \mathcal{I}
\mathcal{H} :	h -dimensional feature space
\mathcal{L}' :	ℓ' -dimensional space after a linear transformation from \mathcal{H}
$\chi_j = \{\mathbf{x}_1^j, \dots, \mathbf{x}_{N_j}^j\}$,	$j = 1, \dots, L$: set of vectors in \mathcal{I} of class j
$\chi = \cup_{j=1}^L \chi_j$:	dataset in \mathcal{I}
\mathbf{X} :	dataset matrix in \mathcal{I} (column vectors \mathbf{x}_i^j)
$\Phi_j = \{\Phi(\mathbf{x}_1^j), \dots, \Phi(\mathbf{x}_{N_j}^j)\}$,	$j = 1, \dots, L$: vectors of class j
	in \mathcal{H}
Θ :	data set matrix in \mathcal{H} (column vectors $\Phi(\mathbf{x}_i^j)$)
$\mathbf{S}_b^x, \mathbf{S}_w^x$:	between and within scatter matrices in \mathcal{I}
$\mathbf{S}_b^y, \mathbf{S}_w^y$:	between and within scatter matrices in \mathcal{L}
$\mathbf{S}_b^{\Phi(x)}, \mathbf{S}_w^{\Phi(x)}$:	between and within scatter matrices in \mathcal{H}
$\mathbf{S}_b^{y'}, \mathbf{S}_w^{y'}$:	between and within scatter matrices in \mathcal{L}'
\mathbf{I}_n :	Identity matrix of $n \times n$ dimension
$\mathbf{1}_{n \times m}$:	a $n \times m$ matrix with elements equal to one

provides numerically stability and controls the complexity of the resultant algorithm [11].

Table I provides the nomenclature used in this letter.

II. LDA

Suppose, in space \mathcal{I} , we have a set of data $\chi = \cup_{j=1}^L \chi_j$, where $\chi_j = \{\mathbf{x}_1^j, \mathbf{x}_2^j, \dots, \mathbf{x}_{N_j}^j\}$, $j = 1, \dots, L$, is the subset of vectors from class j . For a general L -class problem, the input data can be projected into an ℓ , $\ell \leq L - 1$ -dimensional space following a linear transformation of $W : \mathcal{I} \mapsto \mathcal{L}$

$$\mathbf{y} = \mathbf{W}^T \mathbf{x} \quad (1)$$

where the projection directions are determined by the column vectors of the transformation matrix \mathbf{W} [2]. The objective of this transformation is to maximize the class separability with regard to a chosen separability criterion. One of the most widely used criteria is

$$\begin{aligned} J(\mathbf{W}) &= \text{Tr} \{ (\mathbf{S}_w^y)^{-1} \mathbf{S}_b^y \} \\ &= \text{Tr} \{ (\mathbf{W}^T \mathbf{S}_w^x \mathbf{W})^{-1} (\mathbf{W}^T \mathbf{S}_b^x \mathbf{W}) \} \end{aligned} \quad (2)$$

where the within-class matrix \mathbf{S}_w^x and between-class scatter matrix \mathbf{S}_b^x are given respectively by

$$\mathbf{S}_w^x = \sum_{i=1}^L \frac{N_i}{N} \sum_{k=1}^{N_i} (\mathbf{x}_k^i - \mathbf{m}_i)(\mathbf{x}_k^i - \mathbf{m}_i)^T \quad (3)$$

$$\mathbf{S}_b^x = \sum_{i=1}^L \frac{N_i}{N} (\mathbf{m}_i - \mathbf{m})(\mathbf{m}_i - \mathbf{m})^T. \quad (4)$$

Finding a transformation matrix \mathbf{W} to maximize $J(\mathbf{W})$ is straightforward [2]. The optimal matrix \mathbf{W} is indeed formed by the ℓ eigenvectors of matrix $(\mathbf{S}_w^x)^{-1}\mathbf{S}_b^x$ corresponding to its ℓ largest eigenvalues.

III. GDA

The linear discriminant analysis for multiclass problems described above can be combined with the *kernel trick* [7] to obtain a nonlinear multiclass discriminant analysis. That is, all of the input vectors are mapped into a high-, or even infinite, dimensional *feature space* \mathcal{H} via a nonlinear functional vector $\Phi(\mathbf{x}): \mathcal{I} \mapsto \mathcal{H}$, and the linear algorithm is then reformulated in the feature space \mathcal{H} in a form of the dot product of the mapped vectors $\Phi(\mathbf{x})$. Meanwhile, if a kernel function $K(x, y)$ satisfies Mercer's theorem [8], it can be represented as a dot product of two vectors in feature space \mathcal{H} , and the vector in \mathcal{H} is obtained by a nonlinear mapping $\Phi(\mathbf{x})$, which is implicitly determined by the chosen kernel function $K(x, y)$, i.e., $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$. Therefore, the inner product in feature space \mathcal{H} can in practice be replaced by a kernel function evaluation operation, and working explicitly with the high-dimensional vectors $\Phi(\mathbf{x})$ is thus avoided [8], [9]. In other words, once a kernel function $K(x, y)$ satisfying Mercer's theorem is selected, the nonlinear mapping functional vector $\Phi(\mathbf{x})$ is implicitly chosen, and the remaining problem is only reduced to a linear problem in the feature space \mathcal{H} . Therefore, a solution becomes readily available by following the same procedure described in Section II.

We begin our derivation by reformulating (3) and (4) as

$$\mathbf{S}_w^x = \frac{1}{N} \mathbf{X} \mathbf{D}_w \mathbf{X}^T \quad (5)$$

$$\mathbf{S}_b^x = \frac{1}{N} \mathbf{X} \Omega \mathbf{D}_b \Omega^T \mathbf{X}^T \quad (6)$$

where \mathbf{D}_w is a diagonal matrix with diagonal elements $\{N_1 \mathbf{I}_{N_1} - \mathbf{1}_{N_1}, \dots, N_L \mathbf{I}_{N_L} - \mathbf{1}_{N_L}\}$; \mathbf{D}_b is a diagonal matrix with diagonal elements $\{N_1, \dots, N_L\}$; and Ω is given by

$$\Omega = \begin{pmatrix} \frac{1}{N_1} \mathbf{1}_{N_1 \times 1} & & \\ & \ddots & \\ & & \frac{1}{N_L} \mathbf{1}_{N_L \times 1} \end{pmatrix} - \frac{1}{N} \mathbf{1}_{N \times L}. \quad (7)$$

We apply the LDA to the feature space \mathcal{H} after the nonlinear mapping $\Phi(\mathbf{x})$. Thus, the objective is to find the transformation matrix \mathbf{W}_Φ such that the linear transformation

$$\mathbf{y}' = \mathbf{W}_\Phi^T \Phi(\mathbf{x}) \quad (8)$$

maximizes

$$J(\mathbf{W}_\Phi) = \text{Tr} \left\{ (\mathbf{S}_w^{y'})^{-1} \mathbf{S}_b^{y'} \right\} \quad (9)$$

where matrices $\mathbf{S}_w^{y'}$ and $\mathbf{S}_b^{y'}$ are respectively the within-class and between-class scatter matrices of the projected vectors $\mathbf{y}' \in \mathcal{L}'$. According to (2) the criterion has the form

$$J(\mathbf{W}_\Phi) = \text{Tr} \left\{ \left(\mathbf{W}_\Phi^T \mathbf{S}_w^{\Phi(x)} \mathbf{W}_\Phi \right)^{-1} \left(\mathbf{W}_\Phi^T \mathbf{S}_b^{\Phi(x)} \mathbf{W}_\Phi \right) \right\}. \quad (10)$$

In order to utilize the kernel, we need to represent matrices \mathbf{W}_Φ , $\mathbf{S}_w^{\Phi(x)}$, and $\mathbf{S}_b^{\Phi(x)}$ in terms of vector $\Phi(\mathbf{x}) \in \mathcal{H}$. The scatter

matrices in feature space \mathcal{H} can be directly induced from (5) and (6) as

$$\mathbf{S}_w^{\Phi(x)} = \frac{1}{N} \Theta \mathbf{D}_w \Theta^T \quad (11)$$

$$\mathbf{S}_b^{\Phi(x)} = \frac{1}{N} \Theta \Omega \mathbf{D}_b \Omega^T \Theta^T \quad (12)$$

where $\Theta = (\Phi(\mathbf{x}_1), \Phi(\mathbf{x}_2), \dots, \Phi(\mathbf{x}_N))$ is the training set matrix in \mathcal{H} .

A result in [8] suggests that the column vectors of matrix \mathbf{W}_Φ are in a space spanned by the vectors in the training set Θ . Accordingly, any projection direction $\mathbf{w}_j^\Phi \in \mathcal{H}$ can be represented as a linear combination of all the vectors in Θ . That is, vector \mathbf{w}_j^Φ can be expressed as

$$\mathbf{w}_j^\Phi = \sum_{i=1}^N \alpha_j \Phi(\mathbf{x}_i) \quad (13)$$

for certain values of the α_j parameters. Then the linear transformation $\mathbf{y}' = \mathbf{W}_\Phi^T \Phi(\mathbf{x})$ from \mathcal{H} to \mathcal{L}' can be represented as

$$\mathbf{y}' = \begin{pmatrix} \sum_{i=1}^N \alpha_{i,1} \Phi(\mathbf{x}_i)^T \\ \vdots \\ \sum_{i=1}^N \alpha_{i,\ell'} \Phi(\mathbf{x}_i)^T \end{pmatrix} \Phi(\mathbf{x}). \quad (14)$$

Note that this is actually a nonlinear transformation from \mathcal{I} to \mathcal{L}' . Defining

$$\mathbf{A} = \begin{pmatrix} \alpha_{1,1} & \cdots & \alpha_{1,\ell'} \\ \vdots & \ddots & \vdots \\ \alpha_{N,1} & \cdots & \alpha_{N,\ell'} \end{pmatrix}. \quad (15)$$

Equation (14) becomes

$$\mathbf{y}' = (\mathbf{A}^T \Theta^T) \Phi(\mathbf{x}) = (\Theta \mathbf{A})^T \Phi(\mathbf{x}). \quad (16)$$

Therefore, from (8) and (16) we obtain

$$\mathbf{W}_\Phi = \Theta \mathbf{A}. \quad (17)$$

Inserting (17), (11), and (12) into (10), $J(\mathbf{W}_\Phi)$ becomes a function of matrix \mathbf{A} and is expressed in terms of vectors $\Phi(\mathbf{x})$ as

$$J(\mathbf{A}) = \text{Tr} \left\{ \left(\mathbf{A}^T \tilde{\mathbf{S}}_w^{\Phi(x)} \mathbf{A} \right)^{-1} \left(\mathbf{A}^T \tilde{\mathbf{S}}_b^{\Phi(x)} \mathbf{A} \right) \right\} \quad (18)$$

where

$$\mathbf{K} = \Theta^T \Theta \quad (19)$$

$$\tilde{\mathbf{S}}_w^{\Phi(x)} = \mathbf{K} \mathbf{D}_w \mathbf{K} \quad (20)$$

$$\tilde{\mathbf{S}}_b^{\Phi(x)} = \mathbf{K} \Omega \mathbf{D}_b \Omega^T \mathbf{K}. \quad (21)$$

Comparing (2) and (18), we can easily find that both equations have the same form if we note the following corresponding pairs $\mathbf{W} \rightarrow \mathbf{A}$, $\mathbf{S}_w^x \rightarrow \tilde{\mathbf{S}}_w^{\Phi(x)}$, and $\mathbf{S}_b^x \rightarrow \tilde{\mathbf{S}}_b^{\Phi(x)}$. Therefore, the optimization solution to (18) is exactly the same as that to (2). That is, the solution \mathbf{A} is given by those eigenvectors of $(\tilde{\mathbf{S}}_w^{\Phi(x)})^{-1} \tilde{\mathbf{S}}_b^{\Phi(x)}$ corresponding to the ℓ' largest eigenvalues.

There are two important comments about this result. First, because our selected Kernel function meets Mercer's theorem, the

K matrix in (19) is indeed implemented by kernel function evaluation, and working with high-dimensional vectors $\Phi(\mathbf{x}) \in H$ is thus avoided. Second, matrix $\tilde{\mathbf{S}}_w^{\Phi(x)}$ is singular and is thus not invertible in practice. Section IV deals with this implementation issue.

IV. A PRACTICAL ISSUE INVOLVED IN IMPLEMENTING OUR ALGORITHM

There are currently two methods that can be used to circumvent the problem of the noninvertibility of the matrix $\tilde{\mathbf{S}}_w^{\Phi(x)}$. The first method is to replace the inverse matrix $(\tilde{\mathbf{S}}_w^{\Phi(x)})^{-1}$ with a pseudoinverse matrix $\tilde{\mathbf{S}}_w^{\Phi(x)+}$ of matrix $\tilde{\mathbf{S}}_w^{\Phi(x)}$. The pseudoinverse matrix is defined as $\tilde{\mathbf{S}}_w^{\Phi(x)+} = \mathbf{M}^T \mathbf{D}^{-1} \mathbf{M}$, where \mathbf{M} is formed by the eigenvectors of matrix $\tilde{\mathbf{S}}_w^{\Phi(x)}$, and where \mathbf{D}^{-1} is a diagonal matrix formed by the inversion of the nonzero eigenvalues of matrix $\tilde{\mathbf{S}}_w^{\Phi(x)}$ [6], [10]. However, this method does not constrain the complexity of the resultant algorithm, and tends to overfit the training set in some cases. Therefore, we adopt here a different method to solve the problem [4], [5]. In this method, a nonsingular matrix \mathbf{G}_w is introduced from $\tilde{\mathbf{S}}_w^{\Phi(x)}$ as

$$\mathbf{G}_w = \tilde{\mathbf{S}}_w^{\Phi(x)} + \tau \mathbf{I} \quad (22)$$

where $\tau > 0$ and is called the *conditioning coefficient*, and \mathbf{I} is the identity matrix.

The conditioning coefficient τ (22) plays two important roles in our algorithm. First, from (22) we know that the larger τ is, the more significantly \mathbf{G}_w deviates from the original $\tilde{\mathbf{S}}_w^{\Phi(x)}$, but \mathbf{G}_w becomes better conditioned, which makes computing the inverse matrix of \mathbf{G}_w more numerically stable. Therefore, τ should be properly chosen to achieve an optimal tradeoff between theoretical accuracy and numerical stability.

Second, τ in (22) also has the effect of controlling the complexity of the resultant algorithm and thus provides us with a way of avoiding overfitting the training dataset. A properly selected τ will help to achieve an optimal tradeoff between the training error and generalization error.

In our implementation the *conditioning coefficient* τ in (22) is chosen as

$$\tau = \max \left\{ \frac{\lambda_{\max, \tilde{\mathbf{S}}_w^{\Phi(x)}}}{C_{\text{allow}}}, \tau_{\min} \right\} \quad (23)$$

where $\lambda_{\max, \tilde{\mathbf{S}}_w^{\Phi(x)}}$ is the maximal eigenvalue of $\tilde{\mathbf{S}}_w^{\Phi(x)}$; τ_{\min} is a predetermined value used to control the complexity of the resultant algorithm, and which can be obtained using cross-validation methods; C_{allow} is also a predetermined value chosen so that any matrix whose condition is less than this value can be numerically inverted stably, given a specific computational precision. Basically, the first term in (23) provides an estimate of the lower bound of an optimal conditioning coefficient, τ .

The justification for choosing τ according to (23) is as follows. From (22) we know that the condition of matrix $\tilde{\mathbf{S}}_w^{\Phi(x)}$ and \mathbf{G}_w are $C_{\tilde{\mathbf{S}}_w^{\Phi(x)}}$ and $C_{\mathbf{G}_w}$ respectively, which can be calculated as

$$C_{\tilde{\mathbf{S}}_w^{\Phi(x)}} = \frac{\lambda_{\max, \tilde{\mathbf{S}}_w^{\Phi(x)}}}{\lambda_{\min, \tilde{\mathbf{S}}_w^{\Phi(x)}}} \quad (24)$$

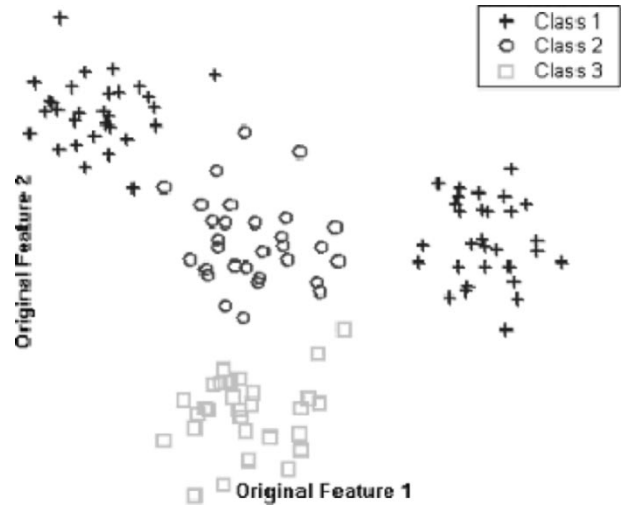


Fig. 1. Three-class problem composed by two-feature vectors.

$$C_{\mathbf{G}_w} = \frac{\lambda_{\max, \mathbf{G}_w}}{\lambda_{\min, \mathbf{G}_w}} = \frac{\lambda_{\max, \tilde{\mathbf{S}}_w^{\Phi(x)} + \tau}}{\lambda_{\min, \tilde{\mathbf{S}}_w^{\Phi(x)} + \tau}}. \quad (25)$$

Let $C_{\mathbf{G}_w} = C_{\text{allow}}$, and substitute it into (24) and (25), we obtain

$$\tau = \frac{\lambda_{\max, \tilde{\mathbf{S}}_w^{\Phi(x)}} (C_{\tilde{\mathbf{S}}_w^{\Phi(x)}} - C_{\text{allow}})}{C_{\tilde{\mathbf{S}}_w^{\Phi(x)}} (C_{\text{allow}} - 1)}. \quad (26)$$

By taking into consideration the facts that $C_{\text{allow}} \gg 1$, and $C_{\tilde{\mathbf{S}}_w^{\Phi(x)}} \gg C_{\text{allow}}$, (26) becomes

$$\tau \cong \frac{\lambda_{\max, \tilde{\mathbf{S}}_w^{\Phi(x)}}}{C_{\text{allow}}}. \quad (27)$$

Thus

$$\tau = \max \left\{ \frac{\lambda_{\max, \tilde{\mathbf{S}}_w^{\Phi(x)}}}{C_{\text{allow}}}, \tau_{\min} \right\}. \quad (28)$$

Fortunately, it is generally not necessary to calculate the coefficient τ exactly. In fact, we can further reduce the computational complexity by replacing $\lambda_{\max, \tilde{\mathbf{S}}_w^{\Phi(x)}}$ in (23) with one of its approximations. For example, we can replace $\lambda_{\max, \tilde{\mathbf{S}}_w^{\Phi(x)}}$ with the one-norm of $\tilde{\mathbf{S}}_w^{\Phi(x)}$ in practice. The rationale behind this replacement is that the one-norm of $\tilde{\mathbf{S}}_w^{\Phi(x)}$ should reflect the scale of the two-norm of $\tilde{\mathbf{S}}_w^{\Phi(x)}$, which is exactly $\lambda_{\max, \tilde{\mathbf{S}}_w^{\Phi(x)}}$.

V. EXPERIMENTS

In this section, we use a synthetic three-class problem, as illustrated in Fig. 1, to demonstrate the performance of our algorithm. A radial basis function is selected as the kernel function in this experiment.

For a three-class problem, our algorithm obtains two eigenvectors, which corresponds to the two nonzero eigenvalues of $(\mathbf{G}_w)^{-1} \tilde{\mathbf{S}}_w^{\Phi(x)}$. Therefore, two new features can be extracted from each original vector. The distribution of the new vectors, which are formed by the two extracted features, is shown in Fig. 2. From this figure, we observe that the extracted features

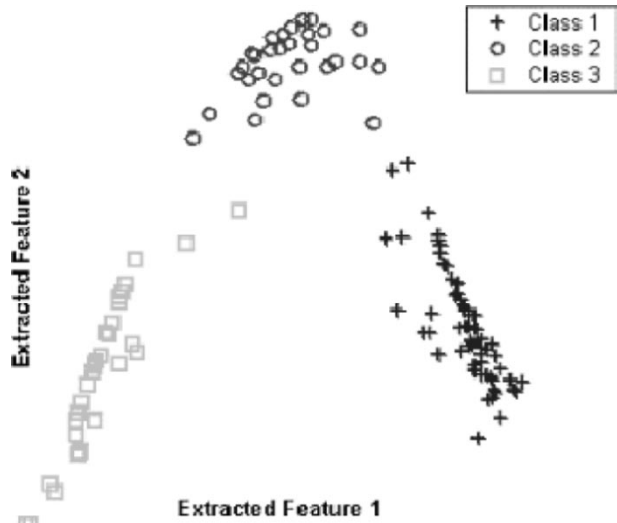


Fig. 2. Distribution of new vectors, which are formed by two extracted features.

are sufficient to readily separate the three classes in this problem.

Experiments of applying this algorithm to benchmark datasets, as well as the performance comparison between this algorithm and alternative algorithms such as support vector machines, are implemented in [11]. We refer the reader to these results due to space limitations.

VI. CONCLUSION

In this letter, we have introduced a new nonlinear multiclass discriminant analysis algorithm. This algorithm relies on the use

of the kernel trick, which permits the efficient computation of the linear discriminant in a high-, or even infinite, dimensional feature space. Our algorithm is a simple and direct extension of the general multiclass linear discriminant analysis. We also suggest a simple way to determine the proper value for the regularization parameter, τ , which can be used to both guarantee the numerical stability and avoid overfitting the training data.

REFERENCES

- [1] R. A. Fisher, "The used of multiple measurements in taxonomic problems," *Ann. Eugenics*, vol. 7, pp. 179–188, 1936.
- [2] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, 2nd ed. New York: Academic, 1990.
- [3] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York: Oxford Univ. Press, 1995.
- [4] S. Mika, G. Rätsch, J. Weston, B. Schölkopf, and K.-R. Muller, "Fisher discriminant analysis with kernels," in *Proc. Neural Networks for Signal Processing IX*, Y.-H. Hu, J. Larsen, E. Wilson, and S. Douglas, Eds: IEEE, 1999, pp. 41–48.
- [5] V. Roth and V. Steinhage, "Nonlinear discriminant analysis using kernel functions," in *Advances in Neural Information Processing Systems*, S. A. Solla, T. K. Leen, and K. -R. Müller, Eds. Cambridge, MA: MIT Press, 1999, vol. 12, pp. 568–574.
- [6] G. Baudat and F. Anouar, "Generalized discriminant analysis using a kernel approach," *Neural Comput.*, vol. 12, no. 10, pp. 2385–2404, 2000.
- [7] M. A. Aizerman, E. M. Braverman, and L. I. Rozonoér, "Theoretical foundations of the potential function method in pattern recognition learning," *Automat. Remote Contr.*, vol. 25, pp. 821–837, 1964.
- [8] S. Saitoh, *Theory of Reproducing Kernels and Its Applications*. Harlow, U.K.: Longman Scientific and Technical, 1988.
- [9] C. J. C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, 1998.
- [10] A. Ruiz and P. E. Lopez-de-Teruel, "Nonlinear kernel-based statistical pattern analysis," *IEEE Trans. Neural Networks*, vol. 12, pp. 16–32, Jan. 2001.
- [11] J. Ma, J. Theiler, and S. Perkins, "Two realizations of a general feature extraction framework," Los Alamos Lab., Los Alamos, NM, Los Alamos Lab. Directed Research and Development (LDRD) Program Tech. Rep., 2002.