Support Vector Machines and the Multiple Hypothesis Test Problem

Daniel J. Sebald, *Member, IEEE,* and James A. Bucklew

***Abstract*—Two enhancements are proposed to the application and theory of support vector machines. The first is a method of multicategory classification based on the binary classification version of the support vector machine (SVM). The method, which is called the M-ary SVM, represents each category in binary format, and to each bit of that representation is assigned a conventional SVM. This approach requires only $\lceil \log_2(K) \rceil$ SVMs, where K is the number of classes. We give an example of classification on an octaphase-shift-keying (8-PSK) pattern space to illustrate main concepts.**

The second enhancement is that of adding equality constraints to the conventional binary classification SVM. This allows pinning the classification boundary to points that are known *a priori* to lie on the boundary. Applications of this method often arise in problems having some type of symmetry. We present one such example where the M-ary SVM is used to classify symbols of a two-user, multiuser detection pattern space.

***Index Terms*—Boundary constraint, equality constrained SVM, M-ary classification, M-ary SVM, multicategory classification, pattern recognition, support vector machine.**

I. Introduction

In a K-class pattern recognition problem (which is also called multiclass or multicategory recognition), the goal is to partition a pattern space into K regions, where each region is assigned to one of K outcomes. It is a generalization of the binary classification problem where the pattern space is partitioned into two subsets. There have been basically three approaches in the literature to solving such K-class problems.

a) Apply well-studied statistical techniques such as regression, estimation, and detection.
b) Use a group of binary classifiers where usually one or both of the two regions for a classifier is associated with just one of the K categories.
c) Incorporate all binary borders into a simultaneously optimized cost function.

The first approach includes a variety of parametric methods requiring *a priori* knowledge of data statistics. The second approach usually requires a large number of classifiers and an additional processing step to resolve ambiguities in region overlaps. The third approach requires the development of a multiclass mathematical programming optimization routine and can become computationally demanding. Several examples of these methods are summarized in Section II.

Our method of K-class recognition (the M-ary SVM) will be shown to have several distinct advantages over these more conventional multicategory classification methods. First, the method makes full use of the binary SVMs ability to model nonlinear classification boundaries by grouping all categories into only two classes then constructing a nonlinear boundary to separate these two more complex classes. Second, the approach is efficient in that it requires only $\lceil \log_2 K \rceil$ classifiers, whereas conventional linear boundary-based multicategory classifiers generally require $O(K)$ or $O(K^2)$ classifiers. Third, the M-ary SVM is rather straightforward to implement, whereas linear classifiers require a variety of *ad hoc* methods for constructing nonlinear boundaries. Fourth, like the binary SVM, the M-ary SVM is data dependent, and statistics about the data need not be known *a priori*. However, as will be shown, with some *a priori* knowledge of the relative position of classes, it may be possible to select a binary numbering pattern that results in SVM models having a smaller number of support vectors and, consequently, more efficient in the recognition stage. The M-ary SVM is presented in Section III along with some examples for an eight-PSK constellation. Contrasting a natural code and Gray code enumeration for the pattern space illustrates some characteristics of the M-ary SVM.

The equality constrained SVM (ECSVM) is a method for forcing the decision boundary of the conventional SVM to pass through chosen points in the pattern space. The SVM discriminant function is linear in the feature space optimized variables. Therefore, setting this function to zero at given points in the pattern space produces linear equality constraints that can be incorporated into the SVM Lagrangian. If the resulting optimization is feasible, the decision boundary passes through the given points. A mathematical derivation of the ECSVM and an example for two-user, multiuser detection are given in Section IV.

We finish this section with an introduction to the main formulae and concepts of the binary SVM pattern classifier [1] because of its central importance to this paper.

Binary SVM Classifier

For the binary classification problem, the training set consists of vectors from the pattern space $\mathbf{x}_i \in \mathbb{R}^N$, $i = 1, 2, \ldots, L$ and to each vector a classification $y_i \in \{-1, +1\}$. Let $(\mathbf{x}_i, y_i)$ represent the binary classification training data. The pattern space data is mapped to a higher dimensional feature space, i.e., $\Phi(\mathbf{x}) \in \mathbb{R}^{N'}$, where an optimal hyperplane

Manuscript received April 20, 2000; revised July 9, 2001. The associate editor coordinating the review of this paper and approving it for publication was Prof. Colin F. Cowan.

D. J. Sebald is at 1553 Adams St., Madison, WI 53711 USA (e-mail: daniel.sebald@ieee.org).

J. A. Bucklew is with the Department of Electrical and Computer Engineering, University of Wisconsin, Madison, WI 53766 USA.

Publisher Item Identifier S 1053-587X(01)09216-9.

$(\mathbf{w}^*, b^*) \in (\mathbb{R}^{N'}, \mathbb{R})$ separates, if possible, the data $\mathbf{x}_i$ according to sign$(f(x))$ where

$$f(\mathbf{x}) = \mathbf{w}^* \cdot \Phi(\mathbf{x}) + b^*. \tag{1}$$

Otherwise, it separates the classes with some allowable *slack*. A margin maximization/error minimization criteria leads to the optimization that minimizes

$$\phi(\mathbf{w}, \boldsymbol{\xi}) = \tfrac{1}{2}(\mathbf{w} \cdot \mathbf{w}) + C \sum_{i=1}^{L} \xi_i \tag{2}$$

under constraints

$$y_i((\mathbf{w} \cdot \Phi(\mathbf{x}_i) + b) \geq 1 - \xi_i, \qquad i = 1, 2, \ldots, L \tag{2'}$$
$$\xi_i \geq 0, \qquad i = 1, 2, \ldots, L \tag{2''}$$

where $\{\xi_i\}$ are slack variables introduced to account for nonseparable data, and C is a constant effectively controlling the shape of the classification boundary when data is nonseparable. The above form is chosen because its dual optimization leads to a quadratic program (QP) solution [1], [2]. The end result is a discriminant function conveniently expressed as a function of the lower dimensional pattern space data

$$f(\mathbf{x}) = \sum_{i \in S} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b^*. \tag{3}$$

The set S is a subset of the indices $\{1, 2, \ldots, L\}$, and the training vectors associated with S are called *support vectors*. In S are margin support vectors that lie on the hyperplane margins and nonmargin support vectors that correspond to errors. The kernel $K(\cdot, \cdot)$ must satisfy the conditions of Mercer's theorem [1], [3] so that it corresponds to some type of inner product in the higher dimensional feature space. The SVM can be shown to have desirable generalization properties, i.e., it classifies well for data that is statistically similar to the training data [1].

II. Multicategory Classifier Overview

We now summarize the main multicategory classification methods to contrast against the new method proposed in Section III. There are a large number of pattern recognition methods overlooked here (for a good overview, see [4]), and in most cases, the strategies given can be used with other forms of classifiers other than the SVM. We have attempted to summarize the fundamental concepts as opposed to exhaustively covering all methods.

As in the binary case, the learning machine is presented with pattern space vectors $\{\mathbf{x}_i\}$, $i = 1, 2, \ldots, L$. Call this training set $\mathcal{T}$. In addition, define $\mathcal{S} = \{1, 2, \ldots, K\}$. In the multiclass case, each training vector is assigned to one of K classes $c_i \in \mathcal{S}$. Let $(\mathbf{x}_i, c_i)$ represent the multicategory classification training data. In the test stage, define $\hat{c}$ to be the learning machine class decision for test vector $\mathbf{x}$.

A. Parametric Strategies

There are a wide variety of pattern classification methods derived from traditional statistics that we loosely call "parametric" approaches. The techniques are too numerous to list, but we describe a few of the well-studied approaches. An obvious attack is along the lines of maximum likelihood, but this requires either knowledge of the noise statistics *a priori* or the requirement that one try to estimate them from some sort of parametric model or even follow some sort of generalized density estimation procedure, *à la* Parzen [5]. Unfortunately, density estimation generally requires large amounts of data and, thus, is limited in application.

Another clear method uses kernels to build a discriminant function for each class from training data as

$$f_k(\mathbf{x}) = \sum_{i=1}^{L} 1_{\{k\}}(c_i) K\left(\frac{\mathbf{x} - \mathbf{x}_i}{h}\right)$$

where

$1_A(\cdot)$ indicator function of set A;
$K(\cdot)$ kernel (not necessarily a Mercer kernel);
h smoothing factor.

Then, the classifier estimate is

$$\hat{c} = \arg\max_{k \in \mathcal{S}} f_k(\mathbf{x}). \tag{4}$$

The drawback of this approach is that the discriminant functions become computationally intensive for even moderate amounts of data since every data point becomes part of the model. Furthermore, the choice of kernel shape and smoothing parameter can sometimes be difficult.

A characteristic shared by many of these approaches is quite generally that they are statistically consistent [4]. However, consistency is a behavior in the limit. Unlike the SVM, there is generally no analysis in the literature regarding the behavior of these methods on small training data sets.

B. Binary Classifier-Based Strategies

The general principle of all binary classifier-based strategies is to build a set of binary classifiers that, when combined in some logical fashion, approximate the desired multicategory classifications. In addition, the overall classifier should have good generalizing capabilities for test data. Depending on the method, we are effectively grouping the training samples into two classes and then constructing discriminant functions based on these groupings. Consider choosing two subsets of $\mathcal{S}$, A, and B such that $A \cap B = \varnothing$. The training sets for the binary classifier are then

$$\mathcal{C}_A = \{\mathbf{x}_i \in \mathcal{T} : c_i \in A\}$$
$$\mathcal{C}_B = \{\mathbf{x}_i \in \mathcal{T} : c_i \in B\}$$

and we associate with each $\mathbf{x}_i \in \mathcal{C}_A \cup \mathcal{C}_B$ the binary training target value

$$y_i = 1_A(c_i) - 1_B(c_i). \tag{5}$$

1) One-Class-Versus-All: In one-class-versus-all [6], binary training classes are constructed by taking one class against all others *combined*, i.e.,

$$A_k = \{k\}$$
$$B_k = \mathcal{S} \backslash A_k$$

for $k = 1, 2, \ldots, K$; then, we train K binary classifiers $f_{A_k B_k}: \mathbb{R}^N \mapsto \mathbb{R}$. The decision $\hat{c}$ is assigned to the class j when $\text{sign}(f_{A_k B_k}(\mathbf{x})) = -1$ for all k except $k = j$. A major problem with this approach is regions of ambiguity when $\text{sign}(f_{A_k B_k}(\mathbf{x})) = +1$ for more than one value of k.

2) Winner-Takes-All: The strategy of winner-takes-all is similar to one-class-versus-all, except the sign function applied to binary classifiers is removed so that each classifier produces a real value. The training sets are assigned similar to that in Section II-B-1. The learning machine decision is then similar to (4). The ambiguities of the one-versus-all technique have now been avoided. However, if a linear binary classifier is used, problems arise when data for one category is not linearly separable from the data of all remaining categories combined [6]. Furthermore, this approach may be precarious for nonlinear classifiers where the discriminant function may not have a direct relationship to distance in the lower dimensional pattern space.

3) Pairwise Classification: In pairwise classification [6], binary training classes are constructed by taking one class against all others *individually*. That is, we construct sets

$$A_k = B_k = \{k\}, \qquad k = 1, 2, \ldots, K$$

and then train a discriminant function $f_{A_k B_m}: \mathbb{R}^N \mapsto \mathbb{R}$ for $k, m \in \{l, \ldots, K\}$ and $k \neq m$, recognizing that $f_{km} = -f_{mk}$. A score function is then constructed by testing all pairwise classes and summing as

$$f_K(\mathbf{x}) = \sum_{\substack{m \in \{1, \ldots, K\} \\ m \neq k}} \text{sign}(f_{A_k B_m}(\mathbf{x})).$$

The class is then declared similar to (4). Tie situations do occur with the pairwise method but are not as prevalent as they are in the one-class-versus-all method. Ties may be resolved with a variation of the winner-takes-all strategy or may be declared to be rejects. The major limitation of this classifier is that it requires $\binom{K}{2} = K(K-1)/2$ binary classifiers. This can be computationally demanding, even for a moderate number of classes.

C. Multiclass Optimization Strategies

1) Multiclass SVM: The multiclass SVM was evidently developed independently by three groups of researchers, as stated in [1]. A generalization of the binary optimization (2) applied to the multicategory training data $(\mathbf{x}_i, c_i)$ is to minimize

$$\phi(\mathbf{w}_1, \ldots, \mathbf{w}_K, \boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_K) = \tfrac{1}{2} \sum_{k=1}^{K} (\mathbf{w}_k \cdot \mathbf{w}_k) + C \sum_{i=1}^{L} \sum_{k \neq c_i} \xi_i^k \tag{6}$$

under constraints

$$(\mathbf{w}_{c_i} \cdot \Phi(\mathbf{x}_i) + b_{c_i}) \geq (\mathbf{w}_k \cdot \Phi(\mathbf{x}_i) + b_k) + 2 - \xi_i^k \tag{6'}$$

$$\xi_i^k \geq 0 \tag{6''}$$

for $i = 1, 2, \ldots, L$ and $k \in \mathcal{S} \backslash c_i$. Solving the dual problem [7] results in discriminant functions expanded as

$$f_k(\mathbf{x}) = \sum_{i:\, c_i = k} A_i K(\mathbf{x}_i, \mathbf{x}) - \sum_{i:\, c_i \neq k} \alpha_i^k K(\mathbf{x}_i, \mathbf{x}) + b_k^*$$

where A_i, α_i^k, and b_k are analogs of Lagrange multipliers for the binary SVM. The class is then declared according to (4).

A linear program (LP) version of the multiclass SVM was developed in [7]. The QP multiclass SVM is quadratic in $(K-1)L$ variables, whereas the LP multiclass SVM has KL variables and KL constraints. Both of the problems become very complex with even a moderate number of classes.

2) Combined Mathematical Programming and SVM: Mangasarian [8] developed an LP-based, piecewise linear separation method for binary problems on linearly nonseparable data using a margin maximization strategy for separable data [9], predating the optimal hyperplane. It may be extended to nonlinear surfaces, but not like the SVM, using a kernel method. This LP approach was then adapted to nonseparable data by minimizing an error cost utilizing the ℓ_1 norm in [10]. The error minimizing concept originally appeared in [11]. This robust linear programming (RLP) technique was enhanced with an ℓ_1 norm of the hyperplane vector added to its optimized cost in [12] and [13] to give it better generalizing capabilities.

Bennett and Mangasarian [14], [15] developed a piece-wise linear separation method for the K-class problem called multicategory discrimination. A robust version of this method called M-RLP was also developed [16]. In this method, an RLP breaks up multiclass data into piece-wise linearly separable or nearly piece-wise linearly separable categories. If there is some subregion that cannot be separated as such, the same process is applied to that subregion. The method is iterated until the categories are appropriately modeled resulting in a tree-like structure. Advantages of the M-RLP are the efficiency and robustness of LP routines compared with QP routines. However, its drawback is that for noisy data, deciding where and when to create subregions is not a straightforward methodology for an algorithm.

Bredensteiner and Bennett [16], [17] have developed a robust linear programming method similar to one-class-versus-all classification (k-RLP) and an SVM-based version of multicategory discrimination (M-SVM). The M-SVM has the advantage that nonlinear boundaries may be constructed in the pattern space, as opposed to just linear boundaries. The limitation of the M-SVM is that it becomes a rather complex optimization problem for even a moderate number of classes.

III. Proposed M-ary Classification

A. M-ary Classification

Support vector machines are able to model classification of rather complex regions for the binary problem [18]–[20]. Therefore, they can be applied to multicategory classification in a more fundamental way than comparing one class against all others. Consider labeling each category in binary format and then using a conventional SVM to model individual bits of that

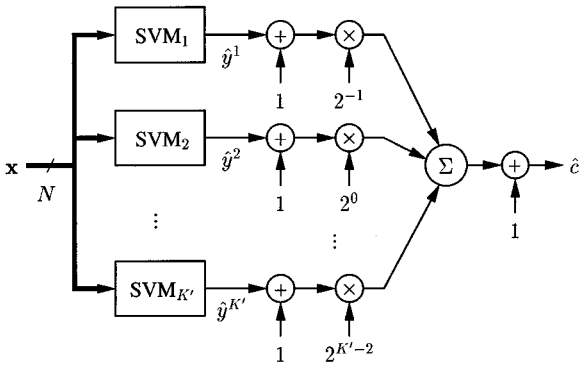

Fig. 1. M-ary nonlinear classification using $K' = \lceil \log_2 K \rceil$ support vector machines.

representation. The process is illustrated in Fig. 1. This strategy can be expressed by assigning

$$A_k = \left\{ j \in \mathcal{S}: \left\lfloor (j-1)2^{-(k-1)} \right\rfloor \text{ is odd} \right\}$$
$$B_k = \mathcal{S} \backslash A_k \tag{7}$$

for $k = 1, \ldots, K'$, where $K' = \lceil \log_2 K \rceil$. Given the multicategory classification problem $(\mathbf{x}_i, c_i)$ of Section II-C1, convert c_i to vector

$$\mathbf{y}_i = [\, y_i^1 \quad y_i^2 \quad \cdots \quad y_i^{K'} \,]^T$$

such that y_i^k is the target value defined by (5) and applied to sets (7).[1] The transformed problem is represented as $(\mathbf{x}_i, \mathbf{y}_i)$, which is meant to be a set of individual binary pattern recognition problems $(\mathbf{x}_i, y_i^k)$, $k = 1, 2, \ldots, K'$. As a generalization of (3), we then have K' discriminant functions

$$f_k(\mathbf{x}) = \sum_{i \in S_k} \alpha_i y_i^k K(\mathbf{x}_i, \mathbf{x}) + b_k^*.$$

In the classification stage, $\hat{y}^k = \text{sign}(f_k(\mathbf{x}))$, and the class is declared according to

$$\hat{c} = 1 + \sum_{k=1}^{K'} (\hat{y}^k + 1) 2^{k-2}. \tag{8}$$

The name "M-ary classification" is given to this method since it uses the symbol representation format of M-ary modulation in communications systems. M-ary does not restrict the number of categories K to be a power of two. The advantages of M-ary classification were given in Section I.

Similar to the multicategory discrimination approach, M-ary SVM attempts to divide the categorization into subproblems. However, rather than a tree structure, the M-ary SVM inherently creates intersections of binary classified patterns. We must point out that the drawback to the M-ary SVM is that the binary classes, as defined by (7), create complex patterns, and the binary SVMs must be designed with enough complexity yet generalize well. However, this continues to be a fundamental theoretical problem of interest to many researchers, e.g., [1], [21].

[1]These types of operations are very straightforward when implemented on a microprocessor. For example, the $\{y_i^k\}$ can be computed with a simple bit-wise test.

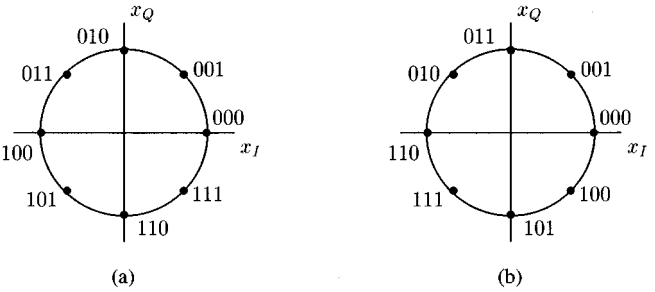

Fig. 2. Signal constellations for 8-PSK signaling using a (a) natural code and (b) Gray code.

B. Illustrative Example[2]

Consider a nondistorted signal space for 8-PSK. In this case, the M-ary classification approach requires only three binary SVMs, whereas the winner-takes-all method requires eight binary SVMs, and the pairwise classifier requires 28 binary SVMs. The constellation using a natural code and a Gray code are shown in Fig. 2. The binary reflected Gray (BRG) code [23] is derived from the natural code as

$$g^k = \begin{cases} n^k, & k = K' \\ n^k \oplus n^{k+1}, & k = 1, 2, \ldots, K'-1 \end{cases}$$

where n^k and g^k are the kth bit of the natural and Gray code, respectively, and $\oplus$ is modulo-2 addition.

A three-bit, M-ary SVM was trained to classify these regions by randomly selecting c_i with equal probability and setting

$$\mathbf{x}_i = \begin{bmatrix} \cos(2\pi c_i/8) + w_i^I \\ -\sin(2\pi c_i/8) + w_i^Q \end{bmatrix}$$

where w_i^I and w_i^Q are independent in-phase and quadrature additive white Gaussian random variables with zero mean and variance $N_0/2$. The signal-to-noise ratio (SNR) is $1/N_0$. A polynomial SVM kernel of the form $K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^d$ was used, with constant $C = 5$ and $L = 200$. The target training values depend on the coding scheme and are assigned

$$y_i^k = \begin{cases} n_i^k, & \text{if natural code} \\ g_i^k, & \text{if Gray code.} \end{cases}$$

Results for a 17-dB SNR are shown in Figs. 3–5. The upper-left plot is the composite classification where each class is represented by a different symbol. The remaining plots show the classification for individual bits where $+$ represents samples of $\mathcal{C}_A$, and $\circ$ represents samples of $\mathcal{C}_B$. Shaded regions are what the SVM classifies as belonging to $\mathcal{C}_A$. In each of the subtitles is the number of margin support vectors followed by the total support vectors.

Fig. 3 shows how the Gray code results in simple classification regions and low model complexity (i.e., small number of support vectors) for each bit and how the composite classifications appear reasonable. For the natural code, Fig. 4 shows the problem when the SVM using a polynomial kernel order of

[2]Examples in this section were done with MATLAB 5.3, with an interface to PATH 3.0 [22] for solving the QP optimization.

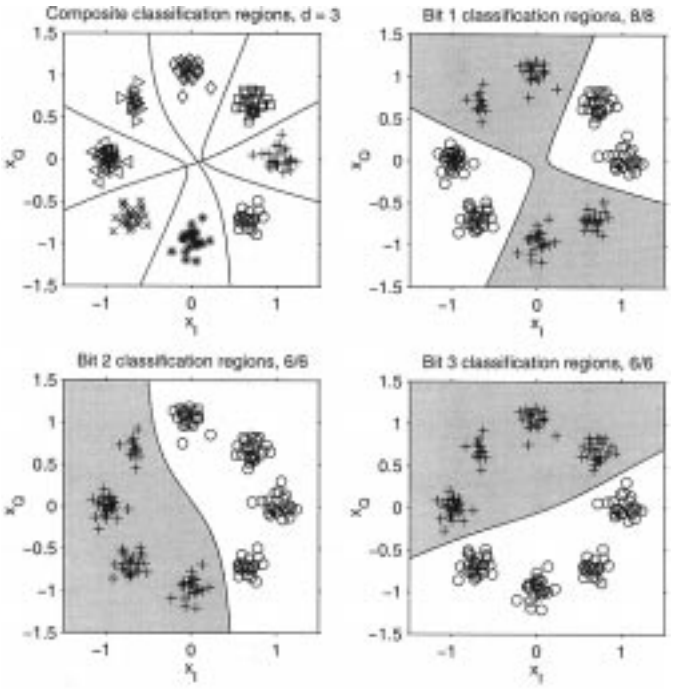

Fig. 3. Result for Gray code.

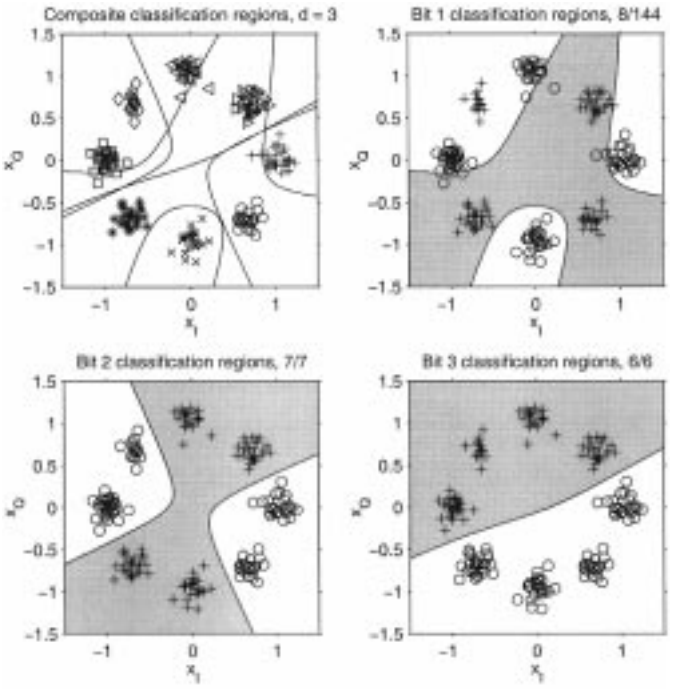

Fig. 4. Result for natural code. The more complex regions result in the data not being separable when $d = 3$. Consequently, the results are poor.

three classifies bit 1. The SVM does not have the necessary complexity for the problem, and an excessive number of nonmargin support vectors indicates that the data is essentially nonseparable in the higher dimensional feature space. However, when $d = 5$, as in Fig. 5, the system performs much better. Note the decrease in number of support vectors for modeling bit 1.

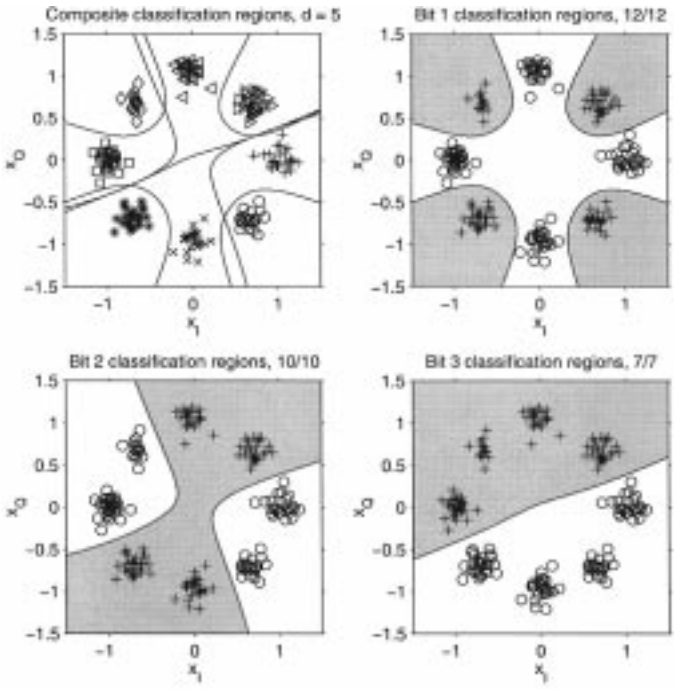

Fig. 5. For the natural code, increasing d to 5 results in the feature space data being separable. Boundaries are now reasonable but with areas of ambiguity near the boundaries.

C. Comments on Gray Codes

The 8-PSK example above suggests advantages of simplifying the pattern space through coding when there is some *a priori* knowledge of the suspected relative location of individual categories. In this particular example, a Gray-coded constellation is beneficial, but more than that, the Gray code should be of the nature that transitions are evenly distributed among the individual bits and that the minimum run (i.e., number of codes without a transition) is maximized. Such codes are called large-gap codes, and Goddyn, *et al.* [24] have applied such codes to photon detection. The generation of specialized Gray codes has been studied in [24]–[28], and the theoretical existence of Gray codes with certain properties has been considered in [29] and [30]. Savage [31] is a good survey of up-to-date Gray code research. The BRG code is always a gap-2 code, but in the case of 8-PSK, the BRG code happens to be the same as the large-gap code [24].

IV. SVM WITH EQUALITY CONSTRAINTS

A. ECSVM

Some classification problems exhibit symmetry, in which case, the optimum decision boundary may be known *a priori* to pass through certain points. We propose adding equality constraints of the form $f(\boldsymbol{\chi}_i) = \tau_i, i = 1, 2, \ldots, M$ to the conventional binary SVM of (2). By setting $\tau_i = 0$, the decision border will pass through point $\boldsymbol{\chi}_i$, provided a solution is feasible. The additional constraints are

$$\mathbf{w} \cdot \Phi(\boldsymbol{\chi}_i) + b = \tau_i, \qquad i = 1, 2, \ldots, M$$

and since they are linear in the feature space variables $\mathbf{w}$, they may be incorporated into the optimization such that duality

theory is still valid. (For a discussion of duality principles, see Luenberger [32].) The convex form of inequality constraints (2′) and (2″) applicable to optimization are

$$\begin{aligned} -(y_i(\mathbf{w}\cdot\Phi(\mathbf{x}_i)+b)-1+\xi_i)\le 0, &\qquad i=1,2,\ldots,L\\ -\xi_i\le 0, &\qquad i=1,2,\ldots,L. \end{aligned}$$

For convenience, define

$$\begin{aligned} f_i &= 1, && i=1,2,\ldots,L\\ f_i &= \tau_{i-L}, && i=L+1,L+2,\ldots,L+M\\ y_i &= -1, && i=L+1,L+2,\ldots,L+M\\ \mathbf{x}_i &= \boldsymbol{\chi}_{i-L}, && i=L+1,L+2,\ldots,L+M. \end{aligned}$$

We now introduce Lagrange multipliers

$$\begin{aligned} \boldsymbol{\alpha} &= [\,\alpha_1 \quad \alpha_2 \quad \cdots \quad \alpha_{L+M}\,]^T\\ \boldsymbol{\beta} &= [\,\beta_1 \quad \beta_2 \quad \cdots \quad \beta_L\,]^T \end{aligned}$$

and construct a Lagrangian

$$\begin{aligned} &L(\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta})\\ &\quad=\tfrac{1}{2}\mathbf{w}\cdot\mathbf{w}+C\sum_{i=1}^{L}\xi_i-\sum_{i=1}^{L}\beta_i\xi_i\\ &\qquad-\sum_{i=1}^{L}\alpha_i(y_i(\mathbf{w}\cdot\Phi(\mathbf{x}_i)+b)-f_i+\xi_i)\\ &\qquad-\sum_{i=L+1}^{L+M}\alpha_i(y_i(\mathbf{w}\cdot\Phi(\mathbf{x}_i)+b)-f_i) \end{aligned} \tag{9}$$

where

$$\begin{aligned} \alpha_i &\ge 0, && i=1,2,\ldots,L && (9')\\ \alpha_i &\in \mathbb{R}, && i=L+1,L+2,\ldots,L+M && (9'')\\ \beta_i &\ge 0, && i=1,2,\ldots,L. && (9''') \end{aligned}$$

The optimizer for (9) is a saddle point minimized with respect to $\mathbf{w}$, b, and $\boldsymbol{\xi}$ and maximized with respect to the Lagrange multipliers.

The dual principle is to solve for the optimum with respect to the variables that are not Lagrange multipliers. The resulting dual functional is then maximized with respect to the Lagrange multipliers. Therefore, continue by applying a theorem by Fermat, i.e.,

$$\frac{\partial L(\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \mathbf{w}}=\mathbf{w}-\sum_{i=1}^{L+M}y_i\alpha_i\Phi(\mathbf{x}_i)=\mathbf{0} \tag{10}$$

$$\frac{\partial L(\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial b}=-\sum_{i=1}^{L+M}y_i\alpha_i=0 \tag{11}$$

$$\frac{\partial L(\mathbf{w},b,\boldsymbol{\xi},\boldsymbol{\alpha},\boldsymbol{\beta})}{\partial \xi_i}=C-\alpha_i-\beta_i=0. \tag{12}$$

Equation (10) implies

$$\mathbf{w}=\sum_{i=1}^{L+M}y_i\alpha_i\Phi(\mathbf{x}_i) \tag{13}$$

whereas (12) implies

$$\beta_i=C-\alpha_i \tag{14}$$

and, along with (9‴)

$$\alpha_i\le C. \tag{15}$$

Substitution of (11) and (13)–(15) into (9) and (9‴) leads to the dual problem, i.e., maximize

$$W(\boldsymbol{\alpha})=\sum_{i=1}^{L+M}\alpha_i f_i-\tfrac{1}{2}\sum_{i,j=1}^{L+M}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j) \tag{16}$$

under constraints

$$\sum_{i=1}^{L+M}\alpha_i y_i=0 \tag{16'}$$

$$\begin{aligned} 0\le\alpha_i\le C, &\qquad i=1,2,\ldots,L && (16'')\\ \alpha_i\in\mathbb{R}, &\qquad i=L+1,L+2\ldots,L+M && (16''') \end{aligned}$$

where the Mercer kernel $K(\mathbf{x}_i,\mathbf{x}_j)$ corresponds to a valid inner product $\Phi(\mathbf{x}_i)\cdot\Phi(\mathbf{x}_j)$.

Equation (16) is a more general form of the familiar QP problem. By substituting (13) into (1) and applying the Mercer kernel method, the resulting discriminant function for pattern recognition takes the same form as (3), but now, the support vector index set S is a subset of $\{1,2,\ldots,L+M\}$.

It is useful to put the dual problem (16) in vector format for purposes of programming. The most common general form is [33]

$$\min\left\{\tfrac{1}{2}\mathbf{z}^T\mathbf{Q}\mathbf{z}+\mathbf{c}^T\mathbf{z}\colon \mathbf{a}_i^T\mathbf{z}=b_i,\ i\in\mathcal{E},\mathbf{a}_i^T\mathbf{z}\le b_i, i\in\mathcal{I}\right\} \tag{17}$$

where

- $\mathbf{z}$ variable to be optimized;
- $\mathcal{E}$ index set for equality constraints;
- $\mathcal{I}$ index set for inequality constraints.

Although the $\{\alpha_i\}$s play the role of Lagrange multipliers, here, they are treated as the optimized variable $\mathbf{z}$ in order to implement constraints (16′)–(16‴). Let

$$\begin{aligned} \mathbf{f} &= [\,f_1 \quad f_2 \quad \cdots \quad f_{L+M}\,]^T\\ \mathbf{y} &= [\,y_1 \quad y_2 \quad \cdots \quad y_{L+M}\,]^T \end{aligned}$$

and the matrix $\mathbf{Y}=\operatorname{diag}\{\mathbf{y}\}$, and arrange the vectors of the pattern space into an $N\times(L+M)$ matrix as

$$\mathbf{X}=[\,\mathbf{x}_1 \quad \mathbf{x}_2 \quad \cdots \quad \mathbf{x}_{L+M}\,].$$

The QP problem is usually programmed as a minimization. Expanding the negative of (16) and rearranging terms leads to

$$\begin{aligned} -W(\boldsymbol{\alpha}) &= -\sum_{i=1}^{L+M}\alpha_i f_i+\tfrac{1}{2}\sum_{i,j=1}^{L+M}\alpha_i\alpha_j y_i y_j K(\mathbf{x}_i,\mathbf{x}_j)\\ &= -\mathbf{f}^T\boldsymbol{\alpha}+\tfrac{1}{2}\boldsymbol{\alpha}^T\mathbf{Y}K(\mathbf{X},\mathbf{X})\mathbf{Y}^T\boldsymbol{\alpha} \end{aligned}$$

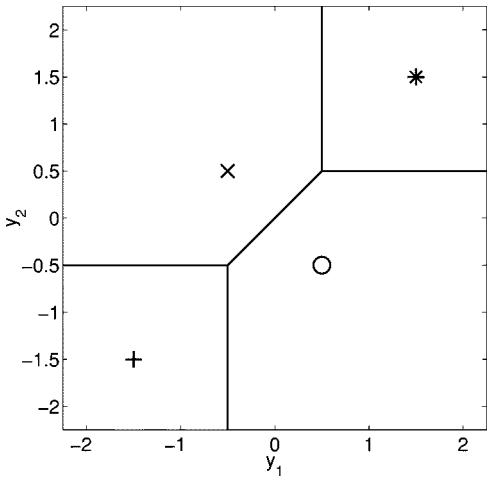

Fig. 6. Pattern space for a noise-free, two-user DS-CDMA system with correlation $\rho = 0.5$. Boundaries show the optimum decision regions under the assumption of additive white Gaussian noise.

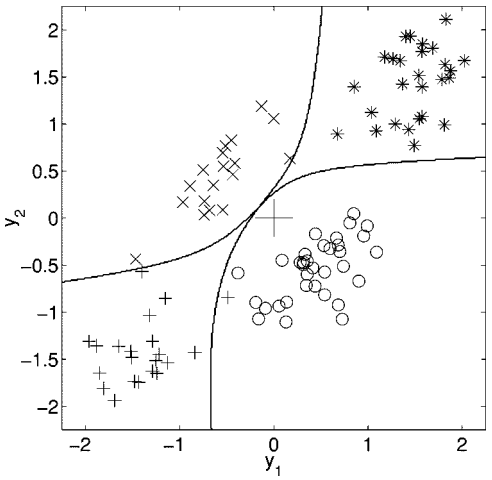

Fig. 7. Example decision boundaries of the M-ary SVM trained with the displayed data for the DS-CDMA space of Fig. 6.

where the matrix Grammian $K(\mathbf{X}, \mathbf{X})$ is a component-wise application of the kernel, i.e., $K(\mathbf{X}, \mathbf{X})_{m,n} = K(\mathbf{x}_m, \mathbf{x}_n)$. Then, we can easily define the following:

$$\mathbf{z} = \boldsymbol{\alpha}, \quad \mathbf{Q} = \mathbf{Y}K(\mathbf{X}, \mathbf{X})\mathbf{Y}^T, \quad \mathbf{c} = -\mathbf{f}.$$

Having defined $\mathbf{z}$ as such, the constraints of (16′)–(16‴) may be expressed compactly as $\mathbf{Az} \leq \mathbf{b}$ with

$$\mathbf{A} = \begin{bmatrix} \mathbf{y}^T \\ \mathbf{I}_{L+M} \\ -\mathbf{I}_{L+M} \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} 0 \\ C\mathbf{1}_{L,1} \\ \infty\mathbf{1}_{M,1} \\ \mathbf{0}_{L,1} \\ \infty\mathbf{1}_{M,1} \end{bmatrix}$$

where

$\mathbf{I}_m$ size m identity matrix;
$\mathbf{1}_{m,n}$ $m \times n$ matrix of ones;
$\mathbf{0}_{m,n}$ $m \times n$ matrix of zeros.

It is taken as understood for this matrix representation that the first constraint is an equality constraint, whereas the remaining $2(L + M)$ constraints are inequality constraints.

B. Illustrative Example[3]

An application of the ECSVM is for the detection of user data in a code division, multiple access (CDMA) modulation scheme. The classification problem can be summarized as trying to match the optimum decision regions shown in Fig. 6. The input variables y_1 and y_2 correspond to the outputs of matched filters having a correlation of $\rho = 0.5$. Each region corresponds to a different combination of user bits: either $\{-1, -1\}$, $\{-1, +1\}$, $\{+1, -1\}$, or $\{+1, +1\}$. Although the optimum boundaries vary with ρ, they always pass through the origin because of the symmetry of the problem. Hence, we use the ECSVM with one equality constraint $[f([0 \quad 0]^T) = 0]$ to pin the decision boundary to the origin.

Fig. 7 shows an example of decision regions for an M-ary SVM trained on the 100 data points shown on the plot. The polynomial kernel was also used for this example, with $d = 5$ and $C = 0.05$. It is evident in Fig. 7 that the classification boundaries do not pass through the origin. On the other hand, the M-ary ECSVM trained on the same data using the same kernel and constants yields the decision regions shown in Fig. 8, where now, the boundaries pass through the origin. In this application, the ECSVM yields noticeably improved performance over the SVM. The QP optimization for the ECSVM requires more processing time than that for the SVM. In addition, the single equality constraint of this CDMA application poses no feasibility difficulties.

[3]Examples in this section were done with MATLAB 5.3 and its QP routine.

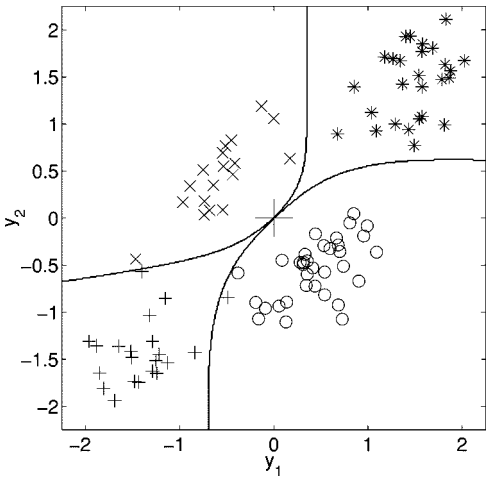

Fig. 8. Example decision boundaries of an M-ary SVM with an equality constraint forcing the border to pass through the origin. The data is the same as that for Fig. 7.

V. CONCLUSIONS

The proposed M-ary SVM is a novel method to perform multicategory classification. It has been contrasted against such approaches as pairwise classification, winner-takes-all strategies, and decision trees. A support vector machine has the advantage that its inherent optimization criteria appropriately adapts a nonlinear model adequately complex for the classification problem at hand. M-ary classification capitalizes on this property by as-

suming a compact representation for classes, whereas individual support vector machines build subregion classifiers of varying complexity. This approach rids us of the need for a secondary processing step, which is a characteristic of many current multicategory classification methods.

The equality constrained SVM is a useful method for forcing the decision boundary of a conventional support vector machine to pass through selected points. The fact that the SVM discriminant function is linear in its feature space variables means that it can be constrained equal to zero at chosen points with a slight modification to the QP optimization. There is the possibility that too many equality constraints could lead to an infeasible solution. However, with a single constraint, this is generally not a problem. The equality constrained SVM can be more computationally demanding in its optimization stage compared against that of the conventional SVM.

REFERENCES

[1] V. N. Vapnik, *Statistical Learning Theory*. New York: Wiley, 1998.

[2] F. S. Hillier and G. J. Lieberman, *Introduction to Mathematical Programming*. New York: McGraw-Hill, 1995.

[3] H. Stark and J. W. Woods, *Probability, Random Processes, and Estimation Theory for Engineers*. Englewood Cliffs, NJ: Prentice-Hall, 1986.

[4] L. Devroye, L. Györfi, and G. Lugosi, *A Probabilistic Theory of Pattern Recognition*. New York: Springer, 1996.

[5] E. Parzen, "On the estimation of a probability density function and the mode," *Ann. Math. Stat.*, vol. 33, pp. 1065–1076, 1962.

[6] U. H.-G. Kreßel, "Pairwise classification and support vector machines," in *Advances in Kernel Methods: Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA: MIT Press, 1999, ch. 15, pp. 255–268.

[7] J. Weston and C. Watkins. (1998) Multi-class support vector machines. Dept. Comput. Sci., Univ. London, Egham, U.K.. [Online]. Available: http://www.cs.rhbnc.ac.uk/jasonw/

[8] O. L. Mangasarian, "Multisurface method of pattern recognition," *IEEE Trans. Inform. Theory*, vol. IT-14, pp. 801–807, Nov. 1968.

[9] ——, "Linear and nonlinear separation of patterns by linear programming," *Oper. Res.*, vol. 13, pp. 444–452, 1965.

[10] K. P. Bennett and O. L. Mangasarian, "Robust linear programming discrimination of two linearly inseparable sets," *Optimi. Methods Softw.*, vol. 1, pp. 23–34, 1992.

[11] F. W. Smith, "Pattern classifier design by linear programming," *IEEE Trans. Comput.*, vol. C-17, pp. 367–372, Apr. 1968.

[12] P. S. Bradley and O. L. Mangasarian, "Feature selection via concave minimization and support vector machines," in *Proc. Machine Learning Fifteenth Int. Conf.*, J. Shavlik, Ed. San Francisco, CA, 1998, pp. 82–90.

[13] K. P. Bennett and E. J. Bredensteiner, "Geometry in learning," in *Geometry at Work*, C. Gorini, E. Hart, W. Meyer, and T. Phillips, Eds. Washington, DC: Math. Assoc. Amer., 1998.

[14] K. P. Bennett and O. L. Mangasarian, "Multicategory separation via linear programming," *Optim. Methods Softw.*, vol. 3, pp. 27–39, 1993.

[15] ——, "Serial and parallel multicategory discrimination," *SIAM J. Optim.*, vol. 4, no. 4, pp. 722–734, 1994.

[16] E. J. Bredensteiner and K. P. Bennett, "Multicategory classification by support vector machines," *Comput. Optim. Appl.*, vol. 12, pp. 53–79, Jan. 1999.

[17] K. P. Bennett, "Combining support vector and mathematical programming methods for classification," in *Advances in Kernel Methods: Support Vector Learning*, B. Schölkopf, C.J. C. Burges, and A. J. Smola, Eds. Cambridge, MA: MIT Press, 1999, ch. 19, pp. 307–326.

[18] O. L. Mangasarian and D. R. Musicant, "Data discrimination via nonlinear generalized support vector machines," Comput. Sci. Dept., Univ. Wisconsin, Madison, WI, Tech. Rep. 99-03, Mar. 1999.

[19] L. Kaufman, "Solving the quadratic programming problem arising in support vector classification," in *Advances in Kernel Methods: Support Vector Learning*, B. Schölkopf, C. J. C. Burges, and A. J. Smola, Eds. Cambridge, MA: MIT Press, 1999, ch. 10, pp. 147–167.

[20] D. J. Sebald and J. A. Bucklew, "Support vector machine techniques for nonlinear equalization," *IEEE Trans. Signal Processing*, vol. 48, pp. 3217–3226, Nov. 2000.

[21] G. Wahba, Y. Lin, and H. Zhang, "Generalized approximate cross validation for support vector machines, or, another way to look at margin-like quantities," Dept. Statist., Univ. Wisconsin, Madison, Tech. Rep. 1006, Apr. 1999.

[22] M. C. Ferris and T. S. Munson, "Interfaces to PATH 3.0: Design, implementation and usage," *Comput. Optim. Appl.*, vol. 12, pp. 207–227, 1999.

[23] F. Gray, "Pulse code communication," U.S. Patent 2 632 058, 1953.

[24] L. Goddyn, G. M. Lawrence, and E. Nemeth, "Gray codes with optimized run lengths," *Util. Math.*, vol. 34, pp. 179–192, 1988.

[25] V. E. Vickers and J. Silverman, "A technique for generating specialized Gray codes," *IEEE Trans. Comput.*, vol. C-29, pp. 329–331, Apr. 1980.

[26] J. P. Robinson and M. Cohn, "Counting sequences," *IEEE Trans. Comput.*, vol. C-30, pp. 17–23, Jan. 1981.

[27] J. E. Ludman, "Gray code generation of MPSK signals," *IEEE Trans. Commun.*, vol. COM-29, pp. 1519–1522, Oct. 1981.

[28] J. E. Ludman and J. L. Sampson, "A technique for generating Gray codes," *J. Statist. Plann. Inference*, vol. 5, pp. 171–180, 1981.

[29] D. G. Wagner and J. West, "Construction of uniform Gray codes," *Congr. Numer.*, vol. 80, pp. 217–223, 1991.

[30] G. S. Bhat and C. D. Savage, "Balanced Gray codes," *Electron. J. Combin.*, vol. 3, no. 1, p. R25, 1996.

[31] C. D. Savage, "A survey of combinatorial Gray codes," *SIAM Rev.*, vol. 39, no. 4, pp. 605–629, 1997.

[32] D. G. Luenberger, *Optimization by Vector Space Methods*. New York: Wiley, 1969.

[33] J. J. Moré and S. J. Wright, *Optimization Software Guide*, ser. Frontiers in Applied Mathematics. Philadelphia, PA: SIAM, 1993, vol. 14.

Daniel J. Sebald (S'89–M'01) received the B.S. degree from the Milwaukee School of Engineering, Milwaukee, WI, in 1987, the M.S. degree from Marquette University, Milwaukee, in 1992, and the Ph.D. degree from the University of Wisconsin, Madison, in 2000, all in electrical engineering.

He is a registered P.E. in the state of Wisconsin and has worked for Camtronics Medical Systems, Hartland, WI; Nicolet Instrument Technologies, Madison; Xyte, Madison; and OB Scientific, Germantown WI. His research interests include signal processing, image processing, communications, real-time DSP, and medical technology.

James A. Bucklew received the Ph.D. degree in electrical engineering from Purdue University, West Lafayette, IN, in 1979. He is currently a Professor with the Department of Electrical and Computer Engineering and the Department of Mathematics at the University of Wisconsin, Madison. He is interested in the general area of statistical signal processing and applied probability and has published well over 100 papers in these areas. He is the author of *Large Deviation Techniques in Decision, Simulation, and Estimation*.

Dr. Bucklew has served in the past as Associate Editor for the IEEE TRANSACTIONS ON INFORMATION THEORY and the IEEE TRANSACTIONS ON SIGNAL PROCESSING.