Published in final edited form as:

IEEE Trans Neural Netw. 2010 May ; 21(5): 701–717. doi:10.1109/TNN.2010.2041069.

Margin-Maximizing Feature Elimination Methods for Linear and Nonlinear Kernel-Based Discriminant Functions

Yaman Aksu,

Electrical Engineering Department, Pennsylvania State University, University Park, PA 16802 USA

David J. Miller,

Electrical Engineering Department, Pennsylvania State University, University Park, PA 16802 USA

George Kesidis [Senior Member, IEEE], and

Electrical Engineering and Computer Science and Engineering Departments, Pennsylvania State University, University Park, PA 16802 USA

Qing X. Yang

Department of Radiology, Pennsylvania State College of Medicine, Hershey, PA 17033 USA

Yaman Aksu: ya1@psu.edu; David J. Miller: djmiller@engr.psu.edu; George Kesidis: kesidis@engr.psu.edu; Qing X. Yang: qyang@hmc.psu.edu

Abstract

Feature selection for classification in high-dimensional spaces can improve generalization, reduce classifier complexity, and identify important, discriminating feature “markers.” For support vector machine (SVM) classification, a widely used technique is recursive feature elimination (RFE). We demonstrate that RFE is not consistent with margin maximization, central to the SVM learning approach. We thus propose *explicit* margin-based feature elimination (MFE) for SVMs and demonstrate both improved margin and improved generalization, compared with RFE. Moreover, for the case of a nonlinear kernel, we show that RFE assumes that the squared weight vector 2-norm is strictly decreasing as features are eliminated. We demonstrate this is not true for the Gaussian kernel and, consequently, RFE may give poor results in this case. MFE for nonlinear kernels gives better margin and generalization. We also present an extension which achieves further margin gains, by optimizing only *two* degrees of freedom—the hyperplane’s intercept and its squared 2-norm—with the weight vector orientation fixed. We finally introduce an extension that allows margin slackness. We compare against several alternatives, including RFE and a linear programming method that embeds feature selection within the classifier design. On high-dimensional gene microarray data sets, University of California at Irvine (UCI) repository data sets, and Alzheimer’s disease brain image data, MFE methods give promising results.

Index Terms

Alzheimer’s; classifier margin; discriminant function; feature elimination; Gaussian kernel; margin maximization; medical imaging; microarray; magnetic resonance imaging (MRI); neurodegenerative; polynomial kernel; recursive feature elimination (RFE); support vector machine (SVM)

I. Introduction

A. Feature Selection in Classification

In high-dimensional domains such as image and image sequence classification, text categorization, and gene microarray classification, one often encounters problems where there are very few labeled training samples, or at any rate few samples *relative to* the (high) dimensionality of the feature measurements for each exemplar/sample. In biomedical imaging and bioinformatics in particular, with training databases derived, e.g., from clinical trials, there may be at most several hundred (patient) samples, each represented by features such as voxels in the hundreds of thousands or gene microarray or text features in the tens of thousands. In these domains, there are compelling reasons for reducing feature dimensionality. First, many features may have at best weak discrimination power. In [25], a type of “curse of dimensionality” (COD) was demonstrated wherein, for fixed sample size, the generalization accuracy may *degrade* as feature dimensionality is increased beyond a certain point. This phenomenon is related to the bias-variance dilemma in statistics [14], which suggests that, for best generalization, model complexity should be matched to available training data resources. Even in domains where generalization accuracy tends to monotonically improve with feature dimensionality, complexity of the classification operation (both computation and memory storage for decision making) may outweigh marginal gains in accuracy achieved by using a large number of features. Finally, in some contexts, it is useful to identify a small subset of features necessary for making good predictions—these “markers,” e.g., anatomical markers in biomedical imaging or gene “biomarkers” in bioinformatics, may shed light on the underlying disease mechanism. Decision making based on a small set of features is also highly interpretable, which is important for explaining how decisions are reached.

There are several approaches for avoiding model overfitting/COD. One is to fit the original high-dimensional data (with M features) using simple models, e.g., naive Bayes models [7], [8]. Another is to limit the amount of model training, e.g., via regularization or early stopping [7]. Support vector machines (SVMs) attempt to avoid overfitting by finding a discriminant function that maximizes the *margin*, i.e., the minimum distance of any sample point to the decision boundary. For a linear SVM, the number of free parameters is upper-bounded by the number of training samples (a subset of which are support vectors at margin distance to the hyperplane), rather than controlled by the feature dimensionality. However, SVMs are *not* immune to the curse of dimensionality [14]. Thus, *feature selection*, wherein only a small subset of the original features are retained, or *feature compaction*, wherein linear or nonlinear transformations map the original features to a new, smaller coordinate space, are often essential for achieving good, *generalizable* classification accuracy. The latter (compaction) approach loses physical interpretation of the original features, which is undesirable when aiming to identify feature “markers.” Thus, we focus here on feature selection.

Unfortunately, there are $2^M - 1$ possible feature subsets, with exhaustive search practically prohibited even for modest M , let alone M in the thousands. Practical feature selection techniques are thus heuristic. There are a variety of methods, exercising a large range of tradeoffs between accuracy and complexity [13]. “Front-end” (or “filtering”) methods select features prior to classifier training, based on evaluation of discrimination power for individual features or small feature groups. “Wrapper” methods use classifier training repeatedly to evaluate the classification accuracy of numerous candidate feature subsets. There are also “embedded” methods, e.g., SVM training with a regularization penalty to suppress irrelevant features. For wrapper methods, there is greedy forward selection, with “informative” features added, backward feature elimination, which starts from the full space and removes features, and more complex bidirectional searches such as simulated annealing

[13]. Backward search starts by assessing joint predictive power of all the features. In principle, one would like to retrain the classifier in conjunction with each backward elimination step that removes a feature subset (optimizing the classifier for the new feature space). However, considering large M and assuming one feature eliminated per step, this requires either M classifier retrains (if retraining is done after a feature elimination) or $M(M-1)/2$ (if retraining is done after *trial-elimination* of every remaining feature). For SVM-based classifier training, even with recent advances that significantly reduce training time [15], [16], it may not be practically feasible to retrain at each step for M in the tens or hundreds of thousands. Thus, for large M , retraining may only be done periodically, after a “batch” of feature eliminations.

B. Feature Selection for SVMs

SVMs have become nearly a standard technique in many domains. There are a number of reasons. First, the SVM objective, maximizing margin, has a theoretical basis tied to achievement of good generalization accuracy [6]. Second, there is a unique, globally optimal solution to the SVM training problem. Third, there are improvements in representation power via nonlinear kernels, which map to a high or even infinite-dimensional feature space and, via the “kernel trick,” do so *without* huge increase in decision making and classifier training complexities. Fourth, SVMs achieve good results on a variety of domains.

Front-end feature selection has been applied for SVMs in numerous prior works, e.g., [19] and [20]. Wrapper-based selection has also been applied in many prior works. Weston *et al.* [26] reduced wrapper complexity by replacing the SVM training objective with an upper bound that is less complex to optimize. A widely used method analyzed in this paper is recursive feature elimination (RFE) [12], wherein at each step one removes the feature with least weight magnitude in the SVM solution. This method is very lightweight and thus easily scales to large M . In [21], a wrapper approach was used, with SVM retraining performed after RFE removed a batch of features.

Embedded feature selection methods, e.g., [27] and [9], reformulate SVM training to encourage feature sparsity in the solution. Different norms and optimization approaches have been investigated, e.g., [3], [9], [18], and [28]. The approach in [3] is based on the same standard SVM formulation used in this paper but builds in feature selection by imposing an upper bound on the number of nonzero weights as an additional constraint on the optimization problem. The approach in [28] is based on the Lasso [24] and uses the ℓ_1 -norm, instead of the ℓ_2 -norm, which encourages feature sparsity. The approach in [9] is based on a hybrid $\ell_1 - \ell_2$ -norm minimization, solved by linear programming; this LP-SVM method uses Newton-type iterations and thus is denoted NLPSVM.

In this paper, we compare performance of our methods with a representative embedded method—NLPSVM [9]. Our paper’s main focus, however, is wrapper-based feature elimination for linear and nonlinear kernel-based classifiers (including SVMs). In [12], Guyon *et al.* essentially argue that the RFE objective for linear SVMs is consistent with the SVM objective of margin maximization. They note that the SVM primal problem poses minimization of the squared weight vector 2-norm subject to (margin-related) constraints on each training point. Eliminating the feature with smallest weight magnitude has the least effect on the squared weight vector 2-norm and, thus, Guyon *et al.* [12] argue, on the SVM solution. In this paper, however, we show experimentally that RFE is not in general in close agreement with margin maximization. The reason is that RFE ignores the margin constraints, focusing solely on minimally reducing the squared weight vector 2-norm. In this work, we first develop a method that explicitly performs margin-based backward feature elimination (MFE) for linear SVMs. We then consider nonlinear kernels. We show that RFE defined for the kernel case [12] assumes that the squared weight vector 2-norm is strictly

decreasing as features are eliminated. We demonstrate experimentally that this assumption is not valid for the Gaussian kernel and that, consequently, RFE may give poor results in this case. MFE for the nonlinear kernel case experimentally gives both better margin and generalization accuracy. We then present an MFE extension which stepwise (greedily) achieves further gains in margin at small additional computational cost. This extension solves an SVM optimization problem to maximize the classifier's margin at each feature elimination step, albeit in a very lightweight fashion by optimizing only over a small set of parameters, very similar to a method suggested in [10]. Finally, we develop an MFE extension that allows margin slackness.

While embedded methods do give a potential alternative to our (wrapper-based) approach, previous studies such as [3], [9], [18], and [28] have not demonstrated superior performance compared to wrapper approaches such as RFE. Here, we compare MFE with both RFE and the embedded method proposed in [9].

The main contributions of this paper are thus: 1) development of novel, margin-based feature selection methods for linear and kernel-based discriminant functions addressing separable and nonseparable contexts; 2) identifying some difficulties with the well-known RFE method, especially in the kernel case; and 3) an objective experimental comparison of several feature selection methods, which also evaluates consistency between a classifier's margin and its generalization accuracy.

Section II gives a brief review of SVMs. Section III discusses limitations of RFE. Sections IV and V develop our new methods. Section VI gives experimental comparisons. Section VII gives the paper conclusions.

II. Brief Review of SVMs

Consider a labeled training set $\{(\underline{x}_n, y_n), n = 1, \dots, N\}$, where $y_n \in \{\pm 1\}$ is the class label and $\underline{x}_n = (x_{n,1}, \dots, x_{n,M}) \in \mathbf{R}^M$ is the n -th data sample. A hyperplane acting as a binary (two-class) decision function in this M -dimensional space is defined by $f(\underline{x}) \equiv \underline{w}^T \underline{x} + b = 0$, $\underline{w} \in \mathbf{R}^M$, $b \in \mathbf{R}$. Denoting $g(\underline{x}_n) \equiv y_n f(\underline{x}_n)$, the signed distance from a data point $\underline{x}_n$ to the decision boundary is $g(\underline{x}_n)/\|\underline{w}\|$. The decision boundary is a *separating* one if it satisfies $g(\underline{x}_n) > 0$ for all $\underline{x}_n$. The margin of the separating decision boundary is thus defined as $\gamma \equiv \min_n g(\underline{x}_n)/\|\underline{w}\|$. An SVM is a linear or generalized linear two-class classifier that learns a separator for the training set with *maximum* margin. The “support vectors,” used to specify the SVM solution, which we denote by $\{\underline{s}_1, \dots, \underline{s}_T\}$ (with index set $\mathcal{S} = \{1, 2, \dots, T\}$), are a subset of the training points at margin distance to the decision boundary. In the linear case, the SVM weight vector solution is $\underline{w} \equiv \sum_{k \in \mathcal{S}} \lambda_{s_k} y_{s_k} \underline{s}_k$, where λ_{s_k} are scalar (positive) Lagrange multipliers. In the generalized linear (nonlinear) case, $f(\underline{x}) \equiv \underline{w}^T \underline{\phi}(\underline{x}) + b = 0$, $\underline{w} \in \mathbf{R}^L$, $b \in \mathbf{R}$, $\underline{\phi}(\underline{x}) \equiv [\phi_1(\underline{x}), \phi_2(\underline{x}), \dots, \phi_L(\underline{x})]^T$, with $\phi_l(\underline{x})$ nonlinear functions of the $\underline{x}$ coordinates. Of particular interest is when inner products between $\underline{\phi}(\underline{x})$ and $\underline{\phi}(\underline{u})$ can be efficiently computed via a positive-definite kernel function, $K(\underline{x}, \underline{u}) \equiv \underline{\phi}^T(\underline{x}) \underline{\phi}(\underline{u})$. In this case, both $\underline{\phi}(\cdot)$ and $\underline{w}$ itself need not be explicitly defined since both the SVM discriminant function $f(\cdot)$ and the SVM weight vector squared 2-norm can be expressed solely in terms of the kernel, i.e.,

$$f(\underline{x}) = \sum_{k \in \mathcal{S}} \lambda_{s_k} y_{s_k} K(\underline{s}_k, \underline{x}) + b \quad (1)$$

$$\|\underline{w}\|^2 = \sum_{k \in \mathcal{S}} \sum_{l \in \mathcal{S}} \lambda_{s_k} y_{s_k} \lambda_{s_l} y_{s_l} K(\underline{s}_k, \underline{s}_l). \quad (2)$$

This approach (the “kernel trick”), where $K(\cdot, \cdot)$ is explicitly specified and provided to the SVM training, is referred to as the “nonlinear kernel case.”

For both linear or nonlinear kernels, the basic SVM training problem is

$$\min_{w,b} \frac{1}{2} \|w\|^2 \quad \text{s.t.} \quad y_n f(x_n) \geq 1, \quad n=1, \dots, N. \quad (3)$$

The SVM training problem can alternatively be posed as

$$\begin{aligned} \min_{w,b,\xi} \quad & \frac{1}{2} \|w\|^2 + C \sum_{n=1}^N \xi_n \\ \text{s.t.} \quad & \xi_n \geq 0, y_n f(x_n) \geq 1 - \xi_n, \quad n=1, \dots, N \end{aligned} \quad (4)$$

so as to allow slackness (ξ) in the margin constraints; (4) allows some support vectors to be practically closer than others to the hyperplane (by nonnegative slackness amounts ξ_n), thus handling both margin violations (i.e., $\xi_n > 0$) and nonseparable data (a classification error occurs for sample n if $\xi_n > 1$). For choosing the SVM training parameter C as well as other SVM hyperparameters in the nonlinear kernel case, the standard practice of using a validation or cross-validation procedure [7] can be employed.

1) Data Dimensionality and Separability—Cover’s linear dichotomy theorem [5] states that the probability that a training set (with points in general position) is linearly separable is very close to 1 if $N \gg M + 1$. As an example, for the gene microarray domain, it is typical to have, e.g., $M \approx 7000$ and N no larger than a few hundred patient samples; in this case, it is *highly probable* that the training set will be separable while eliminating all the way down to a few hundred features.

III. Limitations of RFE

A. Limitations of RFE: Linear Kernel Case

RFE is a stepwise (greedy), backward feature elimination technique for SVMs. Under RFE [12], the index m^* of the first feature to be eliminated is $\arg \min_{m \in \{1, \dots, M\}} |w_m|$, and, more generally, at step i , this same selection rule is applied to the $M - i$ remaining features, with SVM retraining (optionally) applied after a batch of features is eliminated in this way. While [12] does suggest a close tie between the RFE choice and the SVM objective (margin maximization), RFE is equivalent to margin-maximizing feature elimination if and only if (5) below is always satisfied, with RFE’s margin on the right and the margin achieved by an approach which *explicitly* eliminates the feature that preserves maximum margin on the left

$$\max_m \min_n \frac{y_n f(x_n) - y_n x_{n,m} w_m}{\sqrt{\|w\|^2 - w_m^2}} = \min_n \frac{y_n f(x_n) - y_n x_{n,m^*} w_{m^*}}{\sqrt{\|w\|^2 - w_{m^*}^2}}. \quad (5)$$

In Fig. 1, we prove via a simple 2-D counterexample that eliminating features according to RFE is not equivalent to preserving maximum margin. In general, direct margin maximization, achieved by the MFE method we develop in Section IV, leads to significant gains in margin and may also lead to improved generalization accuracy over RFE, as we demonstrate in the sequel.

B. Limitations of RFE: Nonlinear Kernel Case

For the case of a nonlinear kernel, a natural extension of RFE was proposed in [12]. Of particular interest, for the discussion that follows, is the choice of the Gaussian kernel $K(\underline{u}, \underline{v}) = \exp(-\beta \|\underline{u} - \underline{v}\|^2)$, $\beta > 0$. In [12], it was proposed to evaluate the square of the weight vector 2-norm (2) both before and after a candidate feature elimination and, thus, at the i th stage of feature elimination, to remove the feature that minimizes the difference

$$\Delta \|\underline{w}\|^2 = (\|\underline{w}\|^2)^{i-1, m_{i-1}^*} - (\|\underline{w}\|^2)^{i, m_i^*}. \quad (6)$$

This criterion is the natural extension of the linear RFE criterion and is consistent with the objective of reducing the square of the weight vector 2-norm the least, assuming that the square of the weight vector 2-norm is in fact monotonically decreasing as the feature dimensionality is reduced. For example, in the case of the polynomial kernel $K(\underline{u}, \underline{v}) = (1 + \underline{u}^T \underline{v})^d$, the kernel's mapping function maps an original feature vector $\underline{u}$ to a new, finite-dimensional feature vector $\phi(\underline{u})$ whose coordinates $\phi_j(\underline{u})$ are products, raised to powers, of the original feature coordinates. Thus, it is clear for the polynomial case that eliminating an original feature coordinate zeroes out one or more coordinates of $\phi(\underline{u})$ while leaving all others unchanged. This *effects* zeroing (removing) the associated scalar weights. Thus, the squared weight vector 2-norm *is* monotonically decreasing as original features are eliminated. However, we have also considered the Gaussian kernel. Since it is not so easy to analytically evaluate the Gaussian case [22], we have simply measured the squared weight vector 2-norm experimentally and have found it is *neither* monotonically decreasing *nor* monotonically increasing with feature eliminations. Consider the consequences for the RFE objective (6): the RFE-optimal feature elimination (assuming some eliminations increase the weight vector 2-norm) will in fact be the feature whose removal *increases* the squared weight vector 2-norm the most—this is the choice that will decrease (6) as much as possible (only, in this case, making $\Delta \|\underline{w}\|^2$ *negative*).

In Fig. 2, on the *UCI arrhythmia* data set, for one “trial”¹ using the Gaussian kernel, we evaluated both RFE and a modified method we dub RFE-abs which eliminates the feature that results in the smallest *change* (either decrease or increase) in the weight vector squared norm (this method is based on [17]). Note that for standard RFE there is initially a significant rise in the squared weight vector 2-norm as features are eliminated, over a range of feature eliminations (i.e., from 225 to 279 features retained). Over this range, both the margin and test set error rate of standard RFE are worse than those of RFE-abs. (We give results for RFE and RFE-abs for more UCI data sets in Section VI.) We further note, however, that RFE-abs is *itself* quite suboptimal with respect to classifier margin—standard RFE becomes nonseparable (with margin going to zero) when 250 features remain, with RFE-abs becoming nonseparable soon afterwards, when 225 features remain. Also shown in Fig. 2 are results for a kernel-based version of the MFE method, which achieves both greater margin and lower test set error rates, compared with the RFE methods. This MFE approach is developed in the next section.

¹A “trial,” defined in Section IV-A and summarized here, is a random 50%–50% split of the data set into a heldout set and a nonheldout set. For the trial, we train the SVM on a random 90% subset of the nonheldout set, and evaluate on the remaining 10% (validation set). SVM hyperparameters are then chosen to minimize the average validation error, measured over five such training/validation splits. We then train an SVM on the entire nonheldout set to obtain the initial classifier used both by our methods and RFE. Feature elimination is then performed, with margin measured on this (nonheldout) set and error rate measured on the heldout (test) set.

IV. MFE: Direct Margin-Based Feature Elimination

A. MFE for the Linear Kernel Case

Since maximizing margin is the (theoretically motivated) goal of SVM training [6], it is expected that eliminating features to preserve the largest (positive) margin should yield classifier solutions that generalize better than solutions with smaller margin. The example in Fig. 1 illustrates both maximum margin-preserving feature elimination and the fact that RFE does not in general achieve this elimination. Surprisingly, while there are some related approaches [17],² we had not seen direct, MFE previously proposed. In recent work [1], [2], we developed just such a technique.

Our MFE method works from a current classifier that is a separator of the training set and, at each step, eliminates the feature m_{MFE} that preserves the largest (positive) training set margin (achieved by sample n_{MFE}), i.e., see (7) shown at the bottom of the page, with S the set of valid candidate features, whose single eliminations will preserve positive margin. Elimination based on (7) is illustrated in the example in Fig. 1.

$$(m_{\text{MFE}}, n_{\text{MFE}}) = \arg \max_{m \in S \equiv \{m' | y_n' f(x_n') - y_n' x_{n,m}' w_m' > 0, \forall n'\}} \min_n \frac{y_n f(x_n) - y_n x_{n,m} w_m}{\sqrt{\|w\|^2 - w_m^2}} \quad (7)$$

We emphasize that feature selection is most urgently needed when M is very high, and that N can often be *quite* small, such as in medical imaging and bioinformatics (e.g., gene microarray) contexts, and, citing [5] in Section II, we conveyed it is highly probable separability can be achieved by a linear classifier in such cases. Even for some intermediate-to-low dimensional domains (e.g., UCI data), we will demonstrate that the training set is both initially separable and remains separable while a significant number of features are eliminated (with margin used as the feature elimination criterion). Thus, we argue that the set S in (7) is nonempty in many practical cases and, thus, using (positive) margin as the elimination criterion is feasible in practice, especially for high-dimensional domains, where feature selection is also most urgently needed.

On the other hand, to handle the case where the training set is *nonseparable*, we propose (in Section V) an extension of MFE that allows for margin slackness and nonseparability.

We next give pseudocode for our basic MFE method.

1) MFE Algorithm Pseudocode for SVMs: Linear Kernel Case—Notation: $q^{i,m} \equiv$ quantity q at feature elimination step i upon elimination of feature m .

- 0** *Preprocessing:* Let $\mathcal{M}$ be the set of eliminated features, with $\mathcal{M} = \emptyset$ initially. First run SVM training on the full space to find a separating hyperplane $f(\underline{x}) = 0$ (with f parameterized by $\underline{w}, b$), with weight norm-squared $L^{-1,0} \equiv \|\underline{w}\|^2$, where $i = -1$ means before eliminating any features and $m_{-1} = 0$ is a dummy placeholder index value. For each feature m , compute $\delta_n^m = y_n x_{n,m} w_m \forall n$. Recall that $g(\underline{x}_n) \equiv y_n f(\underline{x}_n)$ so that δ_n^m is the Δg quantity $\delta_n^{j,m} \equiv (g_n^{j-1, m_{j-1}} - g_n^{j,m})$ whose value is the same at every elimination step for a given (m, n) pair. Compute

²This reference eliminates features to maximize the average distance to the hyperplane, over all training points, rather than to maximize margin.

$g_n^{-1,0} = y_n b + \sum_{m=1}^M \delta_n^m \forall n$. Set $i \leftarrow 0$. At elimination step i , perform the following operations.

- 1 For each $m \notin \mathcal{M}$, using recursion, compute $g_n^{i,m} = g_n^{i-1,m_{i-1}} - \delta_n^m \forall n$, determine $\mathcal{N}_n^{i,m} = \min_n g_n^{i,m}$. Determine the candidate feature set $S(i) = \{m \notin \mathcal{M} \mid \mathcal{N}_n^{i,m} = 0\}$. Note that δ_n^m need not be computed in this step if stored for all m and n during preprocessing (step 0). If $S(i)$ is empty (the data is nonseparable) then **stop**.
- 2 For $m \in S(i)$, using recursion, compute $L^{i,m} = L^{i-1,m_{i-1}} - w_m^2$, determine $\gamma^{i,m} = \max_{m \in S(i)} \mathcal{N}_n^{i,m} / \sqrt{L^{i,m}}$.
- 3.1 Eliminate feature $m_i \equiv \arg \max_{m \in S(i)} \gamma^{i,m}$, i.e., $\mathcal{M} \rightarrow \mathcal{M} \cup \{m_i\}$.
- 3.2 Keep for the next iteration only the recursive quantities $\{g_n^{i,m_i} \forall n\}$, L^{i,m_i} , associated with the eliminated feature.
- 3.3 $i \rightarrow i + 1$ and go to step 1.

In Fig. 3(a), we demonstrate that MFE achieves both larger margins and better overall generalization performance (test set error rate) than RFE on the gene microarray *Leukemia* data set.³ For this data set, the average number of retained features at which separability was lost under MFE was 90, and thus the curves are shown for the range of 90–7129 features retained. Similar results are achieved on other microarray data sets, as well as data sets from the UCI repository. More extensive evaluations are given in Section VI.

B. MFE for the Nonlinear Kernel Case

To address the suboptimality of RFE (and RFE-abs) described in Section III-B, similar to the pseudocode in Section IV-A1 we propose a recursively-implemented MFE algorithm, now for *kernel-based* SVMs. In this case, the recursion is on the kernel computation. For example, for the Gaussian kernel, denoting $\mathcal{K}_{k,n}^{i,m} \equiv K(\underline{s}_k^{i,m}, \underline{x}_n^{i,m})$ at elimination step i , we have the recursion

$$\mathcal{K}_{k,n}^{i,m} = \mathcal{K}_{k,n}^{i-1,m_{i-1}} \exp(\beta(s_{k,m} - x_{n,m})^2) \quad \forall k, \forall n. \quad (8)$$

Likewise, for the polynomial kernel $K(\underline{u}, \underline{v}) = (\beta \underline{u}^T \underline{v} + 1)^d \equiv (H(\underline{u}, \underline{v}))^d$, and denoting $\mathcal{H}_{k,n}^{i,m} \equiv H(\underline{s}_k^{i,m}, \underline{x}_n^{i,m})$ at elimination step i , we have the recursion

$$\mathcal{H}_{k,n}^{i,m} = \mathcal{H}_{k,n}^{i-1,m_{i-1}} - \beta s_{k,m} x_{n,m} \quad \forall k, \forall n \quad (9)$$

$$\mathcal{K}_{k,n}^{i,m} = (\mathcal{H}_{k,n}^{i,m})^d \quad \forall k, \forall n. \quad (10)$$

These recursively computed kernels, which are used to evaluate both the discriminant function $f(\underline{x}_n)$ via (1) and the weight vector norm via (2), form the basis for a kernel-MFE algorithm whose pseudocode implementation is a simple modification of the linear SVM

³The curves in Fig. 3(a) are the result of averaging over ten “trials”—margin measured on a trial’s nonheldout set and error rate measured on the trial’s heldout (test) set are averaged over ten trials.

pseudocode given in Section IV-A1. Our MFE method works from a current classifier that is a separator of the training set and, at each step i , eliminates the feature m_{MFE} (below) that preserves the largest (positive) training set margin⁴:

$$(m_{\text{MFE}}, n_{\text{MFE}}) = \arg \max_{m \in \mathcal{S} \equiv \{m' | g_n^{i,m'} > 0, \forall n'\}} \min_n \frac{g_n^{i,m}}{\|w\|^{i,m}}. \quad (11)$$

Fig. 2(b) and (c) demonstrates substantial increases in margin and generalization performance (much lower error rate) achieved by MFE for the Gaussian case over both RFE and RFE-abs on the *UCI arrhythmia* data set. The number of retained features at which separability was lost under MFE was 36, and thus the margin and test set error rate curves are shown for the range of 36–279 features retained—notice from the margin curves that the data remain separable under MFE for much longer than under RFE or RFE-abs. Again, we give more extensive results for these methods in Section VI.

C. “Little Optimization” (LO): Further Increases in Margin

For large M , it may not be computationally practical to retrain the SVM in the reduced feature space, in conjunction with each feature elimination step. However, a *type* of classifier retraining at every step that is consistent with margin maximization and yet is exceptionally modest computationally, compared to full SVM retraining, is still possible. The idea is to solve the SVM problem but while optimizing drastically fewer parameters than the full complement of SVM feature weights. Let $(\underline{w}^{-\mathcal{M}}, b)$ denote the linear SVM weight vector (and affine parameter) after a set $\mathcal{M}$ of features are eliminated. Suppose we consider the *new* parameterized weight vector $(A \underline{w}^{-\mathcal{M}}, w_0)$, where A and w_0 are *scalar* parameters to be optimized, with $\underline{w}^{-\mathcal{M}}$ held fixed. That is, we allow adjusting the squared weight vector 2-norm and the affine parameter, with the weight vector orientation fixed. We thus pose the standard SVM training problem, but optimizing only in this 2-D parameter space: $\min_{A, w_0} A^2$ s.t. $y_n(A(\underline{w}^{-\mathcal{M}T} \varphi(\underline{x}_n)) + w_0) \geq 1, n = 1, \dots, N$. In the linear kernel case, this problem is given by (12) and in the nonlinear kernel case, it is given by (13)

$$\begin{aligned} \min_{A, w_0} \quad & A^2 \\ \text{s.t.} \quad & y_n(A(\underline{w}^{-\mathcal{M}T} \underline{x}_n^{-\mathcal{M}}) + w_0) \geq 1, \quad n=1, \dots, N \end{aligned} \quad (12)$$

$$\begin{aligned} \min_{A, w_0} \quad & A^2 \\ \text{s.t.} \quad & y_n \left(A \left(\sum_{k \in \mathcal{S}} \lambda_{s_k} y_{s_k} K(\underline{s}_k^{-\mathcal{M}}, \underline{x}_n^{-\mathcal{M}}) \right) + w_0 \right) \geq 1, \quad n=1, \dots, N. \end{aligned} \quad (13)$$

This problem was previously posed in [10], with the solution achieved by use of Newton iterations. Next, we develop an alternative solution that is advantageous in that it is essentially closed form, requiring very little computation. In particular, the feasible region of the problem is defined by two cones in the (A, w_0) plane (one such cone c^+ is shown in Fig. 4), with the minimum squared weight vector 2-norm (A^2) in each cone achieved at the cone’s tip, which is easily found. Thus, the minimization is performed by identifying the tip of each cone and choosing the one with smaller A^2 . Referring to Fig. 4, we prove this as follows. Denoting a slope $m_n \equiv -\underline{w}^{-\mathcal{M}T} \underline{x}_n^{-\mathcal{M}}$ in the SVM linear case (or

⁴Equation (11) specializes to (7) for the case of a linear kernel.

$m_n \equiv -\sum_{k \in \mathcal{S}} \lambda_{\mathbf{x}_k} y_{\mathbf{x}_k} K(\mathbf{s}_k^{-\mathcal{M}}, \mathbf{x}_n^{-\mathcal{M}})$ in the nonlinear kernel case), we rewrite the constraints (12) [or (13)] as two sets of inequalities (S_1 for class 1 and S_2 for class 2) in the (A, w_0) plane: $S_1 = \{w_0 - m_n A + 1 | y_n = 1, n = 1, \dots, N\}$, $S_2 = \{w_0 - m_n A - 1 | y_n = -1, n = 1, \dots, N\}$. In this plane, each inequality in S_1 (one for each data point in class 1) specifies a line. Let L_1 be the set of these lines associated with S_1 . Let L_2 be defined similarly. Using the figure, we can show that the feasible region of (12) (or (13)) in half-space $A > 0$ is the cone c^+ bounded by the line l_2^+ with maximum slope in L_2 and the line l_1^+ with minimum slope in L_1 , and that the (feasible) minimum A^2 in c^+ is at its tip P . Specifically, by their definitions, lines l_1^+ and l_2^+ are known. l_1^+ intersects l_2^+ at a lower point (P) along l_2^+ than any other $l_1 \in L_1$ (which all pass through $(0, 1)$ and are oriented away from l_1^+ in the counterclockwise arrow direction shown; one such line l_1 is shown as a thin solid line in the figure). Similarly, l_2^+ intersects l_1^+ at a higher point along l_1^+ than any other $l_2 \in L_2$ (which all pass through $(0, -1)$; one such line l_2 is shown as a thin dashed line). Thus, in the half-space $A > 0$ the feasible region is c^+ , and the (feasible) minimum A^2 in c^+ is at its tip P . Similarly, the minimum (feasible) A^2 in the other half-space $A < 0$ is at the tip of a corresponding cone c^- bounded by the line l_1^- with maximum slope in L_1 and the line l_2^- with minimum slope in L_2 .

The intersection point $P = (A_{\text{inter}}, w_{0\text{inter}})$ is computed as follows: $A_{\text{inter}} = 2/(m_{\text{max}} - m_{\text{min}})$ (where m_{max} is the slope of l_2^+ and m_{min} is the slope of l_1^+) and $w_{0\text{inter}} = m_{\text{min}} A_{\text{inter}} + 1$. Further, it takes N additional multiplications and additions to scale $f(\mathbf{x}_n) - w_0$ (i.e., m_n) by A_{inter} and add $w_{0\text{inter}}$, creating g_n , $n = 1, \dots, N$, for use at the next elimination step. This LO thus takes only $N + 2$ multiplications and $N + 2$ additions at each elimination step; it just requires first finding the tips of the cones c^+ and c^- , choosing the tip with minimum A^2 , and then performing N multiplications (scalings) and adds (shifts).

Since LO requires so little computation, it can be performed in conjunction with each (margin-optimizing) feature elimination step. This can take place *after* eliminating a feature (LO-Lite), or can be embedded into the elimination decision (LO-Full).⁵ In both cases, at each step, LO is guaranteed to increase the margin that would have been achieved by basic MFE during that step. However, since feature elimination is performed within a greedy (stepwise-optimal) framework, there is no theoretical guarantee that the margin curve for LO-Lite or LO-Full will lie strictly above the margin curve for basic MFE—LO will in general alter the (greedily chosen) sequence of margin-maximizing feature eliminations. This lack of guarantee in fact applies even if *full* SVM retraining is coupled to the feature eliminations; we dub such a procedure “MFE-Retrain.” In Fig. 3(b), we demonstrate a strict increase in margin (averaged over trials) for MFE-LO over basic MFE on the *UCI hepatitis* data set, during the first five feature elimination steps (when the data is still separable). In this case, MFE-LO also achieves a modest reduction in average test set error at 14 retained features. We give more detailed results of these methods in Section VI.

V. MFE-SLACK: Utilizing Margin Slackness

Recall the two SVM training objectives given in Section II, with (3) choosing the weight vector to maximize margin while strictly enforcing margin constraints, and with (4) allowing for some margin slackness [i.e., ξ_n in (4)], including possible misclassifications (i.e., $\xi_n > 1$). Introducing slackness allows classifier design even when the data set is nonseparable. Moreover, strictly satisfying the margin could potentially lead to over-fitting when training samples at the margin are outliers or even *misabeled* samples. Optimizing the amount of

⁵In LO-Full, we perform candidate elimination of each feature and then perform LO in the resulting reduced space. We then pick the candidate elimination that leads to largest post-LO margin.

slackness (by choosing the parameter C), e.g., via cross validation, may yield classifiers with better generalization than those based on strictly maximizing margin. All of these reasons motivate us in this section to propose a feature elimination extension of MFE that allows for margin slackness.

Since the approach we propose uses (4) as the feature elimination objective, it is instructive to first discuss in more detail the objectives (3) and (4) and their relationship to margin maximization. In particular, recall that, assuming we have a separator, the margin is $\gamma = \min_n g(\underline{x}_n)/\|\underline{w}\|$ and, further, note that $g(\cdot) \equiv yf(\cdot)$ can be amplitude-scaled by an arbitrary nonzero constant ρ without altering the decision boundary. In particular, if we form $\tilde{g} = \rho g$, where $\rho = 1/\min_n g(\underline{x}_n)$, then $\gamma = \min_n g(\underline{x}_n)/\|\underline{w}\| = 1/\|\underline{w}\|$. We thus see the well-known result that, for this special choice of ρ , maximizing margin is equivalent to minimizing the squared weight vector 2-norm. Further, we can define $\xi_n = \max(0, 1 - \tilde{g}(\underline{x}_n))$, i.e., consistent with the constraint $\xi_n = 0$ in (4), zero slackness for correctly classified samples at greater than margin distance and positive slackness for samples at less than margin distance to the decision boundary (including possibly misclassified samples). The latter samples will be referred to as “margin violators” in the sequel. Now consider the SVM objective function in (4)—if C is made sufficiently large, no margin slackness will be tolerated and minimizing (4) *reduces to* minimizing the squared weight vector 2-norm and, thus, to maximizing margin. We thus see that (4) is a generalization of strict margin maximization that *specializes* to strict margin maximization when C is made sufficiently large.

Now let us relate this to feature elimination algorithms. In Section IV, at each elimination step, we chose the pair $(m_{\text{MFE}}, n_{\text{MFE}})$ to strictly maximize margin, with m_{MFE} the eliminated feature and n_{MFE} the sample achieving the post-elimination margin. In this section, we will choose the pair $(m_{\text{MFE-S}}, n_{\text{MFE-S}})$ to minimize the objective (4), with both the weight vector 2-norm and the slackness values evaluated *post*-feature-elimination. In this discrete optimization, for each *candidate* feature for elimination m , we must evaluate *every* (correctly classified) sample $\tilde{n}$ as the potential margin-defining sample associated with this elimination (we will dub such margin-defining samples as “anchor” samples). To do so, as discussed above, we find the value $\rho_{\tilde{n}}$ such that, post-elimination of feature m , $\rho_{\tilde{n}} \tilde{g}_{\tilde{n}} = 1$, and we measure the slackness values relative to this anchor sample, i.e., $\xi_n = \max(0, 1 - \rho_{\tilde{n}} \tilde{g}_n) \forall n$, and we plug into the objective function (4). Candidate anchor samples (and associated induced slackness values) are illustrated by an example in Fig. 5. The pair $(m_{\text{MFE-S}}, n_{\text{MFE-S}})$ that, post-elimination, minimizes (4) over all the discrete choices $\{(m, n)\}$ is selected, which thus determines the eliminated feature $m_{\text{MFE-S}}$.

From the discussion above on the objective functions (3) and (4), it should be clear that, if C is made sufficiently large (and assuming the data is separable), nonzero slackness values will again not be tolerated. In this case, choosing $(m_{\text{MFE-S}}, n_{\text{MFE-S}})$ to minimize (4) *precisely* reduces to choosing the pair to maximize margin, i.e., the discrete optimization problem minimizing (4) *precisely* reduces to the problem solved by the standard MFE algorithm from Section IV. In summary, the method we propose in this section is a natural extension of MFE to incorporate slackness that in fact *precisely* reduces to MFE for large enough C when the data are separable. We next present pseudocode for this algorithm, which we will (justifiably) refer to as “MFE-slack.”

A. MFE-Slack Algorithm Pseudocode for Linear and Nonlinear Kernel SVMs

Notation: $q^{i,mcna} \equiv$ quantity q at elimination step i upon elimination of candidate feature m_c when the candidate anchor is $\underline{x}_{n_a}$.

- 0 *Preprocessing*: Let $\mathcal{M}$ be the set of eliminated features, with $\mathcal{M} = \emptyset$ initially. First, run SVM training on the full space to find the initial hyperplane $f(\underline{x}) = 0$.

The hyperplane need not be a separating one; see Section VI for our training procedure. Recalling that $g(\underline{x}_n) \equiv y_n f(\underline{x}_n)$, compute $\{g_n^{-1,0} \forall n\}$ and $(\|\underline{w}\|^2)^{-1,0}$, where $i = -1$ means before eliminating any features and $m_c = 0$ is a dummy placeholder index value. Set $i \leftarrow 0$. At elimination step i , perform the following operations.

- 1 For each $m_c \notin \mathcal{M}$, compute $(\|\underline{w}\|^2)^{i,m_c}$ using recursion as in basic MFE; for each $m_c \notin \mathcal{M}$ and $\forall n$, compute g_n^{i,m_c} using recursion as in basic MFE.⁶ For each candidate feature (m_c) elimination, we consider any data point to be a *valid* candidate anchor $\underline{x}_{n_a}$ if it is not misclassified under that elimination (i.e., if $g_{n_a}^{i,m_c} \geq 0$). For all such *valid* ($\underline{x}_{n_a}, m_c$) pairs and all candidate features m_c , perform steps 2 and 3.
- 2 To ensure $\xi_{n_a}^{i,m_c,n_a} = 0$ (for anchor point $\underline{x}_{n_a}$) and to meet all the constraints in (4), we perform scaling as follows. Compute $\tilde{g}_n^{i,m_c,n_a} = g_n^{i,m_c} \rho^{i,m_c,n_a}$ for all nonanchors $\underline{x}_n$ ($n = n_a, n = 1, \dots, N$), where $\rho^{i,m_c,n_a} = 1/g_{n_a}^{i,m_c}$. Also, set $\tilde{g}_{n_a}^{i,m_c,n_a}$ to 1.
- 3 Compute $\xi_n^{i,m_c,n_a} = \max(0, 1 - \tilde{g}_n^{i,m_c,n_a})$, $n = 1, \dots, N$.
- 4.1 Perform the discrete double-minimization shown below in (14) to determine the feature m_i to be eliminated, and then eliminate it, i.e., $\mathcal{M} \rightarrow \mathcal{M} \cup \{m_i\}$.⁷

$$(m_i, n_i) = \arg \min_{m_c \in \mathcal{M}} \min_{n_a \in \{n | g_n^{i,m_c} > 0\}} \left(\frac{1}{2} (\|\underline{w}\|^2)^{i,m_c} (\rho^{i,m_c,n_a})^2 + C \sum_{n'=1}^N \xi_{n'}^{i,m_c,n_a} \right) \quad (14)$$

- 4.2 Since the scaling by ρ was only needed for the current elimination step, we do not carry the scaled quantities $\tilde{g}$ to the next elimination step ($i + 1$).
- 4.3 $i \rightarrow i + 1$ and go to step 1.

VI. Results

We performed experiments to compare our methods with RFE and NLPSVM [9].

A. Experimental Protocol for the Initial Classifier Training

We used the following common experimental protocol both for training of the *initial* classifier used by MFE and RFE and for training of the (*final*) NLPSVM classifier.

- Step 1)** The data set is randomly split 50%–50% into a non-heldout (training) set X and a heldout (test) set $\bar{X}$. Each such split defines one “trial.” Steps 2)–4) perform a bootstrap validation procedure to select classifier hyperparameters for each trial, from among a candidate set of hyperparameter values.
- Step 2)** Perform five 90%–10% random splits of X . For each such split, the large subset X_L will be used for training and the small subset X_S will be used for validation.

⁶As a reminder, in the nonlinear kernel case, the recursion is on the kernel computation.

⁷The simple default way to choose C in (14) is to set it to the C value used in (4) to train the initial (pre-elimination) classifier; however, a different C value may be chosen for (14), e.g., based on a cross-validation procedure applied after eliminating a batch of features.

- Step 3)** Perform the following for each candidate for the hyperparameter values: For each bootstrap split, use X_L to train a classifier and evaluate the performance on X_S . Average the validation error rate over the five bootstraps.
- Step 4)** Select the hyperparameter values, from among the set of candidates, that minimize the average validation error rate. Then, retrain the classifier for these hyperparameter values using all of X .

To achieve fair comparisons, our methods, RFE, and NLPSVM all shared precisely the same data—for every trial, the training set X , the test set $\bar{X}$, and the multiple X_L and X_S sets (for the trial's bootstrap splits) were the same for all methods. Furthermore, for every trial, our methods and RFE used the same initial classifier [defined by (4) and determined in Step 4)], obtained by training on all of X .

We emphasize that the above protocol is also used for NLPSVM training. Since for NLPSVM the learned classifier is independent of the initial chosen parameters, use of the above protocol for NLPSVM means that the NLPSVM hyperparameters were chosen to minimize the validation error of the (*final, trained*) NLPSVM classifier. By contrast, for MFE and RFE, the hyperparameters were only chosen to minimize validation error of the *initial* (pre-elimination) classifier (without accounting for subsequent feature eliminations). In this way, the protocol is somewhat favorably biased toward NLPSVM. While it is possible to modify our protocol to choose best hyperparameters consistent with subsequent feature eliminations for MFE and RFE, we have not done this. Despite this disadvantage, we will show that our basic MFE method typically achieved better or competitive generalization (lower *test set* error rate) compared with NLPSVM [9], in our extensive experiments.

Our set of candidate hyperparameter values for the linear kernel case (i.e., our candidate set for the C parameter) was $\{2^0, 2^1, \dots, 2^{10}\}$. We also used this set for NLPSVM's first parameter ν [9], a penalty parameter analogous to C in (4). For the second NLPSVM parameter δ [9], which is added to the Hessian before matrix inversion (as part of the Newton direction computation), we used the value set from [9]: $\{10^{-3}, 10^{-2}, \dots, 10^3\}$. In this paper, we compare with NLPSVM for the linear kernel case only. In the nonlinear kernel case, our set of candidate β parameter values was $\{2^{-30}, 2^{-29}, \dots, 2^{10}\}$ and our C set was the same as in the linear kernel case; thus our hyperparameter grid was of dimension 11×41 .

The SVM training (4) in all experiments was performed using the LIBSVM software [4]. For NLPSVM, we used the code provided in [9], and the following values used therein for additional NLPSVM parameters: $\epsilon = 4 \times 10^{-4}$, $\alpha = 10^3$, $tol = 10^{-3}$, $\text{imax} = 50$.

1) Data Preprocessing Prior to Initial Classifier Training—Categorical features Q whose values are elements of an unordered set of categories (e.g., $Q \in \mathcal{Q} = \{\text{orange, yellow, red}\}$) were mapped to $\text{card}(\mathcal{Q})$ unit vector features, each indicating one of the categories (e.g., $[0 \ 0 \ 1]^T$ for orange, $[0 \ 1 \ 0]^T$ for yellow, $[1 \ 0 \ 0]^T$ for red). As a second step, we normalized all feature values to the $[0, 1]$ range (separately for X and $\bar{X}$).

B. Experimental Protocol for Feature Elimination

Even for high-dimensional domains, we may eliminate all the way down to a few features, in which case at some point the data typically do become nonseparable. To continue the elimination process after loss of separability, we define the “hybrid” MFE/MFE-slack method, where, simply, MFE is used for the steps where the data are separable (including the initial step) and MFE-slack is used for the other steps.

In the sequel, we have broken up our experimental comparisons into three categories: 1) high-dimensional data (Sections VI-C and VI-F), 2) low-to-intermediate dimensioned domains for which the data *is* initially separable (Section VI-D), and 3) nonseparable domains (Section VI-E) which require immediate use of our MFE-slack method. For a given (data set, kernel) pairing [where the kernel is linear, polynomial (of degree 3), or Gaussian], if initial classifier separability was obtained in at least six out of the ten trials, we concluded that the pair most suitably fell into the separable category (i.e., Section VI-C or Section VI-D); otherwise the pair fell into the nonseparable category (i.e., Section VI-E).

1) Stopping Criteria—A criterion for stopping the feature elimination process should first have a classification error rate at the stopping point not much higher than (and, desirably, lower than) the error rate of the initial (pre-elimination) classifier. Second, in order to identify the important, class-defining feature “markers,” it is desirable for the percentage of features eliminated (at the stopping point) to be large.⁸ We defined and evaluated the following stopping criterion.

2) Acceleration Based (Accel)—To robustly detect the “knee” in the validation-set error rate curve for MFE-based methods as features are eliminated, we used the following algorithm to determine a stopping point.

1. On the error sequence $\underline{e} = \{e_i, i = 1, \dots, M\}$, where e_i is the validation-set error rate after eliminating i features, perform first-order autoregressive smoothing with parameter $1/2$.⁹
2. Compute associated velocity $\underline{v}$ and acceleration $\underline{a}$ sequences: $v_i = e_i - e_{i-1}$ and $a_i = v_i - v_{i-1}$, $i = 1, \dots, M$. Compute the running average sequence $\hat{\underline{a}}$ of $\underline{a}$ and the sequence $\underline{d}$ of percent increases in consecutive $\hat{a}_i$ values.
3. Set the percentage threshold t to 200%. Find the first point $\tilde{d}_i$ along the sequence $\underline{d}$ that exceeds t ; if such a point does not exist, lower t by 20% and try again. This high-to-low approach avoids stopping before the “knee” in the curve (to avoid false-positive knee detection). Then, to avoid over-shooting the knee, based on the elimination sequence, restore 0.01 M eliminated features.

For the NLPSVM method, we used the stopping criterion from [9]: stop when either of two conditions is satisfied: 1) $\|\underline{u}^i - \underline{u}^{i+1}\| \leq \text{tol}$, where $\underline{u}^i$ represents an (intermediate) solution for the LP problem at the i th iteration and tol is a user-selected tolerance value, specified in Section VI-A; 2) user-selected maximum number of iterations (imax , specified in Section VI-A) is reached.

C. Experiments on High-Dimensional Separable Data

We used three high-dimensional gene microarray data sets which we obtained from the LIBSVM website [4]: *Leukemia*, *Duke Breast Cancer*, *Colon Cancer*, respectively, with 38, 44, and 62 samples. For the case of the linear kernel, test set classification error rate curves for our methods and RFE are shown in Fig. 6. Also shown is training set margin for one of the data sets. All curves shown are averages over ten trials (and for each trial, the initial SVM was a training set separator). The vertical line indicates the average number of retained features at which separability was lost, over the ten trials, under our “basic MFE” method. Note that, consistent with theory [5], this is a small number, comparable to the number of

⁸If a criterion allows *no* features to remain at the stopping point, this is obviously not a good criterion.

⁹Since the error sequence $\underline{e}$ to be used here must not come from the heldout (test) set $\tilde{X}$, we obtain it from the nonheldout set X in a separate experiment accompanying the main experiment—after a training/validation (T/V) split (70%–30%) of X , an SVM is trained on set T using the SVM hyperparameters that were earlier selected via the protocol in Section VI-A for the main experiment, and the feature elimination order obtained subsequently from set T is applied to set V to obtain $\underline{e}$.

training points. Thus, for Fig. 6(a) and (c), on average MFE was able to eliminate from 7129 features down to approximately 100, with MFE-slack applied thereon. For each data set for the linear kernel case, we see that basic MFE achieved much larger training set margins (on average) than RFE and other MFE variants and this was accompanied by much better generalization performance (lower test set error rate curve).

On the 2000-feature *Colon Cancer* data set, we also eliminated features solely using MFE-slack. For all MFE-slack experiments herein, the MFE-slack C value used [in (14)] for each trial was chosen via the protocol given in Section VI-A for initial (pre-elimination) classifier design. For each trial, it turned out this C was large enough that there were no margin violations at any elimination steps; thus, the MFE/MFE-slack and MFE-slack elimination sequences were identical. However, the two elimination sequences are not always the same in practice when C is chosen using the Section VI-A protocol; we will demonstrate this shortly for other data sets.

Several comments should be made at this point. First, although in Section VI-D we showed that our novel basic MFE achieved both larger margins and better generalization than RFE for the low-dimensional separable case, we caution that the *theoretical* connection between margin and error rate pertains to an upper bound [14] and does not really tell us how our various methods (with different degrees of margin optimization) will relatively perform (i.e., more margin leading to better error rate may not be an accurate rule-of-thumb). Second, as discussed in Section IV-C, the greedy (stepwise) nature of feature elimination by our MFE methods and RFE (including variants performing periodic retraining) does not guarantee dominance of one method over another even with respect to margin; it is theoretically possible that basic MFE gives a solution sequence with a larger margin curve than either MFE-LO or MFE-Retrain, even though these latter methods adjust the weight vector to increase margin at each step.

Keeping these points in mind, in Fig. 6(d), MFE-LO (Full or Lite) did not achieve a higher training set margin curve than basic MFE, unlike our earlier (separable) illustrative example in Fig. 3 and unlike our upcoming (separable) example Fig. 9(d) for the *UCI mfeat* data set. These examples emphatically illustrate our second point in the preceding paragraph. Note further that RFE-Retrain achieves larger margin than MFE-Retrain. With respect to generalization performance, in Fig. 6, basic MFE and MFE-slack outperformed MFE-LO, as well as MFE-Retrain (the latter method did achieve a higher margin curve than basic MFE, as shown).

The inconsistency between the degree of margin optimization and test error generalization performance can perhaps be understood as a type of “overfitting.” This interpretation is also consistent with the good performance of MFE and MFE-slack; the MFE-slack performance will be demonstrated again in Section VI-D and extensively in Section VI-E. We note also that MFE and MFE-slack are the only two among our margin-based methods that, while eliminating features, preserve the initially designed SVM weights. Thus, in this sense, the basic MFE and MFE-slack methods do not “stray” as far from the original SVM solution, and this may have bearing on their performance. Fig. 6 also illustrates that MFE-Retrain outperformed RFE-Retrain in generalization in one of the three data sets (*Colon Cancer*), and these two methods were competitive for the other two data sets. Recall and notice again that both methods are outperformed by the “basic” MFE method that does not retrain during the elimination steps.

For the case of a polynomial kernel of degree 3, average test set error rate curves for our methods and RFE are shown in Fig. 7 for *Colon Cancer* (representative of high-dimensional data sets). Also shown is average training set margin. We see that basic MFE again achieved

much larger training set margins (on average) than RFE and this was accompanied by much better generalization performance (lower test set error rate curve). In this case, MFE/MFE-slack and MFE-slack produced different elimination sequences; this is hardly discernible in Fig. 7, since the generalization performance of the two methods differed only slightly. We also note that basic MFE/MFE-slack and MFE-slack achieved the best generalization performance and were significantly better than the other MFE variants. Comparing Figs. 7(b) and 6(b), we see that better classification results were obtained for the linear kernel than the polynomial kernel.

1) Comparison With NLPSVM—Since basic MFE had the best generalization performance for each data set above, we next compared this method to NLPSVM for the linear kernel case. For each trial t , let $M_{t,NLPSVM}$ denote the number of features selected by NLPSVM at its stopping point, and let $E_{t,NLPSVM}$ be the associated test set error rate. For the same trial, let $M_{t,MFE}$ be the number of features selected by MFE/MFE-slack at its stopping point using our *accel* criterion and let $E_{t,MFE}$ be the associated test set error rate. Let M_{NLPSVM} be the average number of features selected by NLPSVM across all trials and let E_{NLPSVM} be the associated average test set error rate, with M_{MFE} and E_{MFE} the corresponding quantities for MFE/MFE-slack. In Fig. 6, to compare the generalization performance of MFE/MFE-slack and NLPSVM, we plotted (M_{MFE}, E_{MFE}) as a circle and (M_{NLPSVM}, E_{NLPSVM}) as a diamond. It is important to note that generally $M_{t,MFE}$ will vary across trials and thus the point (M_{MFE}, E_{MFE}) , again depicted as a circle, will not necessarily lie on the shown average MFE/MFE-slack curve. Observing these points, notice that the MFE/MFE-slack test set error rate is much lower than the NLPSVM test set error rate for each of the three data sets, albeit achieved with a larger set of selected features than for NLPSVM based on its stopping criterion.

In Fig. 8, we illustrate the excursions that the NLPSVM method takes in practice during its iterations (each computing a different decision boundary) towards a final decision boundary. For a typical trial, the dotted path connects (number of features, test set error rate) points for consecutive NLPSVM iterations from its initial iteration (which the “start” arrow points to) to its final iteration (marked by “finish”).¹⁰ As demonstrated by the zig-zag movements on the dotted path, the number of features is not monotonic with NLPSVM iterations and different iterations may revisit the same number of features. As demonstrated jointly by the left-right shifts (in number of features) and large up-down shifts (in test set error rate), a wide range of test set error rates is achieved by NLPSVM as it performs consecutive iterations. On the other hand, the MFE/MFE-slack test error rate curve trend is flat or decreasing with increasing number of features [an illustrative MFE single trial is shown in Fig. 8(b)]. Thus, even though our heuristic stopping criterion is suboptimal,¹¹ if we *restore* eliminated features starting from the MFE/MFE-slack stopping point we are likely to either improve the error rate or leave it unchanged. As indicated by Fig. 8, this statement is not true for the NLPSVM method. If the number of NLPSVM features is increased, there is no strong likelihood that the error rate will be better, and going back several iterations from the NLPSVM stopping point gives no guarantee of the same features being retained or of comparable error rate performance. That is, for the NLPSVM method, there is much less predictability of generalization performance than for MFE, in the vicinity of the method’s stopping point.

¹⁰An iteration’s test set error rate is associated with its (iteration-specific) feature subset. These features correspond to the coordinates not zeroed-out by NLPSVM in the iteration’s weight vector $\underline{w}$.

¹¹In Fig. 6, for the *Duke Breast Cancer* data set, for example, note that it may be better to stop basic MFE at approximately 350 features.

D. Experiments on Low-Dimensional Separable Data

For low-dimensional separable data, average test set error rate curves for three (data set, kernel) pairings are shown in Fig. 9. Also shown are average training set margin curves for one of the data sets. As previewed in Section VI-C, the curves in Fig. 9 show that our MFE/MFE-slack method achieved better generalization performance than RFE, and also performed best among all of our methods. These results, like those in Section VI-E, are representative of a more extensive study we have conducted involving more data sets and more (data set, kernel) pairings.

We show the MFE-Retrain and RFE-Retrain generalization performances for the linear kernel case for two data sets in Fig. 9(a) and (b), where we again find that MFE-Retrain is either outperforming or performing competitively with RFE-Retrain. Again, both methods are outperformed by “basic” MFE and MFE-slack (which do not retrain during the elimination steps). As was the case for high-dimensional data sets in Section VI-C, MFE/MFE-slack and MFE-slack produced the same elimination sequence for one data set [*hepatitis* in Fig. 9(b)], whereas for other data sets [*dermatology* in Fig. 9(a) and *mfeat* in Fig. 9(c)] the two methods produced different sequences and their generalization performances differed very slightly. In Fig. 9(a) and (b) (i.e., the linear kernel case), observing the MFE/MFE-slack and NLPSVM points (circle and diamond, respectively), the NLPSVM test set error rate is slightly higher than the MFE/MFE-slack test set error rate for these two data sets, and this is achieved with a slightly smaller number of selected features than MFE/MFE-slack.

E. Experiments on Low-Dimensional Nonseparable Data

For low-dimensional nonseparable data, average test set error rate curves are shown in Fig. 10 for six (data set, kernel) pairings. Excluding Fig. 10(d), the curves in Fig. 10 (for five data sets) are averages over ten, ten, six, six, nine trials, respectively, which were the nonseparable trials among the ten trials generated. For Fig. 10(d), although the curves are averages over nine *separable* trials among the ten trials generated, notice from the location of the vertical line that while this *UCI car* data set was initially separable, it immediately lost separability when feature elimination commenced and thus we concluded it is more suitable for discussion in this section than Section VI-D. The curves in Fig. 10 show that MFE-slack consistently achieved better generalization performance than RFE and RFE-abs.

Experiments (not shown) found this same result for additional (UCI data set, kernel) pairings. We also note that for *flag* and *ionosphere* in Fig. 10 [and e.g., Fig. 6(a)], for which we applied RFE retraining, this retraining did improve performance over RFE without retraining. In Fig. 10(a) and (b) (i.e., the linear kernel case), the NLPSVM test set error rate (diamond) is slightly lower than the MFE/MFE-slack test set error rate (circle) albeit achieved with a slightly larger set of selected features.

F. High-Dimensional Feature Space Application: Brain Images

We processed 47 T_1 -weighted 3-D magnetic resonance imaging (MRI) images [12 Alzheimer’s disease (AD), 35 Control]. After segmentation and registration, a Gray Matter tissue density image called “RAVENS” was generated for each by using the HAMMER software [23] and then smoothed by a 5-mm Gaussian filter. After identically cropping all (3-D) RAVENS images to remove nonbrain background, we considered each voxel as a feature and performed SVM classification and feature elimination experiments on a slice-by-slice basis, as follows. First, we split the 47 samples into a training set ($\mathcal{X}$, with eight AD and 25 Control samples) and a test set ($\mathcal{X}$, with four AD and ten Control samples). For *each* of the (approximately) 150 2-D (151×186) slices comprising a 3-D image, we trained a linear SVM, which perfectly classified all training samples as AD or Control. Next, the RFE and basic MFE methods each eliminated one voxel at a step, generating each method’s

ordered set of discriminating voxels for that slice. Training set margins and test set classification error rates were averaged over 55 (out of the 150) of these slice-specific classifiers. Fig. 11 shows that MFE achieved larger margins and generalized better than RFE. In Fig. 12, we show 12 slices that contain the hippocampus,¹² with a foreground of overlaid colors. The background image commonly used for such an overlay is the average over the (spatially registered) brain image population used in the experiment; thus, in this case, it is the average of the RAVENS Gray Matter images. In Fig. 12(a), the orange regions are those determined by MFE to be the most discriminating regions for AD, i.e., these contain the retained voxels for MFE up until the point where the data became nonseparable. Likewise, red indicates the regions found by RFE (with the same number of retained voxels as MFE), and white indicates the regions found by both MFE and RFE. Thus, Fig. 12(a) indicates that MFE is detecting the hippocampus, whereas RFE is not. Fig. 12(b) shows a confidence ranking among the MFE regions that were displayed (in orange and white) in Fig. 12(a). Higher rank for a voxel means it is more discriminating for AD. We indicate the ranking using the following colormap: red = high, yellow = higher, white = highest. The ranking was based simply on the feature elimination order. Notice almost all of the smallest clusters (among the retained voxels which are shown in color) have received the lowest ranking; it is significant that MFE's most discriminating voxels are found in sizeable, spatially compact voxel clusters. It is also significant that, among the retained voxels, those in the hippocampus have almost uniformly received the "higher"-to-"highest" ranking.

VII. Conclusion

In this paper, we presented MFE and several extensions, applicable to SVMs and other linear and nonlinear, kernel-based discriminant functions. In the nonlinear kernel case, we identified shortcomings of RFE and demonstrated improved margin and accuracy achieved by kernel-based MFE. A second extension performs a lightweight SVM training that adjusts the current solution in the reduced feature space to improve margin. In a third approach, MFE-slack, we formulated a simple extension of MFE to incorporate margin slackness. We evaluated on UCI data sets and gene microarray data sets, toward a comprehensive evaluation of the generalization performances of MFE, RFE, and NLPSVM. We demonstrated that MFE and MFE-slack provide better generalization than RFE, and better or competitive generalization compared with NLPSVM. We found that basic MFE, which requires separable data, was always suitable for high-dimensional data but was also applicable for several low-dimensional UCI data sets. We observed that methods which provide further increases in margin beyond that of basic MFE did not necessarily lead to improved generalization. This was attributed to overfitting and is consistent with the good performance achieved by MFE-slack, which introduces margin slackness. Finally, we also gave illustrative results on brain image data.

Acknowledgments

The software for carrying out pipeline processing of the 3-D MRIs in Section VI-F was developed through a collaboration between D. Bigler and Y. Aksu at the Center for NMR Research, Pennsylvania State College of Medicine, Hershey, PA.

This work was supported in part by National Institute of Health (NIH) under Grant R01 AG027771.

References

1. Aksu, Y.; Kesidis, G.; Miller, DJ. Scalable, efficient, stepwise-optimal feature elimination in support vector machines. *Proc. IEEE Workshop Mach. Learn. Signal Process.*; 2007. p. 75-80.

¹² A well-published "marker" brain structure for AD.

2. Aksu, Y.; Miller, DJ.; Kesidis, G. Margin-based feature selection techniques for support vector machine classification. *Proc. IAPR Workshop Cogn. Inf. Process*; 2008. p. 176-181.
3. Chan, AB.; Vasconcelos, N.; Lanckriet, GR. Direct convex relaxations of sparse SVM. *Proc. Int. Conf. Mach. Learn*; 2007. p. 145-153.
4. Chang, C.; Lin, C. LIBSVM: A library for support vector machines, 2001 [Online]. Available: <http://www.csie.ntu.edu.tw/~cjlin/libsvm>
5. Cover TM. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE Trans Electron Comput.* Jun; 1965 EC-14(3):326-334.
6. Cristianini, N.; Shawe-Taylor, J. *An Introduction to Support Vector Machines*. Cambridge, U.K: Cambridge Univ. Press; 2000.
7. Duda, R.; Hart, P.; Stork, G. *Pattern Classification*. 2. New York: Wiley; 2001.
8. Dumais, S.; Platt, J.; Heckerman, D.; Sahami, M. Inductive learning algorithms and representations for text categorization. *Proc. Conf. Inf. Knowl. Manage*; 1998. p. 148-155.
9. Fung G, Mangasarian OL. A feature selection Newton method for support vector machine classification. *Comput Optim Appl.* Jul; 2004 28(2):185-202.
10. Fung G, Mangasarian OL. Multicategory proximal support vector machine classifiers. *Mach Learn.* 2005; 59:77-97.
11. Furey T, Cristianini N, Duffy N, Bednarski DW, Schummer M, Haussler D. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics.* 2000; 16(10):906-914. [PubMed: 11120680]
12. Guyon I, Weston J, Barnhill S, Vapnik V. Gene selection for cancer classification using support vector machines. *Mach Learn.* 2002; 46(1):389-422.
13. Guyon I, Elisseeff A. An introduction to variable and feature selection. *J Mach Learn Res.* 2003; 3:1157-1182.
14. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning*. New York: Springer-Verlag; 2001.
15. Joachims, T. Training linear SVMs in linear time. *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Disc. Data Mining*; 2006. p. 217-226.
16. Keerthi SS, DeCoste D. A modified finite Newton method for fast solution of large scale linear SVMs. *J Mach Learn Res.* 2005; 6:341-361.
17. Kugler, M.; Aoki, K.; Kuroyanagi, S.; Iwata, A.; Nugroho, AS. Feature subset selection for support vector machines using confident margin. *Proc. Int. Joint. Conf. Neural Netw*; 2005. p. 907-912.
18. Mangasarian OL. Exact 1-norm support vector machines via unconstrained convex differentiable minimization. *J Mach Learn Res.* 2006; 7:1517-1530.
19. Mladenic, D.; Brank, J.; Grobelnik, M.; Milic-Frayling, N. Feature selection using linear classifier weights: Interaction with classification models. *Proc. ACM SIGIR Conf. R&D Inf. Retrieval*; 2004. p. 234-241.
20. Peng H, Long F, Ding C. Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans Pattern Anal Mach Intell.* Aug; 2005 27(8):1226-1238. [PubMed: 16119262]
21. Ramaswamy S, Tamayo P, Rifkin R, Mukherjee S, Yeang CH, Angelo M, Ladd C, Reich M, Latulippe E, Mesirov JP, Poggio T, Gerald W, Loda M, Lander ES, Golub TR. Multiclass cancer diagnosis using tumor gene expression signatures. *Proc Nat Acad Sci USA.* Dec; 2001 98(26): 15149-15154. [PubMed: 11742071]
22. Shawe-Taylor, J.; Cristianini, N. *Kernel Methods for Pattern Analysis*. Cambridge, U.K: Cambridge Univ. Press; 2004.
23. Shen D, Davatzikos C. HAMMER: Hierarchical attribute matching mechanism for elastic registration. *IEEE Trans Med Imag.* Nov; 2002 21(11):1421-1439.
24. Tibshirani R. Regression shrinkage and selection via the lasso. *J R Stat Soc B (Methodological).* 1996; 58(1):267-288.
25. Trunk GV. A problem of dimensionality: A simple example. *IEEE Trans Pattern Anal Mach Intell.* Jul; 1979 1(PAMI-3):306-307. [PubMed: 21868861]

26. Weston, J.; Mukherjee, S.; Chapelle, O.; Pontil, M.; Poggio, T.; Vapnik, V. Neural Information Processing Systems. Vol. 13. Cambridge, MA: MIT Press; 2001. Feature selection for SVMs.
27. Weston J, Elisseeff A, Schölkopf B, Tipping M. Use of the zero norm with linear models and kernel methods. J Mach Learn Res. Mar.2003 3:1439–1461.
28. Zhu, J.; Rosset, S.; Hastie, T.; Tibshirani, R. Neural Information Processing Systems. Cambridge, MA: MIT Press; 2003. 1-norm support vector machines.

Biographies

Yaman Aksu received the B.A. degree in physics from Franklin and Marshall College, Lancaster, PA, in 1992, the B.S.E. degree in electrical engineering from Washington University, St. Louis, MO, in 1993, and the M.S.E. degree in electrical/computer engineering (ECE) from the Johns Hopkins University, Baltimore, MD, in 1996. Currently, he is working towards the Ph.D. degree at the Pennsylvania State University, University Park.

From May 1996 to July 2004, he was with Sensor Systems, Inc., Sterling, VA, and its subsidiary Medical Numerics, working on biomedical image processing/analysis systems/software focused on neuroimaging with MRI, such as MEDx, BrainWave (on General Electric Medical Systems MRI scanners), and Digital Image Archive System (DIAS, National Institutes of Health, Bethesda, MD). His research interests include biomedical image analysis, machine learning, applied math, signal processing, and bioinformatics.

David J. Miller received the B.S.E. degree from Princeton University, Princeton, NJ, in 1987, the M.S.E. degree from the University of Pennsylvania, Philadelphia, in 1990, and the Ph.D. degree from the University of California, Santa Barbara, in 1995, all in electrical engineering.

From January 1988 to January 1990, he was with General Atronics Corp., Wyndmoor, PA. From September 1995 to July 2001, he was an Assistant Professor of Electrical Engineering at The Pennsylvania State University, University Park. He is now tenured Full Professor of Electrical Engineering at The Pennsylvania State University. His research interests include machine learning, source coding and coding over noisy channels, image coding and processing, and bioinformatics.

Dr. Miller received the National Science Foundation CAREER Award in 1996. He was General Chair for the 2001 IEEE Workshop on Neural Networks for Signal Processing. He was an Associate Editor for the IEEE Transactions on Signal Processing from 2004 to 2007. He was also Chair of the Machine Learning for Signal Processing Technical Committee, within the IEEE Signal Processing Society, from 2007 to 2009.

George Kesidis (S'90–M'92–SM'03) received the Ph.D. degree in electrical engineering and computer science from the University of California, Berkeley, in 1992.

He was a Professor at the Electrical and Computer Engineering Department, University of Waterloo, Waterloo, ON, Canada, from 1992 to 2000. Since 2000, he has taught at both the Computer Science and Engineering and Electrical Engineering Departments of the Pennsylvania State University. His research experience includes several areas of computer/communication networking and machine learning.

Dr. Kesidis served as the TPC Co-Chair of IEEE INFOCOM 2007 and is now serving on the editorial boards of the *ACM Transactions on Modeling and Computer Simulation (TOMACS)* and the IEEE Communications Surveys and Tutorials.

Qing X. Yang received the Ph.D. degree in physics from the Georgia Institute of Technology, Atlanta, in 1991.

Currently, he is a Professor at the Departments of Radiology and Neurosurgery, Pennsylvania State College of Medicine, Hershey. His research interests include rapid MRI data acquisition method development, high field MRI physics, and clinical applications of quantitative MRI and fMRI and their image analysis in humans and animal models.

Fig. 1.

When an SVM is trained on the three points in (a) ($\underline{x}_1 = (3, 4)$ in class 1, $\underline{x}_2 = (-7, -1)$ and $\underline{x}_3 = (-3, -4)$ in class 2), the decision boundary is $(\underline{w}, b) = ((0.12, 0.16), 0)$ (line shown through origin with slope $-3/4$), with a margin of 5 (to all three points)—neither a vertical separator line through the origin nor a horizontal separator line through 1.5 is the boundary since these achieve only a margin of 3 and 2.5, respectively. Since $w_1 = 0.12 < 0.16 = w_2$, RFE will eliminate the first feature and a threshold of the second feature at zero will become the boundary, resulting in a margin of 1 (distance from $x_{2,2} = -1$ to zero). If instead the second feature is eliminated, a (larger) margin of 3 (distance from $x_{1,1} = 3$ to the boundary at zero) would be obtained. This proves that eliminating features according to RFE is not in general equivalent to eliminating to preserve maximum margin. The latter choice is made by the MFE method proposed in this paper; (b) and (c) also respectively indicate for the two elimination options that, if a new SVM training were performed in the resulting 1-D space, then the SVM trained after the MFE feature elimination step, rather than RFE, would again have the larger margin ($3 > 2.5$).

Fig. 2.

Results for the Gaussian kernel on the *UCI arrhythmia* data set with 279 features. (a) $\|w\|^2$; (b) margin; (c) test error rate.

Fig. 3.

(a) Training set margin, test set error rate, for the *Leukemia* data set, with 7129 features. (b) Training set margin, test set error rate, for the *UCI hepatitis* data set, with 19 features.

Fig. 4. Illustration of C^+ , one of two cones used in the solution of the “little optimization (LO)” problem.

Fig. 5.

In (a), consider the line through the origin with slope $-3/4$ as a decision boundary obtained by training an SVM on a sample set that includes the four shown samples: $\underline{x}_1$ and $\underline{x}_2$ in class 1, $\underline{x}_3$ and $\underline{x}_4$ in class 2. $\underline{x}_1$ and $\underline{x}_3$ are located at a (violated) margin distance of 5 (shown by dashed lines), with $\underline{x}_2$ the margin violator. For the case where m_c is the second feature, the new boundary is the origin and the candidate anchor $\underline{x}_{n_a}$ in (e)–(h), marked by a circle, is $\underline{x}_1$, $\underline{x}_2$, $\underline{x}_3$, and $\underline{x}_4$, respectively. The pseudocode in Section V-A will describe how MFE-slack performs scaling by ρ on all samples to ensure $\xi_{n_a} = 0$ (for anchor $\underline{x}_{n_a}$). In (e)–(h), after scaling, $0 < \xi_n < 1$ for samples marked by a square (i.e., samples closer to the boundary than $\underline{x}_{n_a}$), there are no misclassified samples (i.e., no samples with $\xi_n > 1$). The case where m_c is the first feature is illustrated in (b)–(d). Here, again after scaling, misclassified samples (with $\xi_n > 1$) are marked by a triangle. Notice there is no diagram with $\underline{x}_4$ as candidate anchor because we only consider as anchors the points that are correctly classified—under elimination of feature 1, $\underline{x}_4$ becomes misclassified.

Fig. 6.

Average linear SVM test set error rate curves for three gene microarray data sets (Section VI-C). We truncated the curves from the right to zoom in on the most useful detail in the concluding segment of the feature elimination process—we state the actual “starting number of features” on the x -axis of the plot. In the initial segment (not shown), the curves stay mostly flat, with little or no change in relative performance of the methods. The vertical line indicates where the basic MFE method lost separability (on average). Average training set margin is shown for *Leukemia* in (d). (a) *Leukemia* data set with 7129 features; linear kernel. (b) *Colon Cancer* data set with 2000 features; linear kernel. (c) *Duke Breast Cancer* data set with 7129 features; linear kernel. (d) *Leukemia* data set with 7129 features; linear kernel; margin.

Fig. 7. Average training set margin and average test set error rate curves on the *Colon Cancer* gene microarray data set with 2000 features (Section VI-C) for the polynomial kernel. (a) Average training set margin. (b) Average test set error rate.

Fig. 8. Illustrative single trials for the 2000-feature *Colon Cancer* data set. (a) NLPSVM. (b) MFE/MFE-slack.

Fig. 9.

Average test set classification error rate, (a)–(c), for three separable low-dimensional UCI data sets (Section VI-D). We truncated the *dermatology* graph from the right to zoom in on the most useful detail (see truncation description in Fig. 6). The vertical line indicates the average number of retained features at which separability was lost. Average training set margin is shown for one of the data sets in (d). Initial number of features is stated on the x -axis. (a) *UCI Dermatology* data set with 130 features; linear kernel. (b) *UCI Hepatitis* data set with 19 features; linear kernel. (c) *UCI mfeat* data set with 216 features; polynomial kernel. (d) *UCI mfeat* data set 216 features; polynomial kernel; margin.

Fig. 10.

Average test set classification error rate on six low-dimensional UCI data sets (Section VI-E). Initial number of features is stated on the x -axis. (a) *UCI Flag* data set with 66 features; linear kernel. (b) *UCI Ionosphere* data set with 34 features; linear kernel. (c) *UCI Arrhythmia* data set with 279 features; polynomial kernel. (d) *UCI Car* data set with 21 features; polynomial kernel. (e) *UCI Cylinder* data set with 78 features; Gaussian kernel. (f) *UCI Wisconsin-Diagnosis* data set with 29 features; polynomial kernel.

Fig. 11. Feature elimination results for high-dimensional brain image data, for the linear SVM case. The initial number of features is 28 086.

Fig. 12. Retained voxels (i.e., discriminating regions) for Alzheimer's disease in color (Section VI-F).