MARGIN-BASED FEATURE SELECTION TECHNIQUES FOR SUPPORT VECTOR MACHINE CLASSIFICATION

Yaman Aksu, David J. Miller, George Kesidis

Department of Electrical Engineering
The Pennsylvania State University
University Park, PA 16802
e-mail:yamanaksu@yahoo.com, djmiller@engr.psu.edu, kesidis@engr.psu.edu

ABSTRACT

Feature selection for classification working in high-dimensional feature spaces can improve generalization accuracy, reduce classifier complexity, and is also useful for identifying the important feature “markers”, *e.g.*, biomarkers in a bioinformatics or biomedical context. For support vector machine (SVM) classification, a widely used feature selection technique is *recursive feature elimination* (RFE). In recent work, we demonstrated that the RFE objective is not generally consistent with the margin maximization objective that is central to the SVM learning approach. We thus proposed *explicit* margin-based feature elimination (MFE) for SVMs and demonstrated both improved margin and improved generalization accuracy, compared with RFE for the case of linear SVMs. In this paper, after reviewing MFE, we first introduce an extension which achieves further gains in margin at small computational cost. This extension solves the SVM optimization problem to maximize the classifier's margin at each feature elimination step, albeit in a lightweight fashion by optimizing only *two* degrees of freedom – the weight vector's slope and intercept. We next consider the case of a nonlinear kernel. We show that RFE defined for the nonlinear kernel case assumes that the weight vector length is strictly decreasing as features are eliminated. We demonstrate experimentally that this assumption is not in general valid for the Gaussian kernel and that, consequently, RFE may give poor results in this case. An extension of MFE for the nonlinear kernel case gives both better margin and generalization accuracy. This approach may help nonlinear kernel SVMs to avoid overfitting and, thus, to achieve better results than linear SVMs in some high-dimensional domains where use of nonlinear kernels has not to date been found very favorable.

Keywords: *support vector machines, classifier margin, recursive feature elimination, Gaussian kernel*

1. INTRODUCTION

1.1. Feature Selection in Classification

In high-dimensional classification domains such as text categorization, image and image sequence classification, and classification in genomics and proteomics, one often encounters problems where there are very few labeled training samples, or at any rate very few samples *relative to* the very high-dimensionality of the feature measurements for each exemplar/sample. In bioinformatics in particular, with training databases derived *e.g.* from clinical trials, there may be at most several hundred (patient) samples, each represented by *e.g.* tens of thousands of DNA microarray features. In all of the abovementioned domains as well as many others, there are several compelling reasons for reducing feature dimensionality. First, many of the features may have at best weak discrimination power. In [10], a type of “curse of dimensionality” (COD) was demonstrated, wherein, in the small sample case, the parameter estimation error for the parameters that model the (many) features with modest discrimination power (those with high variance and/or small distance between class means) defeats the generalization accuracy benefit derived from using these features for classification – *i.e* for fixed sample size, the generalization accuracy may *degrade* as the feature dimensionality is increased beyond a certain point. This phenomenon is related to the bias-variance dilemma in statistics [8] which essentially suggests that, for best generalization, model complexity should be matched to the available training data resources. Even in domains where generalization accuracy tends to monotonically improve with increasing feature dimensionality, complexity of the classification operation (both computation and memory storage required for decisionmaking) 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 set of “marker” features that have unusual correlation with the class variable. In document classification, these markers may represent class(topic)-specific key words/terms. In image-based classification, these features may identify the best descriptors for representing images/objects of a certain type. In bioinformatics, gene “biomarkers” may shed light on the underlying disease mechanism and/or help to identify targets for drug therapy. Decisionmaking based on a small set of features is also highly interpretable, which is important *e.g.* in explaining credit card approval decisions.

There are several approaches for avoiding model overfitting/COD. One is to fit the original high-dimensional data (with M features), by using simple models, *e.g.* naive Bayes models [4] or those that share parameters in modeling distributions for multiple features. Another approach is to limit the amount of model training, *e.g.* through use of regularization costs or early stopping [4]. Support vector machines (SVMs) attempt to avoid overfitting by finding a discriminant function that maximizes the margin[1]. In SVMs, the

[1] The minimum distance of any sample point to the decision boundary.

number of free model parameters is upper-bounded by the number of training samples (a subset of which are support vectors with nonzero Lagrange multiplier parameters), rather than controlled by the feature dimensionality. However, whether using linear or nonlinear kernels, SVMs are *not* immune to the curse of dimensionality [8]. Thus, *feature selection*, wherein only a small subset of the original features are retained, is often an essential step for achieving good, *generalizeable* classification accuracy, for SVMs, as well as other classifiers.

Unfortunately, there are $2^M - 1$ possible feature subsets, with exhaustive search practically prohibited even for modest M, let alone M on the order of thousands. Practical feature selection techniques are thus heuristic; there are a variety of such methods, exercising a large range of tradeoffs between accuracy and complexity [7]. Front-end methods apply knowledge of the class labels to evaluate discrimination power of individual genes or small gene groups prior to classifier training. Wrapper-based approaches combine feature selection and classifier training, with the classifier learning algorithm repeatedly applied for different feature subsets and with the best subset chosen based on a specified criterion such as training set error rate. These methods improve predictive power by capturing joint feature effects. Wrapper algorithms entail higher computation than front-end methods because they embed classifier training within the feature search. There is greedy forward selection, with "informative" features added starting from a null set, backward search, which starts from the full space and then eliminates features, bidirectional searches, and more complex algorithms such as simulated annealing [7]. Backward search starts by assessing joint predictive power of all features. In principle, one would like to retrain the classifier in conjunction with each backward elimination step that removes a single feature (optimizing the classifier for the new feature space). However, considering large M, this requires either M classifier retrainings (if retraining is done *after* a feature elimination step) or $\frac{M(M-1)}{2}$ (if retraining is done *before* feature elimination, *i.e.*, after *trial- elimination* of each remaining feature, at every feature elimination step). For SVM- based classifier training, considered in the sequel, *neither* of these is computationally feasible because even a single run of SVM training takes significant computation for large M. Thus, for large M, retraining can only be done periodically, after a "batch" of features has been removed.

1.2. Feature selection for SVMs

SVMs have become nearly a standard classification technique in many domains. There are a number of reasons. First, the SVM objective, maximizing the classifier's margin, has a strong theoretical basis tied to the achievement of good generalization accuracy [3]. Second, there is a unique, globally optimal solution to the SVM training problem. Third, there are improvements in representation power achieved through the use of nonlinear kernels, which map to a high or even infinite-dimensional feature space and, via the "kernel trick", do so *without* requiring a large increase in the complexity of decisionmaking and classifier training. Fourth, SVMs have achieved good results on a variety of domains, including *e.g.* document classification and bioinformatics applications. Finally, precisely because SVMs are so widely used, any improvements to the existing methodology are likely to have a large impact and to be widely disseminated.

A number of feature selection methods have been investigated for SVMs, including some customized ones. Front-end filtering has been applied in numerous prior works. In [11], wrapper complexity was reduced by replacing the SVM training objective with an upper bound that is less complex to optimize. Other approaches are not wrapper-based, but modify the SVM objective to encourage sparse solutions, *e.g.,* [12]. A widely used method is *recursive feature elimination* (RFE) [6], 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 very large M^2. The authors in [6] essentially argue that the RFE objective for linear SVMs is consistent with the SVM objective of margin maximization. They note that the SVM primal optimization problem poses minimization of the square of the weight vector length subject to (margin-related) constraints involving each training point. Eliminating the feature with smallest weight magnitude has the least effect on the weight vector length and, thus, [6] argues, on the SVM solution. In recent work [1], we have shown experimentally, however, that RFE is not in close agreement with margin maximization. The reason is that RFE ignores the margin constraints in the SVM problem, focusing solely on minimally reducing the weight vector length. In this work, we first review our recently proposed *margin-based feature elimination* (MFE) method [1], which explicitly performs margin-optimal backward feature elimination for SVMs. We then introduce an extension which 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 choosing only *two* degrees of freedom – the weight vector's slope and intercept. We then consider the case of a nonlinear kernel. We show that RFE defined for the nonlinear kernel case [6] assumes that the weight vector length 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. An extension of MFE for the nonlinear kernel case gives both better margins and generalization accuracies.

2. MARGIN-BASED FEATURE ELIMINATION IN SVMS

2.1. Brief review of linear SVMs

Consider a labeled training set $\{\underline{x}_n, y_n\}$ for $1 \leq n \leq N$ where N is the number of samples, $y_n \in \{\pm 1\}$ is the class label, and $\underline{x}_n \in \mathbf{R}^M$ is the nth data sample. Assuming the data set is linearly separable (*highly* likely in the small sample, very high-dimensional feature space case), a linear SVM will find a separating hyperplane

$$f(\underline{x}) \quad \equiv \quad \underline{w}^{\mathrm{T}}\underline{x} + b = \sum_{m=1}^{M} w_m x_m + b,$$

$\underline{w} \in \mathbf{R}^M$, $b \in \mathbf{R}$. We denote the perpendicular from $\underline{x}_n$ to the hyperplane by $\underline{p}_n$ and note the standard results $\underline{p}_n \equiv \frac{f(\underline{x}_n)}{||\underline{w}||^2}\underline{w}$

[2] However, as discussed earlier, if classifier retraining is performed for each eliminated feature or, worse, for each candidate feature elimination, computational complexity may be practically infeasible for large M.

(where $||\cdot||$ is the Euclidean norm) and $\frac{y_n f(\underline{x}_n)}{||\underline{w}||}$ for the signed distance from $\underline{x}_n$ to the hyperplane. Denoting $g(\underline{x}_n) \equiv y_n f(\underline{x}_n)$, the hyperplane is a *separating* one if it satisfies $g(\underline{x}_n) > 0$ for all $\underline{x}_n$. The margin of the separating hyperplane f is thus defined as $\gamma \equiv \frac{\min_n g(\underline{x}_n)}{||\underline{w}||}$. We will denote $\mathcal{N} \equiv \min_n g(\underline{x}_n)$, $\mathcal{D} \equiv ||\underline{w}||$, $L \equiv \mathcal{D}^2$ and write $\gamma \equiv \frac{\mathcal{N}}{\mathcal{D}}$. The basic SVM training problem, finding the hyperplane yielding largest margin, is:

$\min_{\underline{w},b} \frac{1}{2}||\underline{w}||^2$
s.t. $y_n(\underline{w}^T\underline{x}_n) + b) \geq 1, n = 1, \ldots, N$

2.2. Limitations of RFE: linear case

Under RFE [6], the index m^* of the first feature to be eliminated is arg $\min_{m \in \{1,2,\ldots,M\}} |w_m|$, and, more generally, at step k, the minimization is performed over the $M-k$ remaining features. While [6] 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 the equation 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:

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

In [2], we prove via a simple 2-dimensional counterexample that eliminating features according to RFE is in fact not equivalent to eliminating according to margin. In general, direct margin maximization leads to significant gains in margin and may also lead to improved generalization accuracy over RFE, as demonstrated experimentally both in [1] and in the sequel.

2.3. Direct margin-based approach

Since maximizing margin is the (theoretically motivated) goal of SVM training [3], eliminating features such that the margin is left as large as possible should be "stepwise superior" to RFE. Surprisingly, while there are some related approaches [9][3], we had not seen direct, margin-based feature elimination previously proposed. In recent work [1], we developed just such a technique, along with its efficient implementation. We note that at each elimination step i, it may appear that the hypothetical margins $\gamma^{(i),m}$ (after eliminating feature m at step i) need to be computed under candidate elimination of *every* remaining feature m and then the maximum over m needs to be found. When M is very large, this will require some computation (albeit not *so* much greater than the (very lightweight) RFE computation). Regardless, we developed a Margin-optimal Feature Elimination (MFE) method that, without sacrificing optimality, only requires explicit margin evaluations for a *subset* of the candidate features and which achieves further computational efficiency via a recursive implementation.

2.3.1. MFE algorithm pseudocode for SVMs: linear case

Below, we will use the following notation: $q^{(i),m} \equiv$ quantity q at feature elimination step i upon elimination of feature m.

0. 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 (parameterized by $\underline{w}, b$), with weight norm-squared $L^{(-1),0} \equiv ||\underline{w}||^2$.[4] For each feature m, compute the following quantities as a preprocessing step:

$$\delta_n^m = y_n x_{n,m} w_m, \forall n \quad \text{and} \quad \delta^{*^m} \equiv \max_n \delta_n^m$$

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 and n. The δ^{*^m} are initially sorted in increasing order to facilitate determination of "candidate feature sets" for elimination. *This sorting step is executed only once.*

Finally, recalling $\mathcal{N} \equiv \min_n g(\underline{x}_n)$, set $i \leftarrow 0$ and compute the following quantities:

$$g_n^{(-1),0} = y_n b + \sum_{m=1}^{M} \delta_n^m \quad \text{and} \quad \mathcal{N}^{(-1),0} = \min_n g_n^{(-1),0}$$

The proposed method [1], performing margin-optimal feature elimination, then takes the following steps:

1. Determine the candidate feature set

$$S(i) \quad = \quad \{m \notin \mathcal{M} \mid \delta^{*^m} \leq \mathcal{N}^{(i-1),m_{i-1}}\}.$$

If $S(i)$ is empty (the data is nonseparable) then **stop**.[5]

2. For $m \in S(i)$, using recursion, compute[6]

$$g_n^{(i),m} = g_n^{(i-1),m_{i-1}} - \delta_n^m \quad \text{and} \quad L^{(i),m} = L^{(i-1),m_{i-1}} - w_m^2,$$

and determine

$$\mathcal{N}^{(i),m} = \min_n g_n^{(i),m} \text{ and } \gamma^{(i),m} = \max_{m \in S(i)} \frac{\mathcal{N}^{(i),m}}{\sqrt{L^{(i),m}}}.$$

3.1. Eliminate feature

$$m_i \quad \equiv \quad \arg\max_{m \in S(i)} \gamma^{(i),m}$$

(which maximizes the resulting margin), i.e., $\mathcal{M} \rightarrow \mathcal{M} \cup \{m_i\}$.

3.2. Keep for the next iteration only the recursive quantities for the eliminated feature: $g_n^{(i),m_i}\ \forall n, \mathcal{N}^{(i),m_i}, L^{(i),m_i}$

3.3. $i \rightarrow i+1$ and go to step 1.

In Figure 1, we demonstrate that MFE achieves both larger margin and better overall test set generalization accuracy than RFE on the UC Irvine *flag* data set[7]. Similar results are achieved on other data sets from the UC Irvine repository. More extensive evaluation on eight data sets from UC Irvine is given in [2].

[3] The method in this paper eliminates features to maximize the average distance to the hyperplane, over all the training points, rather than to maximize margin.

[4] Here, $i = -1$ means before eliminating any features and the 0 is a dummy placeholder index value, m_{-1}.

[5] The set $S(i)$ consists of the features at step i that, if singly eliminated, will preserve a positive margin. Note that $S(i)$ can be very efficiently computed given sorted δ^{*^m} values. For example, a 1-D bisection search can be used. Margin will only be evaluated for features in the set $S(i)$.

[6] Note that the terms δ_n^m and w_m^2 need only be computed for $m \in S(i)$ (and do *not* need to be computed during elimination steps if stored for all m and n during preprocessing (step 0)).

[7] Results are based on averaging over 21 trials created by three randomly chosen training/test splits of the data set and seven randomly chosen training subsets of the training set of each split.

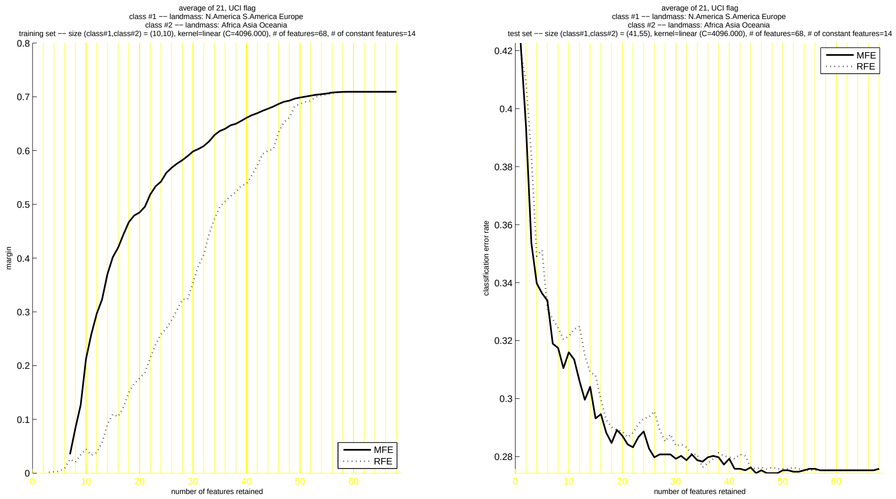

Fig. 1. Margin versus number of retained features (on left), and test error rate versus number of retained features (on right) for UC Irvine *flag* data set.

3. EXTENSIONS OF MARGIN-BASED FEATURE ELIMINATION

3.1. "Little Optimization" (LO): further increases in margin with modest computation

As noted earlier, for large M, it is not computationally practical to retrain the SVM in the reduced feature space, in conjunction with each feature elimination step. However, here we introduce a *type* of classifier retraining at every feature elimination step that is consistent with margin maximization and yet is *exceptionally* modest computationally, compared to full SVM retraining. 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 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 weight vector's length, and the affine parameter, but with the weight vector direction determined solely by the feature elimination steps. We thus pose the standard SVM training problem, but optimizing only in this *two*-dimensional parameter space:

$$\min_{A,w_0} A^2 \text{ s.t. } y_n(A(\underline{w}^{-\mathcal{M}^T}\underline{x}_n^{-\mathcal{M}})+w_0) \geq 1, n = 1, \ldots, N$$

It turns out [2] that solving this minimization problem requires almost *no* computation. In particular, the feasible region is defined by two cones in the (A, w_0) plane, with the minimum weight vector length (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. Since there is virtually no computation required, this "little optimization" can be performed in conjunction with each (margin-optimizing) feature elimination step. At each elimination step, this optimization is guaranteed to increase margin compared to the basic MFE technique described previously[8]. Figure 2 demonstrates improvement in margin (with a modest increase in generalization accuracy in this particular case) provided by the "little optimization" on the UCI *hepatitis* data set. In addition to this form of classifier retraining, for large M, it is practicable to periodically intersperse *full* SVM retrainings after a *batch* of features has been eliminated.

3.2. Limitations of RFE: nonlinear kernel case

For the case of a nonlinear kernel, a natural extension of the RFE method was proposed in [6]. In this case, the weight vector is only *implicitly* defined by the support vectors and the kernel function. However, using the "kernel trick", the squared weight vector length can be expressed *and easily evaluated* as:

$$||\underline{w}||^2 = \sum_{k\in\mathcal{S}}\sum_{l\in\mathcal{S}} \lambda_{\underline{s}_k} y_{\underline{s}_k} \lambda_{\underline{s}_l} y_{\underline{s}_l} K(\underline{s}_k, \underline{s}_l). \qquad (1)$$

Here, $\{\underline{s}_1, ..., \underline{s}_T\}$ is the set of support vectors (a subset of the original training points), with index set $\mathcal{S} = \{1, 2, ..., T\}$, $\lambda_{\underline{s}_k}$ is the (positive) Lagrange multiplier associated with support vector $\underline{s}_k$, and $K(\cdot, \cdot)$ is the kernel function. 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 [6], it was proposed to evaluate the weight vector length (1) both before and after a candidate feature elimination and, at the i-th stage of feature elimination, to remove the feature that minimizes the difference:

$$\Delta||\underline{w}||^2 = (||\underline{w}||^2)^{(i-1),m^*_{i-1}} - (||\underline{w}||^2)^{(i),m^*_i}, \qquad (2)$$

[8] Since this optimization is performed within a greedy (stepwise-optimal) framework, there is no guarantee that the margin curve for MFE applied in conjunction with "little optimization" will lie strictly above the margin curve for the basic MFE method – the "little optimization" will in general alter the (greedily chosen) sequence of margin-maximizing feature eliminations. There is only *guaranteed* strict improvement in the margin curve if the same feature elimination sequence is used by the two methods. However, experimentally we have found that the "little optimization" does typically lead to strictly positive shifts in the margin curve.

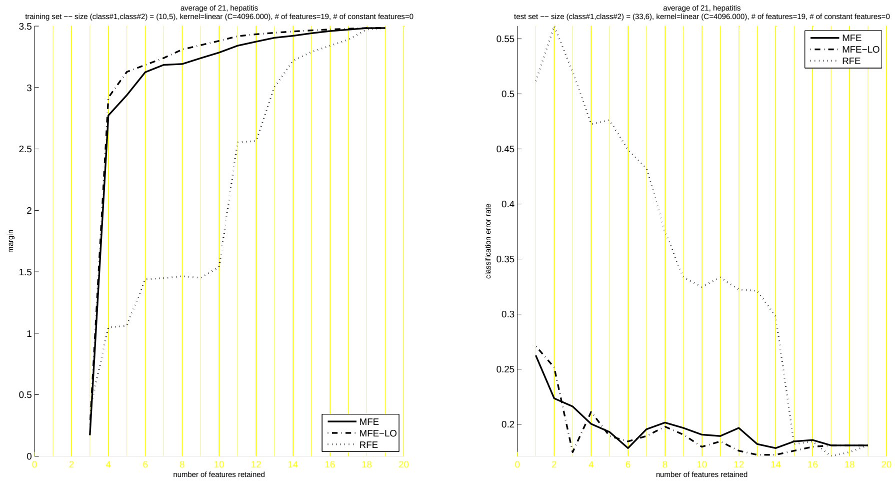

Fig. 2. Margin versus number of retained features (on left), and test classification error rate versus number of retained features (on right) for UC Irvine *hepatitis* data set.

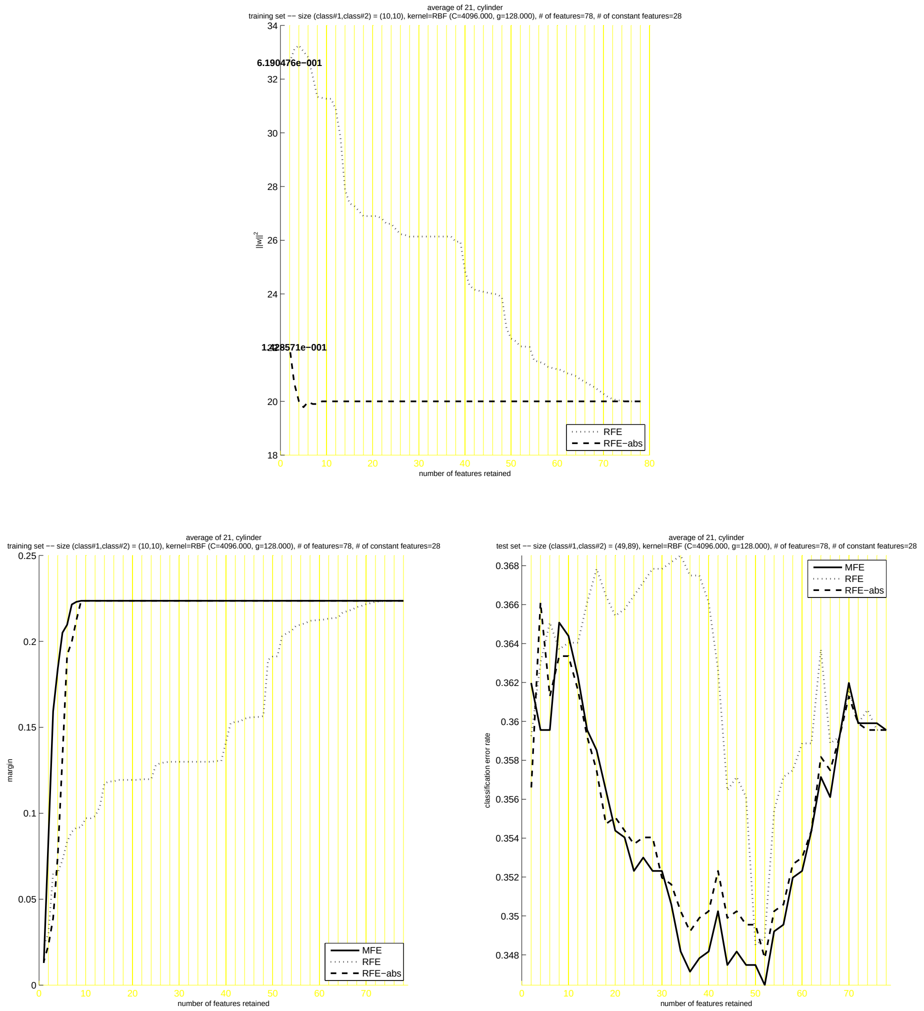

Fig. 3. Weight vector length squared versus number of retained features (at top), and margin versus number of retained features (on left), test classification error rate versus number of retained features (on right) for UC Irvine *cylinder* data set.

i.e. the length before elimination minus the length after elimination. This criterion is the natural extension of the linear RFE criterion and is consistent with the objective of reducing the weight vector length the least, *assuming that the weight vector length 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 function maps an original feature vector $\underline{u}$ to a new, finite-dimensional feature vector $\underline{\phi}(\underline{u})$ whose coordinates $\phi_i(\underline{u})$ are products raised to powers of the original feature coordinates. Since the weight vector length is $||\underline{\phi}(\underline{u})||^2 = \sum_{i=1}^{L(M)} |\phi_i(\underline{u})|^2$, where $L(M)$ is the dimension in the kernel-mapped space, it is clear for the polynomial case that eliminating an original feature coordinate zeroes out some components of $\underline{\phi}(\underline{u})$ while leaving all others unchanged. This effects zeroing (removing) the associated scalar weights. Thus, the weight vector length is monotonically decreasing as features are eliminated. However, we have also considered the Gaussian kernel. It is not so easy to analytically evaluate the Gaussian case. Instead, we have measured the weight vector length experimentally and found it is neither monotonically decreasing *nor* monotonically increasing with feature eliminations. Consider the consequences for the RFE objective (2): the RFE-optimal feature elimination (assuming some eliminations increase weight vector length) will in fact be the feature whose removal *increases* the weight vector length the *most*; this is the choice that will decrease (2) as much as possible (only, in this case, making $\Delta||\underline{w}||^2$ *negative*). In Figure 3, on the UC Irvine *cylinder* data set, for 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 length (This method is based on [9]). Note that standard RFE gives a steep rise in the weight vector length (averaged over different training trials) as features are eliminated, over a range of feature eliminations and, over this range, both margin and generalization accuracy are poor. Clearly, RFE_abs gives margin and generalization accuracy that are superior on this data set. We further note again, however, that RFE_abs is *itself* suboptimal with respect to classifier margin. In the nonlinear kernel case, the distance from a data point $\underline{x}_n$ to the decision boundary is evaluated via:

$$\frac{y_n(w_0 + \sum_{k\in\mathcal{S}} \lambda_{\underline{s}_k} y_{\underline{s}_k} K(\underline{s}_k, \underline{x}_n))}{\sqrt{\sum_{k\in\mathcal{S}} \sum_{l\in\mathcal{S}} \lambda_{\underline{s}_k} y_{\underline{s}_k} \lambda_{\underline{s}_l} y_{\underline{s}_l} K(\underline{s}_k, \underline{s}_l)}}$$

Similar to the pseudocode in Section 2.3.1, we propose a recursively-implemented margin-optimizing feature elimination algorithm 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:

$$\mathcal{K}_{k,n}^{(i),m} = \mathcal{K}_{k,n}^{(i-1),m_{i-1}} exp(\beta(s_{k,m} - x_{n,m})^2), \forall k, \forall n \qquad (3)$$

Figure 3 demonstrates increases in margin and generalization achieved by MFE_kernel over both RFE and RFE_abs on the *cylinder* UC Irvine data set. In some domains, *e.g.* on microarray data sets, it has been reported that linear SVMs achieve accuracy as least as good as nonlinear, kernel-based SVMs [5]. We believe a possible reason is that maximizing margin during feature elimination (which was not done in past work) may be especially an imperative in the nonlinear kernel case, if one is to avoid overfitting in high dimensions, with few training samples. Our MFE_kernel approach may allow not only improved results for large M compared with RFE (and RFE_abs), but it may also allow demonstrable gains in accuracy over linear SVMs as well, *i.e.* our approach may make kernel-based SVMs more attractive for domains with large M and few training samples. This will be explored in future work.

4. CONCLUSIONS

In this paper, we first reviewed our previous work on margin-based feature elimination for SVMs. We then introduced several extensions of this approach. One extension performs a very lightweight SVM training that adjusts the current solution in the reduced feature space to improve the margin. The second extension addresses the nonlinear kernel case. Here, we identified shortcomings of the standard RFE approach and demonstrated improved margin and accuracy achieved by a kernel version of MFE. In future work, we will consider extensions of our feature elimination method that allow for slackness in the margin constraints. At the conference, we will present full technical details for our methods and extensive simulation results, including results for large M.

References

[1] Y. Aksu, G. Kesidis, and D.J. Miller, "Scalable, efficient stepwise-optimal feature elimination in support vector machines", In *IEEE Workshop on MLSP*, 2007.

[2] Y. Aksu, D. J. Miller, G. Kesidis, "Improved Feature Elimination and SVM Optimization Techniques for Linear and Nonlinear Kernels", In preparation for journal submission, 2008.

[3] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines.* Cambridge University Press, 2000.

[4] R. Duda, P. Hart, G. Stork. *Pattern Classification*, Second Edition, John Wiley and Sons, New York, 2001.

[5] T. Furey et al., "Support vector machine classification and validation of cancer tissue samples using microarray expression data", *Bioinformatics*, vol. 16, no. 10, pp. 906-914, 2000.

[6] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. "Gene selection for cancer classification using support vector machines", *Machine Learning*, 46(1):389-422, 2002.

[7] I. Guyon, A. Elisseeff. " An introduction to variable and feature selection", *J. Machine Learning Research*, vol. 3, pp. 1157-1182, 2003.

[8] Hastie T, Tibshirani R, Friedman J. *The Elements of Statistical Learning*, New York: Springer, 2001.

[9] M. Kugler et al. "Feature subset selection for support vector machines using confident margin", Proc. IJCNN, pp. 907-912, 2005.

[10] Trunk GV, "A problem of dimensionality: A simple example", *IEEE Trans. PAMI* vol. 1, pp. 306-307, 1979.

[11] J. Weston et al. Feature selection for SVMs. Advances in Neural Information Processing Systems 13. MIT Press, 2001.

[12] J. Weston, A. Elisseeff, B. Scholkopf, M. Tipping. Use of the zero norm with linear models and kernel methods. J. Mach. Learn. Res. 3 (Mar. 2003), 1439-1461.