Reducing the Amount of Support Vectors of SVM Classifiers using Separable Case Approximation

Dries Geebelen, Johan A.K. Suykens, and Joos Vandewalle

Abstract

In this brief, we present an algorithm, called Separable Case Approximation (SCA), that reduces the amount of support vectors of support vector machine classifiers. The algorithm consists of three steps: first it uses Vapnik's SVM classifier for the non-separable case to find a solution, then it removes misclassified examples from the training set such that the training set becomes separable and finally it separates the data set using Vapnik's SVM classifier for the separable case. We also propose an algorithm, called Smoothed Separable Case Approximation (SSCA), that additionally smoothens the pruned solution. SCA manages to reduce the amount of support vectors by 85% to 95% for most of the benchmarks with only a small decrease in test set accuracy. SSCA reduces the amount of support vectors even more when using a radial basis function (RBF) kernel.

Index Terms

Support vector machine classifier, pruning, sparse approximation.

This work was supported by the Research Council K.U.Leuven (GOA Ambiorics, GOA MaNet, CoE EF/05/006 Optimization in Engineering(OPTEC), IOF-SCORES4CHEM, several PhD/postdoc & fellow grants), the Flemish Government (FWO: PhD/postdoc grants, projects: G0226.06 (cooperative systems and optimization), G0321.06 (Tensors), G.0302.07 (SVM/Kernel), G.0320.08 (convex MPC), G.0558.08 (Robust MHE), G.0557.08 (Glycemia2), G.0588.09 (Brain-machine) research communities (ICCoS, ANMMM, MLDM); G.0377.09 (Mechatronics MPC)), IWT (PhD Grants, Eureka-Flite+, SBO LeCoPro, SBO Climaqs, SBO POM, O&O-Dsquare), the Belgian Federal Science Policy Office (IUAP P6/04 (DYSCO, Dynamical systems, control and optimization, 2007-2011)), EU (ERNSI; FP7-HD-MPC (INFSO-ICT-223854), COST intelliCIS, FP7-EMBOCON (ICT-248940)), Contract Research (AMINAL) and others (Helmholtz: viCERP, ACCM, Bauknecht, Hoerbiger).

D. Geebelen, J.A.K. Suykens and J. Vandewalle are with the Department of Electrical Engineering (ESAT-SCD-SISTA). Katholieke Universiteit Leuven, B-3001 Leuven (Heverlee), Belgium. Phone: +32-16-321709, fax: +32-16-321970, e-mail: dries.geebelen@esat.kuleuven.be.

I. Introduction

Classifying data is a common task in machine learning. Given a set of input vectors $\{x_k\}_{k=1}^N$ along with corresponding class labels $\{y_k\}_{k=1}^N$, the task is to find a static function that defines the relation between input vectors and their class labels. Vapnik's support vector machine (SVM) classifier [1], [2] is a classifier that obtains good accuracy on real life data sets, as shown e.g. in [3] and [4]. Compared to other classification algorithms, the SVM classifier has an important drawback: the run time complexity, as explained in [5] and [7], can be considerably higher due to the large amount of support vectors.

A lot of research has already been done considering the pruning of the SVM solution. In Burges [5] and Schölkopf et al.[6], an algorithm is proposed that approximates the weight vector such that the distance to the original weight vector is minimized. Osuna and Girosi [7] used SVM itself as a regression tool to approximate the decision surface and presented a reformulation of the training problem. Suykens et al. [8] proposed a pruning procedure for least squares support vector machines (LS-SVM) which is based upon the sorted support value spectrum (SVS). De Kruif & De Vries [9] further sophisticated this pruning method. Cawley & Talbot [10] and Licheng Jiao et al. [11] presented a sparse approximation algorithm for LS-SVM that iteratively builds the decision function by adding one basis function at a time.

In this paper, an algorithm, called Seperable Case Approximation (SCA), is proposed that increases the sparsity of support vector machine classifiers. This pruning method can be seen as a post-training filter. SCA first makes the training set separable by removing misclassified examples and then separates the data set using Vapnik's SVM classifier for the separable case. Additionally, we propose an extension called Smoothed Separable Case Approximation (SSCA) that assures smoothness of the pruned solution. For most data sets,

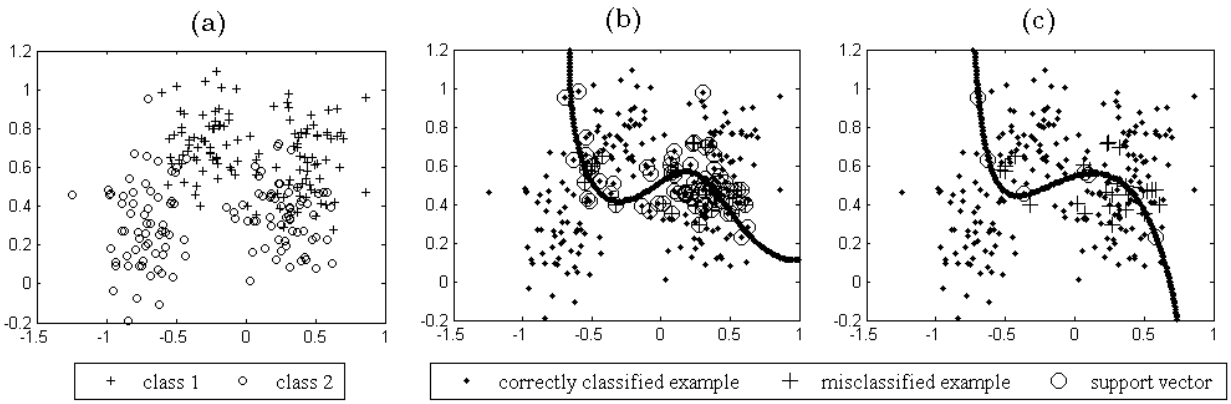


Fig. 1. (a) Training examples of the Ripley data set with corresponding labels; (b) non-pruned solution using Vapnik's SVM classifier for the non-separable case with $\sigma = 1$ and $c = 100$; (c) pruned solution using SCA.

SSCA results in even less support vectors than SCA when using a radial basis function (RBF) kernel. Although these algorithms can be applied to different kinds of SVM classifiers (e.g. Vapnik's SVM classifier and the LS-SVM classifier), only the application on Vapnik's SVM classifier is discussed in this paper.

The remainder of this brief is organized as follows. In Section II the SVM classifier for both the separable and non-separable case are shortly reviewed. The proposed pruning algorithm and its properties are discussed in Section III. In Section IV we explain the extension that additionally smoothens the pruned solution. In Section V the accuracy of the algorithm is measured on real life data sets. Finally, some concluding remarks are made in Section VI.

II. Support Vector Machine Classifier

A. Separable Case

Given a set of N training examples with input vectors $\{x_k \in \mathbb{R}^n\}_{k=1}^N$ and corresponding labels $\{y_k \in \{-1, 1\}\}_{k=1}^N$, Vapnik's SVM classification algorithm builds a linear classifier in

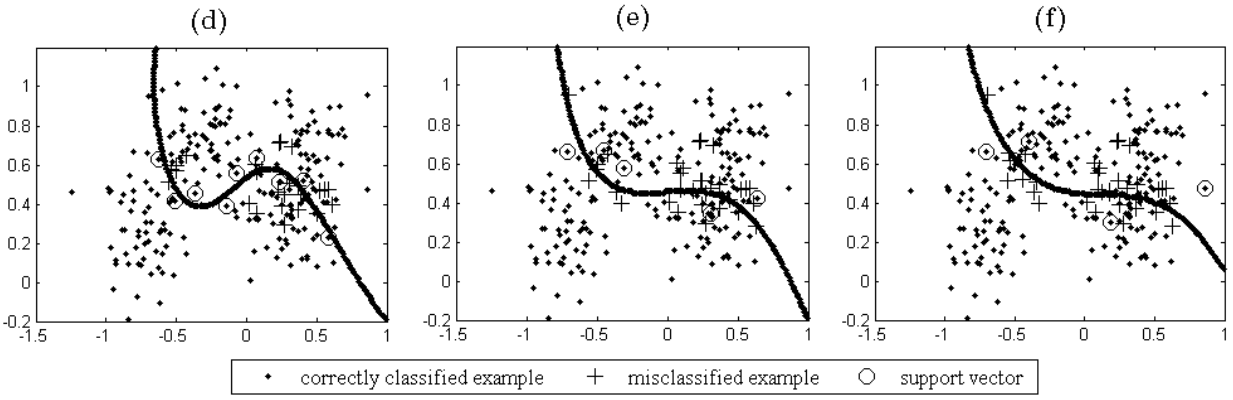


Fig. 2. SSCA applied to the Ripley data set with $\sigma = 1$ and $c = 100$. Parameter d differs per figure: (d) $d = 0.3$; (e) $d = 1.3$; (f) $d = 2.0$.

the feature space

$$\hat{y} = \text{sign}(w^T \varphi(x) + b) \tag{1}$$

where the feature map $\varphi(x)$ maps the input data into a feature space (which can be infinite dimensional). In the separable case, a hyperplane is found that separates the positive from the negative training examples such that the distance between this hyperplane and the training point closest to this hyperplane is maximized. It solves the following primal problem in w and b:

$$\min_{w,b} J_P(w) = \frac{1}{2} w^T w \tag{2}$$

$$\text{such that } y_k[w^T \varphi(x_k) + b] \geq 1, \qquad k = 1, ..., N. \tag{3}$$

The dual problem is

$$\max_{\alpha} J_D(\alpha) = -\frac{1}{2} \sum_{k,l=1}^{N} y_k y_l K(x_k, x_l) \alpha_k \alpha_l + \sum_{k=1}^{N} \alpha_k$$

$$\text{such that } \begin{cases} \sum_{k=1}^{N} \alpha_k y_k = 0 \\ \alpha_k \geq 0, \qquad k = 1, ..., N. \end{cases} \tag{4}$$

Explicit calculations in an infinite or high-dimensional feature space are impossible or computational very expensive. According to Mercer's theorem [12], any positive-definite kernel function can be expressed as

$$K(x,z) = \varphi(x)^T \varphi(z).$$

Due to the kernel trick, we can now work in huge dimensional feature spaces without having to do explicit calculations in this space. A popular kernel function is the radial basis function (RBF) kernel ($K(x,x_k) = \exp(-\|x-x_k\|_2^2/\sigma^2)$). The predicted label (1) of a new data point becomes:

$$\hat{y} = \text{sign}\left(\sum_{k=1}^{N} \alpha_k y_k K(x,x_k) + b\right). \tag{5}$$

B. Non-Separable Case

In the non-separable case, Vapnik's SVM classifier tolerates misclassifications in the training set. This is done by taking additional slack variables (ξ_k) in the problem formulation. In the primal weight space the optimization becomes

$$\min_{w,b} J_P(w) = \frac{1}{2} w^T w + c \sum_{k=1}^{N} \xi_k$$

$$\text{such that } \begin{cases} y_k[w^T \varphi(x_k) + b] \geq 1 - \xi_k & k = 1,...,N \\ \xi_k \geq 0 & k = 1,...,N. \end{cases}$$

where c is a positive real constant. The hyperparameter c determines the trade-off between the margin and the training error. For c going to infinity, the non-separable case converges

to the separable case. The dual after applying the kernel trick is

$$\max_{\alpha} J_D(\alpha) = -\frac{1}{2}\sum_{k,l=1}^{N} y_k y_l K(x_k, x_l)\alpha_k\alpha_l + \sum_{k=1}^{N}\alpha_k$$
$$\text{such that } \begin{cases} \sum_{k=1}^{N}\alpha_k y_k = 0 \\ 0 \leq \alpha_k \leq c \qquad k = 1, ..., N. \end{cases} \tag{6}$$

The predicted label for a new data point is equal to (5).

III. Separable Case Approximation

As stated in the introduction, a drawback of SVM classifiers is the amount of support vectors. For the separable case, the training data points where

$$y_k[w^T\varphi(x_k) + b] = 1$$

are support vectors. This means that all the support vectors are on the hyperplane where $w^T\varphi(x_k) + b = 1$ and the hyperplane where $w^T\varphi(x_k) + b = -1$. For the non-separable case, the support vectors are characterized by

$$y_k[w^T\varphi(x_k) + b] \leq 1.$$

We can divide the latter support vectors in two subsets. The first subset contains all misclassified examples ($y_k[w^T\varphi(x_k) + b] \leq 0$) and the second subset contains the correctly classified training examples close to the decision boundary in the feature space ($0 \leq y_k[w^T\varphi(x_k) + b] \leq 1$). Ideally we reduce the amount of support vectors in both subsets. We propose the following pruning algorithm:

Separable Case Approximation (SCA):

1. Find the non-pruned solution using Vapnik's SVM for the non-separable case (6).

TABLE I

SCA APPLIED TO THE RIPLEY DATA SET. SV IS THE AMOUNT OF SUPPORT VECTORS, $\mathrm{ACC}_{\mathrm{TR}}$ IS THE TRAINING SET ACCURACY AND $\mathrm{ACC}_{\mathrm{TEST}}$ IS THE TEST SET ACCURACY.

	SV	$\mathrm{ACC}_{\mathrm{TR}}(\%)$	$\mathrm{ACC}_{\mathrm{TEST}}(\%)$
non-pruned	78	89.6	89.7
pruned	7	90.0	89.0

TABLE II

SSCA APPLIED TO THE RIPLEY DATA SET. d IS A PARAMETER OF THE SSCA ALGORITHM, SV IS THE AMOUNT OF SUPPORT VECTORS, $\mathrm{ACC}_{\mathrm{TR}}$ IS THE TRAINING SET ACCURACY AND $\mathrm{ACC}_{\mathrm{TEST}}$ IS THE TEST SET ACCURACY, $\|w\|_2$ IS THE 2-NORM OF THE WEIGHT VECTOR.

d	SV	$\mathrm{ACC}_{\mathrm{TR}}(\%)$	$\mathrm{ACC}_{\mathrm{TEST}}(\%)$	$\|w\|_2$
non-pruned	78	89.6	89.7	25
0	7	90.0	89.0	236
0.3	9	89.6	89.9	47
0.7	8	89.2	89.7	26
1	6	88.0	90.1	14
1.3	5	87.2	90.2	11
2.0	4	85.6	89.5	7

2. Remove misclassified examples ($y_k[w^T\varphi(x_k)+b] \leq 0$) such that the training set becomes separable.
3. Find the SVM-solution of the reduced training set for the separable case (4).

In the final solution (after step 3), all training examples are taken into account because they all determine, in the first step, which training examples will be classified correctly. The support vectors, however, cannot be misclassified training examples. We expect the final support vectors to be correctly classified examples close to the original separating hyperplane. In a lot of cases, although not all, most of the support vectors of the pruned solution are correctly classified support vectors of the non-pruned solution. Vapnik's SVM classifier for the separable case will, in general, not cause more support vectors than $p+1$, where p is the number of dimensions of the feature space. This upper bound exists because, as stated in the beginning of this section, all support vectors lie on one of two parallel hyperplanes

TABLE III

PROPERTIES OF THE BENCHMARK DATA SETS. N_{TEST} STANDS FOR THE NUMBER OF TEST DATA POINTS, N_{TR} FOR NUMBER OF THE TRAINING DATA POINTS, N_{INP} FOR THE NUMBER OF INPUT DIMENSIONS AND N_{CLA} FOR THE NUMBER OF CLASSES.

	bld	hea	adu	shu
N_{test}	115	90	33000	43500
N_{tr}	230	180	12222	14500
n_{inp}	6	13	123	9
n_{cla}	2	2	2	7

in the feature space and the degrees of freedom for choosing two parallel hyperplanes in a p dimensional space equals $p+1$. This means that, in general, it is not possible to choose the hyperplanes such that $p+2$ points or more lie on it. Because the final solution of SCA is found using Vapnik's SVM classifier for the separable case, this upper bound applies to SCA as well. A final remark is that the training set accuracy of the pruned result will not be lower because correctly classified examples remain classified correctly. This, however, does not mean that the test set accuracy will be higher.

We conclude this section by applying SCA to the Ripley data set using a SVM-classifier with RBF kernel. The Ripley data set contains 250 training data points and 1000 test data points. The found classifier is visualized in Figure 1 and the resulting accuracies are shown in Table I. We observe a drastic reduction in the amount of support vectors with a slight loss in test set accuracy. As expected, the training set accuracy is higher after the pruning. We can see on Figure 1 that the support vectors of the pruned solution are a subset of the correctly classified support vectors of the non-pruned solution.

IV. Smoothed Separable Case Approximation

A correctly classified example belonging to one class and a correctly classified example belonging to the other class can lie arbitrarily close to each other in the feature space. This implies that, for the solution of the separable case approximation, the hyperplane where

$w^T\varphi(x_k)+b=1$ and the hyperplane where $w^T\varphi(x_k)+b=-1$ can lie arbitrarily close to each other. The weight vector of the pruned solution ($\|w_{pr}\|_2$) can thus become arbitrarily large. The lower $\|w_{pr}\|_2$, the smoother the solution. We propose the following algorithm for adding additional smoothness (w_{npr} and b_{npr} are part of the non-pruned solution):

Smoothed Separable Case Approximation (SSCA):

1. Find the SVM-solution using Vapnik's SVM for the non-separable case (6).
2. Remove examples for which $y_k[w_{npr}^T\varphi(x_k)+b_{npr}]<d$ with $d>0$.
3. Find the SVM-solution of the reduced training set for the separable case (4).

In the reduced training set, all examples satisfy:

$$y_k\left[\left(\frac{w_{npr}}{d}\right)^T\varphi(x_k)+\frac{b_{npr}}{d}\right]\geq 1.$$

Because w_{pr}, the weight vector of the pruned solution, is found according to the maximal margin criterion (2), its 2-norm is the smallest of all weight vectors satisfying (3) for the reduced training set. Hence, the following counts

$$\|w_{pr}\|_2\leq\frac{\|w_{npr}\|_2}{d}. \tag{7}$$

Using SSCA, we can assure additional smoothness. For a lot of data sets, the additional smoothness will also result in a lower amount of support vectors when using a RBF kernel. There are several ways for choosing a good value for d. One could use, for example, a validation set to determine the best value of d. Picking the fixed value 1 for d, however, gives in most cases a good trade off between smoothness and training error. We can assure smoothness because the pruned solution is at least as smooth as the non-pruned solution, as derived in (7). On the other hand, examples that cause no error in the non-pruned solution

will be classified correctly in the pruned solution as well. In practice it is not unlikely that choosing d slightly smaller, for example 0.5, gives better test accuracy as shown in Section V. Similiar as for SCA, an upper bound for the number of support vectors is the number of dimensions in the feature space plus one. Opposed to SCA, the training set accuracy can be lower than that of the non-pruned solution because the correctly classified examples of the non-pruned solution for which $y_k[w_{npr}^T \varphi(x_k) + b_{npr}] < d$ can become misclassified in the pruned solution.

The results for the Ripley data set are shown in Table II and visualized in Figure 2. The additional smoothing reduces the training set accuracy, whereas the test set accuracy remains similar. Due to the already low amount of support vectors, we notice only a small overall decrease in the amount of support vectors for an increasing value of d. We can clearly see $\|w\|_2$ decrease for an increasing value of d.

V. Experimental Results

In this section, we will measure the accuracy of SCA and SSCA on four benchmark data sets: Bupa liver disorder (*bld*), Statlog heart disease (*hea*), Adult (*adu*) and Statlog shuttle (*shu*). All the data sets have been obtained from the UCI benchmark repository [13]. The properties of the data sets are shown in Table III. The data set *shu* contains 7 classes. A one-versus-all approach was taken. LIBSVM [14] is the software used to find both the non-pruned and the pruned solution. For the pruned solution we approximate the separable case by taking 10^8 times the value of c used for the non-pruned solution. The chosen hyperparamaters of data sets *bld*, *hea*, and *adu* are the same as in [4]. In this paper 10-fold crossvalidation is used to determine the optimal hyperparameters. For the data set *shu*, we use the same hyperparameters as in [15]. Here, the hyperparameters are optimized using 30% of the training set as a validation set. As a preprocessing step, all records containing

missing values are removed. For all data sets the average accuracies, average amount of support vectors and the standard deviation of the accuracies are obtained on at least 10 randomizations.

The resulting classifications are summarized in Tabel III. SCA ($d = 0$) and SSCA ($d > 0$) reduce the amount of support vectors significantly for the linear kernel. When using a linear kernel, the feature space equals the input space. As explained in Section III, the number of support vectors will, in general, not exceed the number of dimensions of the feature space plus one. Both $bld(lin)$ and $adu(lin)$ confirm this upperbound. For $bld(rbf)$, $hea(rbf)$ and $adu(rbf)$ the amount of support vector is reduced by 80% to 95% when using SCA. For SSCA this amount is further reduced. When the hyperparamter c of the non-pruned solution is chosen high, like in $shu(rbf)$, the non-separable case approximates the separable case. In this case, the solution found by SCA and SSCA will not differ much from the non-pruned solution and the reduction in the amount of support vectors will be small. The non-pruned solution has on average the highest test set accuracy. SSCA with $d = 0.5$ has slightly lower test set accuracy than SCA for some data sets ($shu(rbf)$ and $bld(lin)$) and slightly higher test set accuracy for others ($bld(rbf)$ and $hea(rbf)$). SSCA with $d = 1$ has on average a slightly worse average test set accuracy than both SCA and SSCA with $d = 0.5$. For all data sets, the training set accuracy of SCA exceeds the one of the non-pruned solution, as expected. For an increasing value of d the training error increases and the generalization error decreases.

VI. Conclusion

In this brief we succeeded to drastically reduce the amount of support vectors of SVM classifiers using Separable Case Approximation (SCA). SCA can be extended to Smoothed Separable Case Approximation (SSCA) in case we want the pruned solution to be smooth.

With SCA we managed to reduce the amount of support vectors by 85% to 95% for most of the benchmarks with only a small decrease in test set accuracy. SSCA reduces the amount of support vectors even more and obtains a slightly better test set accuracy than SCA for some data sets and a slightly worse for others.

References

[1] C. Cortes, V. Vapnik, "Support vector networks," *Mach. Learning,* vol 20, pp. 273-297, 1995.

[2] V. Vapnik, *Statistical learning theory,* New York: John Wiley & Sons, 1998.

[3] D. Meyer, F. Leisch, and K. Hornik, "The support vector machine under test," *Neurocomput.,* vol 55(1-2), pp 169-186, 2003.

[4] T. Van Gestel, J.A.K. Suykens, G. Lanckriet, A. Lambrechts, B. De Moor, and J. Vandewalle, "Benchmarking least squares support vector machine classifiers," *Mach. Learning,* vol 54, no. 1, pp 5-32, 2004.

[5] C.J.C. Burges, "Simplified support vector decision rules," *Proc. 13th Int. Conf. Mach. Learning,* pp. 71-77, 1996.

[6] B. Schölkopf, S. Mika, C.J.C. Burges, P. Knirsch, K. Muller, G. Rätsch, and A.J. Smola "Input space versus feature space in kernel-based methods," *IEEE Trans. Neural Netw.,* vol 10, no. 5, pp. 1000-1017, September 1999.

[7] E. Osuna and F. Girosi, "Reducing the run-time complexity of Support Vector Machines," *Int. Conf. Pattern Recognition*, Brisbane, Australia, August 16-20 1998.

[8] J.A.K. Suykens, J. Debrabanter, L. Lukas, and J. Vandewalle, "Weighted least squares support vector machines: Robustness and sparse approximation," *Neurocomput.,* vol 48, pp. 85-105, 2002.

[9] B.J. De Kruif and T.J.A. De Vries, "Pruning error minimzation in least squares support

vector machines," *IEEE Trans. Neural Netw.,* vol 14, no. 3, pp. 696-702, May 2004.

[10] G.C. Cawley and N.L.C. Talbot, "A greedy training algorithm for sparse least-squares support vector machines," *Proc. Int. Conf. on Artificial Neural Netw.,* Madrid, Spain, August 23-30 2002, vol 2415. *Lecture Notes in Computer Science,* pp. 681-686, Berlin: Springer.

[11] L. C. Jiao, L. F. Bo and L. Wang, "Fast sparse approximation for least squares support vector machine" *IEEE Trans. Neural Netw.,* vol 18, no. 3, pp. 685-697, May 2007.

[12] N. Aronszajn, "Theory of reproducing kernels," *Trans. Amer. Math. Soc.,* vol 686, pp. 337-404, 1950.

[13] C. L. Blake and C.J. Merz (1998), UCI Repository of machine learning databases. Irvine, CA: University of California, Dept. of Information and Computer Science. Available: http://www.ics.uci.edu/∼ mlearn/MLRepository.html.

[14] C.-C. Chang and C.-J. Lin (2001), LIBSVM: a library for support vector machines. Available: http://www.csie.ntu.edu.tw/∼cjlin/libsvm.

[15] C.-W. Hsu and C.-J. Lin, A comparison on methods for multi-class support vector machines, *IEEE Trans. Neural Netw.,* vol 13, pp. 415-425, March 2002.

TABLE IV

SCA ($d=0$) AND SSCA ($d>0$) APPLIED TO BENCHMARK DATA SETS. THE USED KERNELS ARE THE RADIAL BASIS FUNCTION (RBF) AND THE LINEAR KERNEL (LIN). d IS A PARAMETER OF THE ALGORITHM, SV IS THE AVERAGE AMOUNT OF SUPPORT VECTORS, $\mathrm{ACC_{TR}}$ IS THE AVERAGE TRAINING SET ACCURACY AND ITS STANDARD DEVIATION AND $\mathrm{ACC_{TEST}}$ IS THE AVERAGE TEST SET ACCURACY AND ITS STANDARD DEVIATION.

(a) Bupa liver disorder (RBF): $c=44$, $\sigma=9.0$.

d	SV	$\mathrm{ACC_{TR}}(\%)$	$\mathrm{ACC_{TEST}}(\%)$
Non-Pruned	161.0	77.2 ± 1.8	72.5 ± 3.5
0	19.4	78.6 ± 1.7	71.6 ± 3.7
0.5	16.3	76.0 ± 1.9	72.1 ± 3.6
1.0	13.8	74.9 ± 2.1	71.3 ± 3.9
1.5	10.2	72.0 ± 2.8	68.6 ± 4.8

(b) Statlog heart disease (RBF): $c=0.31$, $\sigma=7.15$.

d	SV	$\mathrm{ACC_{TR}}(\%)$	$\mathrm{ACC_{TEST}}(\%)$
Non-Pruned	121.6	86.1 ± 3.0	84.9 ± 1.5
0	24.9	87.5 ± 3.0	84.2 ± 1.6
0.5	23.3	85.2 ± 3.2	84.8 ± 1.8
1.0	16.1	85.0 ± 3.1	84.5 ± 1.9
1.5	5.2	81.7 ± 11.2	80.5 ± 9.0

(c) Adult (RBF): $c=5$, $\sigma=8$.

d	SV	$\mathrm{ACC_{TR}}(\%)$	$\mathrm{ACC_{TEST}}(\%)$
Non-Pruned	11896	84.9 ± 0.1	84.3 ± 0.3
0	587	85.3 ± 0.1	84.3 ± 0.3
0.5	259	84.6 ± 0.1	84.3 ± 0.3
1.0	138	84.3 ± 0.1	84.2 ± 0.3
1.5	76	84.2 ± 0.1	84.1 ± 0.2

(d) Statlog shuttle (RBF): $c=512$, $\sigma=16$.

d	SV	$\mathrm{ACC_{TR}}(\%)$	$\mathrm{ACC_{TEST}}(\%)$
Non-Pruned	589	99.97 ± 0.00	99.87 ± 0.00
0	465	99.98 ± 0.00	99.87 ± 0.00
0.5	477	99.97 ± 0.00	99.86 ± 0.00
1.0	396	99.91 ± 0.00	99.82 ± 0.00
1.5	162	97.77 ± 0.01	97.72 ± 0.01

(e) Bupa liver disorder (Lin): $c=37.2$.

d	SV	$\mathrm{ACC_{TR}}(\%)$	$\mathrm{ACC_{TEST}}(\%)$
Non-Pruned	166.4	70.9 ± 2.2	68.0 ± 4.0
0	7.0	72.3 ± 2.1	68.3 ± 3.9
0.5	7.0	71.0 ± 2.1	68.0 ± 4.1
1.0	6.9	70.5 ± 2.5	67.4 ± 4.1
1.5	6.4	69.5 ± 3.0	65.7 ± 4.6

(f) Adult (Lin): $c=0.005$.

d	SV	$\mathrm{ACC_{TR}}(\%)$	$\mathrm{ACC_{TEST}}(\%)$
Non-Pruned	12967	84.1 ± 0.1	83.9 ± 0.3
0	75	84.2 ± 0.1	83.9 ± 0.3
0.5	69	84.1 ± 0.1	83.9 ± 0.3
1.0	76	84.0 ± 0.1	83.8 ± 0.3
1.5	57	83.9 ± 0.1	83.7 ± 0.3