Journal of Machine Learning Research 2 (2001) 293-297 Submitted 3/01; Published 12/01

Exact Simplification of Support Vector Solutions

Tom Downs TD@ITEE.UQ.EDU.AU
School of Information Technology and Electrical Engineering
University of Queensland,
Brisbane, Q. 4072, Australia

Kevin E Gates KEG@MATHS.UQ.EDU.AU
School of Information Technology and Electrical Engineering
and Department of Mathematics
University of Queensland,
Brisbane, Q. 4072, Australia

Annette Masters AMASTERS@MATHS.UQ.EDU.AU
Department of Mathematics
University of Queensland
Brisbane, Q. 4072, Australia

Editor: Nello Cristianini, John Shaw-Taylor, and Robert C. Williamson

Abstract

This paper demonstrates that standard algorithms for training support vector machines generally produce solutions with a greater number of support vectors than are strictly necessary. An algorithm is presented that allows unnecessary support vectors to be recognized and eliminated while leaving the solution otherwise unchanged. The algorithm is applied to a variety of benchmark data sets (for both classification and regression) and in most cases the procedure leads to a reduction in the number of support vectors. In some cases the reduction is substantial.
Keywords: Support vector machines, kernel methods.

1. Introduction

The time taken for a support vector classifier to compute the class of a new pattern is proportional to the number of support vectors, so if that number is large, classification speed is slow. This can be a problem for some applications and Burges (1996) described a method of speeding up the classification process by approximating the solution using a smaller number of vectors. The "reduced set" of vectors determined by Burges' method are generally not support vectors. When applied to the NIST data set using second and third degree polynomial kernels, speed improvements of an order of magnitude were achieved with very small effects on generalization performance. The method was somewhat refined in (Burges and Schoelkopf, 1997) and again applied to the NIST data set, this time using fifth degree polynomial kernels, and 20-fold gains in speed were achieved, again without significant impairment to generalization performance. The reduced set of vectors used in this method is computed from the original support vector set in such a way that it provides the best approximation to the original decision surface. Unfortunately, determination of this reduced set proved to be computationally expensive and the approach seems not to have been pursued further.

More recently it has been shown (Syed et al., 1999) that the discarding of even a small proportion of the support vectors can lead to a severe reduction in generalization performance. Syed et al. (1999) stated that this implies that the support vector set chosen by the SVM is a minimal set. But Burges (1996) and Burges and Schoelkopf (1997) had already pointed out that there exist non-trivial cases where the reduced set approximation is exact, showing that the support vector set delivered by the SVM is not always minimal. Burges and Schoelkopf offered no explanation for this phenomenon. In this paper we explain circumstances in which a reduced set can be computed that avoids any approximation. The required computation is essentially trivial and basically involves discarding unnecessary support vectors and modifying the Lagrange multipliers of the remaining support vectors so that the solution remains unchanged.

2. Reducing the Number of Support Vectors

2.1 Pattern Classification

Suppose we train an SVM classifier with pattern vectors $\boldsymbol{x}_i$, and that r of these are determined to be support vectors. Denote them by $\boldsymbol{x}_i$, $i = 1,2, \ldots ,r$. The decision surface for pattern classification then takes the form

$$f(\boldsymbol{x}) = \sum_{i=1}^{r} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b \tag{1}$$

where α_i is the Lagrange multiplier associated with pattern $\boldsymbol{x}_i$ and $K(\cdot\,, \cdot)$ is a kernel function that implicitly maps the pattern vectors into a suitable feature space.

Now suppose that support vector $\boldsymbol{x}_k$ is linearly dependent on the other support vectors in feature space, *i.e.*

$$K(\boldsymbol{x}, \boldsymbol{x}_k) = \sum_{\substack{i=1 \\ i \neq k}}^{r} c_i K(\boldsymbol{x}, \boldsymbol{x}_i) \tag{2}$$

where the c_i are scalar constants.

Then the decision surface (1) can be written

$$f(\boldsymbol{x}) = \sum_{\substack{i=1 \\ i \neq k}}^{r} \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + \alpha_k y_k \sum_{\substack{i=1 \\ i \neq k}}^{r} c_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b \tag{3}$$

Now define $\alpha_k y_k c_i = \alpha_i y_i \gamma_i$ so that (3) can be written

$$\begin{aligned} f(\boldsymbol{x}) &= \sum_{\substack{i=1 \\ i \neq k}}^{r} \alpha_i (1 + \gamma_i) y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b \\ &= \sum_{\substack{i=1 \\ i \neq k}}^{r} \bar{\alpha}_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b \end{aligned} \tag{4}$$

$$\text{where } \bar{\alpha}_i = \alpha_i (1 + \gamma_i) \tag{5}$$

Comparing (4) and (1) we see that the linearly dependent support vector is not required in the representation of the decision surface. Note, however, that the Lagrange multipliers must be

modified according to (5) in order to obtain the simplified representation. But this is a very simple modification that can be applied to any linearly dependent support vector that is identified.

2.2 Regression

For regression problems the support vector solution takes the form

$$f(\boldsymbol{x}) = \sum_{i=1}^{r} (\alpha_i - \alpha_i^*) K(\boldsymbol{x}, \boldsymbol{x}_i) + b$$

which can be written

$$f(\boldsymbol{x}) = \sum_{i=1}^{r} \beta_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b \tag{6}$$

where $\beta_i = \alpha_i$ or $\beta_i = -\alpha_i^*$, depending on which constraint is active.

Now suppose support vector $\boldsymbol{x}_k$ is linearly dependent on the other support vectors in feature space according to (2). We then find, as for the classification case, that $\boldsymbol{x}_k$ can be eliminated from (6) and this gives

$$f(\boldsymbol{x}) = \sum_{i=1}^{r} \bar{\beta}_i K(\boldsymbol{x}, \boldsymbol{x}_i) + b \tag{7}$$

where $\bar{\beta}_i = \beta_i + \beta_k c_i$.

3. Results

In order to determine the simplifications that are possible using this method, we need to be able to identify support vectors that are linearly dependent in feature space. This can be done using techniques from elementary linear algebra and we employ the row reduced echelon form (Noble and Daniel, 1988).

The results below indicate the degree of simplification achieved on solutions obtained using standard SVM procedures on several benchmark data sets. We do not give any indication of generalization performance because the simplifications leave performance unchanged.

3.1 Classification

Results for classification were obtained using four data sets arbitrarily selected from the UCI database. We employed the SMO algorithm (Platt, 1999) to generate an initial support vector set and then applied the method described in Section 2.1 to eliminate support vectors that are linearly dependent in feature space. We used linear, polynomial and RBF kernels and the results are detailed in Tables 1 – 3.

Linear Kernel $(\mathbf{x}_i \cdot \mathbf{x}_j)$		Data Sets			
		Contraceptive Method Choice	Diabetes	Haberman	Wisconsin Breast Cancer
SMO	# SVs	28	100	16	21
Modified set	# SVs	7	8	3	9
Reduction in SVs		75.0%	92.0%	81.25%	57.14%

Table 1 : Results using the linear kernel

Polynomial Kernel $(\mathbf{x}_i \cdot \mathbf{x}_j + 1)^d$			Data Sets			
			Contraceptive Method Choice	Diabetes	Haberman	Wisconsin Breast Cancer
d=1	SMO	# SVs	9	79	42	21
	Modified set	# SVs	7	9	4	10
	Reduction in SVs		22.22%	88.61%	90.45%	52.38%
d=2	SMO	# SVs	12	48	87	12
	Modified set	# SVs	11	42	10	11
	Reduction in SVs		8.33%	12.5%	88.51%	8.33%
d=3	SMO	# SVs	23	171	91	17
	Modified set	# SVs	23	141	18	17
	Reduction in SVs		0.0%	17.54%	80.22%	0.0%
d=4	SMO	# SVs	24	106	79	29
	Modified set	# SVs	23	106	29	28
	Reduction in SVs		4.12%	0.0%	63.29%	3.45%
d=5	SMO	# SVs	32	241	46	13
	Modified set	# SVs	31	213	32	11
	Reduction in SVs		3.13%	11.62%	30.44%	15.39%
d=6	SMO	# SVs	28	269	63	14
	Modified set	# SVs	28	165	41	13
	Reduction in SVs		0.0%	38.66%	34.92%	7.14%
d=7	SMO	# SVs	12	239	42	20
	Modified set	# SVs	11	119	38	20
	Reduction in SVs		8.33%	50.21%	9.52%	0.0%

Table 2 : Results using the polynomial kernel

RBF Kernel			Data Sets			
			Contraceptive Method Choice	Diabetes	Haberman	Wisconsin Breast Cancer
σ =1.5	SMO	# SVs	50	48	21	54
	Modified set	# SVs	50	47	21	53
	Reduction in SVs		0.0%	2.08%	0.0%	1.85%
σ =2.0	SMO	# SVs	50	50	23	56
	Modified set	# SVs	50	49	22	53
	Reduction in SVs		0.0%	2.0%	4.35%	5.36%
σ =2.5	SMO	# SVs	183	49	69	86
	Modified set	# SVs	176	48	46	79
	Reduction in SVs		3.83%	2.04%	33.3%	8.14%
σ =3.0	SMO	# SVs	73	134	55	107
	Modified set	# SVs	73	51	30	91
	Reduction in SVs		0.0%	61.94%	45.45%	14.95%
σ =3.5	SMO	# SVs	294	137	35	130
	Modified set	# SVs	207	44	24	114
	Reduction in SVs		29.59%	67.88%	31.43%	12.31%

Table 3 : Results using RBF kernels

3.2 Regression

For the regression case, the initial support vectors were generated using *SVMTorch* (Collobert and Bengio, 2001) on a data set of 4000 points also provided in Collobert and Bengio (2001). Equation (7) was then used to eliminate those support vectors that were identified as linearly dependent. The simplifications obtained are detailed in Table 4.

RBF Kernel	SVMTorch - # SVs	Modified set - #SVs	Reduction in SVs
$\sigma = 900$	597	412	30.99%
$\sigma = 915$	600	408	32.00%
$\sigma = 950$	599	394	34.22%
$\sigma = 1000$	613	384	37.36%
$\sigma = 1050$	625	369	40.96%

Table 4 : Results using RBF kernels for regression

4. Discussion

Tables 1-4 demonstrate that in most cases our procedure leads to a reduction in the number of support vectors and in some cases a substantial reduction. The amount of reduction achievable is both kernel and problem dependent and does not appear to be predictable *a priori*. However, the examples indicate that larger reductions tend to occur with lower degree polynomial kernels and with RBF kernels having larger σ values.

Note that the solutions we obtain are not generally unique. Linear dependence is a collective property of the support vectors and the choice of which support vectors to eliminate is not a unique one. This indicates that those support vectors that Vapnik terms *essential* (Vapnik, 1998) are the ones that are linearly independent before the implementation of our procedure.

We are currently developing an SVM training algorithm that will generate linearly independent support vectors only.

Acknowledgement

This work was supported by the Australian Research Council, Grant Number A49937172.

References

C. J. C. Burges. Simplified support vector decision rules. *Proceedings 13th International Conference on Machine Learning*, Bari, Italy, 1996, pp 71-77.

C. J. C. Burges and B. Schoelkopf. Improving speed and accuracy of support vector learning machines. *Advances in Neural Information Processing Systems*, **9**, MIT Press, 1997, 375-381.

R. Collobert and S. Bengio. SVMTorch: Support Vector Machines for Large-Scale Regression Problems", *Journal of Machine Learning Research,* 1:143-160, 2001.

B. Noble and J. W. Daniel. *Applied Linear Algebra*. 3rd Edition, Prentice-Hall, 1988.

J. Platt. Fast Training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods – Support Vector Learning,* pages 185-208, 1999, MIT Press.

N. A. Syed, H. Liu and K. K. Sung. Incremental Learning with Support Vector Machines. *Proceedings of Workshop on Support Vector Machines at International Joint Conference on Artificial Intelligence,* Stockholm, 1999.

V. N. Vapnik. *Statistical Learning Theory*. Wiley, New York, 1998.