ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 207-212

An Efficient Formulation of Sparsity Controlled Support Vector Regression

Pierre M. L. Drezet† and Robert F. Harrison

Department of Automatic Control & Systems Engineering,
The University of Sheffield, Mappin Street, Sheffield, S1 3JD, UK.

Abstract. Support Vector Regression (SVR) is a kernel based regression method capable of implementing a variety of regularisation techniques. Implementation of SVR usually follows a dual optimisation technique which includes Vapnik's ϵ-insensitive zone. The number of terms in the resulting SVR approximation function is dependent on the size of this zone, but improving sparsity by increasing the size of this zone adversely effects precision. We describe an efficient method of formulating SVR without an ϵ-insensitive zone, that selects a minimum support set for the terms of the approximator. Sparsity can then be traded for increased training error and/or decreased SV regularisation.

1. Introduction

Support vector machine (SVM) training requires constrained quadratic optimisation in as many variables as there are training vectors. Support vector regression (SVR), however, requires twice this number to accommodate Vapnik's ϵ-insensitive loss function [10]. These numbers can lead to substantial computational problems for large data sets, in both training and prediction phase.

Efforts have been made to reduce the computational requirements during training. These include techniques such as data chunking, sequential minimal optimisation, and the Kernel Adatron [7],[9],[6]. These methods are based on the *dual* formulation of the SVM, which for classification has the heuristically appealing property that the support vectors chosen are those that define the *optimal boundary* [10]. Support vectors form the support set of the prediction function, but their number is often far in excess of the minimum required for the solution's basis.

Computational requirements in the prediction mode depend on the number of support vectors selected from the training set. Situations in both SVM classification and SVR which lead to excessive numbers of support vectors are related to the amount of noise in the training data, and the amount of regularisation. This problem is acute in regression problems when the size

†The financial support of the EPSRC and Sheffield University is gratefully acknowledged

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 207-212

of the ϵ-insensitive zone is reduced to zero. In this case all training vectors are included in the support set. To circumvent this, some methods have been suggested for reducing the set as an intermediate stage between training and prediction [5],[2].

The dual formulation is normally chosen because it is computationally cheaper than solving the primal problem. It is also automatically derived in terms of Lagrange multipliers and dot products of training vectors only. This dot product formulation is predisposed to the use of kernel functions to extend the linear foundations of SVM to nonlinear problems. The dot product substitution can also be made in the primal formulation [4], [8], leading to a solution which is not reliant on Kuhn-Tucker constraint conditions [10], to select the support set.

We present an efficient method for SVR without an ϵ-insensitive zone and incorporating independent control of sparsity. The resulting formulation yields a flexible hybrid of *basis pursuit* [3] and Vapnik's SVR. In section 2.1. SVR and sparsity controlled SVR are summarised and in section 2.2. this is reformulated to remove general linear constraints. Section 3. demonstrates computational requirements by example.

2. Support Vector Regression Optimisation

SVR is related to SV classification by simply replacing the loss term associated with training vectors not within the *optimal boundaries* with one for vectors outside an error insensitive zone, of width, ϵ. The objective of SVR is to minimise the following cost function.

$$L(\vec{w}, b) \quad = \quad ||\vec{w}||^2 + CF_\epsilon \left(y_i - (\vec{w}.x_i + b)\right) \tag{1}$$

The function, F_ϵ, is conventionally the ϵ-insensitive loss function, $F_\epsilon(e) = |e| - \epsilon, \quad for \quad |e| > \epsilon$, and 0 otherwise. To accommodate this loss function, equation 1 can be formulated by defining errors as slack variables of linear constraints. Here, least squares error cost will be used.

$$\begin{aligned} & L(\vec{w}, b) && = && ||\vec{w}||^2 + C(||\vec{\xi}||^2 + ||\vec{\xi_i^*}||^2) \\ \text{subject to} \quad & \vec{w}.\vec{x}_i + b && \geq && y_i - \epsilon - \xi_i \\ & \vec{w}.\vec{x}_i + b && \leq && y_i + \epsilon + \xi_i^* \\ & \xi_i, \xi_i^* && \geq && 0, \quad \text{for} \quad i = 1, \ldots, l. \end{aligned} \tag{2}$$

By introducing Lagrange multipliers and substituting derivatives, the dual formulation in terms of Lagrange multipliers and training vectors is given by equation 3. The dual optimisation of SVR has $2l$ variables to be maximised in the quadratic objective, $2l$ box constraints and one linear equality constraint. The dot products,$< x_i, x_j >$, may be substituted for kernel functions, $K(x_i, x_j)$[10].

$$L(\vec{\alpha}, \vec{\alpha}^*) \quad = \quad -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) < \vec{x}_i, \vec{x}_j > + \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)y_i$$

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 207-212

$$- \quad \epsilon\sum_{i=1}^{l}\alpha_i + \alpha_i^* - \sum_{i=1}^{l}\frac{\alpha_i^2 - \alpha_i^{*2}}{C} \tag{3}$$

$$\text{subject to} \quad \sum_{i=1}^{l}\alpha_i - \alpha_i^* = 0$$
$$\alpha_i, \alpha_i^* \geq 0 \quad \text{for} \quad i = 1, \ldots, l$$

The dual optimisation sets the non-negative Lagrange multipliers, α, α^*, to zero if their corresponding constraints are not a limiting factor in the optimisation. Sparsity is thus governed by the proportion of training vectors within the ϵ-insensitive zone.

2.1. Sparsity Controlled SVR (SCSVR)

The derivation of equation 3 shows that the weights, $\vec{w}$ can be represented in terms of Lagrange multipliers and dot products of input vectors only. By incorporating this weight substitution directly into the primal form, without the Lagrangian formulation, the following cost function can be constructed which is suitable for kernel function substitution [4], [8]. The formulation includes an extra objective term $D\sum_i |\alpha_i + \alpha_i^*|$, which encourages sparsity in the solution by selecting one of the multiple solutions with fewest multipliers, α, α^*. The error loss function values, $n \in \{1, 2\}$, are suitable for quadratic programming.

$$L(\vec{\alpha}, \vec{\alpha}^*, b) = \sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) < \vec{x}_i, \vec{x}_j > + C\sum_{i=1}^{l}\xi_i^n$$
$$+ \quad D\sum_{i=1}^{l}\alpha_i + \alpha_i^*$$

$$\text{subject to} \quad \sum_{j=1}^{L}(\alpha_j - \alpha_j^*) < \vec{x}_j, \vec{x}_i > + b \geq y_i - \epsilon - \xi_i$$
$$\sum_{j=1}^{L}(\alpha_j - \alpha_j^*) < \vec{x}_j, \vec{x}_i > + b \leq y_i + \epsilon + \xi_i$$
$$\alpha_i, \alpha_i^*, \xi_i \geq 0 \quad \text{for} \quad i = 1, \ldots, l$$

2.2. SCSVR without ϵ-insensitivity

If ϵ-insensitivity is not required and least squares error loss is chosen, the above equation can be formulated without general linear constraints, except for simple bounds, $\alpha, \alpha^* \geq 0$. The box constraints (equation 5) on α, α^* are required for least modulus (l_1) minimisation of α, α^*. If an l_2 norm is minimised, the optimisation does not require any constraints, but sparsity in the solution is lost. The cost function for minimising the l_1 norm of α, α^* is,

$$L(\vec{\alpha}, \alpha^*, b) = D\sum_{i=1}^{l}|\alpha_i + \alpha_i^*| + \sum_{i,j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) < \vec{x}_i, \vec{x}_j >$$
$$+ \quad C\sum_{i=1}^{l}\left(y_i - \left[\sum_{i=1}^{l}(\alpha_i - \alpha_i^*) < (\vec{x}_i, \vec{x}_i > + b\right]\right)^2 \tag{4}$$

$$\text{subject to} \quad \alpha_i, \alpha_i^* \geq 0 \quad \text{for} \quad i = 1, \ldots, l \tag{5}$$

This formulation can be solved using simple quadratic programming techniques for bounded variables. Alternatively, an iterative method of solving this problem, such as that of [1], [6] can be applied.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 207-212

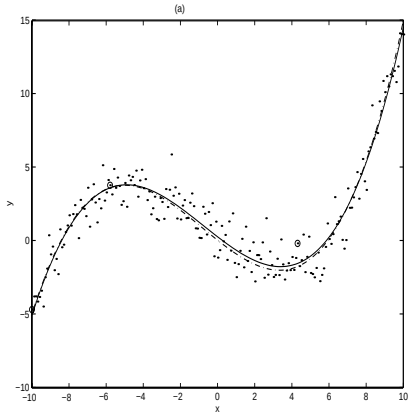

Figure 1: SCSVR with Polynomial Kernel, $(1 + \vec{x}.\vec{y})^3$, for Polynomial function $f(x) = 0.02x^3 + 0.05x^2 - x + e$, e =Gaussian white noise ($\sigma = 1$). 0—support vectors, solid line—estimation, dashed line—true value. Training parameters are as for table 1.

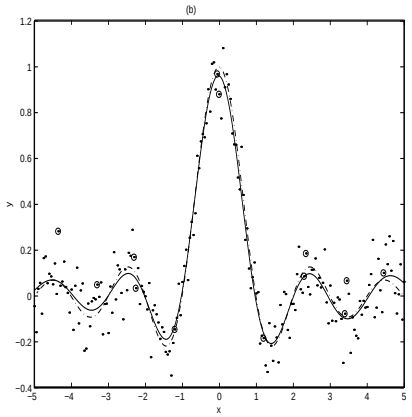

Figure 2: SCSVR with Gaussian kernel, $\exp(|\vec{x} - \vec{y}|^2)$, for sinc function $f(x) = \mathrm{sinc}(x) + e$, e =Gaussian white noise ($\sigma = 0.1$). 0—support vectors, solid line—estimation, dashed line—true value. Training parameters are as for table 1.

3. Performance

Comparing the three formulations of SVR, dual, sparsity controlled (SCSVR$_\epsilon$), sparsity controlled SVR without ϵ (SCSVR), the training requirements vary considerably (table 1). SCSVR can be seen to have at least comparable training requirements to the dual method. SCSVR$_\epsilon$ has the largest, owing to the large number of constraint calculations involved. The emphasis in comparing computational requirements for training is between SCSVR$_\epsilon$ and SCSVR.

The sparsity controlled methods both produce approximation functions which contain less support vectors than the dual methods. Computational requirements during prediction are directly proportional to the number of support vectors selected. In the polynomial case the number is limited by the dimension of the kernel feature space. The Boston housing benchmark was used to further illustrate how sparsity control can decrease training and prediction calculations while maintaining predictive accuracy. All three formulations were implemented using the same NAG quadratic optimisation function in MATLAB, rather than the standard optimisation toolbox.

4. Conclusions

SCSVR attempts to minimise the number of support vectors selected during training without necessarily effecting precision. The combination of sparse ap-

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 207-212

Polynomial function, Polynomial Kernel, $(1 + \vec{x}.\vec{y})^3$,

Method	Training Mflops	Support Vectors	Error
$dual_{\epsilon=0}$	17	101	0.81
$dual_{\epsilon=1}$	5	26	1.1
$SCSVR_{\epsilon=0}$	50	4	0.81
SCSVR	1.5	3	0.81

sinc function, Gaussian Kernel, $exp(|\vec{x} - \vec{y}|^2)$,

Method	Training Mflops	Support Vectors	error
$dual_{\epsilon=0.1}$	39	60	0.35
SCSVR	25	13	0.341

Table 1: Training requirements for SVR formulations for polynomial and sinc functions. ($C = 10$, $\epsilon = 0$, $D = 10^{-3}, 1$ for polynomial and sinc examples respectively). Training calculations are in flops$\times 10^6$. The polynomial data set comprises of 101 vectors, and the sinc, 201. The error value is the sum squared difference to the noise free function.

Boston Housing Benchmark, Gaussian Kernel, $exp(0.2|\vec{x} - \vec{y}|^2)$

Method	Training Mflops	Support Vectors	test error (MSE)
$dual_{\epsilon=0.1}$	246	88	14.0
SCSVR, D=1	195	29	9.1
SCSVR, D=20	66	13	11.5

Table 2: Comparison of # support vectors and prediction accuracy for multi-dimensional regression data ($C = 1000$P. Training calculations are in flops $\times 10^6$. The training data set comprises of 406 vectors. The error value is the mean squared difference to 100 test examples.

proximation and SVR allows the unappealing properties of ϵ-insensitivity to be removed from SVR. From the sparse approximation point of view, the hybrid of SVR and basis pursuit provides extra control of regularisation. The formulation of SCSVR without large numbers of constraints, can also lessen the computational requirements during training below those of the dual formulation. Predictive accuracy of SCSVR compares well with ϵ-insensitive SVR in these examples, this is probably down to the fact that SCSVR improves the empirical risk because the loss function utilises all of the training data. Further work is required to judge SCSVR performance with other types of data and noise, and also to investigate how well it can be optimised computationally.

ESANN'1999 proceedings - European Symposium on Artificial Neural Networks
Bruges (Belgium), 21-23 April 1999, D-Facto public., ISBN 2-600049-9-X, pp. 207-212

References

[1] J. K. Anlauf and M. Biehl. The adatron: an adaptive perceptron algorithm. *Europhysics Letters*, 7(10):687–692, 1989.

[2] J. C. Burgess. Simplified support vector decision rules. In L. Saitta, editor, *Proceedings, 13th International Conference on Machine Learning*, pages 71–77. Morgan Kauffman, 1996.

[3] S. Chen. *Basis Pursuit*. PhD thesis, Department of Statistics, Stanford University, November 1995.

[4] P. Drezet and R. F. Harrison. Directly optimised support vector machines for classification and regression. Research Report 716, University of Sheffield, Dpt. Automatic Control & Systems Engineering, 1998.

[5] T-T. Friess and R.F. Harrison. Linear programming support vector machines for pattern classification and regression estimation; and the sr algorithm: Improving speed and tightness of VC bounds in SV algorithms. Research Report 706, University of Sheffield, Dpt. Automatic Control & Systems Engineering, 1998.

[6] T-T. Friess and R.F. Harrison. Support vector neural networks. Research Report 725, University of Sheffield, Dpt. Automatic Control & Systems Engineering, 1998.

[7] E. Osuna, R. Freund, and F. Girosi. An improved training algorithm for support vector machines. In *IEEE Workshop on Neural Networks and Signal Processing*, pages 276–278, Amelia Island, FL, September 1997.

[8] Edgar E. Osuna and Federico Girosi. Reducing the run-time complexity of support vector machines. In *Advances in Kernel Methods: Support Vector Learning*, chapter 16. MIT Press, Cambridge, MA, 1998.

[9] J. Platt. Sequential minimal optimisation: A fast training algorithm for support vector machines. In *Advances in Kernel Methods: Support Vector Learning*, chapter 12. MIT Press, Cambridge, MA, 1998.

[10] V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, 1995.