SPARSE APPROXIMATION USING LEAST SQUARES SUPPORT VECTOR MACHINES

J.A.K. Suykens, L. Lukas, J. Vandewalle

K.U. Leuven, Dept. of Electrical Eng., ESAT-SISTA
Kardinaal Mercierlaan 94, B-3001 Leuven (Heverlee), Belgium
Email: johan.suykens@esat.kuleuven.ac.be

ABSTRACT

In least squares support vector machines (LS-SVM's) for function estimation Vapnik's ϵ-insensitive loss function has been replaced by a cost function which corresponds to a form of ridge regression. In this way nonlinear function estimation is done by solving a linear set of equations instead of solving a quadratic programming problem. The LS-SVM formulation also involves less tuning parameters. However, a drawback is that sparseness is lost in the LS-SVM case. In this paper we investigate imposing sparseness by pruning support values from the sorted support value spectrum which results from the solution to the linear system.

Keywords. Function estimation, support vector machines, ridge regression, sparse approximation, pruning, radial basis functions.

1. INTRODUCTION

Support vector machines (SVM's) which have been introduced by Vapnik [14] have been successful for solving classification and function estimation problems [9]. It allows the use of several kernel functions including linear, polynomial, splines, radial basis functions and multilayer perceptrons with one hidden layer. In the support vector method of function estimation [14, 15, 16] the problem is formulated in such a way that the resulting model follows from solving a convex optimization problem, more specifically a quadratic programming problem. This is obtained by employing the so-called Vapnik's ϵ-insensitive loss function, formulating the approximation problem as an inequality constrained optimization problem [2] and exploiting the Mercer condition in order to relate the nonlinear feature space mapping to the chosen kernel function. Moreover, the model complexity needed to approximate the underlying function given the training data also follows from the solution to this quadratic programming problem and is given by the number of non-zero support values. Hence, when estimating a support vector machine with radial basis function (RBF) kernel the number of hidden units follows from solving the QP problem. This is an additional advantage of the SVM framework in comparison with regularization theory of RBF networks [7, 10]. The remaining unknown parameters of the SVM and/or tuning parameters in the algorithm are determined using generalization bounds, cross-validation criteria or others.

In this paper we further investigate least squares support vector machines (LS-SVM's) for function estimation, which is related to a ridge regression type of SVM [8] (but with inclusion of an additional bias term in the model). LS-SVM's were also proposed in the context of classification in [11, 12, 13]. Vapnik's SVM formulation is modified in the sense of ridge regression and taking equality instead of inequality constraints in the problem formulation. As a result one solves a linear system instead of a QP problem. An iterative method for solving large scale problems has been discussed in [12]. Due to this choice of a 2-norm one looses sparseness in the resulting LS-SVM.

The aim of this paper is to show that despite the least squares norm (with parametric regularization) one can impose sparseness to the LS-SVM based upon the sorted spectrum of support values. These support values are all non-zero, but indicate the relative importance of how each of the data points contributes to the model. This support value spectrum is used then to achieve sparseness by gradually imposing zero support values (i.e. removing data points from the training set) to the less significant data points. This procedure is similar to pruning techniques in neural networks [6, 5], but pruning is done here based upon the support value spectrum (which is the solution vector to the LS-SVM linear set of equations) instead of on the basis of a Hessian matrix. A simple example on the estimation of a sinc function shows how the LS-SVM sparse approximation procedure retains the same support vectors as the original SVM approach. The link between sparse approximation and SVM's has been established in [3]. In this paper LS-SVM models are re-estimated when pruning the support value spectrum. Other approaches for achieving sparse approximation take into account an additional term in the cost function [3, 9].

This paper is organized as follows. In Section 2 we present LS-SVM's for function estimation. In Section 3 we discuss the sparse approximation procedure for LS-SVM's. In Section 4 an illustrative example is given.

2. LEAST SQUARES SUPPORT VECTOR MACHINES

Consider a given training set of N data points $\{x_k, y_k\}_{k=1}^N$ with input data $x_k \in \mathbb{R}^n$ and output $y_k \in \mathbb{R}$. In feature space SVM models take the form:

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

where the nonlinear mapping $\varphi(\cdot)$ maps the input data into a higher dimensional feature space. Note that the dimension of w is not specified (it can be infinite dimensional). In least squares support vector machines for function estimation the following optimization problem is formulated

$$\min_{w,e} \mathcal{J}(w,e) = \frac{1}{2} w^T w + \gamma \frac{1}{2} \sum_{k=1}^N e_k^2 \tag{2}$$

subject to the equality constraints

$$y_k = w^T \varphi(x_k) + b + e_k, \quad k = 1, ..., N$$

This corresponds to a form of ridge regression [4]. The Lagrangian is given by

$$\mathcal{L}(w,b,e;\alpha) = \mathcal{J}(w,e) - \sum_{k=1}^N \alpha_k \left\{ w^T \varphi(x_k) + b + e_k - y_k \right\} \tag{3}$$

with Lagrange multipliers α_k. The conditions for optimality are

$$\left\{ \begin{array}{lll} \frac{\partial \mathcal{L}}{\partial w} = 0 & \rightarrow & w = \sum_{k=1}^N \alpha_k \varphi(x_k) \\ \frac{\partial \mathcal{L}}{\partial b} = 0 & \rightarrow & \sum_{k=1}^N \alpha_k = 0 \\ \frac{\partial \mathcal{L}}{\partial e_k} = 0 & \rightarrow & \alpha_k = \gamma e_k \\ \frac{\partial \mathcal{L}}{\partial \alpha_k} = 0 & \rightarrow & w^T \varphi(x_k) + b + e_k - y_k = 0 \end{array} \right. \tag{4}$$

for $k = 1..N$. After elimination of e_k, w the solution is given by the following set of linear equations

$$\left[\begin{array}{c|c} 0 & \vec{1}^T \\ \hline \vec{1} & \Omega + \gamma^{-1} I \end{array} \right] \left[\begin{array}{c} b \\ \hline \alpha \end{array} \right] = \left[\begin{array}{c} 0 \\ \hline y \end{array} \right] \tag{5}$$

where $y = [y_1; ...; y_N]$, $\vec{1} = [1; ...; 1]$, $\alpha = [\alpha_1; ...; \alpha_N]$ and the Mercer condition

$$\begin{array}{rcll} \Omega_{kl} & = & \varphi(x_k)^T \varphi(x_l), & k,l = 1, ..., N \\ & = & \Psi(x_k, x_l) & \end{array} \tag{6}$$

has been applied. This finally results into the following LS-SVM model for function estimation

$$y(x) = \sum_{k=1}^N \alpha_k \Psi(x, x_k) + b \tag{7}$$

where α, b are the solution to (5). For the choice of the kernel function $\Psi(\cdot,\cdot)$ one has several possibilities: $\Psi(x, x_k) = x_k^T x$ (linear SVM); $\Psi(x, x_k) = (x_k^T x + 1)^d$ (polynomial SVM of degree d); $\Psi(x, x_k) = \exp\{-\|x - x_k\|_2^2/\sigma^2\}$ (RBF SVM); $\Psi(x, x_k) = \tanh(\kappa x_k^T x + \theta)$ (MLP SVM). In the case that one uses RBF kernels the remaining unknowns (that do not follow from (5)) are σ, γ. Note that in the standard SVM case one needs to determine an additional ϵ accuracy parameter which is related to the ϵ-insensitive loss function.

3. SPARSE APPROXIMATION USING LS-SVM'S

A major difference between the LS-SVM version and Vapnik's version is that sparseness is lost in the LS case. This immediately follows from the choice of the 2-norm and is also revealed by the fact that the support values are proportional to the errors at the data points, namely $\alpha_k = \gamma e_k$. However, by plotting the spectrum of the sorted $|\alpha_k|$ values one can evaluate which data are most significant for contribution to the LS-SVM model. Sparseness is imposed then by gradually omitting the least important data from the training set and re-estimating the LS-SVM (Fig.1):

1. Train LS-SVM based on N points.

2. Remove a small amount of points (e.g. 5% of the set) with smallest values in the $|\alpha_k|$ spectrum.

3. Re-train the LS-SVM based on the reduced training set.

4. Go to 2, unless the performance degrades.

This procedure corresponds in fact to pruning of the LS-SVM. An advantage in comparison with pruning for classical neural network techniques [1, 6, 5] is that now the pruning doesn't involve a computation of a Hessian matrix but is immediately done based upon the physical meaning of the solution vector α.

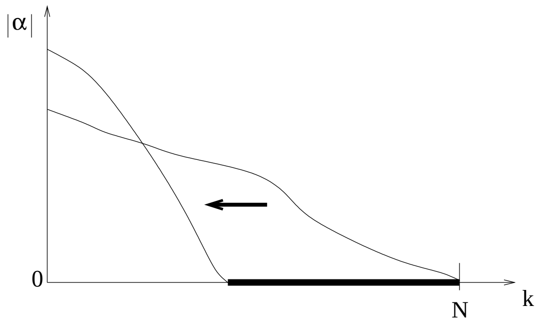

Fig.1: *Pruning of the LS-SVM spectrum.*

4. EXAMPLE

We illustrate the sparse approximation procedure for LS-SVM's on the example of a sinc function for the noiseless (Fig.2) and the noisy case (Gaussian noise with standard dev. 0.5) (Fig.3). Given are $N = 500$ training data points. A RBF kernel has been chosen. The values of $\sigma = 1, \gamma = 10$ were chosen for the noiseless case and $\sigma = 1, \gamma = 1$ in the noisy case. These values were determined from an independent test set. In both cases the number of hidden units can be decreased from 500 to less than 100 without loss of performance. The support values are reduced by 5% in each step with re-estimation of the LS-SVM.

5. CONCLUSIONS

We have proposed a sparse approximation procedure for least-squares support vector machines which prunes the network by gradually removing points from the training set based upon the support value spectrum.

Acknowledgements

This research work was carried out at the ESAT laboratory and the Interdisciplinary Center of Neural Networks ICNN of the Katholieke Universiteit Leuven, in the framework of the FWO project G.0262.97 *Learning and Optimization: an Interdisciplinary Approach*, the Belgian Programme on Interuniversity Poles of Attraction, initiated by the Belgian State, Prime Minister's Office for Science, Technology and Culture (IUAP P4-02 & IUAP P4-24), the Concerted Action Project MIPS (*Modelbased Information Processing Systems*) of the Flemish Community. Johan Suykens is a postdoctoral researcher with the National Fund for Scientific Research FWO - Flanders.

6. REFERENCES

[1] Bishop C.M., *Neural networks for pattern recognition*, Oxford University Press, 1995.

[2] Fletcher R., *Practical methods of optimization*, Chichester and New York: John Wiley and Sons, 1987.

[3] Girosi F., "An equivalence between sparse approximation and support vector machines," *Neural Computation*, 10(6), 1455-1480, 1998.

[4] Golub G.H., Van Loan C.F., *Matrix Computations*, Baltimore MD: Johns Hopkins University Press, 1989.

[5] Hassibi B., Stork D.G., "Second order derivatives for network pruning: optimal brain surgeon," In Hanson, Cowan, Giles (Eds.) *Advances in Neural Information Processing Systems*, Vol.5, pp.164-171, San Mateo, CA: Morgan Kaufmann, 1993.

[6] Le Cun Y., Denker J.S., Solla S.A., "Optimal brain damage," In Touretzky (Ed.) *Advances in Neural Information Processing Systems*, Vol.2, pp.598-605, San Mateo, CA: Morgan Kaufmann, 1990.

[7] Poggio T., Girosi F., "Networks for approximation and learning," *Proceedings of the IEEE*, Vol.78, No.9, pp.1481-1497, 1990.

[8] Saunders C., Gammerman A., Vovk V., "Ridge Regression Learning Algorithm in Dual Variables," *Proc. of the 15th Int. Conf. on Machine Learning ICML-98*, Madison-Wisconsin, 1998.

[9] Schölkopf B., Burges C., Smola A. (Eds.), *Advances in Kernel Methods - Support Vector Learning*, MIT Press, 1998.

[10] Smola A., Schölkopf B., Müller K.-R., "The connection between regularization operators and support vector kernels," *Neural Networks*, 11, 637-649, 1998.

[11] Suykens J.A.K., Vandewalle J., "Least squares support vector machine classifiers," *Neural Processing Letters*, Vol.9, No.3, pp.293-300, June 1999.

[12] Suykens J.A.K., Lukas L., Van Dooren P., De Moor B., Vandewalle J., "Least squares support vector machine classifiers: a large scale algorithm," *European Conference on Circuit Theory and Design, ECCTD'99*, pp.839-842, Stresa Italy, August 1999.

[13] Suykens J.A.K., Vandewalle J., "Multiclass Least Squares Support Vector Machines," *International Joint Conference on Neural Networks IJCNN'99*, Washington DC, July 1999.

[14] Vapnik V., "The nature of statistical learning theory," Springer-Verlag, New-York, 1995.

[15] Vapnik V., "Statistical learning theory," John Wiley, New-York, 1998.

[16] Vapnik V., "The support vector method of function estimation," In *Nonlinear Modeling: Advanced black-box techniques*, Suykens J.A.K., Vandewalle J. (Eds.), Kluwer Academic Publishers, Boston, pp.55-85, 1998.

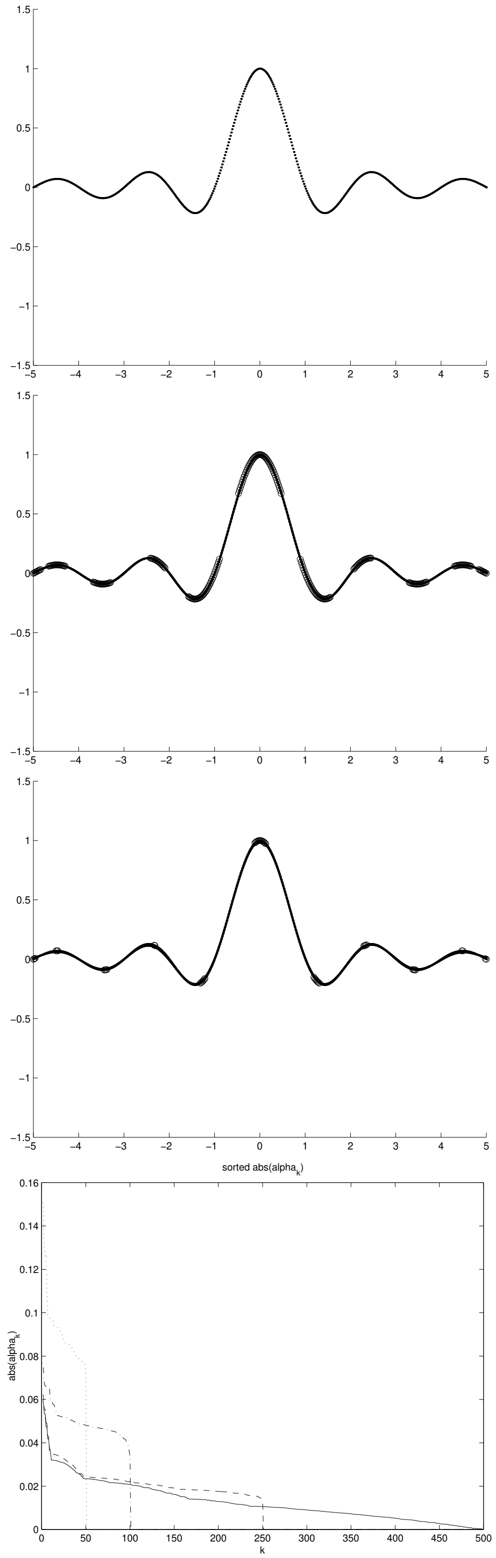

Fig.2: *Sparse approximation by LS-SVM's for sinc function (noiseless case): (a) $N =$ 500 SV; (b) 250 SV; (c) 50 SV; (d) snapshots of shifted SV spectrum while pruning.*

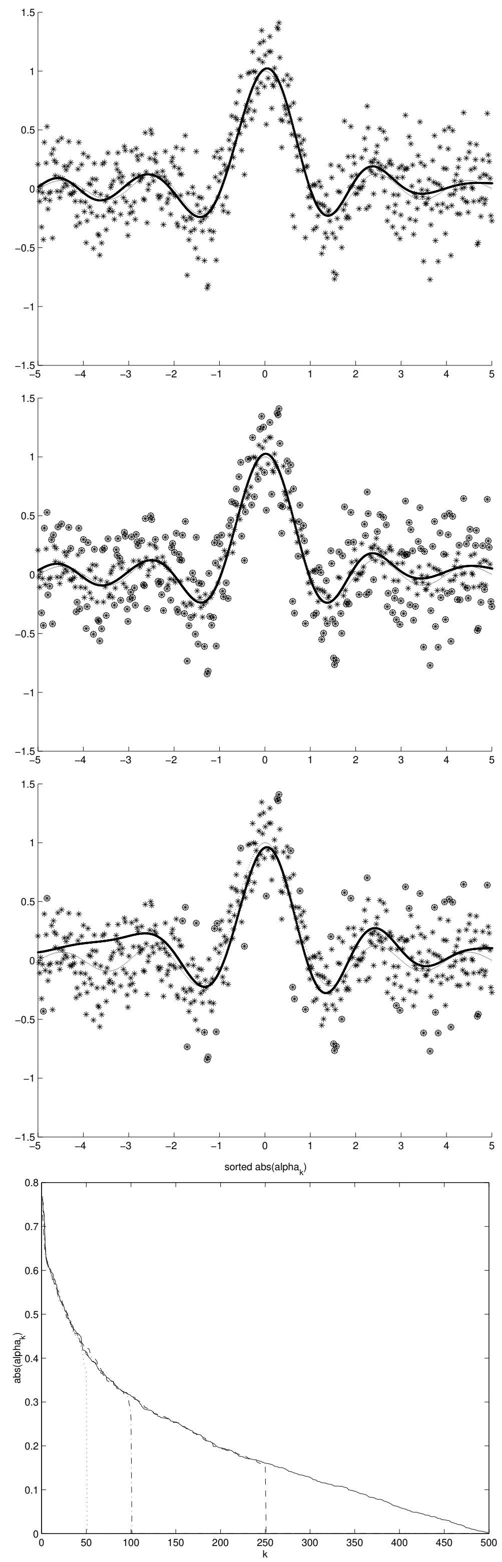

Fig.3: *Sparse approximation by LS-SVM's for sinc function (noisy case): (a) $N = 500$ SV; (b) 250 SV; (c) 50 SV; (d) snapshots of shifted SV spectrum while pruning.*