Multi resolution least squares SVM solver

T. Schouten, J.A.K. Suykens, B. De Moor
Katholieke Universiteit Leuven
Department Electrotechniek, ESAT-SISTA
Kardinaal Mercierlaan 94, B-3001 Leuven, Belgium

Abstract— **We discuss the use of multi resolution analysis (MRA) for fast approximate solution of large linear sets of equations arising in least squares support vector machine (LS-SVM) problems. When LS-SVMs are used on a low dimensional input space, the matrix of the linear set exhibits structure that leads to a sparse approximation in the wavelet domain. The amount of structure decreases with increasing dimensionality. We will illustrate this principle by means of a small example.**

Keywords— **Multi resolution analysis, least squares support vector machines, direct solution of large structured linear problems.**

I. Introduction

The research of this paper is situated in the field of nonlinear function estimation using support vector machines (SVMs) [17][18]. Training of a SVM involves solving a quadratic programming problem. It has been proposed to use least squares support vector machines (LS-SVMs) as a variant of standard SVMs [9][11]. In this method one replaces the ϵ–insensitive loss function proposed by Vapnik which leads to sparse SVM models by a quadratic loss function. Changing the original inequality constraints to equality constraints reduces the problem to the solution of a linear set of equations. Compared to standard SVMs, sparsity of the support value spectrum is lost. It is shown in [12][13] that sparsity can be imposed based on pruning the least significant support vectors. Large scale problems have been treated in [10]. Excellent benchmark results for LS-SVMs are obtained in [16].

In this paper we present a motivation for a method exploiting smoothness of the matrix arising in large LS-SVM problems when the data points are taken from a low dimensional grid, as is usually the case in signal processing. The smooth structure of the linear set is exploited by transforming it to the wavelet domain where it can be approximated by a sparse matrix. This wavelet sparseness is not to be confused with the (inherent) sparseness of a SVM model or the (imposed) sparseness of a LS-SVM model.

We base our work on a method presented in [3]. The method uses smoothness (in a local polynomial sense) of the matrix of a set of equations to construct a direct solver that works in $O(N)$ time. The trick is to replace LU factorization, a method that is $O(N^3)$ for dense matrices, by a banded version in the wavelet domain. The wavelet decomposition used is the non–standard form (NS-form) of the wavelet transform, also used in image compression. The NS-form is particularly efficient for smooth diagonally dominant matrices such that their transform can be approximated by a banded matrix.

The core operations of the method are sparse LU factorization, forward/backward substitution, sparse matrix/matrix and matrix/vector operations and the application of 2D analysis and 1D analysis and synthesis two–scale wavelet transforms. In this paper we will concentrate on the latter.

This paper is organized as follows. Section II is a review of the LS-SVM model and the linear set to be solved. Section III discusses the issue of structure in 1D LS-SVM problems. Section IV describes the solution of the 1D uniform sampled LS-SVM linear set in the wavelet domain. It deals with orthogonal decompositions as used in [3]. Section V will comment on the extension possible by using the lifting scheme. The first part discusses the extension to biorthogonal wavelets on an interval. The second part will comment on the extension to irregular grids.

II. Least Squares SVM's

The LS-SVM model for function estimation has the following representation in feature space

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

where $x \in \mathbb{R}^n, y \in \mathbb{R}$. $\varphi(\cdot)$ is a nonlinear map from the input space to a higher dimensional feature space, which can be infinite dimensional.

Given a training set $\{x_k, y_k\}_{k=1}^N$ one defines now the optimization problem

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

subject to equality constraints

$$y_k = w^T \varphi(x_k) + b + e_k, \quad k = 1, ..., N. \tag{3}$$

The cost function with squared error and regularization corresponds to a form of ridge regression [4]. A similar problem has been studied in [5] but without considering a bias term.

One constructs then the Lagrangian

$$\mathcal{L}(w,b,e;\alpha) = \mathcal{J}(w,e) - \sum_{k=1}^{N} \alpha_k \{w^T \varphi(x_k) + b + e_k - y_k\} \tag{4}$$

Tom Schouten is a Research Assistant with the K.U.Leuven. Email: tom.schouten@esat.kuleuven.ac.be, Johan Suykens is a postdoctoral researcher with the F.W.O. (Fund for Scientific Research-Flanders). Email: johan.suykens@esat.kuleuven.ac.be Bart De Moor is a Research Associate at the F.W.O. Flanders. Email: bart.demoor@esat.kuleuven.ac.be, This research work was carried out at the ESAT laboratory of the Katholieke Universiteit Leuven. Tel: +32-(0)16-321709, Fax: +32-(0)16-321970, This work is supported by several institutions: the Flemish Government (Research Council K.U.Leuven : Concerted Research Action GOA-MIPS and Mefisto-666), the FWO (projects G.0240.99, G.0256.97), and Research Communities(ICCoS and ANMMM), I-WT projects (EUREKA 1562-SINOPSYS, EUREKA 2063-IMPACT, STWW), the Belgian State, Prime Minister's Office - OSTC - (IUAP P4-02 and IUAP P4-24, Sustainable Mobility Programme - Project MD/01/24 1997-2000) and the European Commission (TMR Networks: ALAPEDES and System Identification, Brite/Euram Thematic Network : NICONET). The scientific responsibility is assumed by its authors.

00-47

Schouten T., Suykens J.A.K., De Moor B., ``Multi resolution least squares SVM solver", in *Proc. of the 43rd IEEE Midwest Symposium on Circuits and Systems (MWSCAS2000)*, Lansing, Michigan, Aug. 2000, cdrom p., Lirias number: 178785.

where α_k are Lagrange multipliers.

From the conditions for optimality $\frac{\partial \mathcal{L}}{\partial w} = 0$, $\frac{\partial \mathcal{L}}{\partial b} = 0$, $\frac{\partial \mathcal{L}}{\partial e_k} = 0$ and $\frac{\partial \mathcal{L}}{\partial \alpha_k} = 0$ we find the solution

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

with $y = [y_1; ...; y_N]$, $\vec{1} = [1; ...; 1]$, $\alpha = [\alpha_1; ...; \alpha_N]$. Mercer's condition is applied as

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

with kernel function K, which can be chosen as a linear, polynomial, spline, RBF or MLP. The resulting LS-SVM model for function estimation becomes

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

where α_k, b are the solution to the linear system (5). In the sequel we focus on RBF kernels.

III. Low dimensional input LS-SVM problems

Due to their structure, an obvious advantage of SVMs in general is to overcome high dimensionality without additional cost. The amount of memory required is quadratic in the number of training points and seemingly independent of the input dimension of the problem. Surely this 'dimensional cost' has to manifest itself somewhere in the complexity of the problem, although not at first sight.

If we take a closer look at the resulting set of equations, we see a lot of structure arising in problems with a low input dimension. This structure is present in the form of a 'smooth' set matrix. In the extreme case of 1D regular sampling, orthogonal wavelets can be used to transform the set such that the transformed set can be approximated by a sparse set.

This sparseness is not to be confused with the notion of sparse LS-SVM modeling: it is merely a trick to solve a linear set in another domain by using a non-parametric transform (wavelets). We might even say that the LS-SVM sparseness is what we want to obtain from the work we put into the method and the wavelet sparseness is a 'complexity refund' we get by applying a non-parametric transform on the set of equations arising from the LS-SVM problem formulation. It could be said it pays back the $O(N^2)$ data-explosion in the LS-SVM formulation. This 'refund' only works in low dimensional case and is a (positive) manifestation of the fenomenon usually referred to as 'the curse of dimensionality'.

Now we will adjust the formulation of (5) such that the wavelet transform can be used more efficiently. In its present form it contains a row and a column of ones that will introduce a lot of significant detail coefficients in the wavelet domain and thus decrease compression. For ease of notation we define $Z = \Omega + \gamma^{-1} I$. As presented in [10], we can transform (5) to the following form

$$\left[\begin{array}{c|c} \vec{1}^T Z^{-1} \vec{1} & 0 \\ \hline 0 & Z \end{array}\right] \left[\begin{array}{c} b \\ \hline \alpha + b Z^{-1} \end{array}\right] = \left[\begin{array}{c} \vec{1}^T Z^{-1} y \\ \hline y \end{array}\right] \tag{8}$$

The algorithm to obtain α and b is as follows.

1. $Z\eta = \vec{1}$
2. $Z\nu = y$
3. $s = \vec{1}^T Z^{-1} \vec{1} = \vec{1}^T \eta$
4. $b = \eta^T y / s$
5. $\alpha = \nu - \eta b$

The core operation of this algorithm is the solution of two linear sets defined by matrix $Z \in \mathbb{R}^{N \times N}$ and the right hand sides y and $\vec{1}$. The goal of this paper is to find a way to exploit the smoothness (in a polynomial sense) of the matrix Z, which is a sum of a smooth kernel matrix Ω and a scaled unit matrix $\gamma^{-1} I$, by solving these sets using the NS-form direct solver described in [3]. The solution of this subproblem is the subject of the following sections.

The advantage of converting (5) to (8) is twofold. The bulk of the work is concentrated in solving a set that is symmetric and positive definite, and does not have a row and a column of ones like the original set. The former will allow for the use of cholesky factorization to replace LU factorisation in [3] and will halve the memory requirements. The latter will increase the smoothness of the matrix to be transformed to the wavelet domain. In [10] this was applied for large scale problems, in order to make iterative methods such as the conjugate gradient method applicable to (8).

IV. Orthogonal wavelets

The most structure can be found when the input data to the LS-SVM problem are taken from a regular sampled grid. In this case we can proceed to transform the problem using two kinds of wavelet transforms. Orthogonal wavelets as proposed in [2] and used in the direct solver presented in [3] and biorthogonal wavelets designed and implemented using the lifting scheme [14] [15], which have several advantages over orthogonal wavelets.

We take the solution of the set $Z\nu = y$ as an example. The solution of $Z\eta = \vec{1}$ is analogous and can use the same NS-form triangular factors. The outline of the algorithm is as follows, for details see [3]

1. Compute Cholesky decomposition of the NS-form of Z.
2. Compute the NS-form of the right hand side y.
3. Perform a foreward and backward substitution.
4. Compute ν using the inverse wavelet transform.

By using bounds for off-diagonal decay of the kernel function and its M-th partial derivatives, with M the number of vanishing moments of the wavelet, we can perform the usually expensive computations, like the two–scale wavelet transform and the Cholesky factorization, on banded matrices, which is the reason why we can get $O(N)$ complexity.

We will now concentrate on one of the basic steps: the calculation of the 2D two–scale wavelet transform. This is an algorithmic building block that will be applied recursively in the construction of the NS-form cholesky decomposition of Z. We

t	n_e	n_e/N	error
1e-02	1448	5.66	9.83e-03
1e-03	2101	8.21	1.31e-03
1e-04	2795	10.9	1.42e-04
1e-05	3245	12.7	9.21e-06
1e-06	3580	14.0	1.62e-06
1e-07	4042	15.8	1.31e-07
1e-08	4338	16.9	5.20e-09
1e-09	4522	17.7	1.41e-09
1e-10	5825	22.8	2.31e-10
1e-11	7496	29.3	9.08e-12

TABLE I

EFFECT OF THRESHOLDING ON ERROR AND SPARSENESS OF NF-FORM WAVELET DECOMPOSITION OF Z.

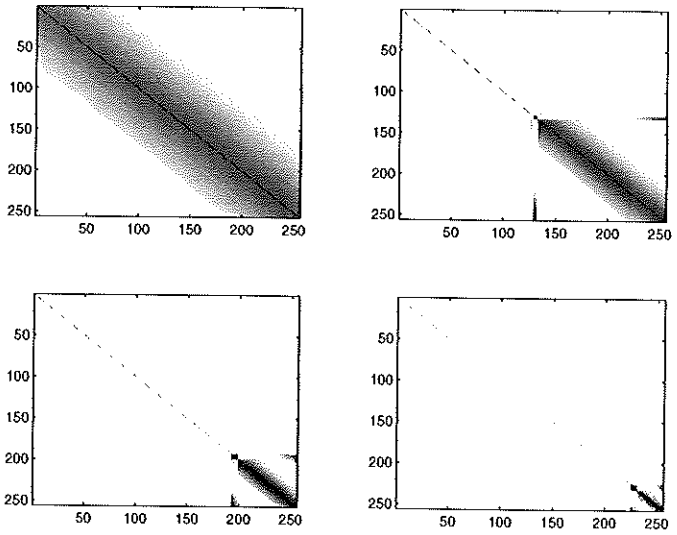

Fig. 1. These images visualise the absolute value of the following matrices. The upper left is the original matrix Z. The upper right is the two–scale transform of Z. The lower matrices visualize the result of the second and third iteration of the NS-form of the 2D wavelet transform.

will use $W = [G|H]$ to represent the two scale orthogonal transform matrix of the wavelet transform. The size of W is determined by the size of the vector or matrix it acts upon. G and H are banded matrices that can be interpreted as subsampling high– and low–pass filters. With W_j we denote the transform matrix acting on the scale coefficients of level $j-1$ to produce scale and detail coefficients for level j. Increasing j corresponds to coarser levels as in [3].

In our application the matrix $Z = \Omega + \gamma^{-1} I$ can be calculated from the x_k, γ and the σ_{RBF}. There is no need to store the N^2 elements of Z. The wavelet transform of Z has the form

$$\begin{bmatrix} A_1 & C_1^T \\ C_1 & Z_1 \end{bmatrix} = W_1^T Z W_1. \qquad (9)$$

A_1 can be interpreted as the detail part of Z, Z_1 as the coarse version of Z. C_1 and C_1^T can be seen to represent vertical and horizontal features of Z.

The transformation of Z can also be written as

$$W^T Z W = W^T(\Omega + \gamma^{-1} I)W = W^T \Omega W + \gamma^{-1} I. \qquad (10)$$

The advantage of this is that the effect of the regularization diagonal $\gamma^{-1} I$ added to the matrix can be excluded from the computation of the transform. The diagonal is preserved on every scale, which can be understood if we compare (9) with (10). $A_1 = G^T \Omega G + \gamma^{-1} I$ and $Z_1 = H^T \Omega H + \gamma^{-1} I$. These are all minor improvements to solve (5) using the method in [3] without need for excessive storage.

Table I contains an illustration of the effect of the threshold level on the relative error of the resulting vector and the sparseness of the approximation of the NS-form Cholesky factors. Note that because our only purpose at this moment is to illustrate the principle without proof, we used a full Cholesky decomposition of thresholded NS-form decomposition, in stead of the original algorithm that uses banded matrices.

We solved $Z\nu = y$, which is one of the two large linear sets of the algorithm in section III, in an approximate way, by discarding all wavelet coefficients with absolute value smaller than t. The wavelet used is the orthogonal Daubechies wavelet with 10 vanishing moments. The number of data points is $N = 256$. The vector y contains a regularly sampled version of the function $\sin(10x)/x$ on the interval $[-1, 1]$. The kernel function used is $K(x_k, x_l) = \exp[-\frac{(x_k - x_l)^2}{\sigma^2}]$ with $\sigma = 0.4$. The tuning parameter $\gamma = 1$. n_e is the total number of non-zero wavelet coefficients. n_e/N can be interpreted as the significant bandwidth around the diagonal for the NS-form matrices of wavelet coefficients. The error is the relative 2–norm of the difference with the exact solution.

Figure 1 contains an example of the original matrix (top left) and its two scale transform (top right). The parameters of the decomposition are the ones associated with the $t = 1e-05$ line in table I. This image clearly shows the sparsity of the wavelet represention of Z: most of the information in Z is captured by the coarse level coefficients (bottom right sub matrix) and by a few wavelet coefficients around the diagonal. When we apply the two–scale transform recursively to the coarse scale part of the previous level decomposition we can construct the NS-form iteratively. Note also that the boundary effects in this example can't be ignored. This is due to the fact that the length of the filters (20) is comparable to the size of the problem on coarser scales and thus decreases locality of the transform. This problem can be addressed using wavelets constructed on the interval.

Next we will comment on some alternatives to the orthogonal wavelet bases presented in [2] and used in [3].

V. BIORTHOGONAL WAVELETS AND THE LIFTING SCHEME

An extension to the approach in [3] could be the use of the lifting scheme to construct biorthogonal wavelets with desired properties on the interval [14][15]. It allows the construction of wavelets on the interval, which eliminates the need for periodic boundary conditions that introduce spurious detail coefficients and thus increases compression.

Using biorthogonal wavelets, one has a primal and a dual

wavelet basis W and $\widetilde{W}$ with the property $W^T\widetilde{W} = W\widetilde{W}^T = I$. This amounts to two possible transformation schemes of Z.

1. $\widetilde{W}^T Z \widetilde{W}$
2. $W^T Z \widetilde{W} = W^T \Omega \widetilde{W} + \gamma^{-1} I$

The first alternative preserves the symmetry of the decomposition. This means the cholesky version of [3] can still be used. Another advantage is that only the dual (analysis) wavelets need to have a lot of vanishing moments. The vanishing moments of the primal (synthesis) wavelets are not important but need to be at least one for stability reasons. The disadvantage is that the regularization diagonal $\gamma^{-1} I$ doesn't survive to coarser scales, its transform has to be included in the intermediate numerical representation. That is the main trade–off for having a four times faster decomposition, compared to the orthogonal case.

The second alternative preserves the survival of $\gamma^{-1} I$ from scale to scale but symmetry is lost, so memory requirements are twice that of the orthogonal case. Both the primal and dual wavelets need a sufficient amount of vanishing moments because they are both involved in the compression of Z. This approach isn't very interesting for our application and will be discarded.

To extend the applicability of a fast 1D LS-SVM we ought to include the case when the data are taken from an irregular grid. The lifting scheme doesn't prohibit this extension, however, there seem to be stability issues when the lifting scheme is applied without caution [7]. Our experiences with this method are inconclusive up till now.

VI. Conclusions

In this paper we have motivated the use of a multi resolution direct solver to exploit smoothness arising in one dimensional LS-SVM problems using an RBF kernel.

References

[1] Beylkin G., Coifman R. and Rokhlin V., *Fast Wavelet Transforms and Numerical Algorithms I.*, Communications on Pure and Applied Mathematics 44 (1991), 141-183.

[2] Daubechies I., *Ten lectures on wavelets*, number 61 in CBMS-NSF Regional Conference Series in Applied Mathematics. SIAM, Philadelphia, 1992.

[3] Gines D. L., Beylkin G. and Dunn J. "LU Factorization of Non-Standard Forms and Direct Multiresolution Solvers," *Applied and Computational Harmonic Analysis, 5*, pp. 156-201, 1998.

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

[5] 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.

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

[7] Simoens J., and Vandewalle S. "On the Stability of Wavelet Bases in the Lifting Scheme," *Report TW 306*, Dept. of Computer Science, K.U.Leuven, 2000.

[8] Suykens J.A.K., Vandewalle J. (Eds.) *Nonlinear Modeling: advanced black-box techniques* Kluwer Academic Publishers, Boston, 1998.

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

[10] 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.

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

[12] Suykens J.A.K., Lukas L., Vandewalle J., "Sparse approximation using least squares support vector machines," to appear *IEEE International Symposium on Circuits and Systems (ISCAS 2000)*, Geneva, Switzerland, May 28-31, 2000.

[13] Suykens J.A.K., Lukas L., Vandewalle J., "Sparse least squares support vector machine classifiers," to appear *European Symposium on Artificial Neural Networks (ESANN 2000)*, Brugge, Belgium, April 26-28, 2000.

[14] Sweldens W., Schröder P., "Building your own wavelets at home," *Wavelets in Computer Graphics*, ACM SIGGRAPH Course Notes, chapter 1-2, pages 17-87. ACM, 1996.

[15] Sweldens W.. "The lifting scheme: A construction of second generation wavelets," *SIAM J. Math. Anal.*, 29(2):511-546, March 1997.

[16] Van Gestel T., Suykens J., Baesens B., Viaene S., Vanthienen J., Dedene G., De Moor B., Vandewalle J., "Benchmarking Least Squares Support Vector Machine Classifiers," submitted for publication.

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

[18] Vapnik V., *Statistical learning theory*, John Wiley, New-York, 1998.