2010

A training algorithm for sparse LS-SVM using compressive sampling

Jie Yang
University of Wollongong, jy962@uow.edu.au

Son Lam Phung
University of Wollongong, phung@uow.edu.au

Abdesselam Bouzerdoum
University of Wollongong, bouzer@uow.edu.au

Publication Details

Yang, J., Bouzerdoum, A. & Phung, S. (2010). A training algorithm for sparse LS-SVM using compressive sampling. 2010 IEEE International Conference on Acoustics, Speech, and Signal Processing (pp. 2054-2057). USA: IEEE.

A training algorithm for sparse LS-SVM using compressive sampling

Abstract
Least Squares Support Vector Machine (LS-SVM) has become a fundamental tool in pattern recognition and machine learning. However, the main disadvantage is lack of sparseness of solutions. In this article Compressive Sampling (CS), which addresses the sparse signal representation, is employed to find the support vectors of LS-SVM. The main difference between our work and the existing techniques is that the proposed method can locate the sparse topology while training. In contrast, most of the traditional methods need to train the model before finding the sparse support vectors. An experimental comparison with the standard LS-SVM and existing algorithms is given for function approximation and classification problems. The results show that the proposed method achieves comparable performance with typically a much sparser model.

Keywords
svm, sampling, compressive, sparse, training, algorithm, ls

Disciplines
Physical Sciences and Mathematics

Publication Details
Yang, J., Bouzerdoum, A. & Phung, S. (2010). A training algorithm for sparse LS-SVM using compressive sampling. 2010 IEEE International Conference on Acoustics, Speech, and Signal Processing (pp. 2054-2057). USA: IEEE.

A TRAINING ALGORITHM FOR SPARSE LS-SVM USING COMPRESSIVE SAMPLING

Jie Yang, Abdesselam Bouzerdoum, Son Lam Phung

School of Electrical, Computer and Telecommunications Engineering
University of Wollongong, Wollongong, NSW 2522, Australia

ABSTRACT

Least Squares Support Vector Machine (LS-SVM) has become a fundamental tool in pattern recognition and machine learning. However, the main disadvantage is lack of sparseness of solutions. In this article Compressive Sampling (CS), which addresses the sparse signal representation, is employed to find the support vectors of LS-SVM. The main difference between our work and the existing techniques is that the proposed method can locate the sparse topology while training. In contrast, most of the traditional methods need to train the model before finding the sparse support vectors. An experimental comparison with the standard LS-SVM and existing algorithms is given for function approximation and classification problems. The results show that the proposed method achieves comparable performance with typically a much sparser model.

Index Terms— Least Squares Support Vector Machine (LS-SVM), Model Selection, Compressive Sampling, Sparse Approximation, Orthogonal Matching Pursuit (OMP)

1. INTRODUCTION

Support Vector Machine (SVM) theory has received great deal of attention since its introduction by Vapnik in the 1990's [1]. Along with kernel methods, SVM is employed as an essential machine learning tool for regression and classification tasks. A variant of SVM, knows as the *Least Squares Support Vector Machine* (LS-SVM), was introduced by Suykens and Vandewalle in 1999 [2]. In LS-SVM, the ϵ-sensitive loss function, used with SVM, is replaced with equality constraints to accelerate the training process; thereby, the quadratic programming problem of SVM is reduced to that of solving a system of linear equations [3]. Another advantage of the LS-SVM formulation is that it involves fewer tuning parameters.

However, the major drawback of LS-SVM is that the solution lacks sparseness in terms of the number of support vectors, which may influence its generalization capacity as well as the training complexity. Several pruning methods have been suggested in order to improve the sparseness of LS-SVM solution. Suykens et al. [4] first proposed removing training samples that have the smallest absolute support values (i.e., Lagrange multipliers). However this method also eliminates training samples near the decision boundary, which has a negative influence on the classifier performance. An improved pruning method was proposed in [5], where a reduced training set comprised of samples near the decision boundary is used to retrain the LS-SVM. In [6], support vectors are eliminated by minimizing the output error after some samples have been deleted. However, the method involves the inversion of a matrix that is often singular or near singular, which requires more computation. An enhancement of the method in [6] was proposed by Kuh and De Wilde [7], in which the support vectors are omitted through regularization so as to accelerate the pruning process. For more on LS-SVM pruning algorithms, the interested reader is referred to the survey presented in [8].

In this paper, we present a Compressive Sampling-based learning algorithm for LS-SVM. Compressive Sampling or Compressed Sensing (CS) [9], which addresses the sparse signal representation, can help in recovering signals that have a sparse representation from a number of measurements/projections lower than the traditional sampling number required by the Shannon/Nyquist sampling theory. Thus, if we consider the LS-SVM model as a sparse structure comprised of support vectors, then CS can be employed to reconstruct this topology. The main advantage of our approach is that the proposed algorithm is capable of iteratively building up the sparse topology, while maintaining the training accuracy of the original larger structure.

The remainder of the paper is organized as follows. Section 2 gives a brief introduction to Compressive Sampling theory. Section 3 presents the proposed approach for training LS-SVM based on CS. Section 4 compares the CS-based algorithm with several other methods on function approximation and classification tasks. Section 5 presents the concluding remarks.

2. COMPRESSIVE SAMPLING

With the rapidly increasing demand on large-scale signal processing, it is not surprising to see a significant research efforts devoted to Compressive Sampling in recent years [9]. Given non-traditional samples in the form of randomized projections, the theory allows us to capture most of the salient information in a signal with a relatively small number of sam-

ples, often far fewer than what is required using traditional sampling schemes. Compressive Sampling algorithms can be divided into two broad categories: (i) Single Measurement Vector (SMV) [10] where the solution is a vector; and (ii) Multiple Measurement Vectors (MMV) [11] where the solution is a two-dimensional array or matrix.

We apply SMV in this paper to achieve a sparse structure for LS-SVM. Mathematically, the SMV problem is expressed as follows. Given a measurement sample $\boldsymbol{y} \in \Re^m$ and a dictionary $\mathcal{D} \in \Re^{m \times n}$ (the columns of $\mathcal{D}$ are referred to as the atoms), we seek a vector solution satisfying:

$$(P): \quad min \left\|\boldsymbol{x}\right\|_0 \quad s.t. \quad \boldsymbol{y} = \mathcal{D}\boldsymbol{x} \tag{1}$$

where $\left\|\boldsymbol{x}\right\|_0$ (known as l_0-norm) is the cardinality or number of nonzero elements in $\boldsymbol{x}$. Several algorithms have been developed including Greedy and Non-convex local optimization algorithm. In this paper, we are only concerned with *Orthogonal Matching Pursuit* (OMP) algorithm; the reader is referred to [10] for more details on its implementation. In particular, the convergence property of OMP is demonstrated by the following two theorems [12]:

Theorem 1 *For any sample $\boldsymbol{y}$, there exists a time function $\beta(t) \in (0,1)$, which depends only on the dictionary, such that the residual error calculated by OMP decays as: $\left\|\boldsymbol{r}_t\right\|^2 \leq \beta(t) \left\|\boldsymbol{r}_{t-1}\right\|^2$, where $\boldsymbol{r}_t = \boldsymbol{y} - \mathcal{D}\boldsymbol{x}_t$ is the reconstruct error after t iterations. The upper limit for the residual error is given by $\left\|\boldsymbol{r}_t\right\|^2 \leq \beta(t) \left\|\boldsymbol{r}_{t-1}\right\|^2 \leq ... \leq \prod_{i=1}^{t} \beta(i) \left\|\boldsymbol{r}_0\right\|^2$.*

Theorem 2 *Given an arbitrary d-sparse signal $\boldsymbol{x} \in \Re^n$ ($n \geq d$) and a random $m \times n$ linearly independent matrix $\mathcal{D}$. OMP can represent $\boldsymbol{x}$ with probability exceeding $1-\delta$ when the following condition is satisfied: $m \geq c\, d \log(n/\delta)$, where c is a positive constant, and $\delta \in (0, 0.36)$.*

3. SPARSE LEARNING ALGORITHMS FOR LS-SVM

Consider the traditional machine learning problem, where a set of N one-dimensional training samples, $\{(x_i, y_i)\}_{i=1}^{N}$, is observed. The aim is to find a mapping $f(\boldsymbol{x})$ so that $f(x_i) \approx y_i, \forall i$. SVM projects the input vectors x_i onto a higher dimensional feature space, using a kernel function $\varphi(x_i)$. A maximal-margin hyperplane is then used to separate the data in the feature space. Therefore, the output is given by

$$f(x_i) = \boldsymbol{w}^T \varphi(x_i) + b \tag{2}$$

where the weight vector $\boldsymbol{w}$ and the bias b are to be estimated from the training data. This usually requires the solution of a quadratic program with inequality constraints. By contrast, in LS-SVM the inequality constraints are replaced with equality constraints, and then the unknown parameters are obtained by solving the following problem:

$$\text{minimize } \mathcal{J}(\boldsymbol{w}, b, \boldsymbol{e}) = \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + \frac{\gamma}{2}\sum_{i=1}^{N} e_i^2 \tag{3}$$

subjects to:

$$y_i = \boldsymbol{w}^T \varphi(x_i) + b + e_i, \quad i = 1, 2, ..., N \tag{4}$$

where $\boldsymbol{e}$ is the error terms and γ is a regularization parameter. Furthermore, this problem can be solved using the Lagrange multiplier method:

$$\text{minimize } \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{e}, \boldsymbol{\alpha}) = \mathcal{J}(\boldsymbol{w}, b, \boldsymbol{e}) + \sum_{i=1}^{N} \alpha_i [y_i - \boldsymbol{w}^T \varphi(x_i) - b - e_i] \tag{5}$$

The Karush-Kuhn-Tucker conditions for the above problem are reduced to a linear system by eliminating $\boldsymbol{w}$ and $\boldsymbol{e}$:

$$\left[\begin{array}{c|c} Q + \gamma^{-1} I_N & 1 \\ \hline \mathbf{1}_N^T & 0 \end{array}\right] \left[\begin{array}{c} \boldsymbol{\alpha} \\ b \end{array}\right] = \left[\begin{array}{c} \boldsymbol{y} \\ 0 \end{array}\right] \tag{6}$$

where $Q_{ij} = \varphi(x_i)^T \varphi(x_j)$, $\mathbf{1}_N^T$ is the N-dimensional row vector whose elements are equal to 1, and I_N is the $N \times N$ identity matrix.

Our aim is to find the optimal and sparse parameter vector $[\boldsymbol{\alpha}\ b]^T$ which satisfies (6). We should note that setting a particular element of $\boldsymbol{\alpha}$ to zero is equivalent to pruning the corresponding training sample. In this regard, the goal of finding a sparse LS-SVM solution, within a given tolerance of accuracy, can be equated to solving (6) by minimizing the l_0-norm of the vector $[\boldsymbol{\alpha}\ b]^T$. The problem can be cast as follows:

$$min \left\| \left[\begin{array}{c} \boldsymbol{\alpha} \\ b \end{array}\right] \right\|_0 s.t. \left[\begin{array}{c} \boldsymbol{y} \\ 0 \end{array}\right] = \underbrace{\left[\begin{array}{c|c} Q + \gamma^{-1} I_N & 1 \\ \hline \boldsymbol{I}^T & 0 \end{array}\right]}_{\mathcal{D}} \left[\begin{array}{c} \boldsymbol{\alpha} \\ b \end{array}\right] \tag{7}$$

Comparing the problem in (7) with (1), we can see that the sparse LS-SVM learning problem is reduced to that of Compressive Sampling, where the dictionary $\mathcal{D}$ is replaced with $\left[\begin{array}{c|c} Q + \gamma^{-1} I_N & 1 \\ \hline \boldsymbol{I}^T & 0 \end{array}\right]$. Therefore, any algorithm that can solve the CS problem in (1) can also be employed to solve the sparse LS-SVM learning problem; hereafter, such an algorithm is referred to as the CLS-SVM. In this paper, however, we concentrate on the OMP algorithm, which has been proven effective for solving the CS problem.

An important question that arises is "will the CLS-SVM algorithm converge?" In fact, its convergence is only influenced by the number of training samples and orthogonality of data. Suppose that we have N_{sv} support vectors with $N > cN_{sv}\log(N/\delta)$. According to $Theorem\ 2$, after N_{sv} iterations, CLS-SVM is guaranteed to find the sparsest solution with probability exceeding $1-\delta$. Therefore, the convergence

of CLS-SVM is guaranteed. Unfortunately, the above claim is built on the success of pursuit algorithms, which depends on the number of training samples and orthogonality of data, and thus convergence is not always guaranteed. However, according to $Theorem$ 1, the OMP method, which has a decreasing residual error, is still known to perform reasonably well [10].

4. EXPERIMENT RESULTS AND ANALYSIS

The proposed CLS-SVM algorithm is tested on three benchmark data sets taken from the UCI benchmark repository: two function approximation problems (*Sinc* function and *Housing* dataset) and one classification problem (*Heart* dataset). In all experiments the data are standardized to zero mean and unit variance. We run all the algorithms 30 times to collect the performance statistics. The degree of sparsity of a solution can be measured as

$$Sparsity = (1 - N_{sv}/N) \times 100\%$$

Here N_{sv} is the number of support vectors and N is the number of training samples.

4.1. Function Approximation

The CLS-SVM algorithm is compared to the methods proposed in [4], [6], and [7] using the sinc data corrupted with zero-mean white Gaussian noise for training. The log mean square error values achieved by the four algorithms are listed in Table 1. Clearly the CLS-SVM method outperforms all the other three methods by achieving the lowest MSE. It is worth noting also that the CLS-SVM algorithm with only 30 support vectors achieves a comparable error to the standard LS-SVM with 100 support vectors. Figure 1 illustrates the sparsity of CLS-SVM solution, where the majority of $\boldsymbol{\alpha}_k/\gamma$ values are nil.

Table 1. The log (MSE) for the *Sinc* function approximation.

	LS-SVM	[4]	[6]	[7]	CLS-SVM
MSE	-4.50	-4.11	-4.25	-4.23	-4.38
N_{sv}	100	30	30	30	30

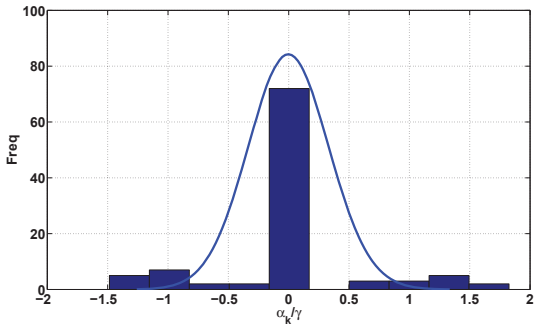

Fig. 1. Histogram of $\boldsymbol{\alpha}_k/\gamma$ values for CLS-SVM.

For the *Housing* Dataset, the CLS-SVM is compared with the methods discussed in [8]; for more details, the reader is referred to [8]. Some parts of results are presented in Table 2. Again for the same degree of sparsity, the CLS-SVM achieves the lowest MSE in all cases. Although, the prediction error of the CLS-SVM increases slowly as the degree of sparsity increases, due to the fact that more support vectors are being eliminated, the degradation is less severe than in the other cases (other results are not shown here due to space limitation, but the same conclusions can be drawn).

Table 2. Results using log MSE for different algorithms on *Housing* Dataset.

Algorithms	log (MSE)			
Sparsity	10%	20%	50%	90%
Random	-1.767	-1.849	-1.413	-0.924
Sv	-1.973	-1.962	-1.793	-1.137
Weighed sv1	-1.985	-1.989	-2.022	-1.810
Weighed sv2	-1.946	-1.456	-1.290	-0.101
Weighed sv3	-1.979	-1.980	-1.986	-1.406
Span	-2.011	-2.059	-1.996	-1.196
CLS-SVM	-3.1498	-3.1175	-3.1295	-2.8799

4.2. Classification

As for the classification tasks, Table 3 shows the Sparsity, Training and Test accuracy for different methods using the *Heart* Dataset. For comparison purposes, we first implemented the CLS-SVM using the sparsest degree achieved by its counterpart [5]. Then we found the sparsest solution that the CLS-SVM can achieve within a range of accuracy. The results from all models are shown in Table 3.

From these results, the following observations can be made. Firstly, the CLS-SVM achieves a significant improvement in test accuracy when using the same sparseness level as its counterparts. Furthermore, if we relax the requirements on the test accuracy and focus on the sparsest architecture, it is interesting that CLS-SVM has a sparser model compared to other methods. For example, with a sparsity of 89%, the CLS-SVM achieves the same test accuracy as [5], which has a sparsity of 76.11%. Overall, the training method with Compressive Sampling into LS-SVM method improves significantly the performance and robustness of the original algorithm.

4.3. Termination Criterion

The CLS-SVM training is terminated if certain stopping conditions are satisfied. These conditions may be considered formally, for example, in terms of a maximum number of suppert vectors. Note that in our algorithm the support vectors are added to the model iteratively. In this subsection, we analyze how the tuning parameter for the number of support vectors

Table 3. Comparison of Classification for the *Heart* Dataset.

Heart	Sparsity	Training (%)	Test (%)
LS-SVM	/	85.44	84.22
[4]	52.22%	82.89	83.33
[5]	76.11%	82.11	85.56
[6]	47.78%	82.89	82.89
CLS-SVM	76.11%	87.78	86.67
CLS-SVM	89.00%	82.22	85.56

(N_{sv}) impacts the performance of CLS-SVM in terms of test accuracy.

Here, the experiment is based on an artificial dataset with two inputs and one output. For the training set, 300 input samples, (x_1, x_2), are generated randomly using a uniform distribution in the range $[-1, 1]$. The corresponding output samples are computed using the function $y = sign[\sin(x_1) + \sin(x_2)]$ to give an output value of $+1$ or -1. The test set samples are generated independently of the training set, but using the same procedure. The CLS-SVM algorithm is run with different values of N_{sv}. Fig. 2 illustrates the classification accuracy as the tuning parameter N_{sv} is varied. Two regions can be highlighted in the figure: ***Zone A*** where the classification accuracy increases sharply with increasing N_{sv}, and ***Zone B*** where the classification accuracy reaches saturation. In this example, it would be better to choose $N_{sv} = 0.35 \times N$ because it corresponds to the sparsest structure with the highest classification accuracy. A similar procedure can be employed in practice using a validation set, where the training is stopped when the saturation region is reached.

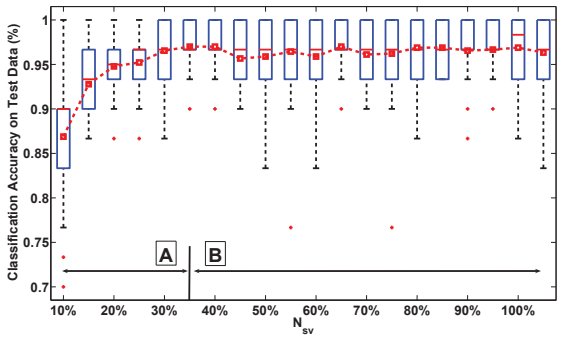

Fig. 2. Classification Accuracy (%) for the CLS-SVM algorithm with different values of N_{sv}

5. CONCLUSION

In this paper we have proposed a new LS-SVM training method based on Compressive Sampling (CS) that offers a better trade-off between computational accuracy and sparseness requirement. We regard the kernel matrix in the LS-SVM model as a dictionary in CS, and then the goal for finding a minimal topology in LS-SVM is simply changed into locating a sparse solution. The main difference between our work and the existing techniques is that the proposed method can locate the sparse topology. However, most of the traditional methods need to train the model before finding the support vectors; consequently, our method can lead to a quick convergence and a much sparser structure.

6. REFERENCES

[1] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer-Verlag, 1995.

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

[3] J. A. K. Suykens *et al.*, *Least Squares Support Vector Machines*, Singapore: World Scientific, 2002.

[4] J. A. K. Suykens, L. Lukas and J. Vandewalle, "Sparse approximation using least squares support vector machines," *Proc. IEEE Int. Symp. Circuits and Systems*, Geneva, Switzerland, May 2000, pp 757–760.

[5] Y. Li, C. Lin and W. Zhang, "Improved sparse least-squares support vector machine classifiers," *Neurocomputing*, vol. 69, no. 13, pp 1655–1658, 2006.

[6] B. J. de Kruif and T. J. deVries, "Pruning error minimization in least squares support vector machines," *IEEE Trans. Neural Nets.*, vol. 14, pp 696–702, 2003.

[7] A. Kuh and P. De Wilde, "Comments on "Pruning Error Minimization in Least Squares Support Vector Machines," *IEEE Trans. Neural Networks*, vol. 18, no. 2, pp 606–609, 2007.

[8] L. Hoegaerts, J. A. K. Suykens, J. Vandewalle, and B. De Moor, "A comparison of pruning algorithms for sparse least squares support vector machines," *Proc. 11th ICONIP*, Calcutta, India, 2004, pp. 22–25.

[9] H. Rauhut, K. Schnass and P. Vandergheynst, "Compressed Sensing and Redundant Dictionaries," *IEEE Trans. Info. Theory*, vol. 54, pp 2210–2219, 2008.

[10] J. A. Tropp and C. Gilbert Anna, "Signal Recovery From Random Measurements Via Orthogonal Matching Pursuit," *IEEE Trans. Information Theory*, vol. 53, pp 4655–4666, 2007.

[11] S. F. Cotter *et al.*, "Sparse solutions to linear inverse problems with multiple measurement vectors," *IEEE Trans. Signal Processing*, vol. 53, pp 2477–2488, 2005.

[12] R. Gribonval and P. Vandergheynst, "On the Exponential Convergence of Matching Pursuits in Quasi-Incoherent Dictionaries," *IEEE Trans. Information Theory*, vol. 52, pp 255–261, 2006.