Zero Norm Least Squares Proximal SVR

Jayadeva[1], Sameena Shah[1], and Suresh Chandra[2]

[1] Dept. of Electrical Engineering, [2] Dept. of Mathematics, Indian Institute of Technology, New Delhi 110016, India
jayadeva@ee.iitd.ac.in, sameena.shah@gmail.com, chandras@maths.iitd.ac.in

Abstract. Least Squares Proximal Support Vector Regression (LSPSVR) requires only a single matrix inversion to obtain the Lagrange Multipliers as opposed to solving a Quadratic Programming Problem (QPP) for the conventional SVM optimization problem. However, like other least squares based methods, LSPSVR suffers from lack of sparseness. Most of the Lagrange multipliers are non-zero and thus the determination of the separating hyperplane requires a large number of data points. Large zero norm of Lagrange multipliers inevitably leads to a large kernel matrix that is inappropriate for fast regression on large datasets. This paper suggests how the LSPSVR formulation may be recast into one that also tries to minimize the zero norm of the vector of Lagrange multipliers, and in effect imposes sparseness. Experimental results on benchmark data show that a significant decrease in the number of support vectors can be achieved without a concomitant increase in the error.

Keywords: SVR, sparse representation, zero-norm, proximal, least squares.

1 Introduction

Support Vector Machines (SVMs) are computationally efficient for classification as well as regression tasks [1]. The elegance of SVMs lies in the fact that the nature of the optimization problem that needs to be solved for both linear and non-linear regression problems remains the same. The optimal hyperplane for the regression problem is determined by solving the problem stated in (1). Usually the dual of (1), a quadratic programming problem (QPP), is solved to obtain the support vectors.

$$\underset{q,q',w,b}{\text{Minimize}} \quad \frac{1}{2} w^T w + \mathrm{C} e^T (q + q')$$

subject to

$$y - (\mathbf{P}w + \mathrm{b}) \leq q,$$

$$(\mathbf{P}w + \mathrm{b}) - y \leq q',$$

$$q, q' \geq 0, \tag{1}$$

where, $w \in \Re^N$ is the weight vector, $\mathrm{b} \in \Re$ is the bias, q is the error for each data point, e is an M dimensional vector of ones, and $\mathrm{C} > 0$ is a parameter that trades off

S. Chaudhury et al. (Eds.): PReMI 2009, LNCS 5909, pp. 38–43, 2009.

accuracy with the complexity of the regressor. **P** is the data matrix containing *M* data points, where each data point $x_i \in \Re^N$, $i=1,...,M$ has a corresponding function value y_i, $i=1,....,M$. LSSVMs [2] for regression solve the following optimization problem.

$$\underset{q,w}{\text{Minimize}} \quad \frac{\mathrm{C}}{2}\left(q^T q\right)+\frac{1}{2}\left(w^T w\right)$$

subject to

$$w^T x_i + b = y_i - q_i, \qquad i=1,..., M \tag{2}$$

The introduction of the l_2 norm of the error variable in the objective function changes the nature of the optimization problem that needs to be solved to obtain the Lagrange multipliers. The solution of LSSVM can be obtained by solving the following system of equations

$$\begin{bmatrix} 0 & e^T \\ e & \mathbf{K}+\dfrac{I}{C} \end{bmatrix}\begin{bmatrix} \mathrm{b} \\ u \end{bmatrix}=\begin{bmatrix} 0 \\ y \end{bmatrix} \tag{3}$$

where, **I** is an identity matrix (*M*x*M*), φ is the nonlinear mapping to higher dimension and **K** is the kernel matrix $K_{ij} = [\phi(P_i)]^T \phi(P_j)$, $i, j = 1, 2, ..., M$.

LSSVMs require the solution of a set of linear equations rather than a QPP, for which the authors propose the use of iterative methods like SOR or conjugate gradient. Though LSSVM is computationally better than conventional SVMs, a disadvantage that least squares methods generally suffer from is the lack of sparseness. The support values are proportional to the errors at all the data points while in conventional SVMs many of them may be zero.

In this paper, we explore the possibility of using the zero norm in conjunction with least squares based methods to impose sparseness. The least squares based method we have chosen is Least Squares Proximal Support Vector Regression (LSPSVR) [3]. LSPSVR has been recently proposed and requires only the inversion of a single, positive definite matrix to obtain the Lagrange multipliers. However, LSPSVR also yields non-sparse solutions. It is important to have most components of β as zero, so as to obtain a kernel matrix of small size. In Section 2 we give the zero norm algorithm and show how it can be used to modify the LSPSVR formulation to obtain a sparse representation. Section 3 contains experimental results obtained for three data sets using this algorithm and finally section 4 is devoted to concluding remarks.

2 Zero Norm LSPSVR

The zero norm of a vector is defined as its number of non- zero components. Formally, the zero norm of a vector *w* is given by $\|w\|_0^0 = card\{w_i | w_i \neq 0\}$, where *card* denotes the set cardinality. The requirement of minimum number of non zero components of β can thus be formulated as the problem of minimizing the zero norm of β.

The l_p norm of the vector w is defined as $\|w\|_p = \left(\sum_{i=1}^{n} w_i^p\right)^{\frac{1}{p}}$. The minimization of the l_p norm of w in conjunction with the constraint $y_i(w.x_i + b) \geq 1$ solves the classical SVM problem. The generalization capabilities of such linear models have been studied for over a decade now. A general belief that has emerged out of this research is that for $p \geq 1$ the minimization of the l_p - norm of w is good for generalization. For $p = 0$, it has been shown in [4] that the problem of minimization of zero norm subject to the constraints $y_i(w.x_i + b) \geq 1$ is NP-hard. Therefore, one needs to use some approximation or local search method for solving the problem. Weston et al. [5] introduced an iterative algorithm, termed as 'Approximation of the zero-norm Minimization (AROM)' that performs a gradient step at each iteration and converges to a local minimum.

AROM Algorithm. AROM [6] solves the following optimization problem.

$$\underset{w}{\text{Minimize}} \quad \| w \|_0^0$$

subject to

$$y_i[w^T x_i + b] \geq 1, \qquad i = 1,...,l \tag{10}$$

i.e. it finds a separating hyperplane with the fewest nonzero elements as the coefficients of the vector $\boldsymbol{w}$. In order to minimize the zero norm of the vector $w = (w_1, w_2, ..., w_N)^T$, a vector of coefficients $z = (z_1, z_2, ..., z_N)^T$ is introduced.

1. Set $z = (1,\ldots,1)^T$
2. Solve

$$\text{Min} \sum_{j=1}^{n} | w_j |$$

subject to:

$$y_i\left(w(x_i * z) + b\right) \geq 1$$

3. Let $\overline{w}$ be the solution of the previous problem. Set $z \leftarrow z * \overline{w}$.
4. Go back to 2 until convergence.

AROM requires the solution of a succession of linear programs combined with a multiplicative update to reduce the number of nonzero components of w, i.e. the dimension of the input data points. Following AROM we introduce a diagonal matrix $\mathbf{Z}$ in the LSPSVR formulation and propose the optimization problem.

Zero Norm LSPSVR

$$\underset{q,w,b}{\text{Minimize}} \quad \mathrm{C} \cdot \frac{1}{2}\left(q^T q\right) + \frac{1}{2}\left(w^T w + \mathrm{b}^2\right)$$

subject to

$$\left(\mathbf{K}(\mathbf{P}, \mathbf{P}^{\mathbf{T}})w + e\,b\right) - y + \mathbf{Z}q = 0\,. \tag{11}$$

The Lagrangian is given by

$$L(w,\mathrm{b},q,\beta)=\frac{C}{2}\|q\|^2+\frac{1}{2}\left\|\begin{bmatrix} w \\ \mathrm{b}\end{bmatrix}\right\|^2-\beta^T\left[\mathbf{K}w+e\,\mathrm{b}-y+\mathbf{Z}q\right] \tag{12}$$

Substituting the KKT conditions, we get

$$\left(\mathbf{K}\mathbf{K}^T+ee^T\right)\beta+\frac{\mathbf{Z}\mathbf{Z}^T\beta}{C}=y\,. \tag{13}$$

On simplifying we obtain $\beta=\left[\mathbf{G}\mathbf{G}^{\mathrm{T}}+\frac{\mathbf{Z}\mathbf{Z}^{\mathbf{T}}}{\mathrm{C}}\right]^{-1}*y$ where $\mathbf{G}=\begin{bmatrix}\mathbf{K} & e\end{bmatrix}$. The form of the equation that needs to be solved to obtain the Lagrange multipliers remains the same as in LSPSVR, and β can still be obtained by just a single matrix inversion, thus preserving the advantage offered by LSPVR. A modification we have introduced is that instead of minimizing the zero norm of weight vector i.e. the number of dimensions, we attempt to minimize the number of patterns. This implies the minimization of the number of components in the error vector $\boldsymbol{q}$. In other words, we attempt to have the same classification performance but such that comparatively large amount of error is contributed by only a few points rather than comparatively smaller amount of error contributed by a large number of points. This is done by multiplying $\mathbf{Z}$ by the error variable instead of the weight vector as done in feature selection. Since the support values for LSPSVR are related to the error vector, $q=\frac{Z^T\beta}{\mathrm{C}}$, therefore the minimization of the zero norm of $\boldsymbol{q}$ implies the minimization of the zero norm of β.

The zero norm LSPSVR algorithm is summarized as below.

1. Set $\mathbf{Z}$ = diagonal matrix of ones.
2. Solve the problem in (11).
3. If $\bar{\beta}$ is the solution of the current problem. Update $\mathbf{Z}\leftarrow\mathbf{Z}*\boldsymbol{q}=\mathbf{Z}*\mathbf{Z}^{\mathbf{T}}*\frac{\bar{\beta}}{\mathrm{C}}$
4. Eliminate all the data vectors i, st. $\mathbf{Z}_{ii}<0$, $\forall i$.
5. Go to Step 2 and iterate till the termination condition is met. There are many possible termination criteria, for example, allowed error bounds, bound on the number of data points, or the required sparseness.

3 Experimental Results

All the program codes were written in MATLAB 7.0 and executed on a Pentium III PC with 256 MB RAM. Co-ordinates of data samples were normalized to lie between zero and one. For each data set, the parameter C was determined by choosing a small tuning set of 20% samples. In all cases, regression was performed by using a polynomial kernel of degree 2, of the form $\mathrm{K}(x_i,x_j)=(x_i.x_j+1)^2$. Figures 1, 2 and 3 show how training and testing errors change as the number of data samples is reduced according to the zero norm LSPSVR algorithm on Comp activ, Census house and Boston housing benchmark data sets [7, 8]. The x-axis in each figure depicts the number

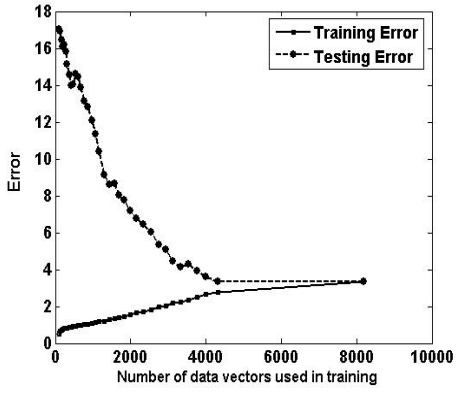

Fig. 1. Training and Testing errors on Comp Active data using 10 fold cross validation

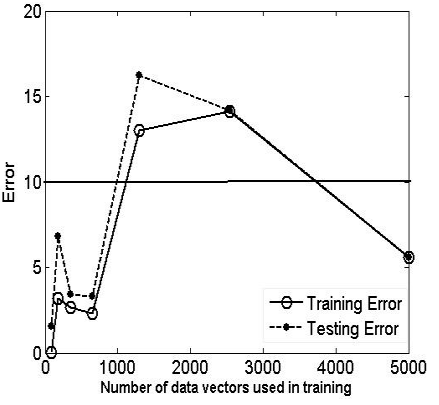

Fig. 2. Training and Testing errors on Census House data using 10 fold cross validation

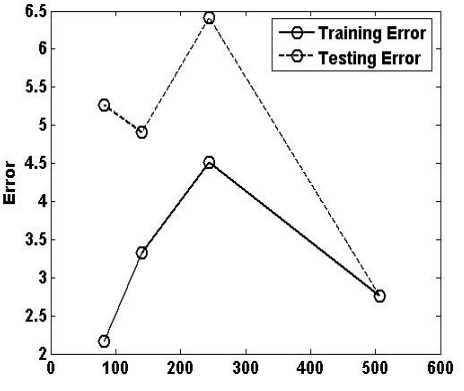

Fig. 3. Training and Testing errors on Housing data using 10 fold cross validation

of data points that were used in training, whereas the *y* axis indicates the average percentage error over 10 sample runs. The testing error was computed over all the data points, whereas the training error was determined as the total error over the data points used in training. The number of training patterns reduces after each iteration, while the number of testing patterns is a constant and equal to the size of the complete data set. Comp active has 8192 points lying in 22 dimensions. Figure 1 shows that the number of data points reduces to around 4000 in the first iteration itself without a significant increase in the error. The figure indicates that it may be best to use the result obtained after three or four iterations, because after that, the training error keeps decreasing while the testing error keeps increasing. Figure 2 shows the results for Census house data set of which we have used the first 5000 data points. We have also eliminated the dimensions whose value is a constant. The number of effective dimensions is thus 120. Although in the first couple of iterations the error goes up, but later on it comes down significantly and is much smaller than the initial value. This shows

that it is possible to achieve better generalization while using fewer support vectors. An important point to note is that the number of data points has been reduced to about 60 from an initial value of 5000. This marks an enormous reduction in the size of the kernel from 5000 x 5000 to 60 x 60. Figure 3 shows the results for zero norm LSPSVR applied on Boston Housing data set that has 506 data points, each of dimension 14. The results show that the training error comes down and the testing error goes up with a marked reduction in the data set to about 16% of its initial size.

4 Conclusion

In this paper we present a novel algorithm that requires a single matrix inversion and a multiplicative update in each iteration to reduce the number of support vectors. Results on benchmark data sets show that it is possible to obtain a significant reduction in the number of support vectors within three or four iterations of the proposed algorithm. At each iteration, a significant number of data points are eliminated. Therefore, for later iterations the size of the data set reduces and problem (11) can be solved much faster. The zero norm LSPSVR approach was able to yield a sparse solution that did not significantly increase the error rate. However, in a couple of cases the generalization error actually reduced below the initial value. This could be because of elimination of outliers, but is an observation that merits further investigation. Since each iteration only requires a single matrix inversion, fast implementations of the proposed scheme are possible. The matrices to be inverted in successive steps are all positive definite, and related to each other, and it is therefore interesting to explore more efficient methods of implementing the proposed scheme. The zero norm approach can be applied to other least squares methods, most notably for imposing sparseness in LSSVM.

References

1. Cristianini, N., Taylor, J.S.: An Introduction to Support Vector Machines and other kernel based learning methods. Cambridge University Press, Cambridge (2000)
2. Suykens, J.: Least Squares Support Vector Machines. In: IJCNN (2003), http://www.esat.kuleuven.ac.be/sista/lssvmlab/
3. Jayadeva, R.K., Chandra, S.: Least Squares Proximal Support Vector Regression. Neurocomputing (communicated)
4. Amaldi, E., Kann, V.: On the approximability of minimizing non zero variables or unsatisfied relations in linear systems. Theoretical Computer Science 209, 237–260 (1998)
5. Weston, J., Elisseeff, A., Scholkopf, B.: Use of the l_0-norm with linear models and kernel methods. Technical report (2001)
6. Weston, J., Elisseeff, A., Scholkopf, B., Tipping, M.: Use of Zero Norm with Linear Models and Kernel Machines. Journal of Machine Learning Research 3, 1439–1461 (2003)
7. Murphy, P.M., Aha, P.M.: UCI Repository of Machine learning Databases (1992), http://www.ics.uci.edu/mlearn/MLRepository.html
8. Data for Evaluating Learning in Valid experiments, http://www.cs.utoronto.ca/~delve