The Korean Communications
in Statistics Vol. 10 No. 2, 2003
pp. 595-601

On Line LS-SVM for Classification[1)]

Daehak Kim[2)], Kwangsik Oh[3)] and Jooyong Shim[4)]

Abstract

In this paper we propose an on line training method for classification based on least squares support vector machine. Proposed method enables the computation cost to be reduced and the training to be performed incrementally. With the incremental formulation of an inverse matrix in optimization problem, current information and new input data can be used for building the new inverse matrix for the estimation of the optimal bias and Lagrange multipliers, so the large scale matrix inversion operation can be avoided. Numerical examples are included which indicate the performance of proposed algorithm.

Keywords : Least Squares Support Vector Machine, Classification, Lagrange multiplier, Inverse matrix, On line training.

1. Introduction

Support vector machine(SVM), introduced by Vapnik(1995, 1998) and coworkers, has experienced rapid development. Despite of many successful application of SVM in classification and function estimation problem, SVM requires to solve a quadratic programming(QP) problem which is time memory expensive. Suykens and Vanderwalle(1999) proposed a modified version of SVM in a least squares sense for classification. In least squares support vector machine(LS-SVM), the solution is given by a linear system instead of a QP problem. The fact that LS-SVM has explicit primal-dual formulations has an advantage of fast computation. Furthermore it is easy to formulate modified version such as the weighted LS-SVM for the robust estimation(Suykens *et al.*, 2002). But the proposed LS-SVM algorithm is trained in batch form, which is not suited to the real application such as on line system identification and control, where the data come in sequentially. So the on line training for the classification is needed urgently in real application.

---

1) This research was supported by the Catholic University of Daegu Research Grant in 2003

2) Professor, Department of Statistical Information, Catholic University of Daegu, Kyungbuk, 712-702, Korea. Email : dhkim@cu.ac.kr

3) Professor, Department of Statistical Information, Catholic University of Daegu, Kyungbuk, 712-702, Korea. Email : ohkwang@cu.ac.kr

4) Adjunct Professor, Department of Statistical Information, Catholic University of Daegu, Kyungbuk, 712-702, Korea. Email : jyshim@cu.ac.kr

Ahmed *et al.*(1999) has brought forth an incremental training algorithm for SVM classification. The basic idea is that only the support vectors are preserved, and those support vectors plus the new coming data are used for training again. The main drawback is that the training is not exactly incremented. It is approximately incremental and the Lagrange multipliers corresponding to the support vectors are not updated incrementally. Cauwenberghs and Poggio(2001) proposed the exact incremental and decremental training for SVM classification. Friess *et al.*(1999) proposed a sequential gradient method for SVM, where the main problem is that the training is not convergent quickly.

In this paper we propose the exact on line(incremental) training for LS-SVM classification, which makes the on line LS-SVM applied for system identification and control possible. In Section 2 we give an overview of LS-SVM classification. In Section 3 we present the on line LS-SVM for classification. In Section 4 we perform the numerical studies with simulated data set and real data set. In Section 5 we give the remarks and conclusions.

2. LS-SVM for classification

Let the training data set D be denoted by $\{\boldsymbol{x}_i, y_i\}_{i=1}^N$, with each input $\boldsymbol{x}_i \in R^d$ and the output y_i which is the binary class labels such that $y_i \in \{-1, +1\}$. The LS-SVM classifier takes the form

$$f(\boldsymbol{x}) = \mathbf{sgn}(\boldsymbol{w}'\phi(\boldsymbol{x}) + b)$$

where the term b is a bias term. Here the feature mapping function $\phi(\cdot): R^d \rightarrow R^{d_f}$ maps the input space to the higher dimensional feature space where the dimension d_f is defined in an implicit way.

The optimization problem is defined with a regularization parameter C as

$$\text{Minimize} \quad \frac{1}{2}\boldsymbol{w}'\boldsymbol{w} + \frac{C}{2}\sum_{i=1}^{N} e_i^2 \tag{1}$$

over $\{\boldsymbol{w}, b, \boldsymbol{e}\}$ subject to equality constraints

$$y_i(\boldsymbol{w}'\phi(\boldsymbol{x}_i) + b) = 1 - e_i, \quad i = 1, \cdots, N.$$

The Lagrangian function can be constructed as

$$L(\boldsymbol{w}, b, e; \alpha) = \frac{1}{2}\boldsymbol{w}'\boldsymbol{w} + \frac{C}{2}\sum_{i=1}^{N} e_i^2 - \sum_{i=1}^{N}\alpha_i(y_i(\boldsymbol{w}'\phi(\boldsymbol{x}_i) + b) - 1 + e_i) \tag{2}$$

where α_i's are the Lagrange multipliers. The conditions for optimality are given by

$$\frac{\partial L}{\partial \boldsymbol{w}} = \mathbf{0} \rightarrow \boldsymbol{w} = \sum_{i=1}^{N} \alpha_i y_i \phi(\boldsymbol{x}_i)$$
$$\frac{\partial L}{\partial b} = 0 \rightarrow \sum_{i=1}^{N} \alpha_i y_i = 0$$
$$\frac{\partial L}{\partial e_i} = 0 \rightarrow \alpha_i = C e_i, \quad i = 1, \cdots, N$$
$$\frac{\partial L}{\partial \alpha_i} = 0 \rightarrow y_i(\boldsymbol{w}'\phi(\boldsymbol{x}_i) + b) - 1 + e_i = 0, \quad i = 1, \cdots, N,$$

with solution

$$\begin{bmatrix} 0 & \boldsymbol{y}' \\ \boldsymbol{y} & \Omega + C^{-1} I \end{bmatrix} \begin{bmatrix} b \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{1} \end{bmatrix} \tag{3}$$

with $\boldsymbol{y} = (y_1, \cdots, y_N)'$, $\mathbf{1} = (1, \cdots, 1)'$, $\boldsymbol{\alpha} = (\alpha_1, \cdots, \alpha_N)'$, and $\Omega = \Omega_{kl}$ where $\Omega_{kl} = y_k y_l \phi(\boldsymbol{x}_k)'\phi(\boldsymbol{x}_l) = y_k y_l K(\boldsymbol{x}_k, \boldsymbol{x}_l)$, $k, l = 1, \cdots, N$, which are obtained from the application of Mercer's conditions(1909). Several choices of the kernel $K(\cdot, \cdot)$ are possible. For Examples,

$$K(\boldsymbol{x}_k, \boldsymbol{x}_l) = (\boldsymbol{x}_k'\boldsymbol{x}_l),$$
$$K(\boldsymbol{x}_k, \boldsymbol{x}_l) = (\boldsymbol{x}_k'\boldsymbol{x}_l + 1)^d,$$
$$K(\boldsymbol{x}_k, \boldsymbol{x}_l) = \exp\{-\|(\boldsymbol{x}_k - \boldsymbol{x}_l)\|^2/(2\sigma^2)\}, \text{ and}$$
$$K(\boldsymbol{x}_k, \boldsymbol{x}_l) = \tanh\{k\boldsymbol{x}_k'\boldsymbol{x}_l + \theta\}.$$

are can be used as kernel function.

Solving the linear equation (3) the optimal bias and Lagrange multipliers, $\hat{b}$ and $\hat{\alpha}_i$'s can be obtained, then the optimal target value for the given $\boldsymbol{x}$ is obtained as

$$f(\boldsymbol{x}) = \mathbf{sgn}\left(\sum_{i=1}^{N} \hat{\alpha}_i y_i K(\boldsymbol{x}, \boldsymbol{x}_i) + \hat{b}\right). \tag{4}$$

Note that in the nonlinear setting, the optimization problem corresponds to finding the flattest function in the feature space, not in the input space.

3. On line LS-SVM for classification

Consider that we have built LS-SVM model based on first N data and that now the new data $\{\boldsymbol{x}_{N+1}, y_{N+1}\}$ is coming in. Denote the equation (3) by $A_N \boldsymbol{a}_N = R_N$, where the subscript N indicates that the current model is based on the first N pairs of data. Then the optimal Lagrange multipliers and bias based on first N pairs of data are obtained from $\boldsymbol{a}_N = A_N^{-1} R_N$ such that $\boldsymbol{a}_N = (\hat{b}, \hat{\alpha}_1, \hat{\alpha}_2, \cdots, \hat{\alpha}_N)'$.

For $N+1$ pairs of data, we have

$$a_{N+1} = A_{N+1}^{-1} R_{N+1} \tag{5}$$

where

$$A_{N+1} = \begin{bmatrix} A_N & b_1 \\ b_2 & c \end{bmatrix}, \quad b_2 = b_1', \quad c = K(\boldsymbol{x}_{N+1}, \boldsymbol{x}_{N+1}) + \frac{1}{C}, \quad R_{N+1} = \begin{bmatrix} R_N \\ 1 \end{bmatrix} \text{ and}$$

$$b_1 = (y_{N+1}, y_1 y_{N+1} K(\boldsymbol{x}_1, \boldsymbol{x}_{N+1}), y_2 y_{N+1} K(\boldsymbol{x}_2, \boldsymbol{x}_{N+1}), \cdots, y_N y_{N+1} K(\boldsymbol{x}_N, \boldsymbol{x}_{N+1}))' .$$

We can have a inverse matrix of the form

$$A^{-1} = \begin{bmatrix} [A_{11} - A_{12} A_{22}^{-1} A_{21}^{-1}]^{-1} & A_{11}^{-1} A_{12} [A_{21} A_{22}^{-1} A_{12} - A_{22}]^{-1} \\ [A_{21} - A_{21} A_{11}^{-1} A_{12}]^{-1} A_{21} A_{11}^{-1} & [A_{22} - A_{21} A_{11}^{-1} A_{12}]^{-1} \end{bmatrix} \tag{6}$$

for some matrix $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ and

$$(A + BCD)^{-1} = A^{-1} - A^{-1} B (C^{-1} + DA^{-1} B)^{-1} DA^{-1} \tag{7}$$

from Muirhead(1982). According to the equation (6), the inverse matrix of A_{N+1} can be changed into

$$A_{N+1}^{-1} = \begin{bmatrix} A_N & b_1 \\ b_2 & c \end{bmatrix} = \begin{bmatrix} [A_N - \frac{1}{c} b_1 b_2]^{-1} & A_N^{-1} b_1 [b_2 A_N^{-1} b_1 - c]^{-1} \\ [b_2 A_N^{-1} b_1 - c]^{-1} b_2 A_N^{-1} & [c - b_2 A_N^{-1} b_2]^{-1} \end{bmatrix} \tag{8}$$

Applying the equation (5) to the upper left submatrix in the equation (8), we have

$$\left[A_N - \frac{1}{c} b_1 b_2\right]^{-1} = A_N^{-1} - A_N^{-1} b_1 \left[-c + b_2 A_N^{-1} b_1\right]^{-1} b_2 A_N^{-1} \tag{9}$$

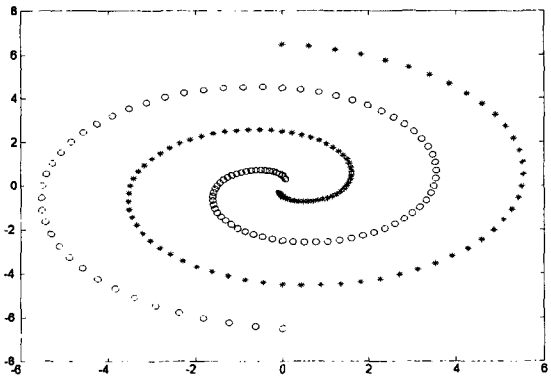
Let $\delta = \left[c - b_2 A_N^{-1} b_1\right]^{-1}$ then the equation (8) can be changed into

$$A_{N+1}^{-1} = \begin{bmatrix} A_N^{-1} & 0 \\ 0 & 1 \end{bmatrix} + \delta \begin{bmatrix} A_N^{-1} b_1 \\ -1 \end{bmatrix} \begin{bmatrix} b_2 A_N^{-1} & -1 \end{bmatrix} \tag{10}$$

With the equation (10) the inversion of matrix is computed through an incremental form, which avoids expensive inversion operation. Thus we can compute the equation (5) to get the optimal Lagrange multipliers $\hat{\alpha}_1, \hat{\alpha}_2, \cdots, \hat{\alpha}_{N+1}$ and bias $\hat{b}$ based on A_N and $\{x_{N+1}, y_{N+1}\}$without inverting A_{N+1} directly. Then we get the on line formulation for LS-SVM for classification.

4. Numerical studies

We illustrate the performance of the proposed algorithm through two data sets - two-spiral data set(simulated data) and Iris data set(real data). The radial basis kernel function with several values of bandwidth parameter σ and the regularization parameter C = 200 are used for the numerical studies. The data points are added one by one and the corresponding Lagrange multipliers and bias are updated every time for the test of incremental formulation in the equation (10).



[Figure 1] Plot of Two-Spiral data set.

The two-spiral data set is the benchmarking data set for the classification and known to be hard for the multi-layer perceptron. The training data with two classes indicated with

different labels are in two dimensional input space. The data set of 200 data points are generated as shown in figure 1. In figure 1, the points indicated by 'o' and '*' are the training data for the binary classes. Each 100 data points are used for the training data set and test data set respectively.

The Iris data set consists of four measurements made on each of 150 flowers. There are three pattern classes - Virginica, Setosa, and Versicolor - corresponding to three different types of Iris. In this case, the reference set consists of 150 feature vectors in 4 space each of which is assigned to one of three classes. In this numerical study, we choose 50 data points of the two classes - Virginica and Setosa - for the training data set and 50 data points for the test data set. Table 1 shows the number of misclassifications for both data sets by the on line and the batch LS-SVM according to the values of σ.

5. Remarks and Conclusions

From the numerical study, we can note the proposed on line classification algorithm derives the satisfying results, whose performance is comparative to the batch algorithm but avoiding a large scale matrix inversion operation, which is an attractive approach to modelling the training data set for large data set.

The on line LS-SVM classification example showed in this paper is limited for the case of binary classification. In future work, we intend to devise the on line LS-SVM applicable to the multi class problem.

[Table 1] The number of misclassifications.

data set	σ	0.2	0.5	0.75	1.0
Spiral data	On line	0	0	0	0
	Batch	0	0	0	2
Iris data	On line	1	0	0	0
	Batch	0	0	0	0

References

[1] Ahmed, S. N., Liu, H. and Sung, K. K. (1999). Incremental Learning with Support Vector Machines, *International Joint Conference on Artificial Intelligence*(IJCAI99), Workshop on Support Vector Machines, Stockholm, Sweden.

[2] Cauwenberghs, G. and Poggio, T. (2001). Incremental and decremental support vector machine learning, In Leen, T. K., Dietterich, T. G. and Tresp, V., editors, *Advances in Neural Information Processing Systems 13* : 409-415, MIT Press.

[3] Friess, T. and Cristianini, N. (1998). The Kernel-Adatron: A Fast and Simple Learning Procedure for support vector machines, *Proceeding of the Fifteenth International Conference on Machine Learning* (ICML), 188-196.

[4] Mercer, J. (1909). Functions of Positive and Negative Type and Their connection with Theory of Integral Equations, *Philosophical Transactions of Royal Society*, A:415-446.

[5] Muirhead, R. B.(1982) *Aspects of Multivariate Statistical Theory.*, John Wiley & Sons, Inc.

[6] Suykens, J.A.K. and Vanderwalle, J. (1999). Least Square Support Vector Machine Classifier, *Neural Processing Letters*, 9, 293-300.

[7] Suykens J.A.K., De Brabanter J., Lukas L., Vandewalle J. (2002). Weighted least squares support vector machines : robustness and sparse approximation, *Neurocomputing, Special issue on fundamental and information processing aspects of neurocomputing*, vol. 48, no. 1-4, 85-105.

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

[9] Vapnik, V. N. (1998). *Statistical Learning Theory*. Springer-Verlag, New York.

[Received May 2003, Accepted July 2003]