Journal of Korean
Data & Information Science Society
2003, Vol. 14, No.4 pp. 965~974

Incremental Multi-classification by Least Squares Support Vector Machine[1)]

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

Abstract

In this paper we propose an incremental classification of multi-class data set by LS-SVM. By encoding the output variable in the training data set appropriately, we obtain a new specific output vectors for the training data sets. Then, online LS-SVM is applied on each newly encoded output vectors. Proposed method will enable the computation cost to be reduced and the training to be performed incrementally. With the incremental formulation of an inverse matrix, the current information and new input data are used for building another new inverse matrix for the estimation of the optimal bias and lagrange multipliers. Computational difficulties of large scale matrix inversion can be avoided. Performance of proposed method are shown via numerical studies and compared with artificial neural network.

Keywords : Multi-class, LS-SVM, Incremental training, Neural network, Classification, Inverse matrix.

1. Introduction

In recent years, chiefly as a consequence of the development of the electronic computer, considerable advances have been made in the practical application of statistical classification analysis. Classification analysis is a multivariate technique concerned with assigning an object into one of several possible categories.

Machine learning based method like support vector machine(SVM) introduced by Vapnik(1995, 1998) are competing modern technology to traditional statistical methods. Despite of many successful application of SVM in the area of statistical

---

1) This research was supported by Catholic University of Daegu research Grants in 2003.
2) Professor, Department of Statistical Information, Catholic University of Daegu,
3) Adjucnt Professor, Department of Statistical Information, Catholic University of Daegu,
4) Professor, Department of Statistical Information, Catholic University of Daegu,

classification, a modified version of SVM was proposed by Suykens and Vanderwalle(1999a) in a least squares sense(LS-SVM) for classification. In LS-SVM the solution is given by a linear system instead of a quadratic programming problem. But their LS-SVM algorithm has also some limit. The data should be trained in batch form, which is not suited to the real application such as an online system identification and control. So incremental online training for the classification is needed urgently in real data application where the data come in sequentially.

On the while 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 these 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 and Cristianini (1998) proposed a sequential gradient method for SVM, where the main problem is that the training is not convergent quickly. Scholkopf et al.(1995) used voting scheme methods based on combining many binary class SVM's for the multi-class pattern recognition. Weston and Watson(1999) generalized the binary class SVM to the multi-class SVM without using combination of binary class SVM's. Suykens and Vandewalle(1999b) proposed a multi-class LS-SVM corresponding to a set of linear equations composed of newly encoded training data sets.

In this paper we propose the exact online(incremental) training method for multi-class LS-SVM. We devide the set of linear equations composed of newly encoded training data sets into a specific number of sets of linear equations and apply online LS-SVM on each set of linear equations.

The rest of paper is organized as follows. In Section 2, we examine the multi-classification method based on LS-SVM. In Section 3, we propose the incremental multi-class LS-SVM. In Section 4 we perform the numerical studies with real data sets. Finally we have concluding remarks in Section 5.

2. Multi-classification by LS-SVM

For the classification of multi-class data sets, let us denote the given training data set of N observation by $\{y_i, \boldsymbol{x}_i\}$, $i=1,\cdots,N$. Here $\boldsymbol{x}_i$ is a multivariate input and y_i is a single output which can takes a value between 0 and $q-1$, where q is the number of classes. For the binary class case, the output can takes only two values like 0 and 1.

For given class number q, we can always find a number m which satisfies

$q \in [2^{m-1}+1, 2^m]$. In order to use LS-SVM for multi-classification, we transform the output y_i, $i=1,\cdots,N$ into a row vector $\boldsymbol{y}_{(i)}{}' = (y_{i1},\cdots,y_{im})$ of m dimension where y_{ik} satisfies $y_i = \sum_{k=1}^{m} y_{ik} 2^{k-1}$. Actually y_{ik}, $i=1,\cdots,N$, $k=1,\cdots,m$ is 0 or 1. With these transformed vectors $\boldsymbol{y}_{(i)}{}'$, $i=1,\cdots,N$, we can construct a matrix Y of size $N \times m$, that is, i-th row of Y is $\boldsymbol{y}_{(i)}{}'$ which consists of $0'$s or $1'$s corresponding to the output y_i. And then, we have another transformation which makes the elements of the matrix Y of $0'$s into $-1'$s. This transformation can make us to apply the binary classification procedure by LS-SVM to multi-classification. Column vectors of Y is expressed as $\boldsymbol{y}^{(k)}$, $k=1,\cdots,m$ to avoid confusion. So the matrix Y can be represented as

$$Y = (\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2)}\cdots, \boldsymbol{y}^{(m)}) = \begin{pmatrix} \boldsymbol{y}_{(1)}{}' \\ \boldsymbol{y}_{(2)}{}' \\ \cdots \\ \boldsymbol{y}_{(N)}{}' \end{pmatrix} = \begin{pmatrix} y_{11} & y_{12} & \cdots & y_{1m} \\ y_{21} & y_{22} & \cdots & y_{2m} \\ \vdots & \vdots & \cdots & \vdots \\ y_{N1} & y_{N2} & \cdots & y_{Nm} \end{pmatrix}.$$

Suykens and Vandewalle(1999b) suggested multi-class LS-SVM based on the formulation

$$\text{Minimize } \frac{1}{2} \sum_{k=1}^{m} \boldsymbol{w}_k{}' \boldsymbol{w}_k + \frac{C}{2} \sum_{i=1}^{N} \sum_{k=1}^{m} e_{i,k}^2 \tag{1}$$

over $\{\boldsymbol{w}_k, b_k, e_{i,k}\}$ subject to equality constraints

$$y_{i,k}(\boldsymbol{w}_k{}' \phi_k(\boldsymbol{x}_i) + b_k) = 1 - e_{i,k}, \quad i=1,\cdots,N, \quad k=1,\cdots,m,$$

where $\boldsymbol{w}_k$ is a weight vector corresponding to the input vectors $\boldsymbol{x}_i$, $i=1,\cdots,N$ and kth column vector $\boldsymbol{y}^{(k)}$ of Y. Bias and errors corresponding to the input vectors $\boldsymbol{x}_i$, $i=1,\cdots,N$ and kth column vector $\boldsymbol{y}^{(k)}$ are denoted by b_k and $e_{i,k}$ respectively. $\phi_k(\boldsymbol{x}_i)$ is the nonlinear feature mapping function of $\boldsymbol{x}_i$, $i=1,\cdots,N$ associated with $\boldsymbol{y}^{(k)}$ and C is a regularization parameter. The lagrangian function can be constructed as

$$\begin{aligned} L(\boldsymbol{w}_k, b_k, e_{i,k}; \alpha_{i,k}) &= \frac{1}{2} \boldsymbol{w}_k{}' \boldsymbol{w}_k + \frac{C}{2} \sum_{i=1}^{N} \sum_{k=1}^{m} e_{i,k}^2 \\ &\quad - \sum_{i=1}^{N} \sum_{k=1}^{m} \alpha_{i,k} \{y_{ik}(\boldsymbol{w}_k{}' \phi_k(\boldsymbol{x}_i) + b_k) - 1 + e_{i,k}\} \end{aligned} \tag{2}$$

where $\alpha_{i,k}{}'$s are the lagrange multipliers. The conditions for optimality are given

by

$$\frac{\partial L}{\partial \boldsymbol{w}_k} = 0 \rightarrow \boldsymbol{w}_k = \sum_{i=1}^{N}\sum_{k=1}^{m} \alpha_{i,k}\phi_k(\boldsymbol{x}_i), k = 1.\cdots, m$$

$$\frac{\partial L}{\partial b_k} = 0 \rightarrow \sum_{i=1}^{N} \alpha_{i,k} y_{ik} = 0, k = 1, \cdots, m$$

$$\frac{\partial L}{\partial e_{i.k}} = 0 \rightarrow \alpha_{i,k} = Ce_{i,k}, \quad i = 1, \cdots, N, \; k = 1, \cdots, m$$

$$\frac{\partial L}{\partial \alpha_{i,k}} = 0 \rightarrow y_{ik}(\boldsymbol{w}_k{}'\phi_k(\boldsymbol{x}_i) + b_k) - 1 + e_{i,k} = 0\,, \quad i = 1, \cdots, N, \; k = 1, \cdots, m$$

and we have a solution

$$\begin{bmatrix} 0_{m\times m} & Y_m{}' \\ Y_m & \Omega_m + C^{-1}\boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{\alpha} \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}. \tag{3}$$

Here, $\boldsymbol{\alpha} = (\boldsymbol{\alpha}_1{}', \cdots, \boldsymbol{\alpha}_m{}')'$ and $\boldsymbol{b}' = (b_1, b_2, \cdots, b_m)$ denote the solutions of (3) and

$$Y_m = \text{Blockdiag}\{\boldsymbol{y}^{(1)}, \boldsymbol{y}^{(2),}\cdots, \boldsymbol{y}^{(m)}\} = \begin{pmatrix} \boldsymbol{y}^{(1)} & 0 & \cdots & 0 \\ 0 & \boldsymbol{y}^{(2)} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & \boldsymbol{y}^{(m)} \end{pmatrix}_{Nm\times m}$$

and

$$\Omega_m = \text{Blockdiag}\{\Omega_{(1)}, \cdots, \Omega_{(m)}\} = \begin{pmatrix} \Omega_{(1)} & 0 & \cdots & 0 \\ 0 & \Omega_{(2)} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & \Omega_{(m)} \end{pmatrix}_{Nm\times Nm}$$

where $\Omega_{(k)}$ is a $N\times N$ matrix with (i,j)th element by $y_{ik}y_{jk}K_k(\boldsymbol{x}_i, \boldsymbol{x}_j)$, $i, j = 1, \cdots, N$, $k = 1, \cdots, m$ which is calculated based on kth kernel function $K_k(\cdot, \cdot)$. Then for given testing input vector $\boldsymbol{x}_t$, it can be classified as y_t by

$$y_t = \sum_{k=1}^{m} y_{tk} \times 2^{k-1}$$

where

$$y_{tk} = 0.5 \times (sign(\sum_{i=1}^{N} \alpha_{i,k} y_{ik} K_k(\boldsymbol{x}_i, \boldsymbol{x}_t)) + b_k + 1), k = 1, \cdots, m.$$

3. Online multi-class LS-SVM

Assume that we have built LS-SVM model based on the first n data for each data set $\{ \boldsymbol{y}_{(i)}, \boldsymbol{x}_i \}$, $i=1,\cdots,n$ where $\boldsymbol{y}_{(i)}$ is defined in the previous section. To apply the multi-class LS-SVM to online LS-SVM, we consider the division of linear system (3) into m sets of linear systems (8).

$$\begin{bmatrix} 0 & \boldsymbol{y}_{(k)}' \\ \boldsymbol{y}_{(k)} & \Omega_{(k)} + C_k^{-1} \boldsymbol{I} \end{bmatrix} \begin{bmatrix} b_k \\ \boldsymbol{\alpha}_k \end{bmatrix} = \begin{bmatrix} 0 \\ \mathbf{1} \end{bmatrix}, k=1,\cdots,m \tag{4}$$

By considering one linear system (3) to m linear systems (4), the regularization parameter $C_k, k=1,\cdots,m$ can be adjusted for each set of linear equations, which can provide better results than equation (3).

Suppose that now the new data $\{y_{n+1}, \boldsymbol{x}_{n+1}\}$ is coming in. Denote the equation (4) by $A_n^{(k)} \boldsymbol{p}_n^{(k)} = R_n^{(k)}, k=1,\cdots,m$, 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

$$\boldsymbol{p}_n^{(k)} = A_n^{(k)-1} R_n^{(k)}, \quad k=1,\cdots,m$$

such that $\boldsymbol{p}_n = (b_k, \alpha_{1,k}, \alpha_{2,k}, \cdots, \alpha_{n,k})'$. For $(n+1)$th pairs of data, we have

$$\boldsymbol{p}_{n+1}^{(k)} = A_{n+1}^{(k)\,-1} R_{n+1}^{(k)} \ , \quad k=1,\cdots,m \tag{5}$$

where $k=1,\cdots,m$

$$A_{n+1}^{(k)\,-1} = \begin{bmatrix} A_n^{(k)} & \boldsymbol{d}_1 \\ \boldsymbol{d}_1' & u \end{bmatrix}^{-1}, \tag{6}$$

$$u = K_k(\boldsymbol{x}_{n+1}, \boldsymbol{x}_{n+1}) + \frac{1}{C_k}, \quad R_{n+1}^{(k)} = \begin{bmatrix} R_n^{(k)} \\ 1 \end{bmatrix}$$

and

$$\boldsymbol{d}_1 = \{ y_{1k} y_{(n+1)k} K_k(\boldsymbol{x}_1, \boldsymbol{x}_{n+1}), \cdots, y_{nk} y_{(n+1)k} K_k(\boldsymbol{x}_n, \boldsymbol{x}_{n+1}) \}'$$

for $k=1,\cdots,m$. We have two famous inverse equations for matrix

$$A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$$

$$A^{-1}=\begin{bmatrix}[A_{11}-A_{12}A_{22}^{-1}A_{21}^{-1}]^{-1} & A_{11}^{-1}A_{12}[A_{21}A_{22}^{-1}A_{12}-A_{22}]^{-1}\\ [A_{21}-A_{21}A_{11}^{-1}A_{12}]^{-1}A_{21}A_{11}^{-1} & [A_{22}-A_{21}A_{11}^{-1}A_{12}]^{-1}\end{bmatrix} \tag{7}$$

and

$$(A+BCD)^{-1}=A^{-1}-A^{-1}B(C^{-1}+DA^{-1}B)^{-1}DA^{-1} \tag{8}$$

According to (7), the equation (6) can be changed into

$$\begin{aligned}A_{n+1}^{(k)\,-1}&=\begin{bmatrix}A_n^{(k)} & \boldsymbol{d}_1\\ \boldsymbol{d}_1' & u\end{bmatrix}^{-1}\\ &=\begin{bmatrix}[A_n^{(k)}-\frac{1}{u}\boldsymbol{d}_1\boldsymbol{d}_1']^{-1} & A_n^{(k)-1}\boldsymbol{d}_1[\boldsymbol{d}_1'A_n^{(k)-1}\boldsymbol{d}_1-u]^{-1}\\ [\boldsymbol{d}_1'A_n^{(k)-1}\boldsymbol{d}_1-u]^{-1}\boldsymbol{d}_1'A_n^{(k)-1} & [u-\boldsymbol{d}_1'A_n^{(k)-1}\boldsymbol{d}_1']^{-1}\end{bmatrix}\end{aligned} \tag{9}$$

for $k=1,\cdots,m$. Applying the equation (8) to the upper left submatrix in the equation(9), we have

$$\begin{aligned}&[A_n^{(k)}-\frac{1}{u}\boldsymbol{d}_1\boldsymbol{d}_1']^{-1}\\ &=A_n^{(k)-1}-A_n^{(k)-1}\boldsymbol{d}_1[-u+\boldsymbol{d}_1'A_n^{(k)-1}\boldsymbol{d}_1]^{-1}\boldsymbol{d}_1'A_n^{(k)-1}\end{aligned} \tag{10}$$

for $k=1,\cdots,m$. Let $\delta_{(k)}=[u-\boldsymbol{d}_1'A_n^{(k)-1}\boldsymbol{d}_1]^{-1}$ then the equation (6) can be changed into

$$A_{n+1}^{(k)\,-1}=\begin{bmatrix}A_n^{(k)-1} & 0\\ 0 & 1\end{bmatrix}+\delta_{(k)}\begin{bmatrix}A_n^{(k)-1}\boldsymbol{d}_1\\ -1\end{bmatrix}[\boldsymbol{d}_1'A_n^{(k)-1}-1] \tag{11}$$

for $k=1,\cdots,m$. With the equation (11) 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 $\alpha_{1,k}, \alpha_{2,k},\cdots, \alpha_{n+1,k}$ and bias b_k for $k=1,\cdots,m$ based on $A_n^{(k)}$ and $\{y_{n+1}, \boldsymbol{x}_{n+1}\}$ without inverting $A_{n+1}^{(k)}$ directly. Then we get the online formulation for multi-class LS-SVM for each new data set. After completion of new coming data set, we have trained multi-class LS-SVM which can be used to the classification of testing data sets.

4. Numerical studies

In this section we illustrate the performance of proposed online multi-classification algorithm. The two famous data sets - Iris data set and Wine data set were considered for the comparison. The gaussian kernel function with

several values of bandwidth parameter σ is used. The regularization parameter C_k's of equation (4) were chosen appropriately by prior guess and used for the numerical studies. The data points are added one by one and the corresponding the lagrange multipliers and bias are updated every time for the incremental formulation in the equation (5).

The Iris data set is the benchmarking data set for the classification. It 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. Since the number of classes q is 3, m should be equal to 2. So the two linear systems of (4) is used for proposed online multi-class LS-SVM. In this case, the reference set consists of 150 feature vectors in 4 dimension. Each of the data point is assigned to one of three classes. In this numerical study, we choose 75 data points of the three classes - Virginica, Setosa and Versicolor - for the training data set and remained 75 data points for the testing data set.

To visualize the training data set we restrict two features which contain the most information of the class, the petal width and petal length. The scatter plot of training data points is shown in figure 1. From the scatter plot of Iris data, we can conjecture class 2 and class 3 can't be separated by linear hyper plane which leads to the use of gaussian kernel function.

Figure 1. Iris data set : The scatter of training data points(Left) and testing data points(Right) for two selected features.

Table 1 shows the number of misclassifications for testing data set by the proposed online multi-class LS-SVM. Proposed method and batch method explained in section 2 provide exactly same results for the Iris data set, so we gave the result from the proposed method only. Comparisons are made according to the 4 values of σ^2 in gaussian kernel function and 5 values of regularization parameter C. In this case we used same regularization parameters $C = C_1 = C_2$ in order to compare both methods.

Table 1. The number of misclassifications for Iris data set

$C \setminus \sigma^2$	0.5	1.0	1.5	2.0
100	2	2	2	3
200	2	2	2	3
300	2	2	2	2
400	2	2	2	2
500	2	2	2	2

For another comparison of proposed method, we employ the artificial neural networks(ANN) of 1 hidden layer with 5, 10, 15, 20 and 25 nodes, respectively. We performed 20 runs for each ANN and obtained the smallest number of misclassifications and the largest number of misclassifications, which are shown in table 2. As seen in table 2 ANN does not provide stable solutions on this data set and all of the numbers of misclassifications are larger than those obtained by the proposed online multi-class LS-SVM in average sense.

Table 2. The number of misclassifications for the Iris data by ANN.

number\node	5	10	15	20	25
smallest	1	5	3	2	4
largest	9	16	13	9	17
average	5	10.5	8	5.5	10.5

The Wine data set is found in UCI machine learning repository, of which data are the results of a chemical analysis of wines grown in the same region in Italy but derived from three different cultivars. An analysis determined the quantities of 13 constituents found in each of the three types of wines, class 1 , 2 and 3 with 59, 71 and 48 data points, respectively. We divide the wine data set into two parts - 128 data points for the training data set and 58 data points for the testing data set. Similarly to the iris data we use gaussian kernel function. And ANNs of 1

hidden layer with 5, 10, 15, 20 and 25 nodes, respectively. are employed for the comparison. Table 3 shows the number of misclassifications for testing data set by the online multi-class LS-SVM and the batch multi-class LS SVM according to the values of σ^2 and $C = C_1 = C_2$. Both methods provide exactly same results also in Wine data set. As seen in table 4 ANN does not provide stable solutions on this data set also and most of numbers of misclassifications are larger than those obtained by the multi-class LS-SVM.

Table 3. The number of misclassifications for Wine data set.

$C \setminus \sigma^2$	0.5	1.0	1.5	2.0
100	2	1	0	0
200	2	1	1	1
300	2	2	3	3
400	2	2	3	3
500	2	2	3	3

Table 4. The number of misclassifications for Wine data set by ANN.

number\node	5	10	15	20	25
smallest	0	0	0	1	1
largest	11	12	13	12	10
average	5.5	6	6.5	6.5	5.5

5. Concluding Remarks

We proposed an online multi-classification by LS-SVM. Performance of proposed method are compared with batch multi-classification by LS-SVM and artificial neural networks, respectively. Through the numerical studies we found that the proposed algorithm derives the satisfying results, whose performance is reasonably well without running a large scale matrix inversion operation, which is attractive approach to modelling the training of large data set. Also hyper parameters in the propose method can be adjusted more efficiently than in a batch method proposed by Suykens and Vandewalle(1999b).

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. Scholkopf, B., Burge, C. and Vapnik, V. (1995). Extracting Support Data a Given Task. *Proceeding of First International Conference on Know1edge Discovery and Data Mining.*
6. Suykens, J.A.K. and Vanderwalle, J. (1999a). Least Square Support Vector Machine Classifier, *Neural Processing Letters*, 9, 293-300.
7. Suykens J.A.K. and Vandewalle, J.(1999b) Multi-class Least Squares Support Vector Machines, *Proceeding of the International Joint Conference on Neural Networks*(IJCNN'99), Washington DC, USA, July. 1999.
8. Suykens J.A.K., De Brabanter J., Lukas L. and 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.
9. Vapnik, V. N. (1995). The Nature of Statistical Learning Theory. *Springer,* New York.
10. Vapnik, V. N. (1998). Statistical Learning Theory. *Springer,* New York.
11. Weston, J. and Watson, C.(1999). Support Vector Machines for multi-class Pattern Recognition, *Proceeding of the Seventh European Symposium on Artificial Neural Networks.*

[received date : Aug. 2003, accepted date : Oct. 2003]