Nonlinear Proximal Support Vector Machine Classifiers Aiming At Large Scale Classification Problems

Xiaoming XU[1],Qiaolin YE[2] , Ning YE[2],Bo WU[2]

[1]College of Information Science and Technology,Nanjing University of Aeronautics & Astronautics,Nanjing,china

[2] School of Information technology,NanJing Forestry University,NanJing ,china

Abstract

In [1], Fung et al, had constructed by a very fast algorithm: PSVM classifier, which mainly makes use of the Sherman-Morrison-Woodbury (SWM) identity [1, 7, 8]. However, for one thing, when handling nonlinear problems, the matrix H in (1) always is of dimension $m \times m$, such that the SWM identity is of no use. For another, for large scale classification problems, its inversion is not feasible and it is not stored. Aiming at the orientation problems, proposed in this paper is new fast algorithm. Experimental results also show LPSVM is fast and feasible to solve large scale classification problems.

Keywords: large scale classification problems; inversion; conjugate gradient method;

1. Introduction

The central idea of Standard vector machines (SVMs)[3,4] is to find an optimal hyperplane which makes the classified examples,in two-class problems, be separated well and enables the distance to the hyperplane to be as large as possible at the same time, a solution of which is to construct a constrained quadratic programming problems [5].

Standard SVMs mostly solve a quadratic program that require considerably longer training time mainly spent on iterative times and time.In contrast,from [1,6],we know that the proximal support vector machine (PSVM) , were implemented wherein each class of points is assigned to the closest of two parallel planes (in input or feature space) that are pushed apart as far as possible. This formulation, which can also be interpreted as regularized least squares and considered in the much more general context of regularized networks leads to an extremely fast and simple algorithm for generating a linear or nonlinear classifier that merely requires the solution of a single system of linear equations[1,6] However, despite of the achieved successes, it still has some flaws. When handling large scale classification problems, nonlinear PSVM will inevitably lead to longer training time due to the fact that we must compute the inversion of the $m \times m$ matrix （$I/v + H'H$） in (1). Aiming at the orientation problems, proposed in this paper is new fast algorithm:a nonlinear Proximal Support Vector Machine Classifier Aiming At Large Scale Classification Problems (LPSVM), which is based on a conjugate gradent method[2].LPSVM avoids the

Proceedings of the 11th Joint Conference on Information Sciences (2008)
Published by Atlantis Press

computation of the inversion of （$I / v + H ' H$) directly and the multiplication between the matrix ,such that it is fast and feasible to solve large scale classification problems.

We now describe the Sherman-Morrison-Wood-bury (SWM) identity that will also be utilized in this paper as in [1,7,8]:

$$(I/v+HH')^{-1}=v(I-H(I/v+H'H)^{-1}H') \quad (1)$$

where v is a positive constant that correspond to the punishment coefficient C in this paper. H is an arbitrary $m \times n$ $(n \ll m)$ matrix.We can invert a $m \times m$ matrix of the formulation (1) into a $n \times n$ matrix by the identity.

However, in kernel feature space, H for PSVM is always a $m \times m$ matrix, such that we should use the left formula in (1) when solving nonlinear problems ,

Which decreases the computational times, especially including the multiplication between the matrices.

2． The Proximal SVM [1,9]

The central idea of SVMs can be expressed by a QP problem:

$$\begin{aligned} &\min \ \frac{1}{2}w'w \\ &subject \ \ to \ \ D(Aw-eb) \geq e \end{aligned} \quad (2)$$

where $D = \begin{bmatrix} y_1 & 0 & 0\ldots0 \\ 0 & y_2 & 0\ldots0 \\ \ldots\ldots & \ldots\ldots & \ldots\ldots \\ 0 & 0 & 0\ldots y_n \end{bmatrix}_{n \times n}$, y_i corresponds to class label of i -th sample, $A \in R^{m \times n}$ .But, in most cases, the formulation (2) has not feasible solution because some data is not separable linearly, such that one introduce ξ to relax the constraints in (2):

$$\begin{aligned} subject \ \ to \ \ D(Aw-eb)+\xi &\geq e \\ \xi &\geq 0 \end{aligned} \quad (3)$$

One minimizes from (2) and (3):

$$\begin{aligned} &\min \ \frac{1}{2}w'w+Ce'\xi \\ &subject \ \ to \ D(Aw-eb)+\xi \geq e \\ &\xi \geq 0 \end{aligned} \quad (4)$$

where the positive constant C determines the trade-off between the empirical error and the complexity term . The optimization problem (4) can also be substituted by (5) as in [7,8,11]:

$$\begin{aligned} &\min \ \frac{1}{2}(w'w+b^2)+\frac{1}{2}C\|\xi\|^2 \\ &subject \ \ to \ \ D(Aw-eb)+\xi \geq e \end{aligned} \quad (5)$$

Make a simple reformulation in (5), one gets PSVM problem:

$$\begin{aligned} &\min \ \frac{1}{2}(w'w+b^2)+\frac{1}{2}C\|\xi\|^2 \\ &subject \ \ to \ \ D(Aw-eb)+\xi = e \end{aligned} \quad (6)$$

Forming the Lagrangian of (6) with multiplier u :

$$\begin{aligned} L(w,b,\xi,u) &= \frac{C}{2}\|\xi\|^2 + \frac{1}{2}\left\|\begin{bmatrix} w \\ b \end{bmatrix}\right\|^2 \\ &-u'(D(Aw-eb)+\xi-e) \end{aligned} \quad (7)$$

and setting partial derivatives concerning the primal variables ,w, b, ξ ,equal to zero, we can get the KKT optimality conditions as follows:

$$\begin{aligned} w - A'D\,u &= 0 \\ b + e'D\,u &= 0 \\ C\xi - u &= 0 \\ D(Aw - eb) + \xi - e &= 0 \end{aligned} \qquad (8)$$

After handling the equations of (8), on e can get

$$u = (\frac{I}{C} + D(AA' + ee')D)^{-1}e = (W)^{-1}e \quad (9)$$

where $W = I/C + HH'$ .Here $H = D[A \;\; -e]$.

In kernel feature space, when making use of the Sherman-Morrison-Wood-bury (SWM) identity is used here, one can get

$$\begin{aligned} &H = D[K \;\; -e] \\ &u = (I/C + D(KK' + ee')D)^{-1}e = (I/C + HH')^{-1}e \\ &u = C(I - H(I/C + H'H)^{-1}H')e \\ &\;\;= C(I - H(W)^{-1}H')\,e \end{aligned} \qquad (10)$$

3. LPSVM In Kernel Feature space

However, in kernel feature space, H is always a $m \times m$ matrix (c.f. linear PSVM), such that the formulation (9) should be used in this paper ,which dec reases the computational times especially including the multiplication between the matrix.

The matrix W in (9) is of dimension $m \times m$, where m is the number of samples. For large scale classification problems, the m will be large, such that the inversion of the matrix is not feasible. To avoid solving the inversion of the W a conjugate gradient method is used [2], which solves $AX = B$ with $A \in R^{m \times m}$ symmetric positive definite and $B \in R^{m}$ as follows:

Conjugate Gradient Algorithm

```
$j = 1; x_0 = 0; it_0 = B;$
While $it_0 \neq 0$
    $j = j + 1;$
    If $j = 1$
        $\rho_1 = it_0$
    Else
        $\beta_j = it'_{j-1}\, it_{j-1} / r'_{j-2}\, r_{j-2}$
        $\rho j = it_j - 1 + \beta_j \rho_{j-1}$
    End
    $\kappa_j = it'_{j-1}\, r_{j-1} / \rho'_j\, A\rho_j$
    $x_j = x_{j-1} + \kappa_j \rho_j$
    $it_j = it_j - 1 - \kappa_j A\rho_j$
End
$x = x_j$                                        (11)
```

We know from [8], W is symmetric positive definite, so (9) can be solved as follows:

LPSVM algorithm

1. Compute matrix H , W .
2. Solve u from $Wu = e$ u using Conjugate Gradient Matlab Code.
3. Compute $b = -e'Du$

We clearly see from (11) that the matrix A is not stored, such that it enables solving the large scale classification problems.

A complete MATLAB code of nonlinear LPSVM is given. From the code, we find u value got by the function for conjugate gradient method, and the Inversion of A in code 3.1 is avoided to solve, such that it can solve large scale classification problems.

Code 3.1 LPSVM MATLAB Code

```
Function [u, bias] = lpsvm(K,D,C,tol,itmax)
  % LPSVM: nonlinear classification
  % [r gamma] = lpsvm (K,D,C);
     [m,n]=size (K); e=ones (m, 1);
     H=D*[K -e];
     A=speye(m)/C+H*H';
      [u,niter,flag]=solveCG(A,e,tol,itmax);①
     h=D*u;
     bias=-sum(h);
```

4. Experiments and analysis

In experiments, for each SVM, Gauss kernel is selected because of its better performance. The optimal C parameter is selected using 5-fold cross-validation over the range C= {1,10,…,∞} , where C is the punishment coefficient.

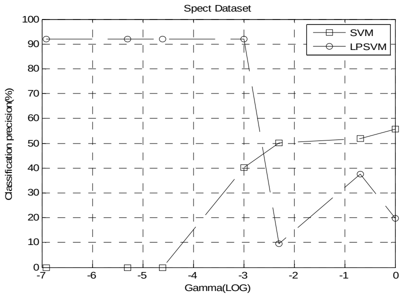
4.1. Comparison with LPSVM and SVM for classification

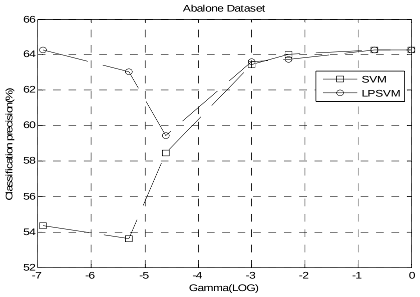
Choose different Kernel parameters Gamma(γ),we compare the testing correctness and running time between LPSVM and PSVM directly for classification.We carry out experiments on eight small datasets: Wine, Spect, Abalone, Yeast and three different Breast Cancer Wisconsin (Diagnostic) data sets from UCI [10].

The experimental results for testing set correctness under different γ 's are shown in Figure 1, and the best classification based on LPSVM and SVM are shown in Table 1. We clearly see from the results that the classification performance of the proposed LPSVM is comparable to that of SVM. Specifically, from the table 1 shown on Abalone, Yeast, WBCD, WPBC and WDBC data sets ,we can find that the best test set correctness when employing the proposed LPSVM are 64.27% , 69.11% , 100% ,88.66% and 94.05% , respectively, which are comparable to those based on SVM

In a word, it is easy to conclude the classification performance of the LPSVM cannot be lower than that of SVM.

[1] *The function for Conjugate Gradient algorithm can download from the website:http://matlabdb.mathematik.uni-stuttgart.de/download.jsp?MC_ID=3&SC_ID=5&MP_ID=168*





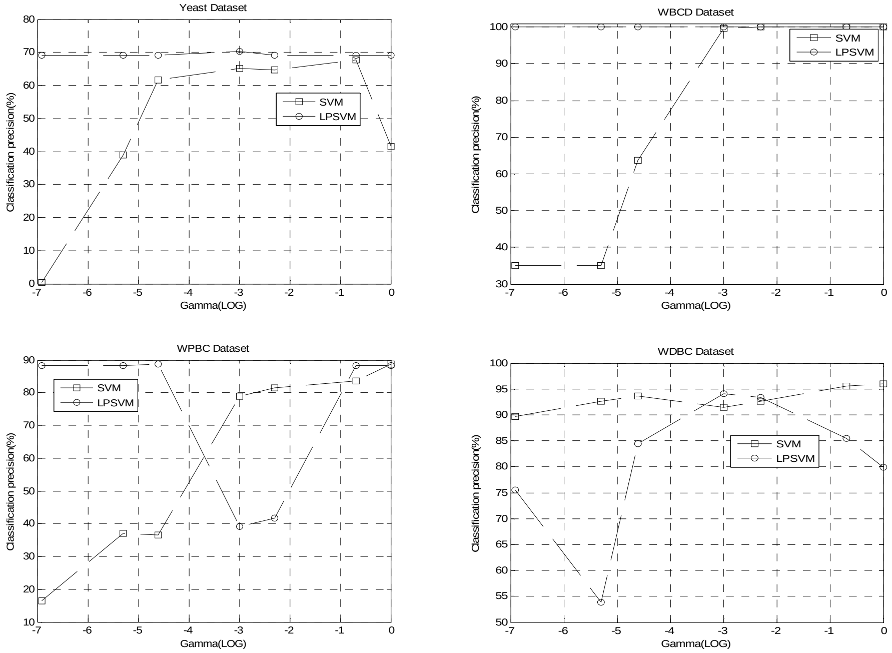


Fig 1: A comparison of the classification performance of the proposed LPSVM and SVM under different Kernel parameters γ 's.

Table 1: The best testing set correctness of the proposed LPSVM on six experiments and by SVM.

Dataset	SVM Test(%)	LPSVM Test(%)
Spect	55.61	91.98
Abalone	64.27	64.27
Yeast	67.68	69.11
WBCD	100	100
WPCD	88.66	88.66
WDBC	95.05	94.05

4.2. Comparison with LPSVM and SVM for Computational time

We report the experimental training time results of LPSVM and SVM on six datasets from[10] in Figure 2.From Figure 2,we can see,when employing LPSVM, that it can yield significantly less training times compared to those spent on standard SVM . For instance, on Abalone ,WBCD and WDBC data sets, SVM spends the least training times of 57.80 (sec.),4.7 (sec.) and 4.6 (sec.) , which are 83.71% ,60% and 61% higher than those of SVM.

As a result, we can conclude LPSVM is fast and feasible when handling large scale classifiation Problems.

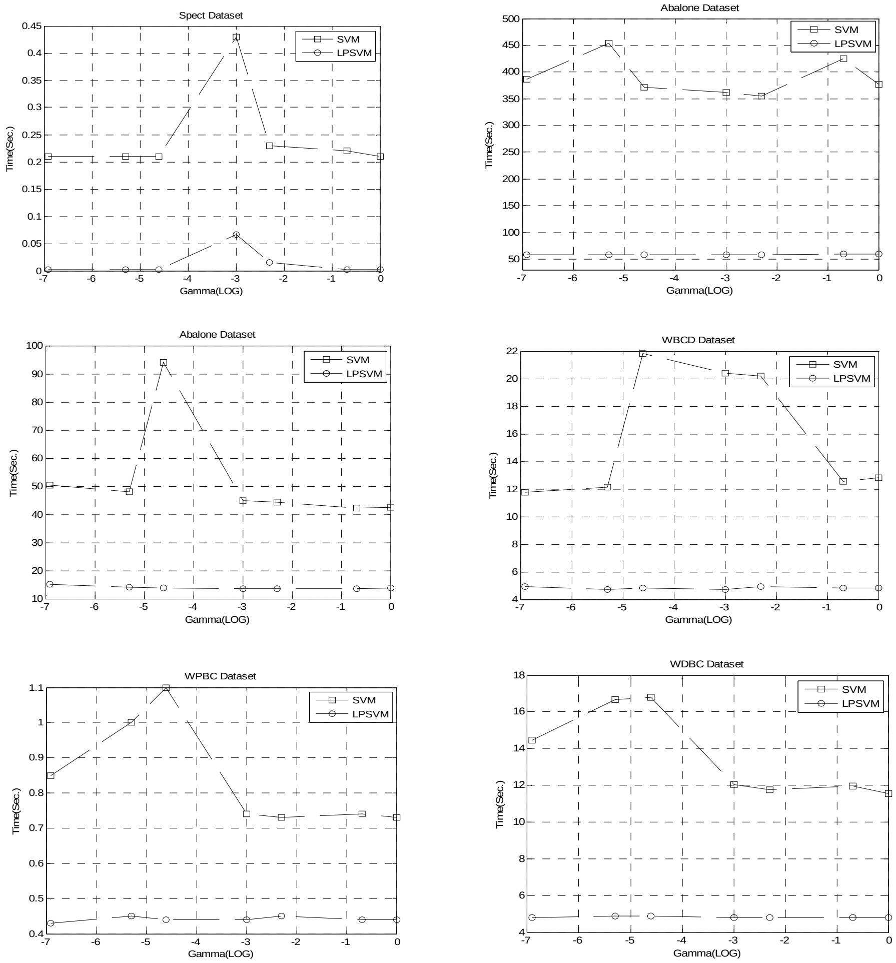


Fig 2: LPSVM and SVM training times using a nonlinear classifier on six datasets.

5. Conclusion and Further Work

PSVM mainly makes use of a property for the Sherman-Morrison-Wood-bury (SWM) identity [1, 7, 8]: a $m\times m$ matrix of the formulation (1) (i.e. $I/v+HH'$, where H is of dimension m × n) in (1) ,can be inverted into a $n\times n$ matrix (i.e. $I/v+H'H$).But,we know,in kernel feature space, H is always of dimension $m\times m$,such that the computational times will increase if we use PSVM algorithm for solving the support value u in (10). What is more, for large scale classification problems, its inversion is not feasible and it is not stored. Aiming at the orientation problems, we propose a new fast algorithm: a nonlinear Proximal Support Vector Machine Classifier Aiming At Large Scale classification Problems (LPSVM), which is based on a conjugate gradent method [2]. LPSVM solves a linear system instead of quadratic programming for SVM case. The performance of the classifiers are turned out to be comparable to SVM, and it is fast and feasible when handling large scale problems due to the fact that it avoids solving the inversion of the matrix W .

Our future work mainly includes the research in multicategory LPSVM, sparse approximation PSVM and the extension of its applicability.

Acknowledgement

The work is supported by the National Defence Basic Research Program in China No. S0500A001, the Program of the National "863" Project in China No. 2002AA411030, and the Scientific and Technological innovation Foundation of Jiangsu province in China.

References

[1] Fung, G. & Mangasarian,O. L. (2001). Proximal support vector machine classifiers. .In F. Provost& R. Srikant (Eds.), Proceedings KDD-2001: Knowledge discovery and data mining (pp. 77–86). San Francisco, CA, New York: Asscociation for Computing Machinery.

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

[3] Bradley, P. S.and Mangasarian, O.L... Massive data discrimination via linear support vector machines . Optimization Methods and Software, 13:1-10.

[4] Burges, C. J. C... A tutorial on support vector machines for pattern recgnition. Data Mining and Knowledge Discovery, 2(2):121-167,1998.

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

[6] Fung, G, M.,Mangasarian, O,L.(2005). Multicategory Proximal Support Vector Machine Classifiers .Machine Learning, Springer.

[7] Mangasarian , O. L. and Musicant, D. R. Active support vector machine classification. Technical Report 00-04, Data Mining Institute, Computer Sciences Department, University of Wisconsin, Madison, Wisconsin, April. Machine learning, to appear, 2000.

[8] Mangasarian , O. L. and Musicant, D.R.(2000). Lagrangian support vector machines. Technical Report 00-06, Data Mining Institute, Computer Sciences Department,University of Wisconsin, Madison, Wisconsin,.Journal of Machine Learning Research, to appear.

[9] Scholkopf, B, Smola,A,J. Optimization, and Beyond. Learning with Kernels MIT Press,2000.

[10] UCI Machine Learning Repository. http : //archive.ics.uci.edu/ml

[11]Mangasarian, O. L... Generalized support vector Machines. In A. Smola, P. Bartlett, B. Scholkopf, and D.Schuurmans, editors, Advances in Large Margin Classifiers,pages 135-146, Cambridge, MA.MIT Press.(2000)