An extended Lagrangian support vector machine for classifications*

YANG Xiaowei[1,2], SHU Lei[1], HAO Zhifeng[1], LIANG Yanchun[3]**, LIU Guirong[2] and HAN Xu[2]

(1. Department of Applied Mathematics, South China University of Technology, Guangzhou 510640, China; 2. Centre for ACES, Department of Mechanical Engineering, National University of Singapore, 119260, Singapore; 3. College of Computer Science and Technology, Jilin University, Changchun 130012, China)

Received May 9, 2003; revised November 21, 2003

Abstract Lagrangian support vector machine (LSVM) cannot solve large problems for nonlinear kernel classifiers. In order to extend the LSVM to solve very large problems, an extended Lagrangian support vector machine (ELSVM) for classifications based on LSVM and SVM^{light} is presented in this paper. Our idea for the ELSVM is to divide a large quadratic programming problem into a series of subproblems with small size and to solve them via LSVM. Since the LSVM can solve small and medium problems for nonlinear kernel classifiers, the proposed ELSVM can be used to handle large problems very efficiently. Numerical experiments on different types of problems are performed to demonstrate the high efficiency of the ELSVM.

Keywords: quadratic programming, support vector machine, decomposition algorithm, LSVM, ELSVM.

Training a support vector machine (SVM) is equivalent to solving a linearly constrained quadratic programming (QP) problem with a number of variables equal to the number of data points. This optimization problem is known to be challenging when the number of data points exceeds a few thousand. Decomposition algorithms are currently the major methods for solving support vector machines. An important issue in the solution process is the selection of the working set, and another one is solving the quadratic programming problem.

In the previous work, Osuna et al.[1] presented a decomposition algorithm and transformed the original quadratic programming into a series of quadratic programming subproblems. From the view of selecting the working set, Joachims[2] proposed an implementation of the decomposition algorithm based on Osuna's idea, which is called SVM^{light}. Platt[3] has also given a sequential minimal optimization (SMO) algorithm that breaks the large QP problems into a series of smallest possible QP subproblems, which can be solved analytically. Keerthi et al.[4] suggested some improvements to Platt's SMO algorithm for SVM classifier design. In order to improve the speed of SVM training without sacrificing the generality performance, Yang et al.[5] presented a preprocessing method based on the set segmentation and k-means clustering. As for the convergence of the decomposition algorithm for support vector machines, Lin[6,7] has given the proofs in detail.

In the decomposition algorithms mentioned above, the matrix that appeared in the dual objective function is not positive definite in general. In order to overcome this difficulty, Mangasarian et al.[8] proposed the Lagrangian support vector machine (LSVM). For a positive semidefinite nonlinear kernel, a single matrix inversion is required in the space of dimension equal to the number of data points classified. Hence, the LSVM cannot handle very large nonlinear classification problems efficiently. In this paper, in order to speed up the convergence of the SVM^{light} and extend LSVM to the very large nonlinear classification problems, an extended LSVM (ELSVM) for classifications is presented.

1 Generalized decomposition algorithm for SVM

In order to describe the problem clearly, some

* Supported by the National Natural Science Foundation of China (Grant No. 19901009), Key Project of Ministry of Education of China (Grant No. 02090), Natural Science Foundation of Guangdong Province, China (Grant No. 970472, 000463), Excellent Young Teachers Program of Ministry of Education of China, "Qian Bai Shi" Talent Young Foundation of Guangdong Province, Science Technology Development Project of Jilin Province of China (20030520) and Natural Science Foundation of South China University of Technology (E5121199)

** To whom correspondence should be addressed. E-mail: xiaoweialdong@163.net or ycliang@public.cc.jl.cn

notations are introduced first. Let l and n be the number and dimension of the training data points, respectively. A data point is denoted by vector $\boldsymbol{x}$ in the n-dimensional real space R^n, $\boldsymbol{x}_+$ denotes the vector in R^n whose negative components are set to zero. The notation $\boldsymbol{A} \in R^{l \times n}$ signifies a real $l \times n$ matrix. $\boldsymbol{A}_i$ and $\boldsymbol{A}_j$ denote the i-th row and j-th column of $\boldsymbol{A}$, respectively. According to the membership of each point $\boldsymbol{A}_i$ in the class $\boldsymbol{A}+$ or $\boldsymbol{A}-$, the component d_{ii} of the $l \times l$ diagonal matrix $\boldsymbol{D}$ is $+1$ or -1. A vector with elements 1 in a real space of arbitrary dimension is denoted by $\boldsymbol{e}$. An identity matrix of arbitrary dimension is denoted by $\boldsymbol{I}$. K and q represent a nonlinear kernel functional and the number of variables in the working set, respectively.

Suppose the training data are

$$(\boldsymbol{x}_1, y_1), \cdots, (\boldsymbol{x}_l, y_l), \quad (\boldsymbol{x}_i \in R^n, y_i \in \{+1, -1\}).$$

An optimal type hyperplane

$$(\boldsymbol{w} \circ \boldsymbol{x}) + b = 0 \tag{1}$$

can be obtained by solving the following optimal problem[9]

$$\min_{(\boldsymbol{w}, b, \xi_i) \in R^{n+1+l}} \Phi(\boldsymbol{w}, \xi) = \frac{1}{2}(\boldsymbol{w} \circ \boldsymbol{w}) + C\sum_{i=1}^{l} \xi_i \tag{2}$$

s. t.

$$y_i[(\boldsymbol{w} \circ \boldsymbol{x}_i) + b] \geqslant 1 - \xi_i, \quad i = 1, 2, \cdots, l, \tag{3}$$

where C is a regularization constant to control the compromise between maximizing the margin and minimizing the number of training set errors, ξ_i are some nonnegative slack variables.

The dual of this optimization problem is

$$\min_{\alpha_i \in R} W(\alpha) = \frac{1}{2}\sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j (\boldsymbol{x}_i \circ \boldsymbol{x}_j) - \sum_{i=1}^{l} \alpha_i \tag{4}$$

s. t.

$$0 \leqslant \alpha_i \leqslant C, \tag{5}$$

$$\sum_{i=1}^{l} \alpha_i y_i = 0. \tag{6}$$

Defining a matrix $\boldsymbol{Q}$ with components $Q_{ij} = y_i y_j (\boldsymbol{x}_i \circ \boldsymbol{x}_j)$, the QP problem (4) ~ (6) can be rewritten as

$$\min_{\alpha \in R^q} W(\alpha) = \frac{1}{2}\alpha^{\mathrm{T}} \boldsymbol{Q} \alpha - \boldsymbol{e}^{\mathrm{T}} \alpha \tag{7}$$

s. t.

$$\alpha^{\mathrm{T}} \boldsymbol{y} = 0, \tag{8}$$

$$0 \leqslant \alpha_i \leqslant C. \tag{9}$$

For the nonlinear kernel classifier, the matrix $\boldsymbol{Q}$ is defined by $Q_{ij} = y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j)$, and the other formulations are not changed.

Decomposing α into two vectors α_B and α_N, fixing α_N and allowing changes only in α_B, the following subproblem can be defined

$$\min_{\alpha_B \in R^q} W(\alpha_B) = \frac{1}{2}\alpha_B^{\mathrm{T}} \boldsymbol{Q}_{BB} \alpha_B + \alpha_N^{\mathrm{T}} \boldsymbol{Q}_{NB} \alpha_B + \frac{1}{2}\alpha_N^{\mathrm{T}} \boldsymbol{Q}_{NN} \alpha_N - \boldsymbol{e}_B^{\mathrm{T}} \alpha_B - \boldsymbol{e}_N^{\mathrm{T}} \alpha_N \tag{10}$$

s. t.

$$\alpha_B^{\mathrm{T}} \boldsymbol{y}_B + \alpha_N^{\mathrm{T}} \boldsymbol{y}_N = 0, \tag{11}$$

$$0 \leqslant \alpha_{Bi} \leqslant C. \tag{12}$$

Because the term $\frac{1}{2}\alpha_N^{\mathrm{T}} \boldsymbol{Q}_{NN} \alpha_N - \boldsymbol{e}_N^{\mathrm{T}} \alpha_N$ is a constant within the defined subproblem, the QP subproblem can be rewritten as

$$\min_{\alpha_B \in R^q} W(\alpha_B) = \frac{1}{2}\alpha_B^{\mathrm{T}} \boldsymbol{Q}_{BB} \alpha_B + \alpha_N^{\mathrm{T}} \boldsymbol{Q}_{NB} \alpha_B - \boldsymbol{e}_B^{\mathrm{T}} \alpha_B \tag{13}$$

s. t.

$$\alpha_B^{\mathrm{T}} \boldsymbol{y}_B + \alpha_N^{\mathrm{T}} \boldsymbol{y}_N = 0, \tag{14}$$

$$0 \leqslant \alpha_{Bi} \leqslant C. \tag{15}$$

To select the working set, Joachims[2] proposes a strategy based on Zoutendijk's method, which uses a first-order approximation to the target function. The idea is to find a steepest feasible direction $\boldsymbol{d}$ of descent, which has only q non-zero elements. The variables corresponding to these elements compose the current working set. This approach leads to the following optimization problem:

$$\min_{d} V(\boldsymbol{d}) = \boldsymbol{g}(\alpha)^{\mathrm{T}} \boldsymbol{d} \tag{16}$$

s. t.

$$\boldsymbol{y}^{\mathrm{T}} \boldsymbol{d} = 0, \tag{17}$$

$$d_i \geqslant 0 \quad \text{for} \quad i: \alpha_i = 0, \tag{18}$$

$$d_i \leqslant 0 \quad \text{for} \quad i: \alpha_i = C, \tag{19}$$

$$1 \leqslant d_i \leqslant 1 \quad \text{and} \quad |\{d_i: d_i \neq 0\}| = q, \tag{20}$$

where $\boldsymbol{g}(\alpha) = \boldsymbol{Q}\alpha - \boldsymbol{e}$, $|\{d_i: d_i \neq 0\}|$ denotes the number of the elements in the set $\{d_i: d_i \neq 0\}$.

2 Lagrangian support vector machine (LSVM)

Using 2-norm instead of 1-norm in the optimal formulation, Mangasarian et al.[8] gave the following reformulation of the SVM

$$\min_{(\boldsymbol{w}, b, \xi) \in R^{n+1+l}} \frac{1}{2}C\xi^{\mathrm{T}}\xi + \frac{1}{2}\boldsymbol{w}^{\mathrm{T}}\boldsymbol{w} + \frac{1}{2}b^2 \tag{21}$$

s. t.

$$\boldsymbol{D}(\boldsymbol{A}\boldsymbol{w} + \boldsymbol{e}b) + \xi > \boldsymbol{e}. \tag{22}$$

The dual of this problem is

$$\min_{0\leqslant\alpha\in R^l}\frac{1}{2}\alpha^T\left(\frac{I}{C}+D(AA^T+ee^T)D\right)\alpha-e^T\alpha. \tag{23}$$

Let

$$H=D[A\quad -e],\quad Q=\frac{I}{C}+HH^T. \tag{24}$$

Then the dual problem (23) becomes

$$\min_{0\leqslant\alpha\in R^l}\frac{1}{2}\alpha^TQ\alpha-e^T\alpha. \tag{25}$$

Let $\gamma=Q\alpha-e$. For the dual problem (25), the KKT necessary and sufficient optimality conditions are the following

$$\alpha\perp\gamma\quad(\alpha\geqslant 0,\ \gamma\geqslant 0), \tag{26}$$

where $\alpha\geqslant 0$ and $\gamma\geqslant 0$ mean that all of elements of the vectors α and γ are nonnegative.

By using the easily established identity between any two real numbers (or vectors), the optimality conditions (26) can be written in the following equivalent form for any positive β

$$Q\alpha-e=((Q\alpha-e)-\beta\alpha)_+. \tag{27}$$

These optimality conditions lead to the following simple iterative scheme, which constitutes the LSVM algorithm

$$\alpha^{i+1}=Q^{-1}(e+((Q\alpha^i-e)-\beta\alpha^i)_+),\quad i=0,1,\cdots, \tag{28}$$

It should be noticed that the LSVM uses the Sherman-Morrison-Woodbury identity[10], which enables us to invert a large $l\times l$ matrix of the form in

$$\left(\frac{I}{C}+HH^T\right)^{-1}=C\left(I-H\left(\frac{I}{C}+H^TH\right)^{-1}H^T\right) \tag{29}$$

by merely inverting a small $(n+1)\times(n+1)$ matrix. In order to guarantee the global linear convergence from any starting point, the following condition should be satisfied

$$0<\beta<\frac{2}{C}. \tag{30}$$

3 Extended Lagrangian support vector machine (ELSVM)

The SVM^{light} uses QP solver for QP subproblems. The LSVM is comparable to or better than other SVM training algorithms for linear classifier and small nonlinear one. But it cannot solve large nonlinear problems. To take advantage of both the SVM^{light} and LSVM, the ELSVM for classifications is presented in this section.

When the decomposition algorithm is applied to the QP problem (23), one can obtain

$$\begin{aligned}&\min_{0\leqslant\alpha\in R^l}\frac{1}{2}\alpha^T\left(\frac{I}{C}+D(AA^T+ee^T)D\right)\alpha-e^T\alpha\\&=\frac{1}{2}\frac{\alpha^T\alpha}{C}+\frac{1}{2}\alpha^TDAA^TD\alpha+\frac{1}{2}\alpha^TDee^TD\alpha-e^T\alpha\\&=\frac{1}{2}\frac{\alpha_B^T\alpha_B}{C}+\frac{1}{2}\frac{\alpha_N^T\alpha_N}{C}+\frac{1}{2}\alpha_B^TD_BA_BA_B^TD_B\alpha_B\\&\quad+\alpha_N^TD_NA_NA_B^TD_B\alpha_B+\frac{1}{2}\alpha_N^TD_NA_NA_N^TD_N\alpha_N\\&\quad+\frac{1}{2}\alpha_B^TD_Be_Be_B^TD_B\alpha_B+\alpha_N^TD_Ne_Ne_B^TD_B\alpha_B\\&\quad+\frac{1}{2}\alpha_N^TD_Ne_Ne_N^TD_N\alpha_N-e_B^T\alpha_B-e_N^T\alpha_N.\end{aligned} \tag{31}$$

Because the term $\frac{1}{2}\frac{\alpha_N^T\alpha_N}{C}+\frac{1}{2}\alpha_N^TD_NA_NA_N^TD_N\alpha_N+\frac{1}{2}\alpha_N^TD_Ne_Ne_N^TD_N\alpha_N-e_N^T\alpha_N$ is a constant within the defined subproblem, the subproblem can be rewritten as

$$\begin{aligned}\min_{0\leqslant\alpha_B\in R^q}&\frac{1}{2}\frac{\alpha_B^T\alpha_B}{C}+\frac{1}{2}\alpha_B^TD_BA_BA_B^TD_B\alpha_B\\&+\frac{1}{2}\alpha_B^TD_Be_Be_B^TD_B\alpha_B-(e_B^T-\alpha_N^TD_NA_NA_B^TD_B\\&-\alpha_N^TD_Ne_Ne_B^TD_B)\alpha_B.\end{aligned} \tag{32}$$

Let

$$H_B=D_B[A_B\quad -e_B],\quad Q_B=\frac{I_B}{C}+H_BH_B^T. \tag{33}$$

Then the dual subproblem (32) becomes

$$\min_{0\leqslant\alpha_B\in R^q}\frac{1}{2}\alpha_B^TQ_B\alpha_B-E_B^T\alpha_B, \tag{34}$$

where

$$E_B^T=e_B^T-\alpha_N^TD_NA_NA_B^TD_B-\alpha_N^TD_Ne_Ne_B^TD_B. \tag{35}$$

Therefore, Eq. (27) can be written as

$$Q_B\alpha_B-E_B=((Q_B\alpha_B-E_B)-\beta\alpha_B)_+ \tag{36}$$

and then

$$\alpha_B^{i+1}=Q_B^{-1}(E_B+((Q_B\alpha_B^i-E_B)-\beta\alpha_B^i)_+). \tag{37}$$

Let

$$G_B=[A_B\quad -e_B],$$
$$Q_B=\frac{I_B}{C}+D_BK(G_B,G_B^T)D_B. \tag{38}$$

For the nonlinear classifier, E_B in the iterative formulation (37) is in the following

$$E_B^T=e_B^T-\alpha_N^TD_NK\left([A_N,-e_N],\begin{bmatrix}A_B^T\\e_B^T\end{bmatrix}\right)D_B. \tag{39}$$

It should be noticed that the strategy selecting sets B and N is almost the same as that adopted in $\mathrm{SVM}^{\text{light}}$ during each iteration. The only difference is that in our algorithm, there are no equality constraints and upper boundary ones in solving the linear programming problem for selecting sets B and N.

Under the assumption that $\boldsymbol{Q}$ is positive definite, Lin[6] has proved that the algorithm of the $\mathrm{SVM}^{\text{light}}$ was asymptotic convergent. In the proposed algorithm, the assumption above is right. In the convergent proof of Ref. [8], replacing $\boldsymbol{Q}$, α^i, $\boldsymbol{I}$, $\boldsymbol{e}$ with $\boldsymbol{Q}_B$, α_B^i, $\boldsymbol{I}_B$, $\boldsymbol{E}_B$ respectively, we could have the same result:

$$\begin{aligned} &\| \boldsymbol{Q}_B \alpha_B^{i+1} - \boldsymbol{Q}_B \bar{\alpha}_B \| \\ &\quad \leqslant \| \boldsymbol{I}_B - \beta \boldsymbol{Q}_B^{-1} \| \circ \| \boldsymbol{Q}_B \alpha_B^i - \boldsymbol{Q}_B \bar{\alpha}_B \|, \end{aligned} \tag{40}$$

where α_B^i is the solution of Eq. (37) and $\bar{\alpha}_B$ the unique solution of Eq. (34). It shows that it is convergent to solve the subproblem (34) via the iteration formulation (37) at the linear rate.

4 Numerical implementation and comparison

The implementation of ELSVM is straightforward. Our experiments were run on a PC, which utilizes a GenuineIntel ~2000 MHz Pentium IV processor with a maximum of 256 MB of memory available. In order to test the speed and effectiveness of the presented algorithm on the linear problem and nonlinear ones, we apply $\mathrm{SVM}^{\text{light}}$, LSVM, and ELSVM on the same data sets available from the UCI Machine Learning Repository 1). All features for all experiments were normalized to the range [−1, +1]. We chose to use the default termination error criterion in $\mathrm{SVM}^{\text{light}}$ of 0.001 and $\beta = \frac{1.9}{C}$. The results for linear problems and those for nonlinear ones are shown in Tables 1 and 2, respectively.

In the linear experiments as shown in Table 1, all the data points are used. The results show that the running time of ELSVM is shorter than that of $\mathrm{SVM}^{\text{light}}$ and longer than that of LSVM for linear classification problems. Their training correctness is almost the same. The reason is that many efforts are spent in selecting the working set for $\mathrm{SVM}^{\text{light}}$ and ELSVM, which takes some time, and LSVM is an iterative method, which requires nothing more complex than inversion of a single matrix with the dimension of input space plus one.

Table 1. Comparison of results training $\mathrm{SVM}^{\text{light}}$, LSVM and ELSVM for linear classification problems

Data ($C=1/l$)	Size $l \times n$	$\mathrm{SVM}^{\text{light}}$		ELSVM		LSVM	
		Training accuracy (%)	Running time	Training accuracy (%)	Running time	Training accuracy (%)	Running time (s)
Ionosphere	351×34	83.23	0.610	77.85	0.110	78.48	0.01
Liver Disorders	345×6	70.72	50.963	68.41	3.254	68.99	0.02
Pima Diabetes	768×8	76.17	297.037	71.61	84.544	69.92	0.04
Tic-Tac-Toe Endgame	958×9	65.66	0.911	65.66	0.451	65.66	0.03
KRKPA7	3196×36	81.15	6.339	84.46	3.975	84.49	0.12

In the nonlinear experiments, the strategy of the tenfold cross validation is used in order to compare testing accuracy between the methodologies and the results are average value of 10 implements. 90% of all the data points were selected randomly for training and the remainings were used for testing. From Table 2, it can be seen that ELSVM is faster than $\mathrm{SVM}^{\text{light}}$ and LSVM cannot work for large nonlinear classification problems. In our algorithm, the QP subproblem is solved by LSVM and LSVM is faster than the regular QP solver for small and medium nonlinear classification problems. Therefore, ELSVM is faster than $\mathrm{SVM}^{\text{light}}$. For a positive semidefinite nonlinear kernel, a single matrix inversion is required in the space of dimension equal to the number of data points classified. Hence, LSVM cannot handle large nonlinear problems.

1) Murphy, P. M. et al. UCI repository of machine learning databases. 1992, Available at http://www.ics.uci.edu/~mlearn/MLRepository.html

Table 2. Comparison of results training SVM^{light}, LSVM and ELSVM for nonlinear classification problems

Data (C=1/ l) l× n	Algorithm	Kernel type	Training accuracy (%)	Testing accuracy (%)	Running time (s)
Ionosphere	SVM^{light}	Linear	83.23	94.29	0.610
351×34		Quadratic	94.62	94.29	1.963
	LSVM	Linear	78.48	82.86	0.010
		Quadratic	95.55	91.43	0.291
	ELSVM	Linear	77.85	82.86	0.110
		Quadratic	95.25	94.29	0.331
Tic-Tac-Toe	SVM^{light}	Linear	65.66	62.50	0.911
Endgame		Quadratic	75.64	75.00	5.067
958×9	LSVM	Linear	65.66	62.50	0.030
		Quadratic	75.87	75.00	3.535
	ELSVM	Linear	65.66	62.50	0.451
		Quadratic	75.99	73.96	5.038
KRKPA7	SVM^{light}	Linear	81.15	84.06	6.339
3196×36		Quadratic	92.90	91.56	28.281
	LSVM	Linear	84.49	85.31	0.120
		Quadratic	—	—	—
	ELSVM	Linear	84.46	83.75	3.975
		Quadratic	95.00	95.83	12.097
Chess-KRK	SVM^{light}	Linear	70.79	71.19	31.074
14001×6		RBF(σ=100)	70.97	71.24	293.672
	LSVM	Linear	—	—	—
		RBF(σ=100)	—	—	—
	ELSVM	Linear	70.77	71.23	23.681
		RBF(σ=100)	70.97	71.24	197.317

5 Conclusions

In this paper, an ELSVM for classification design is presented. The ELSVM is based on both SVM^{light} and LSVM and is easy to implement. In order to test the speed and effectiveness of ELSVM, six UCI data sets are tested. Simulation results show that ELSVM is faster than SVM^{light} for linear and large nonlinear classification problems. Numerical experiments also show that the LSVM is faster than ELSVM for linear classification problems, but for the large nonlinear classification problems, the LSVM fails to work. ELSVM extends the LSVM to very large data sets for nonlinear kernel classifications.

References

1 Osuna, E. et al. An improved training algorithm for support vector machines. Neural Networks for Signal Processing. Proceedings of the IEEE, 1997, 276.

2 Joachims, T. Making large-scale support vector machine learning practical. In: Schölkopf, B. et al. (Eds.), Advances in Kernel Methods-Support Vector Learning. Massachusetts: The MIT Press, 1999, 169.

3 Platt, J. C. Fast training of support vector machines using sequential minimal optimization. In: Schölkopf, B. et al. (Eds.), Advances in Kernel Methods-Support Vector Learning. Massachusetts: The MIT Press, 1999, 185.

4 Keerthi, S. S. et al. Improvements to Platt's SMO algorithm for SVM classifier design. Neural Computation, 2001, 13(3): 637.

5 Yang, X. W. et al. A fast SVM training algorithm based on the set segmentation and k-means clustering, Progress in Natural Science, 2003, 13(10): 750.

6 Lin, C. J. On the convergence of the decomposition method for support vector machines. IEEE Transactions on Neural Networks, 2001, 12(6): 1288.

7 Lin, C. J. Asymptotic convergence of an SMO algorithm without any assumptions. IEEE Transactions on Neural Networks, 2002, 13(1): 248.

8 Mangasarian, O. L. et al. Lagrangian support vector machines. Journal of Machine Learning Research, 2001, 1: 161.

9 Vapnik, V. N. The Nature of Statistical Learning Theory. New York: Springer Verlag, 1995.

10 Golub, G. H. et al. Matrix Computations (3rd edition). Maryland: the John Hopkins University Press, 1996.