Comparison of L1 and L2 Support Vector Machines

Yoshiaki Koshiba
Graduate School of Science and Technology
Kobe University
Rokkodai, Nada, Kobe, Japan
Email: koshiba@chevrolet.eedept.kobe-u.ac.jp

Shigeo Abe
Graduate School of Science and Technology
Kobe University
Rokkodai, Nada, Kobe, Japan
Email: abe@eedept.kobe-u.ac.jp

***Abstract*— In this paper, we compare L1 and L2 support vector machines from the standpoint of training time and the generalization ability. The generalization ability for seven benchmark data sets are almost the same but training time of L1-SVMs is usually shorter than that of L2-SVMs. We also compare the effect of the approximate KKT (Karush-Kuhn-Tucker) conditions using the bias term and the exact KKT conditions. According to the computer experiments, since the approximate KKT conditions give a conservative estimate of violating variables, training time using the approximate KKT conditions is usually shorter.**

I. Introduction

Support vector machines [1], [2], are widely used for pattern classification problems. The advantages of SVMs over conventional methods are high generalization ability especially when the number of training data is small, adaptability to various classification problems by changing kernel functions, and global optimal solution obtained by quadratic programming.

Support vector machines with linear sum of slack variables, which are commonly used, are called L1-SVMs, and SVMs with the square sum of slack variables are called L2-SVMs. Characteristics of both L1- and L2-SVMs have been studied analytically [3], [4], [5], [6], [7]. For instance, dependence of the solutions on the margin parameter C [3] and non-uniqueness of solutions [6], [7] are studied.

A support vector machine is trained by solving the associated dual problem by the quadratic programming technique. But since the number of variables is the number of training data, training time becomes very long for the large number of training data. To overcome this problem, the decomposition technique [9], [10] is usually used. The training is continued until the solution satisfies the Karush-Kuhn-Tucker (KKT) complementarity condition. But since the KKT condition includes a primal variable, detection of the variables that violate the KKT condition is inexact during training. To overcome this problem, in [11], the exact KKT condition is derived for the SMO (Sequential Minimal Optimization) technique.

In this paper, first we extend the exact KKT condition for the SMO to general training of L1- and L2-SVMs. Then by computer experiments, we compare L1- and L2-SVMs from the standpoint of training time and the generalization ability. Since the Hessian matrix of L2-SVMs is positive definite, the associated optimization problem is considered to be more computationally stable than that of L1-SVMs. Using some benchmark data sets, we show that this does not hold for most cases. Then, we compare the training time using the approximate and exact KKT conditions.

This paper is organized as follows. In Section II, we explain L1-SVMs and L2-SVMs. In Section III, we discuss the approximate and exact KKT conditions. Then, in Section IV, we discuss training of SVMs by decomposition techniques, and in Section V, we compare performance of L1- and L2-SVMs by computer simulations.

II. Support Vector Machines

In this section, we describe the theory of L1-SVMs and L2-SVMs for two-class problems.

A. L1 Support Vector Machines

Let training datum be $\mathbf{x}_i$ $(i = 1, ..., M)$ and its label be $y_i = 1$ if $\mathbf{x}_i$ belongs to Class 1, and $y_i = -1$ if Class 2. In SVMs, to enhance linear separability, the input space is mapped into a high dimensional feature space using the mapping function $\mathbf{g}(\mathbf{x})$.

To obtain the optimal separating hyperplane of the L1-SVM in the feature space, we consider the following optimization problem:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \parallel \mathbf{w} \parallel^2 + C \sum_{i=1}^{M} \xi_i, \\ \text{subject to} \quad & y_i(\mathbf{w}^t \mathbf{g}(\mathbf{x}_i) + b) \geq 1 - \xi_i, \\ \text{for} \quad & i = 1, ..., M, \end{aligned} \tag{1}$$

where $\mathbf{w}$ is a weight vector, C is the margin parameter that determines the tradeoff between the maximization of the margin and the minimization of the classification error, ξ_i $(i = 1, ..., M)$ are the nonnegative slack variables and b is a bias term. Introducing the Lagrange multipliers α_i, we obtain the following dual problem:

maximize

$$Q(\alpha) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{M} \alpha_i \alpha_j y_i y_j \mathbf{g}(\mathbf{x}_i)^t \mathbf{g}(\mathbf{x}_j),$$

$$\text{subject to} \quad \sum_{i=1}^{M} y_i \alpha_i = 0, \quad 0 \le \alpha_i \le C. \tag{2}$$

We use the mapping function that satisfies

$$H(\mathbf{x}, \mathbf{x}') = \mathbf{g}(\mathbf{x})^t \mathbf{g}(\mathbf{x}'), \tag{3}$$

where $H(\mathbf{x}, \mathbf{x}')$ is a kernel function. By this selection, we need not treat the variables in the feature space explicitly.

Solving the above dual problem, we obtain the decision function:

$$D(\mathbf{x}) = \sum_{i=1}^{M} \alpha_i^* y_i H(\mathbf{x}_i, \mathbf{x}) + b^*, \tag{4}$$

where an asterisk denotes the optimal solution.

B. L2 Support Vector Machines

L2-SVMs use the square sum of the slack variables ξ_i in the objective function instead of the linear sum of the slack variables. Thus we consider optimization problem as follows:

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} \parallel \mathbf{w} \parallel^2 + \frac{C}{2} \sum_{i=1}^{M} \xi_i^2, \\ \text{subject to} \quad & y_i(\mathbf{w}^t \mathbf{x}_i + b) \ge 1 - \xi_i, \\ \text{for} \quad & i = 1, ..., M. \end{aligned} \tag{5}$$

Introducing the Lagrange multipliers α_i, we obtain the dual problem:

maximize

$$Q(\alpha) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{M} y_i y_j \alpha_i \alpha_j \Big(H(\mathbf{x}_i, \mathbf{x}_j) + \frac{\delta_{ij}}{C} \Big) \tag{6}$$

subject to

$$\sum_{i=1}^{M} y_i \alpha_i = 0, \quad \alpha_i \ge 0 \quad \text{for} \quad i = 1, ..., M, \tag{7}$$

where δ_{ij} is Kronecker's delta function, in which $\delta_{ij} = 1$ for $i = j$ and 0, otherwise. Since $1/C$ is added to the diagonal elements of the Hessian matrix H, the matrix becomes positive define. Therefore, the associated optimization problem is more computationally stable than the L1-SVMs, in which the Hessian matrix is positive semi-definite [7].

III. Stopping Criteria

In this section, we describe the stopping criteria of L1 and L2 dual problems. Since the optimal solution must satisfy Karush-Kuhn-Tucker (KKT) complementarity condition, during training we check the condition, and if all training data satisfy the condition, we terminate training. But since the KKT condition of the dual problem includes the primal variable, the detection of violation becomes inexact. Thus the exact KKT condition is derived for the SMO (Sequential Minimal Optimization) technique [11]. In the following we discuss the approximate method and the exact method, which is an extension of [11].

A. Approximate KKT Conditions

In L1-SVMs, the KKT condition is given by

$$\alpha_i^* (y_i (\mathbf{w}^{*t} \mathbf{x}_i + b^*) - 1 + \xi_i^*) = 0, \tag{8}$$

$$b_i^* \xi_i^* = (C - \alpha_i^*) \xi_i^* = 0. \tag{9}$$

Thus, there are three cases as follows:

1) $\alpha_i^* = 0$. Then $\xi_i^* = 0$. Therefore, $\mathbf{x}_i$ is correctly classified.
2) $0 < \alpha_i^* < C$. Then $y_i(\mathbf{w}^{*t} \mathbf{x}_i + b^*) - 1 + \xi_i^* = 0$ and $\xi_i^* = 0$. Thus, $y_i(\mathbf{w}^* \mathbf{x}_i + b^*) = 1$ and $\mathbf{x}_i$ is a support vector.
3) $\alpha_i^* = C$. Then $y_i(\mathbf{w}^{*t} \mathbf{x}_i + b^*) - 1 + \xi_i^* = 0$ and $\xi_i^* \ge 0$. Therefore $\mathbf{x}_i$ is a bounded support vector and if $0 \le \xi_i^* < 1$, $\mathbf{x}_i$ is correctly classified, and if $\xi_i^* \ge 1$, $\mathbf{x}_i$ is misclassified.

While, in L2-SVMs, the KKT condition is given by

$$y_i \left(\sum_{j=1}^{M} \alpha_j^* y_j \Big(H(\mathbf{x}_j, \mathbf{x}_i) + \frac{\delta_{ij}}{C} \Big) + b^* \right) - 1 = 0. \tag{10}$$

In the above KKT conditions, since b, which is calculated exactly only after the optimal solution is obtained, is included, the violation check becomes inexact.

B. Exact KKT Conditions

To derive the exact KKT condition [11], we redefine the dual objective function of the L1-SVM given by (6) and (7), introducing the Lagrange multipliers δ_i, μ_i, and β:

$$\begin{aligned} Q(\alpha, \delta, \mu, \beta) \quad = \quad & \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{M} \alpha_i \alpha_j y_i y_j H(\mathbf{x}_i, \mathbf{x}_j) \\ & + \sum_{i=1}^{M} \delta_i \alpha_i - \sum_{i=1}^{M} \mu_i (\alpha_i - C) \\ & + \beta \sum_{i=1}^{M} \alpha_i y_i. \end{aligned} \tag{11}$$

Thus, we obtain the KKT conditions as follows.

$$\frac{\partial Q}{\partial \alpha_i} = (F_i + \beta) y_i + \delta_i - \mu_i = 0, \tag{12}$$

$$\delta_i \alpha_i = 0, \quad \delta_i \ge 0, \tag{13}$$

$$\mu_i (\alpha_i - C) = 0, \quad \mu_i \ge 0, \quad \text{for} \quad i = 1, ..., M, \tag{14}$$

where

$$F_i = y_i - \sum_{j=1}^{M} y_j \alpha_j H(\mathbf{x}_i, \mathbf{x}_j). \quad (15)$$

These KKT conditions are reduced to the three cases as follows:

1) For $\alpha_i = 0$,
$$(F_i + \beta) y_i \leq 0. \quad (16)$$
2) For $0 < \alpha_i < C$,
$$(F_i + \beta) y_i = 0. \quad (17)$$
3) For $\alpha_i = C$,
$$(F_i + \beta) y_i \geq 0. \quad (18)$$

These three equations are further simplified to the following two cases.

1) For $i \in I_{up}$,
$$(F_i + \beta) \leq 0, \quad (19)$$
where
$$\begin{aligned} I_{up} &= I_0 \cup I_1 \cup I_2, \\ I_0 &= \{i | 0 < \alpha_i < C\}, \\ I_1 &= \{i | y_i = 1, \alpha_i = 0\}, \\ I_2 &= \{i | y_i = -1, \alpha_i = C\}. \end{aligned} \quad (20)$$
2) For $i \in I_{down}$
$$(F_i + \beta) \geq 0, \quad (21)$$
where
$$\begin{aligned} I_{down} &= I_0 \cup I_3 \cup I_4, \\ I_3 &= \{i | y_i = 1, \alpha_i = C\}, \\ I_4 &= \{i | y_i = -1, \alpha_i = 0\}. \end{aligned} \quad (22)$$

If for any $i \in I_{up}$ and $j \in I_{down}$ there exists β that satisfies

$$-F_j \leq \beta \leq -F_i, \quad (23)$$

the KKT conditions for the dual problem are satisfied. We define

$$F_{max} = \max_{j \in I_{down}} -F_j, \quad F_{min} = \min_{i \in I_{up}} -F_i. \quad (24)$$

Then, since I_0 is included in both I_{up} and I_{down}, (23) is equivalent to

$$F_{max} = F_{min} = \beta. \quad (25)$$

If this equation is satisfied, the solution is optimal. Therefore, we can use this as a stopping criterion for training. We loose (25) to soften a computational burden as follows,

$$F_{min} \geq F_{max} - \tau, \quad (26)$$

where τ is a positive tolerance parameter. Introducing parameter τ, let the τ-violating set V_{KKT} be

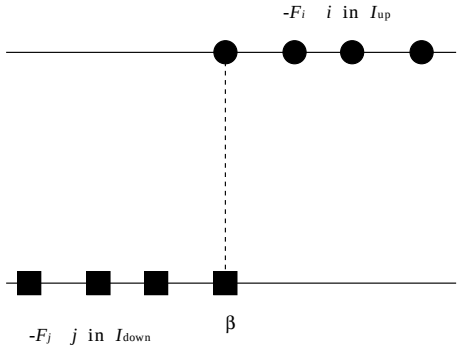

Fig. 1. KKT Conditions Satisfied

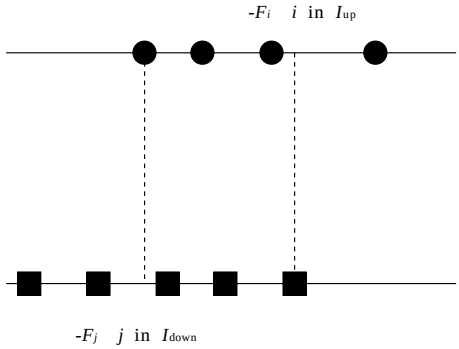

Fig. 2. KKT Conditions Violated

$$\begin{aligned} V_{KKT} = \{\mathbf{x}_i | & F_{min} + \tau < -F_i \text{ for } i \in I_{down}, \\ & F_{max} - \tau > -F_i \text{ for } i \in I_{up}\}. \end{aligned} \quad (27)$$

Fig. 1 shows a case where the KKT conditions are satisfied. The filled circles show $-F_i$ $(i \in I_{up})$ and the filled rectangles show $-F_j$ $(j \in I_{down})$. Fig. 2 shows a case where the KKT conditions are not satisfied. The data between the two dotted lines including the data on the lines violate the KKT conditions.

For L2-SVMs instead of (15), the following equation is obtained:

$$F_i = y_i - \sum_{j=1}^{M} y_j \alpha_j (H(\mathbf{x}_i, \mathbf{x}_j) + \delta_{ij}/C). \quad (28)$$

And furthermore, I_{up} and I_{down} are defined by

$$I_{up} = I_0 \cup I_1, \quad I_{down} = I_0 \cup I_4. \quad (29)$$

The other calculation is similar to those of L1-SVMs.

IV. Decomposition Technique

Since the number of variables of the dual problem is the number of training data, it becomes difficult to solve the

problem for a large number of training data. To overcome this problem decomposition techniques are used [9], [10]. Here, we use variable size chunking discussed in [9]. We divided the training data into the working set W and the fixed set B.

We solve the subproblem for α_i associated with the data in W, fixing the variables associated with the data in B. After the solution of the subprogram is obtained, we delete the variables with non-zero α_i, from W, that satisfy the KKT condition and add F points from B, where F is a fixed integer, that do not satisfy the KKT condition. Then we solve the subproblem. We iterate this procedure until V_{KKT} is empty.

For the approximate KKT condition, we randomly select F points that violate the KKT condition. For the exact KKT condition, first we sort the sets I_{up} and I_{down} in the descending order of KKT violations. Then we alternately select F points from the top of I_{up} and I_{down}.

V. Simulation Experiments

We evaluated the performance of L1-SVMs and L2-SVMs using the iris data [12], [13], the numeral data [14], the thyroid data [15], the blood cell data [16], and hiragana data [17]. Specifications of these data are shown in TABLE I; the numbers of inputs, classes, training data, and test data. Since these benchmark data sets are multiclass problems, we used one-against-all fuzzy SVMs [8] to resolve unclassifiable regions. We set $\tau = 0.01$ for the exact KKT condition. We use the following dot product and polynomial kernels:

$$\text{Dot product kernels}: \quad H(\mathbf{x}, \mathbf{x}') = \mathbf{x}^t \mathbf{x}', \tag{30}$$

$$\text{Polynomial kernels}: \quad H(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^t \mathbf{x}' + 1)^d. \tag{31}$$

TABLE I
Benchmark Data Specification

Data	Inputs	Classes	Training data	Test data
Iris	4	3	75	75
Numeral	12	10	810	820
Thyroid	21	3	3772	3428
Blood cell	13	12	3097	3100
Hiragana-50	50	39	4610	4610
Hiragana-105	105	38	8375	8356
Hiragana-13	13	38	8375	8356

For dot product kernels, the maximum rank of the Hessian matrix for L1-SVMs is the number of input variables plus 1 [7]. Thus, if the working set size exceeds this value, the Hessian matrix for L1-SVMs is positive semi-definite. But the Hessian matrix for L2-SVMs is always positive definite. Thus, for dot product kernels, training of L2-SVMs should be faster than that of L1-SVMs.

The dual problem was solved by combining the primal-dual interior-point method [18] with the variable chunking technique. We ran the c program on a Pentium III 1 GHz PC.

TABLE II shows the results of L1-SVMs and L2-SVMs using the approximate KKT condition. Here, we set $F = 50$. Namely, we added 50 data after the subproblem was solved. From the table, except for the thyroid data, training of the L2-SVM with dot product kernels is slower than that of the L1-SVM with dot product kernels. And for polynomial kernels, in most cases, training of the L1-SVM is faster.

The recognition rates of the test data by the L2-SVM are higher than those by the L1-SVM for 12 cases out of 26. But those by the L1-SVM are higher for 6 cases. Thus the L2-SVM performed better than the L1-SVM, but the difference of the recognition rate is small.

TABLE III shows the results when the exact KKT condition was used. Similar to the approximate KKT condition, in most cases, training time of L1-SVM is shorter than that of the L2-SVM. The recognition rates of the test data by the L2-SVM tend to be better than those by the L1-SVM, but the difference of the recognition rate is small.

Comparing TABLES II and III, training time by the exact KKT condition was not always shorter than that by the approximate KKT condition. But the exact KKT condition for L1-SVMs with dot product kernels performed better than the approximate KKT condition for the thyroid, blood cell, and hiragana-13 data.

To investigate why the exact KKT condition is not always better than the approximate KKT condition, we study the case for the blood cell data with $d = 3$. Fig. 3 shows training time of the approximate and exact KKT conditions for the change of F, namely the number of variables added to the working set W. For the approximate KKT condition, training time decreases as the number of variables added to the working set is increased. But for the exact KKT condition, the shortest training time is around $F = 25$. By this characteristics the selection of the optimum F is difficult for the exact KKT condition.

Fig. 4 shows the working set sizes of the approximate and exact KKT conditions against the number of iterations, when Class 2 is separated from the remaining classes. From the figure, the working set size of the exact KKT condition is larger than that of the approximate KKT condition after the second iteration. This means that the approximate KKT condition estimates the violating variables conservatively. Thus, with the smaller working set size, training by the approximate KKT condition is faster.

VI. Conclusions

In this paper, we evaluated training time and the generalization ability of L1-SVMs and L2-SVMs. As a result of the experiment, training of L2-SVMs was not always faster than that of L1-SVMs, and the difference of the generalization abilities between the two is small. Further, we compared the training time using the exact KKT condition and the approximate KKT conditions, and showed that training by

TABLE II

PERFORMANCE OF L1-SVM AND L2-SVM USING THE APPROXIMATE KKT CONDITION

Data	Kernel	L1-SVM (%)	Time (s)	L2-SVM (%)	Time (s)
Iris	Dot	96.00 (97.33)	**0.06**	**97.33** (98.67)	0.07
(C=5000)	$d=2$	94.67 (100)	0.03	94.67 (100)	0.03
	$d=3$	94.67 (100)	0.03	94.67 (100)	0.03
Numeral	Dot	99.27 (100)	**0.89**	**99.39** (100)	1.15
(C=2000)	$d=2$	99.39 (100)	**0.87**	99.39 (100)	1.11
	$d=3$	99.51 (100)	**0.88**	99.51 (100)	1.10
	$d=4$	99.51 (100)	**0.94**	99.51 (100)	1.15
Thyroid	Dot	**95.82** (96.58)	12234	94.22 (94.67)	**3880**
(C=10000)	$d=2$	**97.14** (98.75)	2951	96.47 (98.38)	**888**
	$d=3$	**97.49** (99.31)	**59**	97.26 (99.10)	326
	$d=4$	**97.43** (99.34)	**38**	97.35 (99.23)	152
Blood cell	Dot	87.23 (91.02)	**925**	**87.87** (90.64)	1094
(C=2000)	$d=2$	92.97 (96.67)	**35**	**93.48** (97.05)	76
	$d=3$	93.19 (98.22)	**34**	**93.71** (98.55)	57
	$d=4$	92.68 (98.93)	**32**	**93.42** (99.00)	47
Hiragana-50	Dot	93.95 (97.81)	**302**	**94.12** (98.48)	474
(C=5000)	$d=2$	99.24 (100)	**191**	99.24 (100)	233
	$d=3$	**99.31** (100)	**205**	99.26 (100)	234
	$d=4$	**99.33** (100)	**195**	99.28 (100)	255
Hiragana-105	Dot	97.03 (97.50)	**1951**	**97.45** (98.08)	4424
(C=2000)	$d=2$	100 (100)	**903**	100 (100)	1066
	$d=3$	100 (100)	**964**	100 (100)	1102
Hiragana-13	Dot	91.92 (93.77)	**1094**	**96.47** (97.41)	1518
(C=1000)	$d=2$	98.56 (98.96)	1205	**98.72** (99.26)	**948**
	$d=3$	98.74 (99.12)	**586**	**98.84** (99.26)	998
	$d=4$	98.71 (99.04)	**724**	**98.77** (99.13)	1164

TABLE III

PERFORMANCE OF L1-SVM AND L2-SVM USING THE EXACT KKT CONDITION

Data	Kernel	L1-SVM (%)	Time (s)	L2-SVM (%)	Time (s)
Iris	Dot	96.00 (97.33)	0.08	**97.33** (98.67)	0.08
(C=5000)	$d=2$	94.67 (100)	**0.03**	94.67 (100)	0.05
	$d=3$	94.67 (100)	**0.03**	94.67 (100)	0.05
Numeral	Dot	99.27 (100)	**1.38**	**99.39** (100)	1.68
(C=2000)	$d=2$	99.39 (100)	**1.44**	99.39 (100)	2.10
	$d=3$	99.51 (100)	**1.69**	99.51 (100)	2.33
	$d=4$	99.51 (100)	**1.75**	99.51 (100)	2.27
Thyroid	Dot	**95.74** (97.24)	**154**	94.19 (94.67)	48200
(C=10000)	$d=2$	**97.14** (98.81)	**113**	96.44 (98.33)	1529
	$d=3$	**97.52** (99.26)	**106**	97.08 (99.07)	309
	$d=4$	**97.46** (99.31)	**73**	97.32 (99.23)	122
Blood cell	Dot	87.98 (91.51)	**272**	**88.45** (91.12)	3270
(C=2000)	$d=2$	93.00 (96.74)	**80**	**93.48** (97.06)	91
	$d=3$	93.26 (98.22)	**63**	**93.71** (98.55)	64
	$d=4$	92.65 (98.93)	**51**	**93.42** (99.00)	63
Hiragana-50	Dot	93.99 (97.81)	**314**	**94.12** (98.46)	539
(C=5000)	$d=2$	99.24 (100)	**203**	99.24 (100)	224
	$d=3$	**99.31** (100)	**208**	99.26 (100)	233
	$d=4$	**99.33** (100)	**213**	99.28 (100)	240
Hiragana-105	Dot	97.04 (97.55)	**2112**	**97.47** (98.10)	4353
(C=2000)	$d=2$	100 (100)	**868**	100 (100)	961
	$d=3$	100 (100)	**868**	100 (100)	979
Hiragana-13	Dot	94.57 (95.68)	**798**	**96.72** (97.39)	1900
(C=1000)	$d=2$	98.50 (98.93)	**687**	**98.73** (99.24)	1045
	$d=3$	98.72 (99.08)	**708**	**98.86** (99.27)	1005
	$d=4$	98.70 (99.01)	**776**	**98.78** (99.13)	1213

the exact KKT condition was not always faster than by the approximate KKT condition.

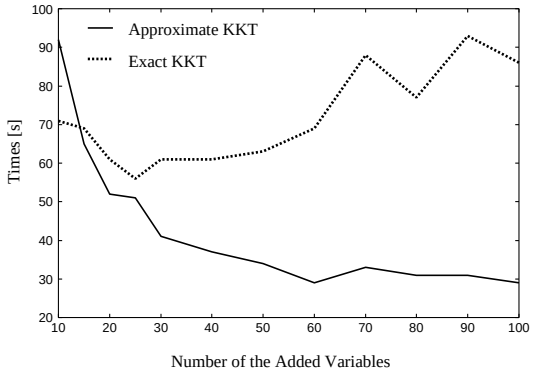

Fig. 3. Relationship between the Number of Variables Added and Training Time

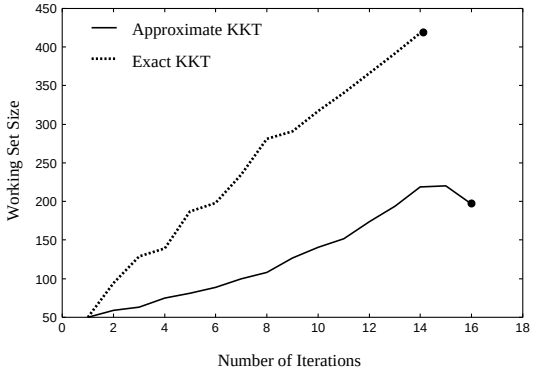

Fig. 4. Relationship between the Working Set Size and Number of Iterations

REFERENCES

[1] V. Vapnik, *Statistical Learning Theory*, John Wiley & Sons, 1998.

[2] V. Cherkassky and F. Mulier, *Learning from Data: Concepts, Theory, and Methods*, John Wiley & Sons, 1998.

[3] M. Pontil and A. Verri, "Properties of Support Vector Machines," *Neural Computation*, Vol. 10, No. 4, pp. 955–974, 1998.

[4] R. Rifkin, M. Pontil, and A. Verri, "A Note on Support Vector Machine Degeneracy," *Proc. 10th International Conference on Algorithmic Learning Theory, (ALT'99). (Lecture Notes in Artificial Intelligence Vol.1720)*, pp. 252–263, 1999.

[5] R. Fernández, "Behavior of the Weights of a Support Vector Machine as a Function of the Regularization Parameter C," *Proc. 8th International Conference on Artificial Neural Networks (ICANN'98)*, Vol. 2, pp. 917–922, 1998.

[6] C. J. C. Burges and D. J. Crisp, "Uniqueness of the SVM Solution," In S. A. Solla, T. K. Leen, and K.-R. Müller, editors, *Advances in Neural Information Processing Systems 12*, pp. 223-229, MIT Press, 2000.

[7] S. Abe, "Analysis of Support Vector Machines," *Neural Networks for Signal Processing XII—Proc. 2002 IEEE Signal Processing Society Workshop*, pp. 89–98, 2002.

[8] T. Inoue and S. Abe, "Fuzzy Support Vector Machines for Pattern Classification," *Proc. IJCNN'01*, pp. 1449–1454, 2001.

[9] C. Saunders, M. O. Stitson, J. Weston, L. Bottou, B. Schölkopf, and A. Smola, "Support Vector Machine Reference Manual," Technical Report CSD-TR-98-03, Royal Holloway, University of London, London, 1998.

[10] E. Osuna, R. Freund, and F. Girosi, "An Improved Training Algorithm for Support Vector Machines," *Neural Networks for Signal Processing VII—Proceedings of the 1997 IEEE Signal Processing Society Workshop*, pp. 276–285, 1997.

[11] S. S. Keerthi and E. G. Gilbert, "Convergence of a Generalized SMO Algorithm for SVM Classifier Design," *Machine Learning*, Vol. 13, pp. 637–649, 2001.

[12] R. A. Fisher, "The Use of Multiple Measurements in Taxonomic Problems," *Annals of Eugenics*, Vol. 7, pp. 179–188, 1936.

[13] J. C. Bezdek et al., "Will the Real Iris Data Please Stand up?" *IEEE Transactions on Fuzzy Systems*, Vol. 7, No. 3, pp. 368–369, 1999.

[14] H. Takenaga et al., "Input Layer Optimization of Neural Networks by Sensitivity Analysis and Its Application to Recognition of Numerals," *Electrical Engineering in Japan*, Vol. 111, No. 4, pp. 130–138, 1991.

[15] S. M. Weiss and I. Kapouleas, "An Empirical Comparison of Pattern Recognition, Neural Nets, and Machine Learning Classification Methods," *Proc. IJCAI-99, Workshop ML3*, pp. 55–60, 1999.

[16] A. Hashizume, J. Motoike, and R. Yabe, "Fully Automated Blood Cell Differential System and Its Application," *Proc. IUPAC 3rd International Congress on Automation and New Technology in the Clinical Laboratory*, pp. 297–302, Kobe, Japan, 1988.

[17] S. Abe, *Pattern Classification, Neuro-fuzzy Methods and Their Comparison*, Springer-Verlag, London, 2001.

[18] R. J. Vanderbei, "LOQO: An Interior Point Code for Quadratic Programming," Technical Report SOR-94-15, Princeton University, 1998.