<table>
<thead>
<tr>
<th>Title</th>
<th>Comparison of L1 and L2 Support Vector Machines</th>
</tr>
</thead>
<tbody>
<tr>
<td>Author(s)</td>
<td>Koshiba, Yoshiaki / Abe, Shigeo</td>
</tr>
<tr>
<td>Citation</td>
<td>Neural Networks, 2003. Proceedings of the International Joint Conference on, 3:2054-2059</td>
</tr>
<tr>
<td>Issue date</td>
<td>2003-07</td>
</tr>
<tr>
<td>Resource Type</td>
<td>Conference Paper / 会議発表論文</td>
</tr>
<tr>
<td>Resource Version</td>
<td>author</td>
</tr>
<tr>
<td>DOI</td>
<td>10.1109/IJCNN.2003.1223724</td>
</tr>
<tr>
<td>URL</td>
<td><a href="http://www.lib.kobe-u.ac.jp/handle_kernel/90000225">http://www.lib.kobe-u.ac.jp/handle_kernel/90000225</a></td>
</tr>
</tbody>
</table>

PDF issue: 2020-09-09
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 $\gamma$ [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 computation is computationally stable than that of L1-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.

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 $x_i$ ($i = 1, ..., M$) and its label be $y_i = 1$ if $x_i$ 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 $\phi(x)$.

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

$$\text{minimize} \quad \frac{1}{2} \| w \|^2 + C \sum_{i=1}^{M} \xi_i,$$

subject to $y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i$, for $i = 1, ..., M$, \hspace{1cm} (1)

where $w$ 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, $\xi_i$ ($i = 1, ..., M$) are the nonnegative slack variables and $b$ is a bias term. Introducing the Lagrange multipliers $\alpha_i$, we obtain the following dual problem:

maximize $Q(\alpha) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{M} \alpha_i \alpha_j y_i y_j g(x_i)^T g(x_j)$,
subject to \( \sum_{i=1}^{M} y_i \alpha_i = 0, \quad 0 \leq \alpha_i \leq C. \) \hfill (2)

We use the mapping function that satisfies
\[
H(x, x') = g(x)^T g(x'),
\]
where \( H(x, x') \) 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:
\[
D(x) = \sum_{i=1}^{M} \alpha_i^* y_i H(x_i, x) + b^*,
\]
where an asterisk denotes the optimal solution.

### B. L2 Support Vector Machines

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

minimize \( \frac{1}{2} \| w \|^2 + \frac{C}{2} \sum_{i=1}^{M} \xi_i^2, \)
subject to \( y_i(w^T x_i + b) \geq 1 - \xi_i, \)
for \( i = 1, \ldots, M. \) \hfill (5)

Introducing the Lagrange multipliers \( \alpha_i \), we obtain the dual problem:

maximize \( Q(\alpha) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{M} y_i y_j \alpha_i \alpha_j \left( H(x_i, x_j) + \frac{\delta_{ij}}{C} \right) \)
subject to \( \sum_{i=1}^{M} y_i \alpha_i = 0, \quad \alpha_i \geq 0 \) for \( i = 1, \ldots, M, \) \hfill (6)

where \( \delta_{ij} \) 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 definite. 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
\[
\alpha_i^*(y_i(w^* x_i + b^*) - 1 + \xi_i^*) = 0,
\]
\[
b_i^* \xi_i^* = (C - \alpha_i^*) \xi_i^* = 0. \hfill (9)
\]

Thus, there are three cases as follows:

1) \( \alpha_i^* = 0 \). Then \( \xi_i^* = 0 \). Therefore, \( x_i \) is correctly classified.
2) \( 0 < \alpha_i^* < C \). Then \( y_i(w^* x_i + b^*) - 1 + \xi_i^* = 0 \) and \( \xi_i^* = 0 \). Thus, \( y_i(w^* x_i + b^*) = 1 \) and \( x_i \) is a support vector.
3) \( \alpha_i^* = C \). Then \( y_i(w^* x_i + b^*) - 1 + \xi_i^* = 0 \) and \( \xi_i^* \geq 0 \). Therefore \( x_i \) is a bounded support vector and if \( 0 \leq \xi_i^* < 1 \), \( x_i \) is correctly classified, and if \( \xi_i^* \geq 1 \), \( x_i \) is misclassified.

While, in L2-SVMs, the KKT condition is given by
\[
y_i \left( \sum_{j=1}^{M} \alpha_j y_j \left( H(x_j, x_i) + \frac{\delta_{ij}}{C} \right) + b^* \right) - 1 = 0. \hfill (10)
\]

In the above KKT conditions, since \( b_i \), 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 \( \delta, \mu_i, \) and \( \beta \):

\[
Q(\alpha, \delta, \mu, \beta) = \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{M} \alpha_i \alpha_j y_i y_j H(x_i, x_j) + \sum_{i=1}^{M} \delta_i \alpha_i - \sum_{i=1}^{M} \mu_i (\alpha_i - C) + \beta \sum_{i=1}^{M} \alpha_i y_i. \hfill (11)
\]

Thus, we obtain the KKT conditions as follows.

\[
\frac{\partial Q}{\partial \alpha_i} = (F_i + \beta)y_i + \delta_i - \mu_i = 0, \hfill (12)
\]
\[
\delta_i = 0, \quad \delta_i \geq 0, \hfill (13)
\]
\[
\mu_i (\alpha_i - C) = 0, \quad \mu_i \geq 0, \quad \text{for } i = 1, \ldots, M. \hfill (14)
\]
where

\[ F_i = y_i - \sum_{j=1}^{M} y_j \alpha_j H(x_i, x_j). \]  

(15)

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

1) For \( \alpha_i = 0 \),

\[ (F_i + \beta)y_i \leq 0. \]  

(16)

2) For \( 0 < \alpha_i < C \),

\[ (F_i + \beta)y_i = 0. \]  

(17)

3) For \( \alpha_i = C \),

\[ (F_i + \beta)y_i \geq 0. \]  

(18)

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

1) For \( i \in I_{up} \),

\[ (F_i + \beta) \leq 0, \]  

where

\[ I_{up} = I_0 \cup I_1 \cup I_2, \]

\[ I_0 = \{ i | 0 < \alpha_i < C \}, \]

\[ I_1 = \{ i | y_i = 1, \alpha_i = 0 \}, \]

\[ I_2 = \{ i | y_i = -1, \alpha_i = C \}. \]  

(20)

2) For \( i \in I_{down} \),

\[ (F_i + \beta) \geq 0, \]  

where

\[ I_{down} = I_0 \cup I_3 \cup I_4, \]

\[ I_3 = \{ i | y_i = 1, \alpha_i = C \}, \]

\[ I_4 = \{ i | y_i = -1, \alpha_i = 0 \}. \]  

(22)

If for any \( i \in I_{up} \) and \( j \in I_{down} \) there exists \( \beta \) that satisfies

\[ -F_j \leq \beta \leq -F_i, \]  

(23)

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

\[ F_{\max} = \max_{j \in I_{down}} -F_j, \quad F_{\min} = \min_{i \in I_{up}} -F_i. \]  

(24)

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

\[ F_{\max} = F_{\min} = \beta. \]  

(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,

\[ F_{\min} \geq F_{\max} - \tau, \]  

(26)

where \( \tau \) is a positive tolerance parameter. Introducing parameter \( \tau \), let the \( \tau \)-violating set \( V_{KKT} \) be

\[ V_{KKT} = \{ x_i | F_{\min} + \tau < -F_i \text{ for } i \in I_{down}, \]

\[ F_{\max} - \tau > F_i \text{ for } i \in I_{up} \}. \]  

(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:

\[ F_i = y_i - \sum_{j=1}^{M} y_j \alpha_j \frac{H(x_i, x_j) + \delta_{ij}}{C}. \]  

(28)

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

\[ I_{up} = I_0 \cup I_1, \quad I_{down} = I_0 \cup I_4. \]  

(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 \( \alpha_i \) 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 \( \alpha_i \), 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:

- **Dot product kernels:** \[ H(x, x') = x^T x', \]
- **Polynomial kernels:** \[ H(x, x') = (x^T x')^d. \]

<table>
<thead>
<tr>
<th>Data</th>
<th>Inputs</th>
<th>Classes</th>
<th>Training data</th>
<th>Test data</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>4</td>
<td>3</td>
<td>75</td>
<td>75</td>
</tr>
<tr>
<td>Numerals</td>
<td>12</td>
<td>10</td>
<td>810</td>
<td>820</td>
</tr>
<tr>
<td>Thyroid</td>
<td>21</td>
<td>3</td>
<td>3772</td>
<td>3428</td>
</tr>
<tr>
<td>Blood cell</td>
<td>13</td>
<td>12</td>
<td>3097</td>
<td>3100</td>
</tr>
<tr>
<td>Hiragana-50</td>
<td>50</td>
<td>39</td>
<td>4610</td>
<td>4610</td>
</tr>
<tr>
<td>Hiragana-105</td>
<td>105</td>
<td>38</td>
<td>8375</td>
<td>8356</td>
</tr>
<tr>
<td>Hiragana-13</td>
<td>13</td>
<td>38</td>
<td>8375</td>
<td>8356</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Data</th>
<th>Kernel</th>
<th>L1-SVM Time (%)</th>
<th>L2-SVM Time (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris (C=5000)</td>
<td>Dot</td>
<td>96.00 (97.33)</td>
<td>97.23 (98.67)</td>
</tr>
<tr>
<td>d = 2</td>
<td>94.67 (100)</td>
<td>94.67 (100)</td>
<td></td>
</tr>
<tr>
<td>d = 4</td>
<td>94.67 (100)</td>
<td>94.67 (100)</td>
<td></td>
</tr>
<tr>
<td>Numeral (C=2000)</td>
<td>Dot</td>
<td>99.27 (100)</td>
<td>99.39 (100)</td>
</tr>
<tr>
<td>d = 2</td>
<td>99.39 (100)</td>
<td>99.39 (100)</td>
<td></td>
</tr>
<tr>
<td>d = 3</td>
<td>99.51 (100)</td>
<td>99.51 (100)</td>
<td></td>
</tr>
<tr>
<td>d = 4</td>
<td>99.51 (100)</td>
<td>99.51 (100)</td>
<td></td>
</tr>
<tr>
<td>Thyroid (C=10000)</td>
<td>Dot</td>
<td>97.14 (98.75)</td>
<td>96.47 (98.38)</td>
</tr>
<tr>
<td>d = 2</td>
<td>97.49 (99.31)</td>
<td>97.26 (99.10)</td>
<td></td>
</tr>
<tr>
<td>d = 4</td>
<td>97.43 (99.34)</td>
<td>97.35 (99.23)</td>
<td></td>
</tr>
<tr>
<td>Blood cell (C=2000)</td>
<td>Dot</td>
<td>87.23 (91.02)</td>
<td>87.87 (90.64)</td>
</tr>
<tr>
<td>d = 2</td>
<td>92.97 (96.67)</td>
<td>93.48 (97.05)</td>
<td></td>
</tr>
<tr>
<td>d = 3</td>
<td>93.19 (98.22)</td>
<td>93.71 (98.55)</td>
<td></td>
</tr>
<tr>
<td>d = 4</td>
<td>92.68 (98.93)</td>
<td>93.42 (99.00)</td>
<td></td>
</tr>
<tr>
<td>Hiragana-50 (C=5000)</td>
<td>Dot</td>
<td>93.95 (97.81)</td>
<td>94.12 (98.48)</td>
</tr>
<tr>
<td>d = 2</td>
<td>99.24 (100)</td>
<td>99.24 (100)</td>
<td></td>
</tr>
<tr>
<td>d = 3</td>
<td>99.31 (100)</td>
<td>99.26 (100)</td>
<td></td>
</tr>
<tr>
<td>d = 4</td>
<td>99.33 (100)</td>
<td>99.28 (100)</td>
<td></td>
</tr>
<tr>
<td>Hiragana-105 (C=2000)</td>
<td>Dot</td>
<td>97.03 (97.50)</td>
<td>97.45 (98.08)</td>
</tr>
<tr>
<td>d = 2</td>
<td>100 (100)</td>
<td>100 (100)</td>
<td></td>
</tr>
<tr>
<td>d = 3</td>
<td>100 (100)</td>
<td>100 (100)</td>
<td></td>
</tr>
<tr>
<td>d = 4</td>
<td>97.78 (99.04)</td>
<td>98.77 (99.13)</td>
<td></td>
</tr>
<tr>
<td>Hiragana-13 (C=1000)</td>
<td>Dot</td>
<td>91.92 (93.77)</td>
<td>96.47 (97.41)</td>
</tr>
<tr>
<td>d = 2</td>
<td>98.56 (98.96)</td>
<td>98.72 (99.26)</td>
<td></td>
</tr>
<tr>
<td>d = 3</td>
<td>98.74 (99.12)</td>
<td>98.84 (99.26)</td>
<td></td>
</tr>
<tr>
<td>d = 4</td>
<td>98.71 (99.04)</td>
<td>98.77 (99.13)</td>
<td></td>
</tr>
</tbody>
</table>

### TABLE III
**Performance of L1-SVM and L2-SVM Using the Exact KKT Condition**

<table>
<thead>
<tr>
<th>Data</th>
<th>Kernel</th>
<th>L1-SVM Time (%)</th>
<th>L2-SVM Time (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris (C=5000)</td>
<td>Dot</td>
<td>96.00 (97.33)</td>
<td>97.33 (98.67)</td>
</tr>
<tr>
<td>d = 2</td>
<td>94.67 (100)</td>
<td>94.67 (100)</td>
<td></td>
</tr>
<tr>
<td>d = 4</td>
<td>94.67 (100)</td>
<td>94.67 (100)</td>
<td></td>
</tr>
<tr>
<td>Numeral (C=2000)</td>
<td>Dot</td>
<td>99.27 (100)</td>
<td>99.39 (100)</td>
</tr>
<tr>
<td>d = 2</td>
<td>99.39 (100)</td>
<td>99.39 (100)</td>
<td></td>
</tr>
<tr>
<td>d = 3</td>
<td>99.51 (100)</td>
<td>99.51 (100)</td>
<td></td>
</tr>
<tr>
<td>d = 4</td>
<td>99.51 (100)</td>
<td>99.51 (100)</td>
<td></td>
</tr>
<tr>
<td>Thyroid (C=10000)</td>
<td>Dot</td>
<td>97.54 (97.24)</td>
<td>94.19 (94.61)</td>
</tr>
<tr>
<td>d = 2</td>
<td>97.14 (98.81)</td>
<td>96.44 (99.03)</td>
<td></td>
</tr>
<tr>
<td>d = 3</td>
<td>97.52 (99.26)</td>
<td>97.08 (99.07)</td>
<td></td>
</tr>
<tr>
<td>d = 4</td>
<td>97.46 (99.31)</td>
<td>97.32 (99.23)</td>
<td></td>
</tr>
<tr>
<td>Blood cell (C=2000)</td>
<td>Dot</td>
<td>87.98 (91.51)</td>
<td>88.45 (91.12)</td>
</tr>
<tr>
<td>d = 2</td>
<td>93.00 (96.74)</td>
<td>93.48 (97.06)</td>
<td></td>
</tr>
<tr>
<td>d = 3</td>
<td>93.26 (98.22)</td>
<td>93.71 (98.55)</td>
<td></td>
</tr>
<tr>
<td>d = 4</td>
<td>92.65 (98.93)</td>
<td>93.42 (99.00)</td>
<td></td>
</tr>
<tr>
<td>Hiragana-50 (C=5000)</td>
<td>Dot</td>
<td>93.99 (97.81)</td>
<td>94.12 (98.46)</td>
</tr>
<tr>
<td>d = 2</td>
<td>99.24 (100)</td>
<td>99.24 (100)</td>
<td></td>
</tr>
<tr>
<td>d = 3</td>
<td>99.31 (100)</td>
<td>99.26 (100)</td>
<td></td>
</tr>
<tr>
<td>d = 4</td>
<td>99.33 (100)</td>
<td>99.28 (100)</td>
<td></td>
</tr>
<tr>
<td>Hiragana-105 (C=2000)</td>
<td>Dot</td>
<td>97.04 (97.55)</td>
<td>97.47 (98.10)</td>
</tr>
<tr>
<td>d = 2</td>
<td>100 (100)</td>
<td>100 (100)</td>
<td></td>
</tr>
<tr>
<td>d = 3</td>
<td>100 (100)</td>
<td>100 (100)</td>
<td></td>
</tr>
<tr>
<td>d = 4</td>
<td>98.72 (99.08)</td>
<td>98.86 (99.27)</td>
<td></td>
</tr>
<tr>
<td>Hiragana-13 (C=1000)</td>
<td>Dot</td>
<td>91.57 (95.68)</td>
<td>96.72 (97.39)</td>
</tr>
<tr>
<td>d = 2</td>
<td>98.50 (98.93)</td>
<td>98.73 (99.24)</td>
<td></td>
</tr>
<tr>
<td>d = 3</td>
<td>98.72 (99.08)</td>
<td>98.86 (99.27)</td>
<td></td>
</tr>
<tr>
<td>d = 4</td>
<td>98.70 (99.01)</td>
<td>98.78 (99.13)</td>
<td></td>
</tr>
</tbody>
</table>

---

The exact KKT condition was not always faster than 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


