Supplementary Materials for "Warm Start for Parameter Selection of Linear Classifiers"

Bo-Yu Chu
Dept. of Computer Science
National Taiwan Univ., Taiwan
r02222047@ntu.edu.tw

Chia-Hua Ho
Dept. of Computer Science
National Taiwan Univ., Taiwan
b95082@csie.ntu.edu.tw

Cheng-Hao Tsai
Dept. of Computer Science
National Taiwan Univ., Taiwan
r01922025@csie.ntu.edu.tw

Chieh-Yen Lin
Dept. of Computer Science
National Taiwan Univ., Taiwan
r01944006@csie.ntu.edu.tw

Chih-Jen Lin
Dept. of Computer Science
National Taiwan Univ., Taiwan
cjlin@csie.ntu.edu.tw

I. PROOFS

This section includes all proofs of theorems in the paper. First we show a useful lemma.

Lemma 1 *If $\boldsymbol{w}_\infty$ exists, then for any $C > 0$, the norm of the optimal solution $\boldsymbol{w}_C$ is upper bounded by $\|\boldsymbol{w}_\infty\|$.*

$$\|\boldsymbol{w}_C\| \le \|\boldsymbol{w}_\infty\|, \forall C > 0. \tag{I.1}$$

Proof. We prove the result by contradiction. If a $\boldsymbol{w}_C$ satisfies $\|\boldsymbol{w}_C\| > \|\boldsymbol{w}_\infty\|$, then

$$\frac{f(\boldsymbol{w}_\infty)}{C} = \frac{\|\boldsymbol{w}_\infty\|^2}{2C} + L(\boldsymbol{w}_\infty) < \frac{\|\boldsymbol{w}_C\|^2}{2C} + L(\boldsymbol{w}_\infty) \le \frac{f(\boldsymbol{w}_C)}{C}, \tag{I.2}$$

where the last inequality is from the definition of W_∞ in (2.1). Results in (I.2) contradict the fact that $\boldsymbol{w}_C$ is the optimal solution of (2). □

I.1 Proof of Theorem 1

We first show that $\boldsymbol{w}_\infty$ exists. Because $W_\infty \neq \phi$, we can consider any $\bar{\boldsymbol{w}} \in W_\infty$ and define the following bounded region.

$$\{\boldsymbol{w} \mid \|\boldsymbol{w}\| \le \|\bar{\boldsymbol{w}}\|\} \cap W_\infty. \tag{I.3}$$

The continuity of $L(\boldsymbol{w})$ implies that W_∞ is closed. Therefore, the new region defined in (I.3) is compact. This property implies that the minimum value of (7) is attained. Furthermore, $L(\boldsymbol{w})$ is convex, so W_∞ is convex as well. The strict convexity of $\|\boldsymbol{w}\|^2$ implies that $\boldsymbol{w}_\infty$ is unique.

Next, by Lemma 1, (I.1) implies that

$$\boldsymbol{w}_C \in S, \forall C > 0, \tag{I.4}$$

where S is the following compact set

$$S \equiv \{\boldsymbol{w} \mid \|\boldsymbol{w}\| \le \|\boldsymbol{w}_\infty\|\}.$$

We then show that

$$\lim_{C\to\infty} \frac{f(\boldsymbol{w}_C)}{C} = L(\boldsymbol{w}_\infty). \tag{I.5}$$

This result follows from taking the limit on

$$\begin{aligned} L(\boldsymbol{w}_\infty) \le L(\boldsymbol{w}_C) &\le \frac{f(\boldsymbol{w}_C)}{C} \\ &\le \frac{f(\boldsymbol{w}_\infty)}{C} = \frac{\|\boldsymbol{w}_\infty\|^2}{2C} + L(\boldsymbol{w}_\infty). \end{aligned}$$

Finally, if the result in (7) does not hold, then there is a sequence $\{\boldsymbol{w}_{C_j}\}_{j=1}^\infty$ and a positive number δ such that

$$\|\boldsymbol{w}_{C_j} - \boldsymbol{w}_\infty\| \ge \delta, \forall j = 1, 2, \ldots. \tag{I.6}$$

Because S is compact, there exists a convergent subsequence $\{\boldsymbol{w}_{C_j}\}, j \in J$ such that

$$\lim_{j\in J, j\to\infty} \boldsymbol{w}_{C_j} = \boldsymbol{w}^*. \tag{I.7}$$

From (I.4), $\boldsymbol{w}^* \in S$, so $\|\boldsymbol{w}^*\| \le \|\boldsymbol{w}_\infty\|$. This property implies

$$\boldsymbol{w}^* \notin W_\infty. \tag{I.8}$$

Otherwise, $\|\boldsymbol{w}^*\| \le \|\boldsymbol{w}_\infty\|$ and $\boldsymbol{w}^* \in W_\infty$ indicates that $\boldsymbol{w}^*$ is a solution of (7). However, (I.6) and the uniqueness of the solution of (7) cause a contradiction.

From (I.8),

$$L(\boldsymbol{w}^*) > L(\boldsymbol{w}_\infty).$$

However, (I.5), (I.7), and the continuity of $L(\boldsymbol{w})$ imply that

$$L(\boldsymbol{w}^*) = L(\boldsymbol{w}_\infty),$$

so we have a contradiction. Therefore, the proof is complete.

I.2 Proof of Theorem 2

It is sufficient to prove that there exists a $\boldsymbol{w}^*$ such that

$$L(\boldsymbol{w}^*) = \inf_{\boldsymbol{w}} L(\boldsymbol{w}).$$

The optimization problem $\inf_{\boldsymbol{w}} L(\boldsymbol{w})$ can be rewritten as

$$\begin{aligned} &\min_{\boldsymbol{w},\boldsymbol{\xi}} \ \|\boldsymbol{\xi}\|^2 \\ &\text{subject to } y_i \boldsymbol{w}^T \boldsymbol{x}_i \ge 1 - \xi_i, i = 1, \ldots, l. \end{aligned} \tag{I.9}$$

Note that $(\boldsymbol{w}, \boldsymbol{\xi}) = (\boldsymbol{0}, \boldsymbol{e})$ satisfies the constraints, so (I.9) is feasible. Besides, the feasible region of (I.9) is a polyhedral, and the objective function $\|\boldsymbol{\xi}\|^2$ is convex quadratic. By [5, Corollary 27.3.1], the infimum value is attained.

I.3 Proof of Theorem 3

From Theorem 1, we only need to show that under condition (9), there exists a $\boldsymbol{w}^*$ such that

$$L(\boldsymbol{w}^*) = \inf_{\boldsymbol{w}} L(\boldsymbol{w}). \tag{I.10}$$

Because

$$\inf_{\boldsymbol{w}} L(\boldsymbol{w}) \le L(\boldsymbol{0}),$$

there exists a sequence $\{\boldsymbol{w}_k\}$ such that

$$L(\boldsymbol{w}_k) \le L(\boldsymbol{0}), \forall k,$$

and

$$\lim_{k\to\infty} L(\boldsymbol{w}_k) = \inf_{\boldsymbol{w}} L(\boldsymbol{w}).$$

Then

$$0 \le \log(1 + e^{-y_i \boldsymbol{w}_k^T \boldsymbol{x}_i}) \le L(\boldsymbol{0}), \forall i, k,$$

so there are a subset J and constants $L_1, \ldots, L_l$, such that

$$\lim_{k\in J, k\to\infty} \log(1 + e^{-y_i \boldsymbol{w}_k^T \boldsymbol{x}_i}) = L_i, \forall i. \tag{I.11}$$

If $L_i \neq 0, \forall i$, then

$$\lim_{k\in J, k\to\infty} YX\boldsymbol{w}_k = \boldsymbol{v}, \tag{I.12}$$

where X and Y are defined in (6), and

$$\boldsymbol{v} = \begin{bmatrix} -\log(e^{L_1} - 1) \\ \vdots \\ -\log(e^{L_l} - 1) \end{bmatrix}.$$

We prove $L_i \neq 0, \forall i$ later. If it is true, from (I.12), we show that there exists $\boldsymbol{w}^*$ such that

$$YX\boldsymbol{w}^* = \boldsymbol{v}. \tag{I.13}$$

and therefore $\inf_{\boldsymbol{w}} L(\boldsymbol{w})$ is attained. Otherwise, because

$$\min_{\boldsymbol{w}} \|YX\boldsymbol{w} - \boldsymbol{v}\|^2$$

attains a minimum $\hat{\boldsymbol{w}}$ following from [5], if no $\boldsymbol{w}^*$ exists, we have

$$YX\hat{\boldsymbol{w}} \neq \boldsymbol{v}.$$

However, (I.12) implies that we can always find $\boldsymbol{w}_k$ such that

$$\|YX\boldsymbol{w}_k - \boldsymbol{v}\| < \|YX\hat{\boldsymbol{w}} - \boldsymbol{v}\|,$$

a situation that violates the optimality of $\hat{\boldsymbol{w}}$.

The remaining task is to prove that $L_i \neq 0, \forall i$. If this result does not hold, then there exists an index i such that $L_i = 0$. From (I.11),

$$e^{-y_i \boldsymbol{w}_k^T \boldsymbol{x}_i} \to 0 \text{ and } y_i \boldsymbol{w}_k^T \boldsymbol{x}_i \to \infty \text{ as } k \to \infty, k \in J. \tag{I.14}$$

Thus J must be an infinite set. We have

$$\|\boldsymbol{w}_k\| \to \infty. \tag{I.15}$$

Otherwise, the boundedness of $\|\boldsymbol{w}_k\|$ and $|y_i \boldsymbol{w}_k^T \boldsymbol{x}_i| \le \|\boldsymbol{w}_k\| \|\boldsymbol{x}_i\|$ violate (I.14). From (I.15), $\boldsymbol{w}_k \neq \boldsymbol{0}$ after k is large enough, so we can consider a sequence $\{\boldsymbol{w}_k/\|\boldsymbol{w}_k\|\}$. Because this sequence is in a compact set, there exists a subset J' of J and a point $\hat{\boldsymbol{w}}$ such that

$$\lim_{k\to\infty, k\in J'} \frac{\boldsymbol{w}_k}{\|\boldsymbol{w}_k\|} = \hat{\boldsymbol{w}}. \tag{I.16}$$

From (9), there is an instance $\boldsymbol{x}_r$ such that

$$y_r \bar{\boldsymbol{w}}^T \boldsymbol{x}_r = -\epsilon < 0.$$

With (I.16), we can further find a subset J'' of J' such that

$$y_r \boldsymbol{w}_k^T \boldsymbol{x}_r \le -\frac{\epsilon}{2} \|\boldsymbol{w}_k\| < 0, \forall k \in J''.$$

Then as $k \to \infty$,

$$\begin{aligned} L(\boldsymbol{w}_k) &\ge \log(1 + e^{-y_r \boldsymbol{w}_k^T \boldsymbol{x}_r}) \\ &\ge \log(1 + e^{\epsilon \|\boldsymbol{w}_k\|/2}) \\ &\to \infty \end{aligned}$$

following from (I.15). This result violates the fact that

$$L(\boldsymbol{w}_k) \to \inf_{\boldsymbol{w}} L(\boldsymbol{w}) \le L(\boldsymbol{0}).$$

Therefore, $L_i \neq 0, \forall i$ and the proof is complete.

I.4 Proof of Theorem 4

From the optimality condition (see, for example, Section 3.4 in [6]),

$$\frac{(\alpha_C)_i}{C} = 2\max(0, 1 - y_i \boldsymbol{w}_C^T \boldsymbol{x}_i), \forall i, \tag{I.17}$$

and

$$\frac{(\alpha_C)_i}{C} = \frac{e^{-y_i \boldsymbol{w}_C^T \boldsymbol{x}_i}}{1 + e^{-y_i \boldsymbol{w}_C^T \boldsymbol{x}_i}}, \forall i \tag{I.18}$$

for L2 and logistic losses, respectively. With $\boldsymbol{w}_C \to \boldsymbol{w}_\infty$ by Theorems 2 and 3 and the continuity of max and exponential functions, taking the limit of (I.17) and (I.18) gives the desired results.

I.5 Proof of Theorem 5

Because data are not separable, Theorem 3 implies that $\boldsymbol{w}_\infty$ exists. Then from the Definition 1, there exists an instance $\boldsymbol{x}_i$ such that

$$y_i \boldsymbol{w}_\infty^T \boldsymbol{x}_i < 0.$$

From Theorem 4, clearly $(\alpha_C)_i \to \infty$ as $C \to \infty$.

I.6 Proof of Theorem 6

The existence of C^*, $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ and the optimality of $\boldsymbol{\alpha}_C$ have been proved in [2, Theorem 3]. From Theorem 1 and (4),

$$\lim_{C\to\infty} \frac{\boldsymbol{w}_C}{C} = \boldsymbol{0} = (YX)^T \boldsymbol{v}_1.$$

Then

$$\boldsymbol{w}_C = (YX)^T \boldsymbol{v}_2 = \boldsymbol{w}_\infty, \forall C \ge C^*.$$

I.7 Proof of Theorem 7

See the paper.

I.8 Proof of Theorem 8

We prove (25) by contradiction. Firstly, we show that if

$$\boldsymbol{w}_{C_1} = \boldsymbol{w}_{C_2}, \tag{I.19}$$

then

$$\boldsymbol{w}_{C_1} = \boldsymbol{w}_{C_2} = \boldsymbol{w}_\infty. \tag{I.20}$$

Because $f(\boldsymbol{w})$ is convex, the gradient at the optimal solution is zero. Therefore, from (I.19),

$$\begin{aligned}\nabla f(\boldsymbol{w}_{C_1}) &= \boldsymbol{w}_{C_1} + C_1\nabla L(\boldsymbol{w}_{C_1}) = \boldsymbol{0},\\ \nabla f(\boldsymbol{w}_{C_2}) &= \boldsymbol{w}_{C_2} + C_2\nabla L(\boldsymbol{w}_{C_2})\\ &= \boldsymbol{w}_{C_1} + C_2\nabla L(\boldsymbol{w}_{C_1}) = \boldsymbol{0}.\end{aligned}$$

Then,

$$\begin{aligned}\nabla L(\boldsymbol{w}_{C_1}) &= \frac{(\boldsymbol{w}_{C_1} + C_1\nabla L(\boldsymbol{w}_{C_1})) - (\boldsymbol{w}_{C_1} + C_2\nabla L(\boldsymbol{w}_{C_1}))}{C_1 - C_2}\\ &= \boldsymbol{0}.\end{aligned}$$

By the convexity of $L(\boldsymbol{w})$, $\boldsymbol{w}_{C_1}$ is an optimal solution of $L(\cdot)$, so $\boldsymbol{w}_{C_1} \in W_\infty$. By Lemma 1, $\|\boldsymbol{w}_{C_1}\| \le \|\boldsymbol{w}_\infty\|$, so by the definition of $\boldsymbol{w}_\infty$ in (7), $\boldsymbol{w}_{C_1} = \boldsymbol{w}_{C_2} = \boldsymbol{w}_\infty$.

By the assumption $\|\boldsymbol{w}_\infty\| \neq 0$ and the fact that $\boldsymbol{w}_\infty$ minimizes $L(\boldsymbol{w})$,

$$\nabla f(\boldsymbol{w}_\infty) = \boldsymbol{w}_\infty + C_1\nabla L(\boldsymbol{w}_\infty) = \boldsymbol{w}_\infty \neq \boldsymbol{0}. \tag{I.21}$$

Hence $\boldsymbol{w}_{C_1} \neq \boldsymbol{w}_\infty$, a contradiction to (I.20). Therefore, (I.19) does not hold, and hence $\boldsymbol{w}_{C_1} \neq \boldsymbol{w}_{C_2}$.

For (26), we can obtain the result by taking limit to the following equation.

$$\begin{aligned}\frac{\|\nabla f(\boldsymbol{w}_{C/\Delta}; C)\|}{\|\nabla f(\boldsymbol{0}; C)\|} &= \frac{\|\boldsymbol{w}_{C/\Delta} + C\nabla L(\boldsymbol{w}_{C/\Delta})\|}{C\|\nabla L(\boldsymbol{0})\|}\\ &= \frac{(C - C/\Delta)\|\nabla L(\boldsymbol{w}_{C/\Delta})\|}{C\|\nabla L(\boldsymbol{0})\|}\\ &= \frac{\Delta - 1}{\Delta}\frac{\|\nabla L(\boldsymbol{w}_{C/\Delta})\|}{\|\nabla L(\boldsymbol{0})\|}.\end{aligned} \tag{I.22}$$

When $C \to 0$, $\boldsymbol{w}_{C/\Delta} \to 0$. By the continuity of $\nabla L(\cdot)$,

$$\lim_{C\to 0} \nabla L(\boldsymbol{w}_{C/\Delta}) = \nabla L(\boldsymbol{0}).$$

Therefore, (I.22) converges to $(\Delta - 1)/\Delta$ when $C \to 0$.

For (27), because $\{\boldsymbol{w}_C\}$ converges to $\boldsymbol{w}_\infty$ and $\nabla L(\boldsymbol{w}_\infty) = 0$, the continuity of $\nabla L(\cdot)$ implies that (I.22) converges to zero as C goes to ∞.

II. CV ACCURACY UNDER SMALL REGULARIZATION PARAMETER

In this section, we explain that the CV accuracy tends to be fixed when C is close to zero. Firstly, we prove the following lemma.

Lemma 2 *If $L(\boldsymbol{w})$ is nonnegative,*

$$\lim_{C\to 0} f(\boldsymbol{w}_C) = \lim_{C\to 0} \|\boldsymbol{w}_C\| = 0. \tag{II.1}$$

Proof. Since

$$0 \le \frac{\|\boldsymbol{w}_C\|^2}{2} \le f(\boldsymbol{w}_C) \le f(\boldsymbol{0}),$$

by taking limit to both sides, we have

$$0 \le \lim_{C\to 0} \frac{\|\boldsymbol{w}_C\|^2}{2} \le \lim_{C\to 0} f(\boldsymbol{w}_C) \le \lim_{C\to 0} f(\boldsymbol{0}) = \lim_{C\to 0} CL(\boldsymbol{0}) = 0.$$

Therefore, (II.1) follows. □

Then we discuss the three losses separately. For L1-loss SVM, Lemma 2 implies that when C is small enough,

$$\|\boldsymbol{w}_C^T\boldsymbol{x}_i\| < 1 \text{ for all } i = 1, \dots, l.$$

Hence, $\xi(\boldsymbol{w}_C; \boldsymbol{x}_i, y_i) > 0$. By the KKT condition, $\alpha_i = C$ for all i. With (4), for any test instance $\boldsymbol{x}$,

$$\operatorname{sgn}(\boldsymbol{w}_C^T\boldsymbol{x}) = \operatorname{sgn}(\sum_{i=1}^{l} y_i C\boldsymbol{x}_i^T\boldsymbol{x}) = \operatorname{sgn}(\sum_{i=1}^{l} y_i\boldsymbol{x}_i^T\boldsymbol{x})$$

is independent of C.

For L2-loss SVM, by the KKT condition,

$$\alpha_i = 2C\max(1 - y_i\boldsymbol{w}_C^T\boldsymbol{x}_i, 0), \forall i = 1, \dots, l. \tag{II.2}$$

When $\boldsymbol{w}_C$ is close to zero, (II.2) becomes

$$\alpha_i = 2C(1 - y_i\boldsymbol{w}_C^T\boldsymbol{x}_i),$$

and α_i is close to $2C$. Therefore, for any test instance $\boldsymbol{x}$,

$$\begin{aligned}\operatorname{sgn}(\boldsymbol{w}_C^T\boldsymbol{x}) &= \operatorname{sgn}(\sum_{i=1}^{l} y_i\alpha_i\boldsymbol{x}_i^T\boldsymbol{x})\\ &= \operatorname{sgn}(\sum_{i=1}^{l} 2C(1 - y_i\boldsymbol{w}_C^T\boldsymbol{x}_i)y_i\boldsymbol{x}_i^T\boldsymbol{x})\\ &= \operatorname{sgn}(\sum_{i=1}^{l} (1 - y_i\boldsymbol{w}_C^T\boldsymbol{x}_i)y_i\boldsymbol{x}_i^T\boldsymbol{x}).\end{aligned} \tag{II.3}$$

If

$$\sum_{i=1}^{l} y_i\boldsymbol{x}_i^T\boldsymbol{x} \neq 0,$$

then because $\|\boldsymbol{w}_C\| \to 0$ from Lemma 2, when C is small enough,

$$\text{(II.3)} = \operatorname{sgn}(\sum_{i=1}^{l} y_i\boldsymbol{x}_i\boldsymbol{x}).$$

The prediction is almost independent of C.

Similarly, for logistic regression, we have the following equality from the KKT condition.

$$\alpha_i = \frac{C}{1 + e^{y_i\boldsymbol{w}_C^T\boldsymbol{x}_i}}, \tag{II.4}$$

where the value is close to $C/2$ when $\|\boldsymbol{w}_C\|$ is small. For any test instance $\boldsymbol{x}$,

$$\begin{aligned}\operatorname{sgn}(\boldsymbol{w}_C^T\boldsymbol{x}) &= \operatorname{sgn}(\sum_{i=1}^{l} y_i\alpha_i\boldsymbol{x}_i^T\boldsymbol{x})\\ &= \operatorname{sgn}\left(\sum_{i=1}^{l} \frac{C}{1 + e^{y_i\boldsymbol{w}_C^T\boldsymbol{x}_i}} y_i\boldsymbol{x}_i^T\boldsymbol{x}\right)\\ &= \operatorname{sgn}\left(\sum_{i=1}^{l} \frac{y_i\boldsymbol{x}_i^T\boldsymbol{x}}{1 + e^{y_i\boldsymbol{w}_C^T\boldsymbol{x}_i}}\right)\\ &= \operatorname{sgn}(\sum_{i=1}^{l} y_i\boldsymbol{x}_i^T\boldsymbol{x})\end{aligned}$$

if

$$\sum_{i=1}^{l} y_i\boldsymbol{x}_i^T\boldsymbol{x} \neq 0,$$

and C is small. The prediction is almost independent of C.

Note that the result is slightly different from [3, Case 1], where they consider the decision and the loss functions with a bias term:

$$\boldsymbol{w}^T\boldsymbol{x} + b.$$

They prove that, for L1-loss SVM, the decision function always outputs the major class when C is small. Although their result also implies that the CV accuracy is fixed when C is small, the value may be different from ours here. When $\boldsymbol{w}_C$ is small, the bias term b dominates the decision value $\boldsymbol{w}_C^T\boldsymbol{x} + b$, so the major class is predicted. On the other hand, our decision function does not have a bias term, so the results still depend on $\boldsymbol{w}_C^T\boldsymbol{x}$.

III. A DETAILED COMPARISON USING PRIMAL NEWTON AND DUAL COORDINATE DESCENT METHODS

We begin with describing some implementation details and then give a detailed comparison.

III.1 Implementation Details

We slightly adjust solvers in LIBLINEAR for the purpose of experiments.

III.1.1 Primal-based Stopping Condition

As mentioned in Section 4, we need to make primal Newton and dual coordinate descent (CD) methods have the same stopping condition to have a fair comparison. While the dual solver's stopping condition can check either the optimality condition of the dual or the primal variables, the primal solver only has the primal variables. Therefore we modified our dual solver to check the optimality condition (28) of the primal variables, so both solvers have the same stopping condition.

However, it is not trivial to evaluate the primal stopping condition (28) for the dual CD method. CD is an efficient method that takes only $O(nl)$ operations for updating all variables once (called an outer iteration in [1]), while evaluating the stopping condition (28) has the same time complexity. If we check the stopping condition in each iteration, a large portion of the training time is used for this extra condition check instead of solving the optimization problem. Therefore, we only check (28) once in every k iterations. We use $k = 10$ in our experiments.

However, the setting of checking the primal-based stopping condition once per k iterations may still have a huge affection on the training time when the number of iterations is small. For example, if the dual solver's default stopping condition can be reached in two iterations, the training time becomes five times because at least 10 iterations are needed. Although this situation seldom happens because the dual solver's stopping condition is usually stricter than (28), we decide to keep the original condition and use it along with (28).

III.1.2 Practical Implementation of (24)

When we introduced (24) as the stopping condition for the parameter selection procedure, we assume that $\boldsymbol{w}_C$ is the optimal solution for minimizing $f(\boldsymbol{w}; C)$. Practically, we have only an approximate solution $\tilde{\boldsymbol{w}}_C$, so $\boldsymbol{w}_{\Delta^{t-1}C}$ in (24) must be replaced by $\tilde{\boldsymbol{w}}_{\Delta^{t-1}C}$.

If a primal-based solver is used, we then have the following property. At $\Delta^{-2}C$, the initial solution is

$$\bar{\boldsymbol{w}}_{\Delta^{-2}C} = \tilde{\boldsymbol{w}}_{\Delta^{-3}C}.$$

Because (24) has the same form as the stopping condition (28) for solving the optimization problem under a fixed regularization parameter,[1] it implies that $\bar{\boldsymbol{w}}_{\Delta^{-2}C}$ is immediately returned as the approximate solution without any iteration. Therefore,

$$\tilde{\boldsymbol{w}}_{\Delta^{-2}C} = \bar{\boldsymbol{w}}_{\Delta^{-2}C} = \tilde{\boldsymbol{w}}_{\Delta^{-3}C}.$$

By the same reason, we have

$$\tilde{\boldsymbol{w}}_{\Delta^{-3}C} = \tilde{\boldsymbol{w}}_{\Delta^{-2}C} = \tilde{\boldsymbol{w}}_{\Delta^{-1}C} = \tilde{\boldsymbol{w}}_C. \tag{III.1}$$

Therefore, in our implementation, we simply check the number of times where the initial and the returned solutions of the optimization solver are the same. That is, if the count reaches three for a continuous sequence of $\Delta^{-2}C$, $\Delta^{-1}C$, and C, then the procedure for the parameter selection stops.

If the dual solver is used, the situation is different. At $\Delta^{-3}C$, we have an approximate dual solution $\tilde{\boldsymbol{\alpha}}_{\Delta^{-3}C}$ and the corresponding primal solution $\tilde{\boldsymbol{w}}_{\Delta^{-3}C}$ with

$$\tilde{\boldsymbol{w}}_{\Delta^{-3}C} = \sum\nolimits_{i=1}^{l} y_i(\tilde{\alpha}_{\Delta^{-3}C})_i\boldsymbol{x}_i.$$

Assume $\tilde{\boldsymbol{w}}_{\Delta^{-3}C}$ satisfies (24) with $t = -2$:

$$\|\nabla f(\tilde{\boldsymbol{w}}_{\Delta^{-3}C}; \Delta^{-2}C)\| \le \epsilon\|\nabla f(\mathbf{0}; \Delta^{-2}C)\|. \tag{III.2}$$

Then the procedure continues to find an approximate solution $\tilde{\boldsymbol{\alpha}}_{\Delta^{-2}C}$. The dual initial solution is

$$\bar{\boldsymbol{\alpha}}_{\Delta^{-2}C} = \Delta\tilde{\boldsymbol{\alpha}}_{\Delta^{-3}C}. \tag{III.3}$$

Because we check the primal-based stopping condition (28) for the dual coordinate descent method, with (III.3) it is like that we start with

$$\bar{\boldsymbol{w}}_{\Delta^{-2}C} = \Delta\tilde{\boldsymbol{w}}_{\Delta^{-3}C}$$

in checking this condition. It is less likely that

$$\|\nabla f(\Delta\tilde{\boldsymbol{w}}_{\Delta^{-3}C}; \Delta^{-2}C)\| \le \epsilon\|\nabla f(\mathbf{0}; \Delta^{-2}C)\|$$

holds because we have assumed in (III.2) that $\tilde{\boldsymbol{w}}_{\Delta^{-3}C}$ is a good approximate solution for minimizing $f(\boldsymbol{w}; \Delta^{-2}C)$ and we have the property that $\{\boldsymbol{w}_C\}$ converges to $\boldsymbol{w}_\infty$. Therefore, the optimization procedure does not stop in the beginning. Instead, it takes several steps before reaching the stopping criterion. Then

$$\tilde{\boldsymbol{w}}_{\Delta^{-2}C} \ne \tilde{\boldsymbol{w}}_{\Delta^{-3}C} \tag{III.4}$$

and $\tilde{\boldsymbol{w}}_{\Delta^{-2}C}$ is a better approximate solution than $\tilde{\boldsymbol{w}}_{\Delta^{-3}C}$ at $\Delta^{-2}C$. By the convergence of $\{\boldsymbol{w}_C\}$, $\tilde{\boldsymbol{w}}_{\Delta^{-2}C}$ tends to be a better solution than $\tilde{\boldsymbol{w}}_{\Delta^{-3}C}$ for minimizing the function $f(\boldsymbol{w}; \Delta^{-1}C)$. That is, $\tilde{\boldsymbol{w}}_{\Delta^{-2}C}$ more easily satisfies

$$\|\nabla f(\tilde{\boldsymbol{w}}_{\Delta^{-2}C}; \Delta^{-1}C)\| \le \epsilon\|\nabla f(\mathbf{0}; \Delta^{-1}C)\|,$$

which is the next condition in (24) to be checked. Therefore, we expect that the parameter-selection procedure stops earlier if the dual solver is used, and we will verify this result in Section III.2.1.

Because of (III.4), right after C is increased and before optimization solver is called, we must check (24) with gradient evaluations. In contrast, the implementation is easier if we apply a primal-based optimization method of using (28) as the stopping condition. The reason is that we can take the advantage of (III.1) by checking the difference of $\tilde{\boldsymbol{w}}$ vectors without gradient evaluations.

[1] Note that here we assume that (24) has been slightly modified to have the term $\min(l^+, l^-)/l$ in (28).

III.1.3 Maximum Iterations

Another implementation issue is when the solver should stop if the solver's stopping condition can hardly be reached. For example, in Section 4 we have shown that the dual CD method has lengthy iterations when C is large. To avoid unreasonable long training time, all solvers in LIBLINEAR stop when a maximal number of iterations is reached even if the stopping condition is not satisfied. To make (28) as the stopping condition used for both primal and dual solvers in most cases, we increase the default 1,000 maximum iterations to 10,000 and 100,000, respectively. Note that the limit for dual is higher because usually its number of iterations is higher than that of primal.

III.1.4 An Improvement of the Newton Method

When solving the optimization problem in step 5.1.1 of Algorithm 1, a trust region Newton method [4] computes the Newton direction $\boldsymbol{s}$ in each iteration by solving the following trust-region sub-problem.

$$\begin{aligned} \min_{\boldsymbol{s}} \quad & \nabla f(\boldsymbol{w})^T \boldsymbol{s} + \frac{1}{2}\boldsymbol{s}^T \nabla^T f(\boldsymbol{w}) \boldsymbol{s} \\ \text{subject to} \quad & \|\boldsymbol{s}\| \le \delta, \end{aligned} \tag{III.5}$$

where $\boldsymbol{w}$ is the current iterate and δ is the size of the trust region. A Conjugate Gradient (CG) method is used to approximately solve (III.5). CG is an iterative procedure that is terminated by LIBLINEAR if either $\boldsymbol{s}$ exceeds the trust region or

$$\|\nabla f(\boldsymbol{w}) + \nabla^2 f(\boldsymbol{w})\boldsymbol{s}\| \le \epsilon_{\text{CG}} \|\nabla f(\boldsymbol{w})\|, \tag{III.6}$$

where $\epsilon_{\text{CG}} = 0.1$.

When $\boldsymbol{w}$ is close to the optimal $\boldsymbol{w}_C$, $\|\nabla f(\boldsymbol{w})\|$ is small. Then the stopping condition becomes stricter. Therefore, for a Newton method starting with initial $\boldsymbol{w} = \boldsymbol{0}$, CG stopping condition (III.6) is loose in the beginning, but is tight in the end. Now with warm start, because $\{\boldsymbol{w}_C\} \to \boldsymbol{w}_\infty$, the initial $\bar{\boldsymbol{w}} = \boldsymbol{w}_C$ is close to $\boldsymbol{w}_{\Delta C}$ for a large C and the CG stopping condition is tight in the beginning. For a truncated Newton method such as the trust region Newton method, early directions are not good enough, so there is no need to accurately solve (III.5). Therefore, under the warm-start setting, the original $\epsilon_{\text{CG}} = 0.1$ in LIBLINEAR may cause a too tight condition.

To understand the performance under different ϵ_{CG} values, in Table I, we show the CV time and the CV rate under $\epsilon_{\text{CG}} = 0.1$ and 0.5 for some data sets. The columns in Table I are defined as follows.

- stop C: the last C that the parameter-selection procedure checked. That is, the C value which satisfies the termination criterion (24) of the parameter-selection procedure.
- stop time: The cumulative CV time (in seconds) from $C_{\min}$ to the stop C.
- best rate: The best CV rate achieved by checking from $C_{\min}$ to the stop C.
- total time: The total CV time (in seconds) from $C_{\min}$ to $C_{\max}$.

We see that ϵ_{CG} does not affect the found best CV rate and the corresponding C very much, but in document data such as yahoo-japan and yahoo-korea, the CV time is dramatically reduced by using $\epsilon_{\text{CG}} = 0.5$. The setting of $\epsilon_{\text{CG}} = 0.1$ causes too many CG steps in the first several (outer) iterations.

Table I: Comparison of primal Newton method with $\epsilon_{\text{CG}} = 0.1$ or 0.5.

Data set	ϵ_{CG}	stop $\log_2 C$	stop time	best rate	total time
a9a	0.1	-2	1.87e+01	84.77	2.30e+01
	0.5	0	2.18e+01	84.80	2.42e+01
covtype_scale	0.1	-4	6.40e+02	75.66	7.22e+02
	0.5	-4	6.37e+02	75.67	7.17e+02
german_scale	0.1	-1	1.74e−01	77.10	1.98e−01
	0.5	-1	2.00e−01	77.10	2.20e−01
ijcnn1	0.1	3	1.03e+01	92.43	1.11e+01
	0.5	3	9.41e+00	92.42	1.01e+01
rcv1_test	0.1	4	8.85e+02	97.75	9.33e+02
	0.5	4	8.30e+02	97.75	8.79e+02
webspam (unigram)	0.1	7	1.42e+03	92.57	1.59e+03
	0.5	4	1.23e+03	92.62	1.28e+03
yahoo-japan	0.1	3	1.79e+03	92.69	1.79e+03
	0.5	3	1.27e+03	92.68	1.27e+03
yahoo-korea	0.1	5	1.60e+04	86.89	1.60e+04
	0.5	5	8.40e+03	86.86	8.40e+03

Although for some data sets the total CV time increases instead, the difference is relatively minor.

As a result, we believe that using a larger $\epsilon_{\text{CG}} = 0.5$ is a generally better setting when warm start is applied. For consistency, we use this setting for all experiments with/without warm start. That is, when standard LIBLINEAR is used in Section 4, we change ϵ_{CG} to 0.5 from the default 0.1.

III.2 Comparison Results

We check the performance of both two-class and multi-class problems in LIBSVM data sets.[2] Because of the large amount of data sets, we run experiments on many machines at the same time. Although these machines have different computation capability, we ensure that the primal and the dual solvers run on the same machine for any data set. By the same reason, training time of the six selected data sets presented in the paper may be different from the time shown here.

The data statistics are in Tables II and III.

III.2.1 Two-class Problems

We apply our parameter-selection procedure on data sets listed in Table II. The columns are defined in Section III.1.4. We terminate the process if it does not stop in three days. In this case, we indicate "not finished" in the table.

We have some observations from Table IV. Firstly, because the optimization problem is not exactly solved, the best CV rates of the two solvers may be different on some data sets. For few cases the stopping condition (28) is too loose to get the true CV. We will discuss this issue in Section III.3. Secondly, the dual CD method usually terminates with a smaller C. This result verifies our expectation in Section III.1.2. Finally, for data sets with more instances than features, the primal Newton method is more competitive. On the other hand, for sparse document data sets, which have many features, the dual CD method is usually fast if we compare the training time from $C_{\min}$ to the best C. In addition to the fact that the dual CD method is more effi-

[2] `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`

Table II: Two-class data statistics: Density is the average ratio of non-zero features per instance.

Data set	l	n	density
a1a	1,605	119	11.649%
a2a	2,265	119	11.651%
a3a	3,185	122	11.365%
a4a	4,781	122	11.365%
a5a	6,414	122	11.366%
a6a	11,220	122	11.368%
a7a	16,100	122	11.369%
a8a	22,696	123	11.277%
a9a	32,561	123	11.276%
australian	690	14	100.000%
australian_scale	690	14	87.443%
breast-cancer	683	10	100.000%
breast-cancer_scale	683	10	100.000%
cod-rna	59,535	8	100.000%
colon-cancer	62	2,000	100.000%
covtype	581,012	54	21.998%
covtype_scale	581,012	54	22.121%
diabetes	768	8	100.000%
diabetes_scale	768	8	99.854%
duke	44	7,129	100.000%
epsilon_normalized	400,000	2,000	100.000%
fourclass	862	2	100.000%
fourclass_scale	862	2	99.594%
german	1,000	24	100.000%
german_scale	1,000	24	95.837%
gisette_scale	6,000	5,000	99.100%
heart	270	13	100.000%
heart_scale	270	13	96.239%
ijcnn1	49,990	22	59.091%
ionosphere_scale	351	34	88.411%
KDD2010-a	8,407,752	20,216,830	0.000%
KDD2010-b	19,264,097	29,890,095	0.000%
leu	38	7,129	100.000%
liver-disorders	345	6	100.000%
liver-disorders_scale	345	6	99.082%
madelon	2,000	500	100.000%
mushrooms	8,124	112	18.750%
news20	19,996	1,355,191	0.034%
rcv1_test	677,399	47,236	0.155%
rcv1_train	20,242	47,236	0.157%
real-sim	72,309	20,958	0.245%
skin_nonskin	245,057	3	100.000%
sonar_scale	208	60	99.992%
splice	1,000	60	100.000%
splice_scale	1,000	60	100.000%
svmguide1	3,089	4	100.000%
svmguide3	1,243	22	99.495%
url_combined	2,396,130	3,231,961	0.004%
w1a	2,477	300	3.823%
w2a	3,470	300	3.878%
w3a	4,912	300	3.885%
w4a	7,366	300	3.892%
w5a	9,888	300	3.881%
w6a	17,188	300	3.888%
w7a	24,692	300	3.890%
w8a	49,749	300	3.883%
webspam (trigram)	350,000	16,609,143	0.022%
webspam (unigram)	350,000	254	33.517%
yahoo-japan	176,203	832,026	0.016%
yahoo-korea	460,554	3,052,939	0.011%

Table III: Multi-class data statistics: Density is the average ratio of non-zero features per instance.

Data set	l	n	#classes	density
acoustic	78,823	50	3	100.000%
acoustic_scale	78,823	50	3	100.000%
aloi	108,000	128	1,000	23.982%
aloi_scale	108,000	128	1,000	23.982%
combined	78,823	100	3	100.000%
combined_scale	78,823	100	3	100.000%
connect-4	67,557	126	3	33.333%
covtype	581,012	54	7	21.998%
covtype_scale	581,012	54	7	22.222%
covtype_scale01	581,012	54	7	22.121%
dna_scale	2,000	180	3	25.342%
glass_scale	214	9	6	99.844%
iris_scale	150	4	3	97.833%
letter_scale	15,000	16	26	100.000%
mnist	60,000	780	10	19.218%
mnist8m	8,100,000	784	10	25.388%
mnist8m_scale	8,100,000	784	10	25.388%
mnist_scale	60,000	780	10	19.218%
news20	15,935	62,061	20	0.129%
news20_scale	15,935	62,061	20	0.129%
pendigits	7,494	16	10	87.182%
poker	25,010	10	10	100.000%
protein	17,766	357	3	28.999%
rcv1_test_multiclass	518,571	47,236	53	0.137%
rcv1_train_multiclass	15,564	47,236	51	0.140%
satimage_scale	4,435	36	6	98.990%
sector	6,412	53	105	617.218%
sector_scale	6,412	55,197	105	0.295%
segment_scale	2,310	19	7	94.484%
seismic	78,823	50	3	100.000%
seismic_scale	78,823	50	3	100.000%
shuttle_scale	43,500	9	7	99.771%
svmguide2	391	20	3	100.000%
svmguide4	300	10	6	100.000%
usps	7,291	256	10	100.000%
vehicle_scale	846	18	4	98.023%
vowel	528	10	11	99.943%
vowel_scale	528	10	11	100.000%
wine_scale	178	13	3	99.870%

cient to handle document data, another reason of the faster training is that it stops the parameter-selection procedure at a smaller C. However, if we check the total time, the primal Newton method may still be competitive; see, for example, KDD2010-a. The result is consistent with the known property that a first-order optimization method like the dual CD method is slower when C is large.

III.2.2 Multi-class Problems

LIBLINEAR implements a one-versus-the-rest approach for multi-class problems, so several two-class problems are solved. We terminate the parameter-selection procedure if the stopping condition (28) holds for all two-class problems.

By the same setting for binary data sets, we check the performance on multi-class problems listed in Table III and present results in Table V. The observations of multi-class problems are consistent to those of two-class problems.

III.3 Strictness of the Stopping Condition

In Tables IV and V, we can find some data sets where the dual CD method has much higher CV accuracy than the primal Newton method. In Section III.2.1, we suspected that the stricter stopping condition for the dual CD method causes such results. We conduct experiments to check data sets with this problem by varying the stopping tolerance ϵ for the primal Newton method. We include one data set (ijcnn) that does not suffer from this problem in this experiment as a comparison. Results are in the left columns of Table VI. The best CV rates are similar for ijcnn under different ϵ values. However, for other data sets, the best CV rate increases when we use a smaller ϵ. This observation implies that for these data sets, the stopping tolerance itself is also a parameter that must be tuned.

To alleviate the above problem, we can always use a strict stopping tolerance, but training time may increase for other data sets. Besides, it is hard to find a good stopping tolerance that is small enough for all data sets. Therefore, we propose an interactive setting to help users improve their model if they think the stopping tolerance is not strict enough. The procedure is as follows.

1. The parameter-selection procedure stops at $\tilde{C}$ and outputs $\boldsymbol{w}^k_{\tilde{C}}, k = 1, \ldots, K$ for all K CV folds.
2. If users think the procedure stops too early with inaccurate CV accuracy because of the stopping tolerance, they can specify a stricter stopping condition to run the procedure again. They do not need to start from $C_{\min}$. Instead, the parameter-selection procedure starts at $\tilde{C}$ and uses $\boldsymbol{w}^k_{\tilde{C}}$ as the initial solution for training.
3. If the problem of using a too large tolerance still occurs, users can go back to step 1 and repeat the process.

We present the result of the above interactive procedure in the right columns of Table VI. A concern on our procedure is that the best C may be smaller than the initial $\tilde{C}$ considered in step 2. However, a comparison between left and right columns in Table VI shows that the proposed interactive procedure can effectively select a C value with CV accuracy close to the best, while requires less training time than the setting of always using a small tolerance.

However, $\boldsymbol{w}^k_{\tilde{C}}$ is not always necessary to output, and saving and loading K models also make the implementation more complicated. Therefore, we also consider using $\mathbf{0}$ as the initial solution in step 2. The results are in Table VII.[3] Table VII shows that the improvement without using $\boldsymbol{w}^k_{\tilde{C}}$ is still very effective. Therefore, we include this version in our released parameter-selection tool.

III.4 Summary

In summary of the comparison between the primal Newton method and the dual CD method, we have the following observations.

1. Although the dual CD method can solve large document data sets more efficiently than the primal Newton method, the advantage is weakened when C is very large. The situation can be very serious for some problems. To avoid having such bad situations, we choose the primal Newton method in our tool.
2. The CG stopping tolerance ϵ_{CG} chosen for the primal Newton method of solving a single optimization problem may be too tight when we solve a sequence of problems using warm start.
3. Although the primal Newton method needs a stricter stopping tolerance on some data sets, we designed an interactive utility to effectively alleviate this problem.

IV. EXPERIMENTAL RESULTS OF L2-LOSS SVM

We conduct experiments on L2-loss SVM under the same setting as logistic regression. Figure I is the CV accuracy and CV training/validation time under different regularization parameters. Note that the best C tends to be smaller than that of logistic regression. The reason is that L2-loss function gives a larger penalty for a wrong prediction. Table VIII is the initial function values of the primal and dual solvers. Figure II demonstrates the training time versus the relative difference from the optimal objective value under the best C values found by Algorithm 1. The comparison of cumulative running time with/without warm start is in Figure III.

References

[1] C.-J. Hsieh, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundararajan. A dual coordinate descent method for large-scale linear SVM. In *ICML*, 2008.

[2] W.-C. Kao, K.-M. Chung, C.-L. Sun, and C.-J. Lin. Decomposition methods for linear support vector machines. *Neural Comput.*, 16(8):1689–1704, 2004.

[3] S. S. Keerthi and C.-J. Lin. Asymptotic behaviors of support vector machines with Gaussian kernel. *Neural Comput.*, 15(7):1667–1689, 2003.

[4] C.-J. Lin, R. C. Weng, and S. S. Keerthi. Trust region Newton method for large-scale logistic regression. In *ICML*, 2007.

[5] R. T. Rockafellar. *Convex Analysis*. Princeton University Press, Princeton, NJ, 1970.

[6] H.-F. Yu, F.-L. Huang, and C.-J. Lin. Dual coordinate descent methods for logistic regression and maximum entropy models. *MLJ*, 85:41–75, 2011.

[3] The experiments of Tables VI and VII are conducted on different machines, so the times are different.

Table IV: Comparison of primal Newton method and dual CD method (binary data sets).

	dual				primal			
Data set	stop $\log_2 C$	stop time	best rate	total time	stop $\log_2 C$	stop time	best rate	total time
a1a	5	1.05	83.24	2.47	10	0.57	83.18	0.57
a2a	5	2.57	82.08	5.23	6	1.04	82.03	1.12
a3a	4	2.30	83.61	5.36	8	1.75	83.58	1.79
a4a	3	2.25	84.40	4.84	8	1.46	84.44	1.52
a5a	4	8.63	84.44	16.02	4	4.60	84.44	5.21
a6a	1	12.62	84.28	23.39	3	8.20	84.30	9.52
a7a	0	7.23	84.57	15.02	6	5.48	84.56	6.04
a8a	0	20.62	84.51	40.65	3	16.08	84.53	18.95
a9a	-1	21.00	84.79	37.95	4	11.93	84.80	13.56
australian	-16	0.39	85.80	129.38	-12	0.43	68.99	0.62
australian_scale	4	0.19	86.96	0.32	6	0.08	86.81	0.09
breast-cancer	-37	0.09	94.73	1,013.96	-35	0.05	65.01	0.13
breast-cancer_scale	4	0.15	96.78	0.22	6	0.06	96.63	0.07
cod-rna	-13	71.52	93.36	22,225.90	-10	37.93	87.56	54.00
colon-cancer	3	1.24	83.87	1.42	6	1.93	83.87	2.02
covtype	-21	140.44	75.58	not finished	-17	92.88	61.30	235.85
covtype_scale	-1	456.23	75.66	714.25	0	453.72	75.67	537.97
diabetes	-3	2.67	68.36	22.89	-4	0.14	67.97	0.18
diabetes_scale	7	0.24	77.34	0.27	6	0.12	77.34	0.13
duke	1	1.50	88.64	1.80	6	1.53	88.64	1.63
epsilon_normalized	6	11,074.39	89.80	15,451.41	6	13,605.46	89.81	14,500.33
fourclass	-7	0.14	73.78	0.32	-4	0.06	73.78	0.08
fourclass_scale	5	0.10	68.68	0.14	7	0.04	68.68	0.05
german	0	1.94	77.20	13.73	2	0.21	76.50	0.24
german_scale	4	0.42	77.20	0.61	5	0.24	77.10	0.27
gisette_scale	0	225.62	97.22	263.80	0	225.92	97.27	261.16
heart	1	12.89	84.07	58.98	1	0.06	83.70	0.07
heart_scale	6	0.28	83.33	0.36	8	0.19	83.33	0.23
ijcnn1	4	43.50	92.46	60.54	6	35.74	92.42	37.67
ionosphere_scale	8	1.40	84.62	1.88	10	0.49	84.33	0.49
KDD2010-a	-3	13,711.68	88.24	71,712.26	2	54,877.52	88.23	56,312.56
KDD2010-b	-3	31,599.50	88.89	102,894.63	2	98,611.87	88.85	101,344.49
leu	2	1.56	89.47	1.87	4	1.80	92.11	2.01
liver-disorders	-3	0.95	69.57	2.62	0	0.17	70.14	0.22
liver-disorders_scale	10	0.26	66.67	0.26	10	0.05	66.09	0.05
madelon	-8	9,233.95	60.30	128,458.04	-4	48.09	60.30	54.02
mushrooms	1	3.31	99.98	3.95	4	2.30	99.96	2.61
news20	10	178.54	96.56	178.54	10	257.85	96.46	257.85
rcv1_test	3	1,420.24	97.77	1,944.22	7	1,462.53	97.75	1,524.17
rcv1_train	9	71.11	97.05	75.25	10	87.48	97.02	87.48
real-sim	8	80.84	97.53	90.04	10	71.36	97.53	71.36
skin_nonskin	-17	117.46	90.66	337.21	-15	80.31	90.71	129.74
sonar_scale	10	6.27	74.04	6.27	10	0.45	74.04	0.45
splice	3	3.18	80.80	8.23	7	0.97	80.70	1.01
splice_scale	6	0.70	72.70	0.85	6	0.40	72.70	0.44
svmguide1	-11	0.81	84.49	17.96	-7	0.40	83.39	0.59
svmguide3	8	1.60	80.05	3.08	10	0.83	79.57	0.83
url_combined	-8	7,308.55	99.40	12,439.65	-4	7,175.31	97.75	8,691.55
w1a	10	8.76	97.46	8.76	10	2.34	97.50	2.34
w2a	10	6.42	97.38	6.42	10	1.02	97.38	1.02
w3a	10	40.97	97.68	40.97	10	5.17	97.70	5.17
w4a	10	21.65	97.81	21.65	10	2.23	97.81	2.23
w5a	10	28.92	97.87	28.92	10	3.34	97.86	3.34
w6a	9	97.88	98.07	121.84	10	18.57	97.99	18.57
w7a	8	36.50	98.19	54.53	10	11.45	98.20	11.45
w8a	7	62.88	98.37	104.72	10	29.34	98.34	29.34
webspam (trigram)	1	17,397.35	99.63	26,008.73	5	23,469.62	98.83	25,024.18
webspam (unigram)	1	622.97	92.80	1,179.12	7	674.11	92.62	703.39
yahoo-japan	9	978.55	92.69	1,099.30	10	1,501.85	92.68	1,501.85
yahoo-korea	5	4,301.03	87.35	10,690.05	9	5,964.98	87.28	6,039.39

Table V: Comparison of primal Newton method and dual CD method (multi-class data sets).

	dual				primal			
Data set	stop $\log_2 C$	stop time	best rate	total time	stop $\log_2 C$	stop time	best rate	total time
acoustic	0	230.37	68.07	370.82	6	193.92	67.84	205.82
acoustic_scale	4	422.51	70.53	693.26	3	202.46	69.85	234.97
aloi	-3	79,725.37	85.92	not finished	10	163,364.21	86.83	163,364.21
aloi_scale	-7	55,116.27	41.16	not finished	10	192,875.19	86.81	192,875.19
combined	1	465.78	80.36	736.76	10	565.93	80.14	565.93
combined_scale	-3	300.03	80.49	781.66	4	304.88	79.59	334.32
connect-4	-2	204.50	75.78	385.94	0	193.89	75.68	232.62
covtype	-19	1,216.15	70.56	not finished	-14	884.40	61.02	1,885.82
covtype_scale	5	7,145.51	71.53	8,867.22	9	3,837.61	71.53	3,872.09
covtype_scale01	2	3,866.98	71.53	5,808.50	7	2,712.17	71.49	2,814.53
dna_scale	7	8.42	95.00	20.79	10	1.82	95.15	1.82
glass_scale	10	1.11	64.49	1.11	10	0.19	64.95	0.19
iris_scale	10	0.14	88.00	0.14	10	0.06	88.00	0.06
letter_scale	9	154.85	68.09	163.32	10	78.15	68.07	78.15
mnist	-18	320.88	91.30	246,983.17	-10	414.50	91.15	648.09
mnist8m	-21	40,992.31	86.26	not finished	-19	56,154.13	85.72	149,764.99
mnist8m_scale	-9	156,704.34	86.28	not finished	1	156,364.96	85.72	180,256.27
mnist_scale	-2	957.83	91.29	5,912.23	8	947.18	91.12	976.39
news20	10	962.99	83.43	962.99	10	738.04	83.43	738.04
news20_scale	10	640.42	84.37	640.42	10	703.26	84.34	703.26
pendigits	-6	26.76	93.42	4,377.59	2	10.53	92.94	12.25
poker	10	1,100.21	49.96	1,100.21	10	90.50	49.96	90.50
protein	4	51.35	68.49	201.80	9	43.90	68.51	44.58
satimage_scale	8	26.89	83.52	32.61	8	13.08	83.40	13.49
sector	3	13.20	0.98	29.27	3	5.15	0.98	14.07
sector_scale	10	1,623.44	92.69	1,623.44	10	2,219.52	92.58	2,219.52
segment_scale	8	16.30	92.38	29.02	10	4.49	92.25	4.49
seismic	3	321.56	70.94	473.28	6	322.09	70.57	341.66
seismic_scale	-3	247.53	70.92	545.83	2	273.39	70.13	313.80
shuttle_scale	10	315.67	92.71	315.67	10	119.03	92.47	119.03
svmguide2	10	0.42	83.12	0.42	10	0.23	83.63	0.23
svmguide4	10	3.96	57.67	3.96	10	0.29	50.67	0.29
usps	4	344.92	94.84	3,996.14	9	186.11	94.71	188.75
vehicle_scale	10	16.93	78.96	16.93	10	2.05	79.20	2.05
vowel	7	7.82	44.70	10.54	10	2.44	45.08	2.44
vowel_scale	10	11.31	45.64	11.31	10	3.37	45.64	3.37
wine_scale	10	0.74	99.44	0.74	10	0.28	99.44	0.28

Table VI: Parameter selection under different ϵ and the effectiveness of the interactive utility. The initial solution is $\boldsymbol{w}^k_{\tilde{C}}$ when a new ϵ is used.

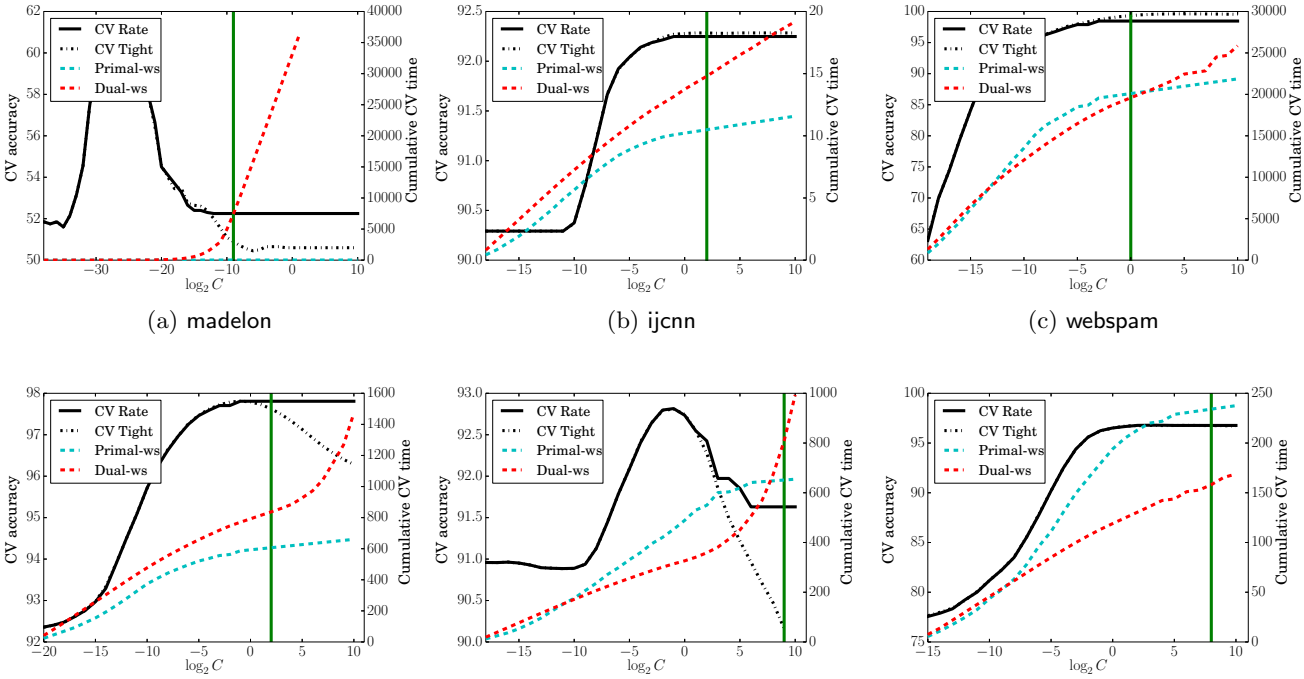
		from $C_{\min}$				interactive			
Data set	ϵ	$\log_2 \tilde{C}$ best	$\log_2 C$ best	CV rate	time	$\log_2 \tilde{C}$ best	$\log_2 C$ best	CV rate	time
	1e-02	-43.0	-16.0	68.55	0.04	-43.0	-16.0	68.55	0.04
australian	1e-03	-43.0	-5.0	81.45	0.06	-13.0	-5.0	81.45	0.02
	1e-04	-43.0	-3.0	85.94	0.08	0.0	0.0	85.36	0.01
	1e-02	-57.0	-57.0	65.01	0.02	-57.0	-57.0	65.01	0.02
	1e-03	-57.0	-57.0	65.01	0.03	-35.0	-35.0	65.01	0.00
breast-cancer	1e-04	-57.0	-57.0	65.01	0.03	-32.0	-32.0	65.01	0.00
	1e-05	-57.0	-8.0	93.27	0.07	-29.0	-8.0	93.27	0.03
	1e-06	-57.0	-2.0	94.44	0.08	-5.0	-1.0	94.44	0.01
	1e-02	-38.0	-13.0	87.56	3.88	-38.0	-13.0	87.56	3.88
cod-rna	1e-03	-38.0	-11.0	87.65	4.80	-10.0	-7.0	88.41	0.70
	1e-04	-38.0	0.0	93.43	8.01	-4.0	0.0	93.43	1.30
	1e-02	-46.0	-23.0	61.06	105.39	-46.0	-23.0	61.06	105.39
covtype	1e-03	-46.0	-12.0	70.51	183.96	-20.0	-13.0	70.21	50.16
	1e-04	-46.0	-7.0	75.24	395.74	-10.0	-7.0	75.24	111.87
	1e-02	-18.0	3.0	92.43	6.80	-18.0	3.0	92.43	6.80
ijcnn1	1e-03	-18.0	5.0	92.46	7.88	6.0	6.0	92.45	0.59
	1e-04	-18.0	4.0	92.45	10.17	9.0	9.0	92.45	0.44
	1e-02	-31.0	-9.0	97.79	2,367.36	-31.0	-9.0	97.79	2,367.36
url_combined	1e-03	-31.0	-2.0	98.98	6,861.13	-6.0	-4.0	98.72	2,156.13
	1e-04	-31.0	2.0	99.46	20,350.75	-1.0	2.0	99.46	13,170.86

Table VII: Parameter selection under different ϵ and the effectiveness of the interactive utility. The initial solution is 0 when a new ϵ is used.

		from $C_{\min}$				interactive			
dataset	ϵ	$\log \tilde{C}$ best	$\log C$ best	CV rate	time	$\log \tilde{C}$ best	$\log C$ best	CV rate	time
	1e-02	-43.0	-16.0	68.55	0.07	-43.0	-16.0	68.55	0.07
australian	1e-03	-43.0	-5.0	81.45	0.11	-13.0	-5.0	81.45	0.04
	1e-04	-43.0	-3.0	85.94	0.17	0.0	1.0	85.36	0.04
	1e-02	-57.0	-57.0	65.01	0.04	-57.0	-57.0	65.01	0.04
	1e-03	-57.0	-57.0	65.01	0.05	-35.0	-35.0	65.01	0.00
breast-cancer	1e-04	-57.0	-57.0	65.01	0.05	-33.0	-33.0	65.01	0.01
	1e-05	-57.0	-8.0	93.27	0.13	-28.0	-8.0	93.27	0.05
	1e-06	-57.0	-2.0	94.44	0.15	-5.0	-5.0	94.00	0.02
	1e-02	-38.0	-13.0	87.56	9.69	-38.0	-13.0	87.56	9.69
cod-rna	1e-03	-38.0	-11.0	87.65	11.83	-10.0	-10.0	87.68	1.52
	1e-04	-38.0	0.0	93.43	21.41	-8.0	0.0	93.43	7.25
	1e-02	-46.0	-23.0	61.06	210.87	-46.0	-23.0	61.06	210.87
covtype	1e-03	-46.0	-12.0	70.51	457.31	-20.0	-13.0	70.22	186.39
	1e-04	-46.0	-7.0	75.24	990.42	-10.0	-7.0	75.24	463.35

Table VIII: Difference between the initial and optimal function values. L2-loss SVM is used. The approach that is closer to the optimum is boldfaced.

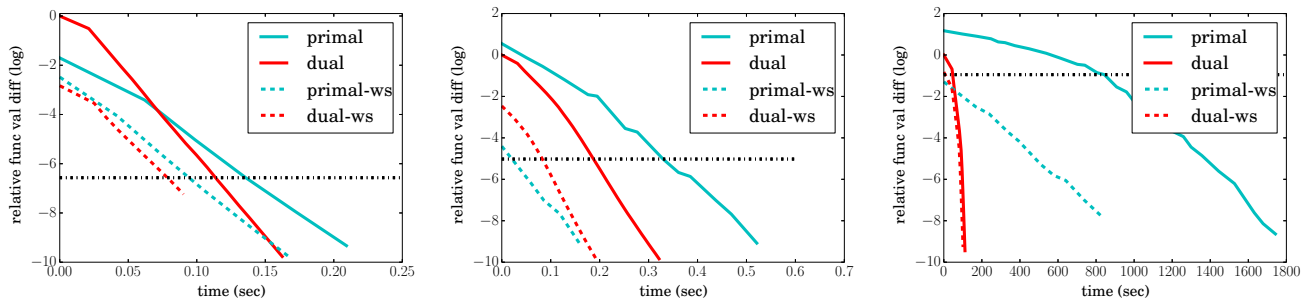
$\log_2 C$	primal	dual	$\|\bar{\boldsymbol{w}}\|^2/2$	primal	dual	$\|\bar{\boldsymbol{w}}\|^2/2$	primal	dual	$\|\bar{\boldsymbol{w}}\|^2/2$
	madelon			ijcnn			webspam		
−4	**5.83e−05**	−1.11e+00	8.05e−03	**9.89e−01**	−1.40e+01	1.50e+01	**8.54e+01**	−2.59e+02	3.45e+02
0	**0.00e+00**	−7.80e+00	8.26e−03	**1.15e−01**	−1.96e+01	1.98e+01	**6.35e+02**	−1.80e+03	2.43e+03
4	**0.00e+00**	−5.91e+01	8.28e−03	**7.55e−03**	−2.02e+01	2.02e+01	**3.80e+03**	−1.32e+04	1.70e+04
8	**0.00e+00**	−4.99e+02	8.28e−03	**5.06e−04**	−2.02e+01	2.02e+01	**1.24e+04**	−6.78e+04	7.99e+04
	rcv1			yahoo-japan			news20		
−4	**1.18e+02**	−4.90e+02	6.08e+02	**6.87e+01**	−1.14e+02	1.83e+02	**3.66e+01**	−6.45e+01	1.01e+02
0	**8.60e+02**	−2.47e+03	3.33e+03	**1.48e+03**	−1.99e+03	3.47e+03	**1.81e+02**	−7.53e+02	9.34e+02
4	**7.01e+03**	−1.97e+04	2.67e+04	**9.79e+03**	−3.56e+04	4.54e+04	**7.81e+01**	−2.33e+03	2.41e+03
8	**2.33e+04**	−1.28e+05	1.51e+05	**8.34e+03**	−1.25e+05	1.33e+05	**3.41e+01**	−2.74e+03	2.78e+03



(a) madelon (b) ijcnn (c) webspam

(d) rcv1 (e) yahoo-japan (f) news20

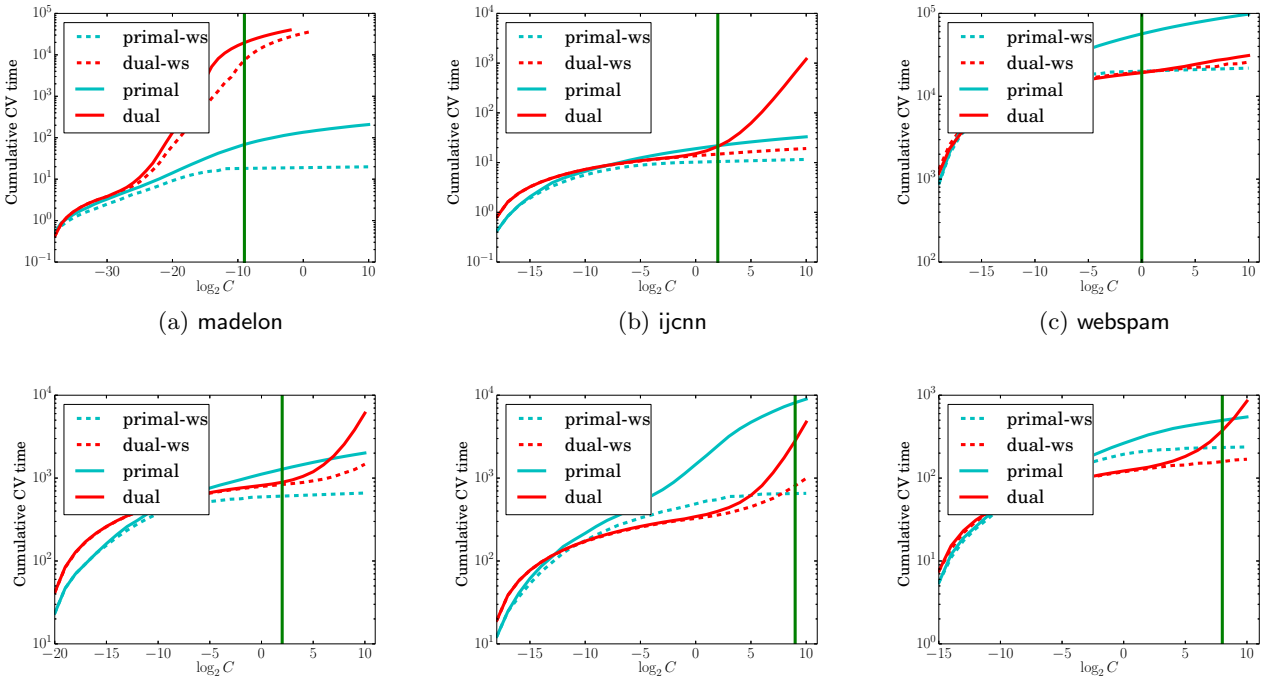
Figure I: CV accuracy and training time using L2-loss SVM with warm start. The two CV curves and the left y-axis are the CV accuracy in percentage (%). The dashed lines and the right y-axis are the cumulative training time in the CV procedure in seconds. The vertical line indicates the last C value checked by Algorithm 1.



(a) madelon (b) ijcnn (c) webspam

(d) rcv1 (e) yahoo-japan (f) news20

Figure II: Objective values versus training time using L2-loss SVM and the best C found by Algorithm 1. The solid lines correspond to settings without applying warm start, where default initial points in LIBLINEAR are used. Primal-ws and Dual-ws are primal and dual solvers with warm-start settings, respectively, and the initial point is obtained by (13) and (14). The horizontal line indicates that the condition (28) with LIBLINEAR's default $\epsilon = 10^{-2}$ has been established.



(a) madelon (b) ijcnn (c) webspam

(d) rcv1 (e) yahoo-japan (f) news20

Figure III: Training time (in seconds) using L2-loss SVM with/without warm-start techniques. The vertical line indicates the last C value checked by Algorithm 1. Because the training time quickly increases when C becomes large, the y-axis is log-scaled.