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

ABSTRACT

In linear classification, a regularization term effectively remedies the overfitting problem, but selecting a good regularization parameter is usually time consuming. We consider cross validation for the selection process, so several optimization problems under different parameters must be solved. Our aim is to devise effective warm-start strategies to efficiently solve this sequence of optimization problems. We detailedly investigate the relationship between optimal solutions of logistic regression/linear SVM and regularization parameters. Based on the analysis, we develop an efficient tool to automatically find a suitable parameter for users with no related background knowledge.

Keywords

warm start; regularization parameter; linear classification

1. INTRODUCTION

Linear classifiers such as logistic regression and linear SVM are commonly used in machine learning and data mining. Because directly minimizing the training loss may overfit the training data, the concept of regularization is usually applied. A linear classifier thus solves an optimization problem that involves a parameter (often referred to as C) to balance the training loss and the regularization term. Selecting the regularization parameter is an important but difficult practical issue. An inappropriate setting may not only cause overfitting or underfitting, but also a lengthy training time.

Several reasons make parameter selection a time-consuming procedure. First, usually the search involves sweeping the following sequence of parameters

$$C_{\min}, \Delta C_{\min}, \Delta^2 C_{\min}, \ldots, C_{\max}, \tag{1}$$

where Δ is a given factor. At each parameter the performance must be estimated by, for example, cross validation (CV). Thus a sequence of training tasks are conducted, and we may need to solve many optimization problems. Secondly, if we do not know the reasonable range of the parameters, we may need a long time to solve optimization problems under extreme parameter values.

In this paper, we consider using warm start to efficiently solve a sequence of optimization problems with different regularization parameters. Warm start is a technique to reduce the running time of iterative methods by using the solution of a slightly different optimization problem as an initial point for the current problem. If the initial point is close to the optimum, warm start is very useful. Recently, for incremental and decremental learning, where a few data instances are added or removed, we have successfully applied the warm start technique for fast training [20]. Now for parameter selection, in contrast to the change of data, the optimization problem is slightly modified because of the parameter change. Many considerations become different from those in [20]. As we will show in this paper, the relationship between the optimization problem and the regularization parameter must be fully understood.

Many past works have applied the warm-start technique for solving optimization problems in machine learning methods. For kernel SVM, [5, 15] have considered various initial points from the solution information of the previous problem. While they showed effective time reduction for some data sets, warm start for kernel SVM has not been widely deployed because of the following reasons.

- An initial point obtained using information from a related optimization problem may not cause fewer iterations than a naive initial point (zero in the case of kernel SVM).
- When kernels are used, SVM involves both regularization and kernel parameters. The implementation of a warm-start technique can be complicated. For example, we often cache frequently used kernel elements to avoid their repeated computation, but with the change of kernel parameters, maintaining the kernel cache is very difficult.

In fact, [15] was our previous attempt to develop warm-start techniques for kernel SVM, but results are not mature enough to be included in our popular SVM software LIBSVM [2]. In contrast, the situation for linear classification is simpler because

- the regularization parameter is the only parameter to be chosen, and
- as pointed out in [20] and other works, it is more flexible to choose optimization methods for linear rather than kernel. We will show that for different optimization methods, the effectiveness of warm-start techniques varies.

KDD'15, August 11–14, 2015, Sydney, NSW, Australia.

ACM 978-1-4503-3664-2/15/08 . . . $15.00.
http://dx.doi.org/10.1145/2783258.2783332.

In this work we focus on linear classification and make warm start an efficient tool for the parameter selection.

Besides warm-start techniques, other methods have been proposed for the parameter selection of kernel methods. For example, Hastie et al. [9] show that for L1-loss SVM, solutions are a piece-wise linear function of regularization parameters. They obtain the regularization path by updating solutions according to the optimality condition. This approach basically needs to maintain and manipulate the kernel matrix, a situation not applicable for large data sets. Although this difficulty may be alleviated for linear classification, we still see two potential problems. First, the procedure to obtain the path depends on optimization problems (e.g., primal and dual) and loss functions. Second, although an approximate solution path can be considered [8], the regularization path may still contain too many pieces of linear functions for large data sets. Therefore, in the current study, we do not pursue this direction for the parameter selection of linear classification. Another approach for parameter selection is to minimize a function of parameters that approximates the error (e.g., [18, 3, 4]). However, minimizing the estimation may not lead to the the best parameters, and the implementation of a two-level optimization procedure is complicated. Because linear classifiers involves a single parameter C, simpler approaches might be appropriate.

While traditional linear classification considers L2 regularization (see formulations in Section 1.1), recently L1 regularization has been popular because of its sparse model. Warm start may be very useful for training L1-regularized problems because some variables may remain to be zero after the change of parameters. In [13, 7], warm start is applied to speed up two optimization methods: interior point method and GLMNET, respectively. However, these works consider warm start as a trick without giving a full study on parameter selection. They investigate neither the range of parameters nor the relation between optimization problems and parameters. Another work [19] proposes a screening approach that pre-identifies some zero elements in the final solution. Then a smaller optimization problem is solved. They apply warm-start techniques to keep track of nonzero elements of the solutions under different parameters. However, the screening approach is not applicable to L2-regularized classifiers because of the lack of sparsity. While warm start for L1-regularized problems is definitely worth investigating, to be more focused we leave this topic for future studies and consider only L2-regularized classification in this work.

To achieve automatic parameter selection, we must identify a possible range of parameter values and decide when to stop the selection procedure. None of the work mentioned above except [19] has discussed the range of the regularized parameter. The work [21] finds a lower bound of C values for kernel SVM by solving a linear program, but for linear SVM, the cost may be too high. Another study [14] proposes using discrete optimization for the automatic parameter selection of any classification method. Their procedure stops if better parameters cannot be found after a few trials. By targeting at L2-regularized linear classification, we will derive useful properties to detect the possible parameter range.

In this work we develop a complete and automatic parameter selection procedure for L2-regularized linear classifiers with the following two major contributions. First, by carefully studying the relationship between optimization problems and regularization parameters, we obtain and justify an effective warm-start setting for fast cross validation across a sequence of regularization parameters. Second, we provide an automatic parameter selection tool for users with no background knowledge. In particular, users do not need to specify a sequence of parameter candidates.

This paper is organized as follows. In the rest of this section, we introduce the primal and dual formulations of linear classifiers. In Section 2, we discuss the relation between optimization problems and regularization parameters. Based on the results, we propose an effective warm-start setting to reduce the training time in Section 3. To verify our analysis, Section 4 provides detailed experiments. The conclusions are in Section 5. This research work has lead to a useful parameter-selection tool[1] extended from the popular package LIBLINEAR [6] for linear classification. Because of space limitation, we give proofs and more experiments in supplementary materials.[1]

1.1 Primal and Dual Formulations

Although linear classification such as logistic regression (LR) and linear SVM have been well studied in literatures (e.g., a survey in [22]), for easy discussion we briefly list their primal and dual formulations by mainly following the description in [20, Section 2]. Consider (label, feature-vector) pairs of training data $(y_i, \boldsymbol{x}_i) \in \{-1, 1\} \times R^n, i = 1, \ldots, l$. A linear classifier obtains its model by solving the following optimization problem.

$$\min_{\boldsymbol{w}} f(\boldsymbol{w}) \equiv \frac{1}{2}\|\boldsymbol{w}\|^2 + CL(\boldsymbol{w}), \tag{2}$$

where

$$L(\boldsymbol{w}) \equiv \sum\nolimits_{i=1}^{l} \xi(\boldsymbol{w}; \boldsymbol{x}_i, y_i),$$

is the sum of training losses, $\xi(\boldsymbol{w}; \boldsymbol{x}, y)$ is the loss function, and C is a user-specified regularization parameter to balance the regularization term $\|\boldsymbol{w}\|^2/2$ and the loss term $L(\boldsymbol{w})$. LR and linear SVM consider the following loss functions.

$$\xi(\boldsymbol{w}; \boldsymbol{x}, y) \equiv \begin{cases} \log(1 + e^{-y\boldsymbol{w}^T\boldsymbol{x}}) & \text{logistic (LR) loss,} \\ \max(0, 1 - y\boldsymbol{w}^T\boldsymbol{x}) & \text{L1 loss,} \\ \max(0, 1 - y\boldsymbol{w}^T\boldsymbol{x})^2 & \text{L2 loss.} \end{cases} \tag{3}$$

As indicated in [20], these loss functions have different differentiability, so applicable optimization methods may vary.

Instead of solving problem (2) of variable $\boldsymbol{w}$, it is well known that the optimal $\boldsymbol{w}$ can be represented as a linear combination of training data with coefficient $\boldsymbol{\alpha} \in R^l$.

$$\boldsymbol{w} = \sum\nolimits_{i=1}^{l} y_i \alpha_i \boldsymbol{x}_i. \tag{4}$$

Then one can solve an optimization problem over $\boldsymbol{\alpha}$. An example is the following dual problem of (2).

$$\begin{aligned} \max_{\boldsymbol{\alpha}} \quad & f^D(\boldsymbol{\alpha}) \equiv \sum\nolimits_{i=1}^{l} h(\alpha_i, C) - \frac{1}{2}\boldsymbol{\alpha}^T\bar{Q}\boldsymbol{\alpha} \\ \text{subject to} \quad & 0 \le \alpha_i \le U, \forall i = 1, \ldots, l, \end{aligned} \tag{5}$$

where $\bar{Q} = Q + D \in R^{l\times l}$, $Q_{ij} = y_i y_j \boldsymbol{x}_i^T \boldsymbol{x}_j$, D is diagonal with $D_{ii} = d, \forall i$, and

$$U = \begin{cases} C \\ \infty \end{cases} \quad d = \begin{cases} 0 & \text{for L1-loss SVM and LR,} \\ \frac{1}{2C} & \text{for L2-loss SVM.} \end{cases}$$

[1] `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/warm-start/`.

Following [20], we have $h(\alpha_i, C) =$

$$\begin{cases} \alpha_i & \text{for L1-loss and L2-loss SVM,} \\ C\log C - \alpha_i \log \alpha_i - (C-\alpha_i)\log(C-\alpha_i) & \text{for LR.} \end{cases}$$

We refer to (2) as the primal problem.

We define some notations for later use.

$$Y \equiv \begin{bmatrix} y_1 & & \\ & \ddots & \\ & & y_l \end{bmatrix} \text{ and } X \equiv \begin{bmatrix} \boldsymbol{x}_1^T \\ \vdots \\ \boldsymbol{x}_l^T \end{bmatrix} \in R^{l\times n}. \quad (6)$$

Further, $\boldsymbol{e} \equiv [1, 1, \ldots, 1]^T \in R^l$ and $\mathbf{0} \equiv [0, 0, \ldots, 0]^T$.

2. OPTIMIZATION PROBLEMS AND REGULARIZATION PARAMETERS

This section studies the relationship between the optimal solution and the regularization parameter C. We focus on the case of large C because the optimization problems become more difficult.[2] We separately discuss primal and dual solutions in Sections 2.1 and 2.2, respectively. All proofs are in the supplementary materials.

2.1 Primal Solutions

Because $f(\boldsymbol{w})$ is strongly convex in $\boldsymbol{w}$, an optimal solution of (2) exists and is unique [1, Lemma 2.33]. We define $\boldsymbol{w}_C$ as the unique solution under parameter C, and W_∞ as the set of points that attain the minimum of $L(\boldsymbol{w})$.

$$W_\infty \equiv \{\boldsymbol{w} \mid L(\boldsymbol{w}) = \inf_{\boldsymbol{w}'} L(\boldsymbol{w}')\}.$$

We are interested in the asymptotic behavior of the solution $\boldsymbol{w}_C$ when $C \to \infty$. The following theorem shows that $\{\boldsymbol{w}_C\}$ converges to a point in W_∞.

Theorem 1 *Consider any nonnegative and convex loss function $\xi(\boldsymbol{w}; \boldsymbol{x}, y)$. If $W_\infty \neq \phi$, then*

$$\lim_{C\to\infty} \boldsymbol{w}_C = \boldsymbol{w}_\infty, \text{ where } \boldsymbol{w}_\infty = \arg\min_{\boldsymbol{w}\in W_\infty} \|\boldsymbol{w}\|^2. \quad (7)$$

Next we check if Theorem 1 is applicable to the three loss functions in (3). It is sufficient to prove that $W_\infty \neq \phi$.

2.1.1 L1-loss and L2-loss SVM

For L1 loss, the asymptotic behavior of $\{\boldsymbol{w}_C\}$ was studied in [12]. Theorem 3 of [12] proves that there exist C^* and $\boldsymbol{w}^*$ such that $\boldsymbol{w}_C = \boldsymbol{w}^*, \forall C \geq C^*$. Later in Theorem 6, we prove the same result by Theorem 1 and properties in [11]. To see if $W_\infty \neq \phi$ needed by Theorem 1 holds, we have that $\inf_{\boldsymbol{w}} L(\boldsymbol{w})$ can be written as the following linear program.

$$\min_{\boldsymbol{w},\boldsymbol{\xi}} \sum_{i=1}^l \xi_i$$
$$\text{subject to } \xi_i \geq 1 - y_i \boldsymbol{w}^T \boldsymbol{x}_i, \quad \xi_i \geq 0, \quad \forall i.$$

It has a feasible solution $(\boldsymbol{w}, \boldsymbol{\xi}) = (\mathbf{0}, \boldsymbol{e})$, and the objective value is non-negative, so from [17, Theorem 4.2.3 (i)], a minimum is attained and $W_\infty \neq \phi$.

For L2-loss SVM, the situation is similar. The following theorem shows that $W_\infty \neq \phi$.

Theorem 2 *If L2 loss is used, then $W_\infty \neq \phi$ and $\{\boldsymbol{w}_C\}$ converges to $\boldsymbol{w}_\infty$.*

[2]This has been mentioned in, for example, [10].

2.1.2 Logistic Regression

For LR, the situation is slightly different from that of linear SVM because W_∞ may not exist. We explain below that if the data set is separable, then it is possible that

$$\inf_{\boldsymbol{w}} L(\boldsymbol{w}) = 0 \quad (8)$$

and no minimum is attained because $L(\boldsymbol{w}) > 0, \forall \boldsymbol{w}$. For separable data, generally there exists a vector $\boldsymbol{w}$ such that

$$y_i \boldsymbol{w}^T \boldsymbol{x}_i > 0, \forall i.$$

Then (8) holds because

$$L(\Delta \boldsymbol{w}) = \sum_{i=1}^l \log(1 + e^{-y_i \Delta \boldsymbol{w}^T \boldsymbol{x}_i}) \to 0 \text{ as } \Delta \to \infty.$$

The above discussion indicates that only if data are not separable may we have a non-empty W_∞. The following definition formally defines non-separable data.

Definition 1 *A data set is not linearly separable if for any $\boldsymbol{w} \neq \mathbf{0}$, there is an instance $\boldsymbol{x}_i$ such that*

$$y_i \boldsymbol{w}^T \boldsymbol{x}_i < 0. \quad (9)$$

We have the following theorem on the convergence of $\{\boldsymbol{w}_C\}$.

Theorem 3 *If LR loss is used and the non-separable condition (9) holds, then $\boldsymbol{w}_\infty$ exists and $\{\boldsymbol{w}_C\}$ converges to $\boldsymbol{w}_\infty$.*

2.2 Dual Solutions

Let $\boldsymbol{\alpha}_C$ be any optimal solution of the dual problem (5). Subsequently we will investigate the relationship between $\boldsymbol{\alpha}_C$ and C. An important difference from the analysis for primal solutions is that $\boldsymbol{\alpha}_C$ may not be unique. This situation occurs if the dual objective function is only convex rather than strictly convex (e.g., L1 loss). Therefore, we analyze L2 and LR losses first because their $\boldsymbol{\alpha}_C$ is unique.

2.2.1 L2-loss SVM and Logistic Regression

The following theorem shows the asymptotic relationship between $\boldsymbol{\alpha}_C$ and C.

Theorem 4 *If L2 loss is used, then*

$$\lim_{C\to\infty} \frac{(\alpha_C)_i}{C} = 2\max(0, 1 - y_i \boldsymbol{w}_\infty^T \boldsymbol{x}_i), i = 1, \ldots, l. \quad (10)$$

If LR loss is used and the non-separable condition (9) is satisfied, then

$$\lim_{C\to\infty} \frac{(\alpha_C)_i}{C} = \frac{e^{-y_i \boldsymbol{w}_\infty^T \boldsymbol{x}_i}}{1 + e^{-y_i \boldsymbol{w}_\infty^T \boldsymbol{x}_i}}, i = 1, \ldots, l. \quad (11)$$

From Theorem 4, $\|\boldsymbol{\alpha}_C\|$ is unbounded for non-separable data because the right-hand side in (10) and (11) is non-zero. Therefore, we immediately have the following theorem.

Theorem 5 *If the problem is not linearly separable, then*

$$\|\boldsymbol{\alpha}_C\| \to \infty \text{ as } C \to \infty.$$

Theorem 4 indicates that for non-separable data, $\boldsymbol{\alpha}_C$ is asymptotically a linear function of C. In Section 3.1, we will use this property to choose the initial solution after C is increased. For separable data, the right-hand side in (10) and (11) may be zero, so the asymptotic linear relationship between $\boldsymbol{\alpha}_C$ and C may not hold. For simplicity, we omit giving detailed analysis on separable data. Further, such data are often easier for training.

2.2.2 L1-loss SVM

Although for L1 loss, the dual optimal $\boldsymbol{\alpha}_C$ may not be unique, there exists a solution path from the earlier work in [11].[3] We extend their result to have the following theorem.

Theorem 6 *If L1 loss is used, then there are vectors $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$, and a threshold C^* such that after $C \geq C^*$,*

$$\boldsymbol{\alpha}_C \equiv \boldsymbol{v}_1 C + \boldsymbol{v}_2 \tag{12}$$

is a dual optimal solution. The primal solution

$$\boldsymbol{w}_C = (YX)^T \boldsymbol{\alpha}_C$$

is a constant vector the same as $\boldsymbol{w}_\infty$, where X and Y are defined in (6)*. Further,*

$$(YX)^T \boldsymbol{v}_1 = \mathbf{0} \text{ and } (YX)^T \boldsymbol{v}_2 = \boldsymbol{w}_\infty.$$

The following extension of Theorem 6 shows that, similar to Theorem 4, most elements of $\boldsymbol{\alpha}_C$ become a multiple of C.

Theorem 7 *Assume L1 loss is used. There exists a C^* such that after $C \geq C^*$, any dual optimal solution $\boldsymbol{\alpha}_C$ satisfies*

$$(\alpha_C)_i = C \text{ if } y_i \boldsymbol{w}_\infty^T \boldsymbol{x}_i < 1,$$
$$(\alpha_C)_i = 0 \text{ if } y_i \boldsymbol{w}_\infty^T \boldsymbol{x}_i > 1.$$

This theorem can be easily obtained by the fact that $\boldsymbol{w}_C = \boldsymbol{w}_\infty, \forall C \geq C^*$ from Theorem 6 and the optimality condition. Like Theorem 5, we immediately get the unboundedness of $\{\boldsymbol{\alpha}_C\}$ from Theorem 7.

3. WARM START FOR PARAMETER SELECTION

In this section, we investigate issues in applying the warm-start strategy for solving a sequence of optimization problems under different C values. The purpose is to select the parameter C that achieves the best CV accuracy.

3.1 Selection of Initial Solutions

We consider the situation when

$$C \text{ is increased to } \Delta C,$$

where $\Delta > 1$. At the current C, let $\boldsymbol{w}_C$ be the unique primal optimal solution, while $\boldsymbol{\alpha}_C$ be any dual optimal solution. We then discuss suitable initial solutions $\bar{\boldsymbol{w}}$ and $\bar{\boldsymbol{\alpha}}$ for the new problem with the parameter ΔC.

From Theorems 1 and 6, $\boldsymbol{w}_C$ is closed to (or exactly the same as) $\boldsymbol{w}_\infty$ after C is large enough, so naturally

$$\bar{\boldsymbol{w}} = \boldsymbol{w}_C \tag{13}$$

can be an initial solution. If a dual problem is solved, from Theorem 4,

$$\bar{\boldsymbol{\alpha}} = \Delta \boldsymbol{\alpha}_C \tag{14}$$

is suitable for L2 or LR loss. We explain that (14) is also useful for L1 loss although in Theorem 6, $\boldsymbol{\alpha}_C$ is not a multiple of C, and $\boldsymbol{\alpha}_C$ is only one of the optimal solutions. Instead, we consider Theorem 7, where $\boldsymbol{\alpha}_C$ is any optimal solution rather than the specific one in (12) of Theorem 6. Because in general $\boldsymbol{y}_i \boldsymbol{w}_\infty^T \boldsymbol{x}_i \neq 1$, most of $\boldsymbol{\alpha}_C$'s elements are either zero or C, and hence they are multiples of C. Thus, (14) is a reasonable initial solution. Note that if $\boldsymbol{\alpha}_C$ is feasible for (5) under C, then $\bar{\boldsymbol{\alpha}}$ also satisfies the constraints under ΔC. Approaches similar to (14) to set the initial point for warm start can be found in some previous works such as [5].

[3] In [11], the loss function has an additional bias term: $\max(0, 1 - y(\boldsymbol{w}^T \boldsymbol{x} + b))$. A careful check shows that their results hold without b.

3.2 A Comparison on the Effectiveness of Primal and Dual Initial Solutions

In this subsection, we show that from some aspects, warm start may be more useful for a primal-based training method.

First, from the primal-dual relationship (4), we have

$$\boldsymbol{w}_C = \sum_{i=1}^{l} (\alpha_C)_i y_i \boldsymbol{x}_i. \tag{15}$$

However, by Theorems 1 and 5, interestingly

$$\boldsymbol{w}_C \to \boldsymbol{w}_\infty \text{ but } \|\boldsymbol{\alpha}_C\| \to \infty.$$

Therefore, on the right-hand side of (15) apparently some large values in $\boldsymbol{\alpha}_C$ are cancelled out after taking $y_i \boldsymbol{x}_i$ into consideration. The divergence of $\boldsymbol{\alpha}_C$ tends to cause more difficulties in finding a good initial dual solution.

Next, we check primal and dual initial objective values after applying warm start. The initial $\bar{\boldsymbol{\alpha}}$ based on (14) gives the following dual objective value. For simplification, we use $\boldsymbol{w}$ and $\boldsymbol{\alpha}$ to denote $\boldsymbol{w}_C$ and $\boldsymbol{\alpha}_C$, respectively. Then,

$$\begin{aligned}
& h(\Delta\boldsymbol{\alpha}, \Delta C) - \frac{1}{2}\Delta^2 \boldsymbol{\alpha}^T \left(Q + \frac{D}{\Delta}\right) \boldsymbol{\alpha} \\
&= \Delta h(\boldsymbol{\alpha}, C) - \frac{1}{2}\Delta^2 \boldsymbol{\alpha}^T \left(Q + \frac{D}{\Delta}\right) \boldsymbol{\alpha} && (16) \\
&= \Delta \left(\frac{1}{2}\boldsymbol{\alpha}^T (Q + D)\boldsymbol{\alpha} + \frac{1}{2}\boldsymbol{w}^T \boldsymbol{w} + C \sum_{i=1}^{l} \xi(\boldsymbol{w}; \boldsymbol{x}_i, y_i)\right) \\
&\quad - \frac{1}{2}\Delta^2 \boldsymbol{\alpha}^T \left(Q + \frac{D}{\Delta}\right) \boldsymbol{\alpha} && (17) \\
&= C\Delta \sum_{i=1}^{l} \xi(\boldsymbol{w}; \boldsymbol{x}_i, y_i) + (\Delta - \Delta^2/2)\boldsymbol{w}^T \boldsymbol{w}, && (18)
\end{aligned}$$

where (16) is from

$$h(\Delta\boldsymbol{\alpha}, \Delta C) = \Delta h(\boldsymbol{\alpha}, C)$$

in Eq. (31) of [20], (17) is from that primal and dual optimal objective values are equal, and (18) is from that at optimum,

$$\boldsymbol{\alpha}^T Q \boldsymbol{\alpha} = \boldsymbol{w}^T \boldsymbol{w}.$$

Note that

$$\Delta - \Delta^2/2$$

is a decreasing function after $\Delta \geq 1$. If $\Delta = 2$, we have

$$\begin{aligned}
(18) &= 2C \sum_{i=1}^{l} \xi(\boldsymbol{w}; \boldsymbol{x}_i, y_i) \\
&\leq \text{primal or dual optimal objective value at } 2C \\
&\leq \frac{1}{2}\bar{\boldsymbol{w}}^T \bar{\boldsymbol{w}} + 2C \sum_{i=1}^{l} \xi(\bar{\boldsymbol{w}}; \boldsymbol{x}_i, y_i). && (19)
\end{aligned}$$

Note that $\bar{\boldsymbol{w}} = \boldsymbol{w}$ from (13). If $\bar{\boldsymbol{w}}$ is close to $\boldsymbol{w}_\infty$, then the primal initial objective value should be close to the optimum. In contrast, from (19) we can see that the dual initial objective value lacks a $\boldsymbol{w}^T \boldsymbol{w}/2$ term. Therefore, it should be less close to the optimal objective value.

Another difference between primal- and dual-based approaches is on the use of high-order (e.g., Newton) or low-order (e.g., gradient) optimization methods. It has been pointed out in [20] that a high-order method takes more advantages of applying warm start. The reason is that if an initial solution is close to the optimum, a high-order optimization method leads to fast convergence. Because the primal problem is unconstrained, it is easier to solve it by a high-order optimization method. In Section 4, we will experimentally confirm results discussed in this Section.

3.3 Range of Regularization Parameters

In conducting parameter selection in practice, we must specify a range of C values that covers the best choice. Because small and large C values cause underfitting and overfitting, respectively, and our procedure gradually increases C, all we need is a lower and an upper bound of C.

3.3.1 Lower Bound of C

We aim at finding a C value so that for all values smaller than it, underfitting occurs. If the optimal $\boldsymbol{w}$ satisfies

$$y_i \boldsymbol{w}^T \boldsymbol{x}_i < 1, \forall i, \tag{20}$$

then we consider that underfitting occurs. The reason is that every instance imposes a non-zero L1 or L2 loss.

We then check for what C values, (20) holds. For L1 and LR losses, if

$$C < \frac{1}{l \max_i \|\boldsymbol{x}_i\|^2}, \tag{21}$$

then from (4) and the property $\alpha_i \le C$,

$$y_i \boldsymbol{w}^T \boldsymbol{x}_i \le |\boldsymbol{w}^T \boldsymbol{x}_i| \le \sum\nolimits_{j=1}^{l} |\alpha_j| |\boldsymbol{x}_j^T \boldsymbol{x}_i| < 1.$$

For L2-loss SVM, if

$$C < \frac{1}{2l \max_i \|\boldsymbol{x}_i\|^2}, \tag{22}$$

then

$$\begin{aligned} y_i \boldsymbol{w}^T \boldsymbol{x}_i &\le \|\boldsymbol{w}\| \|\boldsymbol{x}_i\| \le \sqrt{\|\boldsymbol{w}\|^2 + 2CL(\boldsymbol{w})} \max_j \|\boldsymbol{x}_j\| \quad (23) \\ &\le \sqrt{2f(\boldsymbol{0})} \max_j \|\boldsymbol{x}_j\| \le \sqrt{2Cl} \max_j \|\boldsymbol{x}_j\| < 1, \end{aligned}$$

where (23) is from that $\boldsymbol{w}$ minimizes $f(\cdot)$.

The value derived in (21) and (22) can be considered as the initial $C_{\min}$ for the parameter search. One concern is that it may be too small so that many unneeded optimization problems are solved. In Section 4.4, we show that optimization methods are very fast when C is small, so the cost of considering some small and useless C values is negligible.

3.3.2 Upper Bound of C

Clearly, an upper bound should be larger than the best C. However, unlike the situation in finding a lower bound, where optimization problems with small C values are easy to solve, wrongly considering a too large C value can dramatically increase the running time.

In an earlier study [14] on parameter selection for general classification problems, they terminate the search procedure if CV accuracy is not enhanced after a few trials. While the setting is reasonable, the following issues occur.

- For small C values, the CV accuracy is stable, so the selection procedure may stop pre-maturely. See more discussions in Section II of the supplementary materials.
- The CV accuracy may not be a monotonic increasing function before the best C. If the CV accuracy is locally stable or even becomes lower in an interval, than the procedure may stop before the best C.

For linear classification, based on results in Section 2, we will describe a setting to terminate the search process by checking the change of optimal solutions.

In Section 2, we showed that $\{\boldsymbol{w}_C\}$ converge to $\boldsymbol{w}_\infty$, and for LR and L2-loss SVM, $\{\boldsymbol{\alpha}_C/C\}$ converges. Therefore, if for several consecutive C values, $\{\boldsymbol{w}_C\}$ or $\{\boldsymbol{\alpha}_C/C\}$ does not change much, then $\boldsymbol{w}_C$ should be close to $\boldsymbol{w}_\infty$ and there is no need to further increase the C value; see more theoretical support later in Theorem 8. We thus propose a setting of increasing C until either

- $C \le \bar{C}$, where $\bar{C}$ is a pre-specified constant, or
- optimal $\boldsymbol{w}_C$ or $\boldsymbol{\alpha}_C/C$ is about the same as previous solutions.

Deriving an upper bound $\bar{C}$ is more difficult than a lower bound in Section 3.3.1. However, with the second condition, a tight upper bound may not be needed. Thus in our experiment, we simply select a large value.

For the second condition of checking the change of optimal solutions, our idea is as follows. For optimal $\boldsymbol{w}_{C/\Delta}$ or $\boldsymbol{\alpha}_{C/\Delta}$ at C/Δ, we see if it is a good approximate optimal solution at the current C. For easy description, we rewrite $f(\boldsymbol{w})$ as

$$f(\boldsymbol{w}; C)$$

to reflect the regularization parameter. We stop the procedure for parameter selection if

$$\|\nabla f(\boldsymbol{w}_{\Delta^{t-1}C}; \Delta^t C)\| \le \epsilon \|\nabla f(\boldsymbol{0}; \Delta^t C)\| \quad \text{for } t = -2, -1, 0, \tag{24}$$

where ϵ is a pre-specified small positive value. The condition (24) implies that $\boldsymbol{w}_{\Delta^{t-1}C}$, the optimal solution for the previous $\Delta^{t-1}C$, is an approximate solution for minimizing the function $f(\boldsymbol{w}; \Delta^t C)$. We prove the following theorem to support our use of (24). In particular, we check the relationship between $\boldsymbol{w}_{C/\Delta}$ and $\nabla f(\boldsymbol{w}; C)$

Theorem 8 *Assume $L(\boldsymbol{w})$ is continuously differentiable.*

1. *If non-separable condition (9) holds and $\|\boldsymbol{w}_\infty\| > 0$, then*

$$\boldsymbol{w}_{C_1} \ne \boldsymbol{w}_{C_2}, \forall C_1 \ne C_2. \tag{25}$$

2. *We have*

$$\lim_{C \to 0} \frac{\|\nabla f(\boldsymbol{w}_{C/\Delta}; C)\|}{\|\nabla f(\boldsymbol{0}; C)\|} = \frac{\Delta - 1}{\Delta}, \text{ and} \tag{26}$$

$$\lim_{C \to \infty} \frac{\|\nabla f(\boldsymbol{w}_{C/\Delta}; C)\|}{\|\nabla f(\boldsymbol{0}; C)\|} = 0. \tag{27}$$

The result (25) indicates that in general $\boldsymbol{w}_{C/\Delta} \ne \boldsymbol{w}_C$, so (24) does not hold. One exception is when C is large, $\boldsymbol{w}_{C/\Delta} \approx \boldsymbol{w}_C \approx \boldsymbol{w}_\infty$ from Theorem 1. Then (24) will eventually hold and this property is indicated by (27). On the contrary, when C is small, from (26) and the property that

$$\Delta \ge \frac{1}{1-\epsilon} \text{ implies } \frac{\Delta - 1}{\Delta} \ge \epsilon,$$

if Δ is not close to one, (24) does not hold. Therefore, our procedure does not stop pre-maturely.

Note that (24) can be applied regardless of whether a primal-based or a dual-based optimization method is used.

Algorithm 1 A complete procedure for parameter section.

1. Given K as number of CV folds.
2. Initialize $C_{\min}$ by (21) or (22), $C_{\max}$ by a constant.
3. Initialize $C_{\text{best}} \leftarrow C_{\min}$, best CV accuracy $A \leftarrow -\infty$.
4. For each CV fold k, give initial $\bar{\boldsymbol{w}}^k$.
5. For $C = C_{\min}, \Delta C_{\min}, \Delta^2 C_{\min}, \ldots, C_{\max}$:
   5.1. For each CV fold $k = 1, \ldots, K$:
      5.1.1. Use all data except fold k for training.
      Apply warm start with the initial point $\bar{\boldsymbol{w}}^k$.
      Obtain the solution $\boldsymbol{w}_C^k$.
      5.1.2. Predict fold k by $\boldsymbol{w}_C^k$.
   5.2. Obtain CV accuracy using results obtained in 5.1.
      If the new CV accuracy $> A$:
      $A \leftarrow$ the new CV accuracy.
      $C_{\text{best}} \leftarrow C$.
   5.3. If (24) is satisfied:
      break
   5.4. For each CV fold k, $\bar{\boldsymbol{w}}^k \leftarrow \boldsymbol{w}_C^k$.
6. Return C_{best}.

If a dual-based method is considered, an optimal $\boldsymbol{w}_C$ is returned by (4) and we can still check (24).

3.4 The Overall Procedure

With all results ready, we propose in Algorithm 1 a practically useful procedure for selecting the parameter of a linear classifier. We evaluate the CV accuracy from the smallest parameter $C_{\min}$, and gradually increase it by a factor Δ; see the sequence of C values shown in (1). In the CV procedure, if the kth fold is used for validation, then all the remaining folds are for training. Therefore, several sequences of optimization problems are solved. Although it is possible to separately handle each sequence, here we consider them together. That is, at each C, the training/prediction tasks on all folds are conducted to obtain the CV accuracy. Then either the procedure is terminated or we go to the next ΔC. Regarding the storage, all we need is to maintain the K vectors of $\boldsymbol{w}$, where K is the number of CV folds.

In Algorithm 1, we see the change of primal solutions is checked for terminating the search process. Although both primal- and dual-based optimization methods can be used (from $\boldsymbol{\alpha}_C$, primal $\boldsymbol{w}_C$ can be easily generated), we expect that a primal-based method is more suitable because of the following properties.

- Primal solution $\boldsymbol{w}_C$ is unique regardless of the loss function, but dual solution $\boldsymbol{\alpha}_C$ may not be unique for L1 loss.
- Primal solutions converge as C increases, but dual solutions diverge (Sections 2 and 3.2).
- After applying warm start to have initial solutions, the primal objective value tends to be closer to the optimum than the dual (Section 3.2).
- The warm start strategy is more effective for a high-order optimization approach (e.g., Newton method). Because the primal problem (2) has no constraints, it is easier to design a high-order solver. ([20] and Section 3.2).

Further, [20, Section 5] has pointed out that implementation issues such as maintaining $\boldsymbol{\alpha}$ make dual solvers more challenging to support warm start. We will detailedly compare implementations using primal- and dual-based methods in Section 4 and supplementary materials.

Table 1: Data statistics: Density is the average ratio of non-zero features per instance.

Data set	l: #instances	n: #features	density	best C
madelon	2,000	500	100.00%	2^{-25}
ijcnn	49,990	22	59.09%	2^5
webspam	350,000	16,609,143	0.02%	2^2
rcv1	677,399	47,236	0.15%	$> 2^{10}$
yahoo-japan	176,203	832,026	0.02%	2^3
news20	19,996	1,355,191	0.03%	$> 2^{10}$

4. EXPERIMENTS

We conduct experiments to verify the results discussed in Sections 2 and 3, and check the performance of Algorithm 1. Because of the space limitation, we present only results of LR, but leave L2-loss SVM in the supplementary materials.[4] We consider six data sets madelon, ijcnn, webspam (trigram version), news20, rcv1, and yahoo-japan with statistics in Table 1,[5] where the last column is the best C with the highest CV accuracy. We tried some large C to empirically confirm that all these sets are not separable.

We consider $C_{\min}$ to be the largest 2^n that satisfies (21). Following the discussion in (19), we use $\Delta = 2$, so the sequence of C values in (1) contains only powers of two. For $C_{\max}$, we set it to be a large number 2^{10}, because as indicated in Section 3.3, another condition (24) will be mainly used to stop our procedure.

Our implementations are extended from LIBLINEAR [6], and we use five-fold CV. We consider two optimization methods in our experiments. One is a dual coordinate descent method [10] for solving the dual problem, while the other is a Newton method [16] for solving the primal problem. At $C_{\min}$, because of no prior information for warm start, the default initial point in LIBLINEAR is used; see Step 4 in Algorithm 1. The two optimization methods have their respective stopping conditions implemented in LIBLINEAR. To fairly compare them, we modify the condition of the dual-based method to be the same as the primal one:[6]

$$\|\nabla f(\boldsymbol{w})\| \le \epsilon \frac{\min(l^+, l^-)}{l} \|\nabla f(\mathbf{0})\|, \tag{28}$$

where l^+ and l^- are the numbers of instances labelled $+1$ and -1, respectively. This condition is related to (24) used for terminating the parameter search. If not specified, the default $\epsilon = 10^{-2}$ in LIBLINEAR is used in our experiments. We also change some settings of the solvers. Details are in Section III of supplementary materials. Experiments are conducted on two four-core computers with 2.0GHz/32GB RAM and 2.5GHz/16GB RAM for webspam and other data sets, respectively.

4.1 CV Accuracy and Training Time

We investigate in Figure 1 the relation between $\log_2 C$ (x-axis) and CV accuracy (y-axis on the left). The purpose is to check the convergence of $\{\boldsymbol{w}_C\}$ proved in Section 2. We

[4] L1-loss SVM is not considered because the primal-based optimization method considered here cannot handle non-differentiable losses.

[5] All data sets except yahoo-japan are available at `http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets`.

[6] For any dual-based optimization method, we can easily obtain an approximate primal solution by (4). On the contrary, we may not be able to modify a primal-based method to use the stopping condition of a dual-based method because from a primal $\boldsymbol{w}$ it is difficult to generate a dual $\boldsymbol{\alpha}$.

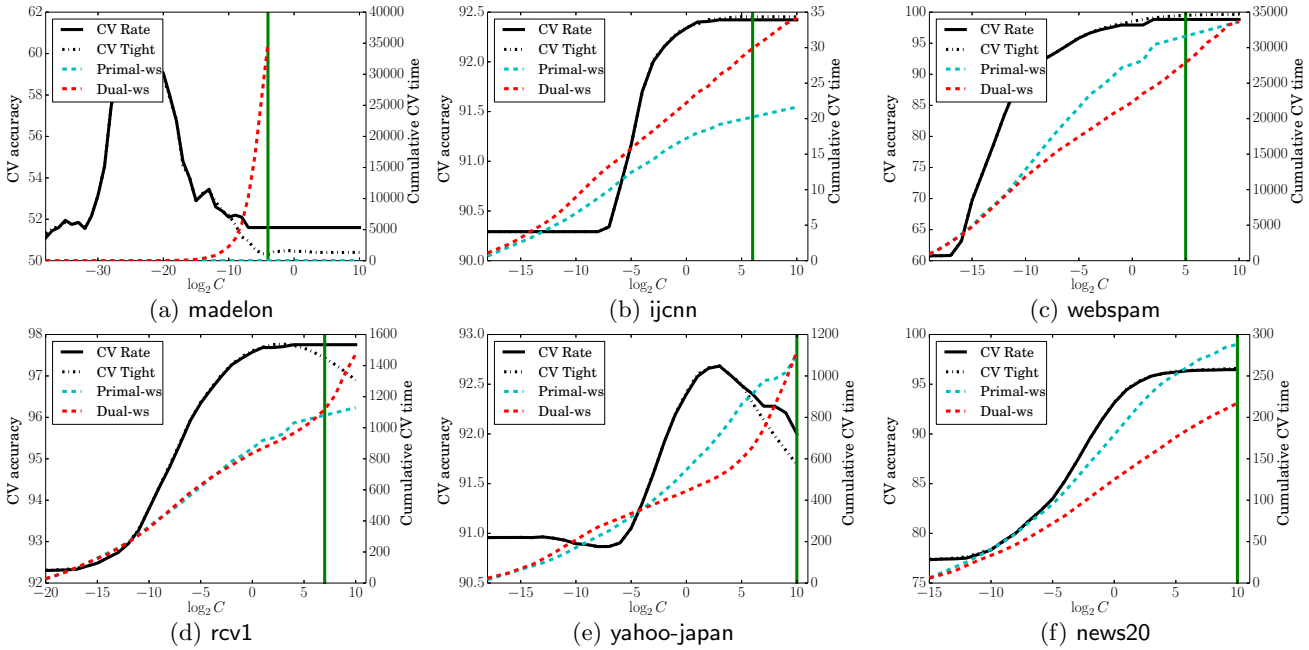


Figure 1: CV accuracy and training time using LR with warm start. The two CV curves and the left y-axis are the CV accuracy in percentage (%). The dashed lines and the right y-axis are the cumulative training time in the CV procedure in seconds. The vertical line indicates the last C value checked by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

show CV rates of using (28) with $\epsilon = 10^{-2}$ and 10^{-6} as the stopping condition for finding $\boldsymbol{w}_C$; see "CV Rate" and "CV Tight" in Figure 1, respectively. We have the following observations. First, for most problems, the CV accuracy is stabilized when C is large. This result confirms the theoretical convergence of $\{\boldsymbol{w}_C\}$. However, for the data set yahoo-japan, the CV accuracy keeps changing even when C is large. This situation may be caused by the following reasons.

- $\boldsymbol{w}_C$ is not close enough to $\boldsymbol{w}_\infty$ yet even though a large $C = 2^{10}$ has been used.
- Training a problem under a large C is so time-consuming that the obtained $\boldsymbol{w}_C$ is still far away from the optimum; see the significant difference of CV rates between using loose and strict stopping tolerances for madelon, rcv1, and yahoo-japan.[7] Later in this section we will discuss more about the relation between training time and C.

The above observation clearly illustrates where the difficulty of parameter selection for linear classification lies. The area of large C values is like a danger zone. Usually the best C is not there, but if we wrongly get into the region, not only does the lengthy training time may occur, but also the obtained CV rates may be erroneous.

The second observation is that when C is small, the CV accuracy is almost flat; see explanation in Section II of the supplementary materials. Therefore, if we use the method in [14] to check the change of CV accuracy, the search procedure may stop too early. In contrast, we explained in Section 3.3.2 that our method will not stop until $\boldsymbol{w}_C$ is close to $\boldsymbol{w}_\infty$.

[7]It is surprising to see that for rcv1, a loose condition gives flat CV rates, a situation closer to the convergence of $\{\boldsymbol{w}_C\}$, but a strict condition does not give that. The reason is that because of using warm start, the initial $\bar{\boldsymbol{w}}$ from $\boldsymbol{w}_C$ immediately satisfies the loose stopping condition at ΔC, so both the optimal solution and CV rate remain the same.

In Figure 1, we also show the cumulative CV training and validation time of Algorithm 1 from $C_{\min}$ to the current C; see the dashed lines and the y-axis on the right. The curve of cumulative time is up-bended because both solvers become slower as C increases. Except news20, this situation is more serious for the dual solver.[8] A reason is that our dual solver is a low-order optimization method. When C is large, the problem becomes harder to solve, and a high-order method such as the Newton method for the primal problem tends to perform better. See more discussion in Section 4.3.

To remedy the problem of lengthy training when C is large, recall in Section 3.3.2 we proposed a method to stop the procedure according to the stopping condition of the solver. The ending point of our procedure is indicated by a vertical line in Figure 1. Clearly, we successfully obtain CV accuracy close to the highest in the entire range.

Figure 1 confirms that we rightly choose $C_{\min}$ smaller than the best C. Although $C_{\min}$ tends to be too small, in Figure 1, the training time for small C values is insignificant.

4.2 Initial Objective Values

We verify our discussion on primal and dual initial objective values in (19). If $\boldsymbol{w}_C \approx \boldsymbol{w}_{\Delta C} \approx \boldsymbol{w}_\infty$, then the initial objective value of primal solvers should be close to the optimum, while the dual initial objective value is asymptoticly smaller than the primal by $\|\bar{\boldsymbol{w}}_\infty\|^2/2$.

In Table 2, we show the difference between initial and optimal objective values.

$$f(\bar{\boldsymbol{w}}) - f(\boldsymbol{w}_C) \text{ and } f^D(\bar{\boldsymbol{\alpha}}) - f^D(\boldsymbol{\alpha}_C).$$

Note that $f(\boldsymbol{w}_C)$ and $f^D(\boldsymbol{\alpha}_C)$ are equal. Because the optimal $\boldsymbol{w}_C$ and $\boldsymbol{\alpha}_C$ are difficult to compute, we obtain their

[8]Note that warm start has been applied. If not, the time increase at large C values is even more dramatic.

Table 2: Difference between the initial and optimal function values. Logistic regression 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	**2.51e−03**	−1.09e−01	4.57e−02	**1.30e+01**	−4.55e+01	5.85e+01	**1.63e+02**	−3.86e+02	5.49e+02
0	**2.62e−04**	−1.45e+00	5.72e−02	**1.31e+01**	−1.90e+02	2.03e+02	**1.12e+03**	−3.26e+03	4.38e+03
4	**0.00e+00**	−1.29e+01	5.82e−02	**1.50e+00**	−2.64e+02	2.66e+02	**8.28e+03**	−2.33e+04	3.16e+04
8	**0.00e+00**	−1.17e+02	5.83e−02	**9.86e−02**	−2.71e+02	2.71e+02	**5.32e+04**	−1.76e+05	2.29e+05
	rcv1			yahoo-japan			news20		
−4	**3.23e+02**	−1.04e+03	1.36e+03	**6.19e+01**	−1.27e+02	1.89e+02	2.19e+01	**−1.43e+01**	3.62e+01
0	**1.60e+03**	−6.02e+03	7.62e+03	**9.00e+02**	−1.40e+03	2.30e+03	**3.78e+02**	−6.32e+02	1.01e+03
4	**1.20e+04**	−3.33e+04	4.53e+04	**1.85e+04**	−2.66e+04	4.51e+04	**2.26e+03**	−7.91e+03	1.02e+04
8	**9.53e+04**	−2.70e+05	3.66e+05	**1.21e+05**	−4.07e+05	5.28e+05	**5.33e+03**	−3.64e+04	4.17e+04

approximations by running a huge number of iterations. We show results of $C = 2^{-4}, 2^0, 2^4, 2^8$, where $\bar{\boldsymbol{w}}$ and $\bar{\boldsymbol{\alpha}}$ are obtained by solving problems of $C = 2^{-5}, 2^{-1}, 2^3, 2^7$ and applying (13) and (14). We use $\epsilon = 10^{-6}$ in (28) for this experiment to ensure that the solution of the previous problem is accurate enough. In Table 2, we also show $\|\bar{\boldsymbol{w}}\|^2/2$ to see if, as indicated in (19), $f^D(\bar{\boldsymbol{\alpha}}) - f^D(\boldsymbol{\alpha}_C)$ is close to $-\|\bar{\boldsymbol{w}}\|^2/2$ when C is large.

From Table 2, except some rare situations with small C values, primal solvers have function values closer to the optimal value than the dual solvers. Further, as C increases, $f^D(\boldsymbol{\alpha}_C) - f^D(\bar{\boldsymbol{\alpha}})$ becomes close to $\|\bar{\boldsymbol{w}}\|^2/2$ for most problems. Note that from (19),

$$f(\bar{\boldsymbol{w}}) - f(\boldsymbol{w}_C) = \frac{1}{2}(\|\bar{\boldsymbol{w}}\|^2 - \|\boldsymbol{w}_C\|^2) + C\sum_{i=1}^{l}\left(\xi(\bar{\boldsymbol{w}}; \boldsymbol{x}_i, y_i) - \xi(\boldsymbol{w}_C; \boldsymbol{x}_i, y_i)\right), \text{ and} \quad (29)$$

$$f^D(\bar{\boldsymbol{\alpha}}) - f^D(\boldsymbol{\alpha}_C) \quad (30)$$

$$= -\frac{1}{2}\|\boldsymbol{w}_C\|^2 + C\sum_{i=1}^{l}\left(\xi(\bar{\boldsymbol{w}}; \boldsymbol{x}_i, y_i) - \xi(\boldsymbol{w}_C; \boldsymbol{x}_i, y_i)\right).$$

For their first term, $\|\bar{\boldsymbol{w}}\|^2 - \|\boldsymbol{w}_C\|^2 \approx 0$ in (29) as $C \to \infty$, but in (30), it converges to $-\|\boldsymbol{w}_\infty\|^2/2$. The situation of the second term is unclear because $\xi(\bar{\boldsymbol{w}}; \boldsymbol{x}_i, y_i) - \xi(\boldsymbol{w}_C; \boldsymbol{x}_i, y_i) \to 0$ as $C \to \infty$. However, $C\left(\xi(\bar{\boldsymbol{w}}; \boldsymbol{x}_i, y_i) - \xi(\boldsymbol{w}_C; \boldsymbol{x}_i, y_i)\right)$ in Table 2 is relatively smaller than $-\|\boldsymbol{w}_C\|^2/2$, and therefore, the primal initial objective value is closer to the optimal objective value than the dual. Finally, the dual solver fails on the data set madelon because of the slow convergence.

4.3 Effectiveness of Warm-start Strategies

In Figure 2, we compare the running time with/without implementing warm start. Each subfigure presents training time versus the following relative difference to the optimal objective value

$$\frac{f(\boldsymbol{w}) - f(\boldsymbol{w}_C)}{f(\boldsymbol{w}_C)} \text{ and } \frac{f^D(\boldsymbol{\alpha}) - f(\boldsymbol{\alpha}_C)}{f(\boldsymbol{\alpha}_C)},$$

where $\boldsymbol{w}_C$ or $\boldsymbol{\alpha}_C$ is an optimal solution and $\boldsymbol{w}$ or $\boldsymbol{\alpha}$ is any iterate in the optimization process. We use the best C value before the vertical line in Figure 1. The initial point of warm start is by (13) and (14), which use solutions at $C/2$.

If warm start is not applied, results in Figure 2 are consistent with past works such as [10]:

- If the number of features is much smaller than instances and C is not large (madelon and ijcnn), a primal-based method may be suitable because of a smaller number of variables. For the opposite case of more features, a dual-based method may be used (webspam and yahoo-japan).
- If C is large, a high-order optimization approach such as Newton methods is more robust (news20).

In practice, a stopping condition is imposed to terminate the optimization procedure (e.g., running up to the horizontal line in Figure 2, which indicates that the condition (28) with LIBLINEAR's default $\epsilon = 10^{-2}$ has been established).

After applying warm start, both primal and dual solvers become faster. Therefore, warm start effectively reduces the training time. However, Figure 2 focuses on the convergence behavior under a given C. What we are more interested in is the total training time of the parameter-selection procedure. This will be discussed in Section 4.4.

4.4 Performance of the Parameter-selection Procedure

We compare the running time with/without applying warm-start techniques. Figure 3 presents the cumulative CV training and validation time from $C_{\min}$ to the current C. We can make the following observations.

- The total running time (log-scaled in Figure 3) is significantly reduced after applying warm start on both primal- and dual-based optimization methods.
- While the dual coordinate descent method is faster when C is small, its training time dramatically increases for large C values. This result corresponds to our earlier discussion that a low-order optimization method is not suitable for hard situations such as when C is large. Even though warm start significantly reduces the training time, for running up to a large C, it is generally less competitive with the primal Newton method with warm start. Therefore, using a high-order optimization method is a safer option in parameter selection for avoiding lengthy running time.

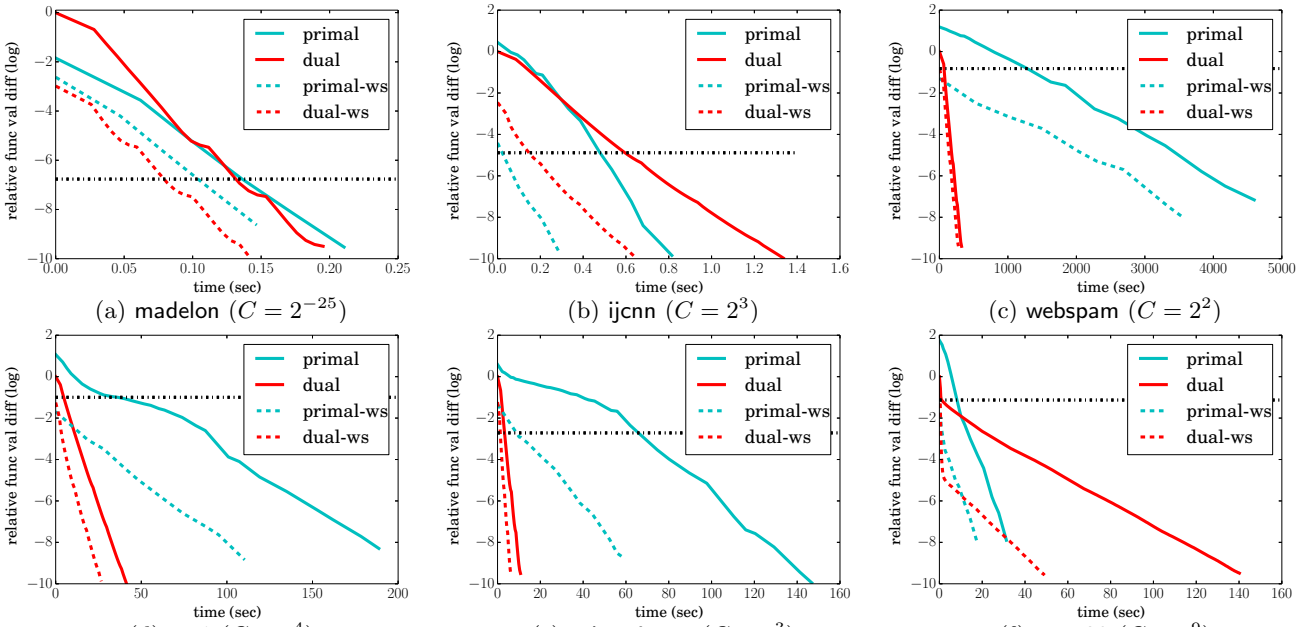
4.5 CV Folds and Models

In Algorithm 1, data are split to K folds for the CV procedure. For each fold k, we maintain a vector $\boldsymbol{w}^k$ that traces initial and optimal solutions all the way from $C_{\min}$ to $C_{\max}$. To reduce the storage, an idea is to maintain only one vector across all folds. If fold 1 serves as a validation set, the training set includes

$$\text{fold } 2,\ \text{fold } 3, \ldots,\ \text{fold } K. \quad (31)$$

Next, for validating fold 2, the training set becomes

$$\text{fold } 1,\ \text{fold } 3, \ldots,\ \text{fold } K. \quad (32)$$



(a) madelon ($C = 2^{-25}$) (b) ijcnn ($C = 2^{3}$) (c) webspam ($C = 2^{2}$)

(d) rcv1 ($C = 2^{4}$) (e) yahoo-japan ($C = 2^{3}$) (f) news20 ($C = 2^{9}$)

Figure 2: Objective values versus training time using LR 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.

Table 3: CV accuracy using a dedicated $\boldsymbol{w}$ for each fold and a shared $\boldsymbol{w}$ for all folds. See details in Section 4.5. The set yahoo-japan is used. The highest CV rate is boldfaced.

C	K models	one model
2^1	92.60	92.66
2^3	**92.69**	92.80
2^5	92.59	92.56
2^7	92.34	94.27
2^9	92.21	**98.01**

The two training sets differ in only two folds: fold 1 and fold 2. We have a scenario of incremental and decremental learning [20], where fold 2 is removed, but fold 1 is added. Then warm start can be applied. Specifically, the optimal solution after training (31) can be used as an initial solution for (32). Although this technique may reduce the storage as well as the training time, we show that practically some difficulties may occur.

In Table 3, we compare the two approaches of using K and one vectors for storing the solutions. The approach of maintaining one vector gives much higher CV accuracy and a larger best C value. An investigation shows that two reasons together cause an over-estimation.

- In training folds 1, 3, ..., K to validate fold 2, our initial point from training folds 2, 3, ..., K contains information from the validation set.
- in training folds 2, 3, ..., K, we obtain only an approximate solution rather than the optimum.

That is, the initial point is biased toward fitting fold 2; this issue should be fixed if we obtain the optimum of training folds 1, 3, ..., K, but in practice we do not. This experiment shows that in applying the warm start technique, we often conveniently assume that optimal solutions are exactly obtained. This assumption is of course incorrect because of numerical computation. While solving optimization problems more accurately may address the issue, the training time also goes up, a situation that contradicts the goal of applying the warm start technique. Our experiences indicate that while warm start is very useful in machine learning, its practical implementation must be carefully designed.

5. CONCLUSIONS

Although we have studied many issues on the parameter selection for linear classifiers, there are some future works.

- Training tasks in the CV procedure under a given C are independent. It is interesting to make a parallel implementation and investigate the scalability.
- Our method can be easily extended to L1-regularized problems. However, the relationship between optimization problems and regularization parameters must be studied because the primal optimal solution $\boldsymbol{w}$ may not be unique.

In conclusion, based on this research work, we have released an extension of LIBLINEAR for parameter selection. It is an automatic and convenient procedure for users without background knowledge on linear classification.

6. ACKNOWLEDGMENTS

This work was supported in part by the National Science Council of Taiwan via the grant 101-2221-E-002-199-MY3.

7. REFERENCES

[1] J. F. Bonnans and A. Shapiro. *Pertubation analysis of optimization problems*. Springer-Verlag, 2000.

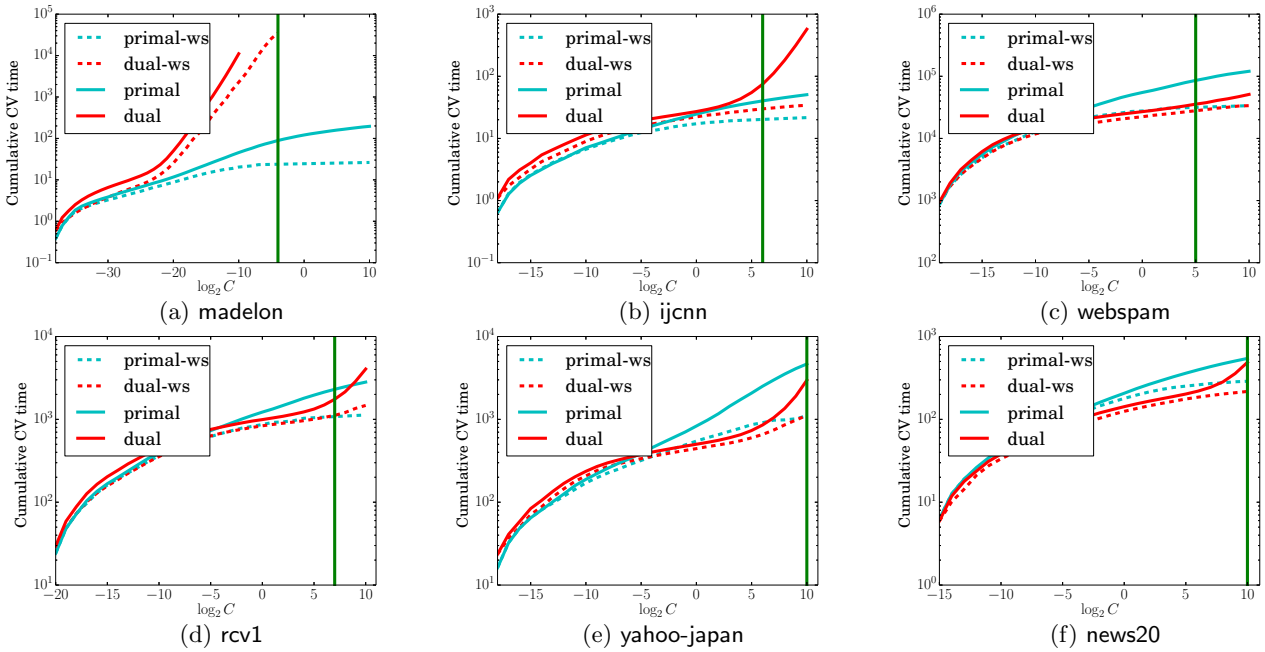


Figure 3: Training time (in seconds) using LR with/without warm-start techniques. The vertical line indicates the last C value checked by Algorithm 1. Because the training time quickly increases when C becomes large, the y-axis is log-scaled.

[2] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM TIST*, 2(3):27:1–27:27, 2011.

[3] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee. Choosing multiple parameters for support vector machines. *MLJ*, 46:131–159, 2002.

[4] K.-M. Chung, W.-C. Kao, C.-L. Sun, L.-L. Wang, and C.-J. Lin. Radius margin bounds for support vector machines with the RBF kernel. *Neural Comput.*, 15:2643–2681, 2003.

[5] D. DeCoste and K. Wagstaff. Alpha seeding for support vector machines. In *KDD*, 2000.

[6] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. LIBLINEAR: a library for large linear classification. *JMLR*, 9:1871–1874, 2008.

[7] J. H. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *JSS*, 33:1–22, 2010.

[8] J. Giesen, M. Jaggi, and S. Laue. Approximating parameterized convex optimization problems. *TALG*, 9:10:1–10:17, 2012.

[9] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *JMLR*, 5:1391–1415, 2004.

[10] 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.

[11] 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.

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

[13] S.-J. Kim, K. Koh, M. Lustig, S. Boyd, and D. Gorinevsky. An interior point method for large-scale l1-regularized least squares. *IEEE J-STSP*, 1:606–617, 2007.

[14] R. Kohavi and G. H. John. Automatic parameter selection by minimizing estimated error. In *ICML*, 1995.

[15] J.-H. Lee and C.-J. Lin. Automatic model selection for support vector machines. Technical report, 2000.

[16] C.-J. Lin, R. C. Weng, and S. S. Keerthi. Trust region Newton method for large-scale logistic regression. *JMLR*, 9:627–650, 2008.

[17] J. Matoušek and B. Gärtner. *Understanding and Using Linear Programming*. Springer, 2007.

[18] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *NIPS*. 2012.

[19] R. Tibshirani, J. Bien, J. Friedman, T. Hastie, N. Simon, J. Taylor, and R. J. Tibshirani. Strong rules for discarding predictors in lasso-type problems. *JRSSB*, 74:245–266, 2012.

[20] C.-H. Tsai, C.-Y. Lin, and C.-J. Lin. Incremental and decremental training for linear classification. In *KDD*, 2014.

[21] Z. Wu, A. Zhang, C. Li, and A. Sudjianto. Trace solution paths for SVMs via parametric quadratic programming. *DMMT*, 2008.

[22] G.-X. Yuan, C.-H. Ho, and C.-J. Lin. Recent advances of large-scale linear classification. *PIEEE*, 100:2584–2603, 2012.