Parameter Selection for Linear Support Vector Regression

Jui-Yang Hsia, Chih-Jen Lin

Abstract—In linear support vector regression (SVR), regularization parameter and error sensitivity parameter are used to avoid overfitting the training data. A proper selection of parameters is very essential for obtaining a good model, but the search process may be complicated and time-consuming. In an earlier work by Chu et al. (2015), an effective parameter-selection procedure by using warm-start techniques to solve a sequence of optimization problems has been proposed for linear classification. We extend their techniques to linear SVR, but address some new and challenging issues. In particular, linear classification involves only the regularization parameter but linear SVR has an extra error sensitivity parameter. We investigate the effective range of each parameter and the sequence in checking the two parameters. Based on this work, an effective tool for the selection of parameters for linear SVR has been available for public use.

I. INTRODUCTION

Support vector regression (SVR) is a linear regression model commonly used in machine learning and data mining. It extends least-square regression by considering an $\epsilon$-insensitive loss function. Further, to avoid overfitting the training data, the concept of regularization is usually applied. An SVR thus solves an optimization problem that involves two parameters: the regularization parameter (often referred to as $C$) and the error sensitivity parameter (often referred to as $\epsilon$). This work aims to derive an effective strategy for selecting these two parameters. Note that we focus on linear SVR rather than kernel SVR, which involves also kernel parameters.

Parameter selection of a learning method is part of the broader subject of automated machine learning (autoML). In general we solve an optimization problem over parameters, where many global optimization algorithms can be applied (e.g., \cite{9,12,13,19,20,21}). Approaches specific to parameter selection for machine learning have also been available (e.g., \cite{13,22}). Further, methods specially designed for support vector machines (SVM) have been proposed (e.g., \cite{11,2,3,4,5,7,11,15,17,23,25,26,27}). Most of them focus on classification rather than regression. Further, methods suitable for linear SVM may not be effective for kernel SVM, and vice versa. A more detailed review of past works is given in supplementary materials.

Among all existing studies, we are interested in the work \cite{2}, which applies a warm-start technique for the parameter selection of linear classification (L2-regularized logistic regression and L2-loss SVM). Because the only parameter is the regularization parameter $C$, their strategy is to sequentially check cross-validation (CV) accuracy at the following parameters

$$C_{\text{min}}, C_{\text{min}}\Delta, C_{\text{min}}\Delta^2, \cdots,$$

where $\Delta > 1$ is a given constant to control the increase of the parameter and $C \leq C_{\text{min}}$ is shown to be not useful. The search procedure stops after the performance cannot be further improved. Between two consecutive parameters, they consider a warm-start technique for fast training. Specifically, the solution of the optimization problem under the current $C$ is used as the initial solution in solving the next problem with the parameter $C\Delta$. Although the idea is simple, \cite{2} must solve some issues in order to finish a now widely used parameter-selection tool in the popular package LIBLINEAR \cite{6} for linear classification.

In this work, we aim to extend the procedure in \cite{2} for SVR. However, because of the difference between classification and regression, and the extra parameter $\epsilon$, some modifications must be made. Further, we must solve the following new challenges.

- We find that deriving a suitable $C_{\text{min}}$ for regression is more difficult than classification.
- For classification that involves only one parameter, the search sequence shown in \cite{1} can be a reasonable choice. For SVR with two parameters, more options are possible. For example, we can consider a sequence of $\epsilon$ values, and for every fixed $\epsilon$, we run a sequence in \cite{1}. Alternatively, we can consider a sequence of $C$ values first and for every $C$ we check a sequence of $\epsilon$ values.
- Because the search space of $C \in (0, \infty)$ is huge, it is a common practice to consider a sequence in \cite{1} by exponentially increasing the $C$ value. However, depending on the data, $\epsilon$ may be in a small interval, so a linear increase/decrease of the $\epsilon$ value might be more suitable.

In this work, we thoroughly investigate the above issues. Our final recommended setting is to check a sequence of $C$ values for every fixed $\epsilon$ value.

We choose to extend the classification work in \cite{2} for linear SVR rather than consider some existing parameter-selection works for kernel SVR (e.g., \cite{23,25}) because of the following reasons. First, the procedure is simpler because we directly check a grid of $(C, \epsilon)$ points. Note that kernel SVR involves more parameters, so a grid search may not be feasible and more sophisticated approaches are needed. Second, while checking $(C, \epsilon)$ points may be time-consuming, by effective warm-start techniques in \cite{2}, the overall procedure is practically viable.

This paper is organized as follows. In the next section, we introduce the formulations of SVR and discuss how to obtain an approximate solution of its optimization problem. In Section \cite{11} we discuss the relationship between solutions of optimization problems and SVR parameters. In particular, we identify a suitable range of $C$ and $\epsilon$. In Section \cite{IV}...
we discuss the procedure to search parameters and show details of our implementation. Section [V] experimentally confirms the viability of the proposed approach. Supplementary materials are available at https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/warm-start/.

Our proposed procedure has been included in the package LIBLINEAR [6] (after version 2.30) for the parameter selection of linear regression.

II. SVR OPTIMIZATION PROBLEM

Consider training data \((y_i, x_i) \in \mathbb{R} \times \mathbb{R}^n\), for \(i = 1, \ldots, l\), where \(y_i \in \mathbb{R}\) is the target value and \(x_i \in \mathbb{R}^n\) is the feature vector. We use \(l\) to denote the number of training instances and let \(n\) be the number of features in each instance. Linear SVR [24] finds a model \(w\) such that \(w^T x_i\) is close to the target value \(y_i\). It solves the following problem with a given regularization parameter \(C > 0\) and an error sensitivity parameter \(\epsilon \geq 0\).

\[
\min_{w} f(w; C, \epsilon), \quad \text{where} \quad f(w; C, \epsilon) \equiv \frac{1}{2} \|w\|^2 + CL(w; \epsilon).
\]

In (2), \(\|w\|^2 / 2\) is the regularization term and \(L(w; \epsilon)\) is the sum of training losses defined as

\[
L(w; \epsilon) = \left\{ \begin{array}{ll}
\sum_{i=1}^{l} \max(0, |w^T x_i - y_i| - \epsilon, 0) & \text{L1 loss,} \\
\sum_{i=1}^{l} \max(0, |w^T x_i - y_i| - \epsilon, 0)^2 & \text{L2 loss.}
\end{array} \right.
\]

SVR employs the \(\epsilon\)-insensitive loss so that small losses of some instances are ignored. That is, the loss is zero if \(|w^T x_i - y_i| \leq \epsilon\).

The objective function \(f(w; C, \epsilon)\) is strongly convex, so the unique optimal solution exists and we denote it as

\[
w_{C, \epsilon} = \arg\min_{w} f(w; C, \epsilon).
\]

Because L1 loss is not differentiable and L2 loss is differentiable but not twice differentiable, different optimization methods have been proposed for training SVR. A detailed study is in [8], which considers two types of approaches: Newton methods for L2-loss SVR and dual coordinate descent methods for L1- and L2-loss SVR. These methods were extended from studies for linear classification (e.g., [10], [16]). For the parameter selection of classification problems, [2] recommends a Newton method after some careful evaluations. An important reason is that a Newton method possesses some advantages under a warm-start setting for training linear classification problems. Therefore, we follow [2] to consider a Newton method to solve each SVR problem in the parameter-selection procedure.

A Newton method iteratively finds Newton directions by considering the second-order approximation of the objective function. Details of the Newton method considered are in [8], [16]. Because differentiability is required, here we consider only L2-loss SVR and investigate its parameter selection.1

1Note that Newton method requires second derivative, but the L2-loss function is not twice differentiable. We follow [18] to consider the generalized second derivative.

Because of the nature of numerical computation, in practice we only obtain an approximate solution \(\tilde{w}_{C, \epsilon}\) of \(w_{C, \epsilon}\), returned from the optimization procedure. In an iterative optimization process a stopping condition must be imposed for the finite termination. For the Newton method considered in this work, we assume the stopping condition is that \(w_{C, \epsilon}\) satisfies

\[
\|\nabla f(\tilde{w}_{C, \epsilon}; C, \epsilon)\| \leq \tau \|\nabla f(0; C, \epsilon)\|, \quad (3)
\]

where \(\tau \in (0, 1)\) is the stopping tolerance. Clearly, (3) is related to the optimal condition \(\nabla f(w_{C, \epsilon}; C, \epsilon) = 0\), but we further consider a relative setting to compare with the gradient at the zero point, which is a common initial point of the optimization procedure. The condition (3) plays a role in the parameter-selection procedure, where details are in Section [IV].

A. Warm-start Techniques for Parameter Selection

While many parameter-selection strategies are available, the approach in [2] is a conservative but reliable setting of checking the cross validation (CV) performance under different parameter values. The training set is randomly split into several folds. Each time one fold is used for validation, while other folds are considered for training. Therefore, many SVR optimization problems must be solved.

To reduce the running time, [2] considers a warm-start strategy to solve closely related optimization problems. We extend their setting for linear SVR. Suppose \(w_{C_1, \epsilon_1}\) is the optimal solution under \(C = C_1\) and \(\epsilon = \epsilon_1\). If \((C_1, \epsilon_1)\) is slightly changed to \((C_2, \epsilon_2)\), we use \(w_{C_1, \epsilon_1}\) as the initial solution for solving the new optimization problem. The idea behind such a warm-start strategy is as follows. For optimization techniques such as Newton methods, they iteratively generate a sequence \(\{w_k\}_{k=0}^{\infty}\) converging to the optimum. Because a small change of parameters may not cause a significant change of the optimization problem, the optimal solution of the original problem can be a good starting point for the new problem. Then the number of optimization iterations may be significantly reduced in comparison with that without warm-start (e.g., using 0 as the initial solution).

We divide the parameter-selection problem for SVR into two parts. One is the search range of each parameter. The other is the design of the search procedure. We study the first in Section [III] and the second in Section [IV].

III. RANGE OF PARAMETERS

We check the range of a parameter by assuming that the other is fixed. To simplify the notification, if \(\epsilon\) is fixed, we denote

\[
w_C = w_{C, \epsilon}, \quad \bar{w}_C = \tilde{w}_{C, \epsilon},
\]

\[
L(w) = L(w; \epsilon), \quad f(w; C) = f(w; C, \epsilon).
\]

Similarly, if \(C\) is fixed, we have

\[
w_\epsilon = w_{C, \epsilon}, \quad \bar{w}_\epsilon = \tilde{w}_{C, \epsilon}, \quad f(w; \epsilon) = f(w; C, \epsilon).
\]

For a suitable parameter range we hope that first parameters achieving the best performance are within it and second the range should be as small as possible. We follow [2] to identify
parameters that should not be considered. For example, if a parameter setting leads to a model that does not learn enough information from the training data, then underfitting occurs and such parameters should not be used.

A. Zero Vector is a Trivial Model

We begin with showing that the zero vector leads to a model that may not learn enough information from the training data. First, because

\[ f(w_{C,\epsilon}; C, \epsilon) \leq f(0; C, \epsilon) \text{ and } \|w_{C,\epsilon}\| \geq \|0\|, \]

we have

\[ L(w_{C,\epsilon}, \epsilon) \leq L(0, \epsilon). \]

The larger training loss indicates that 0 may not learn more from the training data than any \( w_{C,\epsilon} \). Second, the following theorem shows that the learnability of \( w_{C} \) deteriorates as \( C \) approaches zero and \( w_{C} \) eventually goes to the zero point.

**Theorem 1.** If \( C_1 > C_0 \), then

\[ \|w_{C_1}\| \geq \|w_{C_0}\| \text{ and } L(w_{C_1}) \leq L(w_{C_0}). \]

Further,

\[ \lim_{C \to 0} w_C = 0. \]

Note that proofs of all theorems are in the supplementary materials. From the discussion, we can treat 0 as a trivial model that underfits the training data. For any with \( L(w) \approx L(0) \), \( w \) may have not learned enough information from the training data.

B. Parameter \( C \)

We fix \( \epsilon \) and discuss the upper and lower bounds for parameter \( C \).

1) **Lower Bound of \( C \):** From the discussion in Section III-A we check under which \( C \) values the training loss \( L(w_C) \) is close to \( L(0) \) by proving the following theorem.

**Theorem 2.** Consider \( L2 \) loss. For \( 0 \leq \delta_1 < 1 \), we have

\[ L(w_C) \geq (1 - \delta_1) \times L(0) \quad \forall C \leq C_{\text{min}}, \]

where \( C_{\text{min}} \) is defined as

\[ C_{\text{min}} = \begin{cases} \frac{\delta_1^2 L(0)}{8n \sum_{i=1}^{\infty} |y_i|^2 (\max_i \|x_i\|)^2} & \text{if } L(0) > 0 \quad \overset{\square}{=} \; (4) \\ \infty & \text{if } L(0) = 0. \end{cases} \]

Therefore, by choosing a \( \delta_1 \) close to 0, \( C_{\text{min}} \) can be a lower bound for the parameter \( C \).

2) **Upper Bound of \( C \):** We first check properties of \( \{w_C\} \) when \( C \) is large. Let \( W_\infty \) be the set of points that attain the minimum of \( L(w) \).

\[ W_\infty \equiv \{ w \mid L(w) = \inf_{w'} L(w') \}. \]

For classification problems, \[2\] has discussed the convergence property of \( \{w_C\} \) as \( C \to \infty \). We extend their results here for regression.

**Theorem 3.** Consider any non-negative and convex loss function. If \( W_\infty \neq \emptyset \), then

\[ \lim_{C \to \infty} w_C = w_\infty, \quad \text{where } w_\infty = \arg \min_{w \in W_\infty} \|w\|^2. \]

If \( L2 \) loss is used, then \( W_\infty \neq \emptyset \).

Because \( w_\infty \) is a model without using regularization, overfitting tends to occur and the performance is often not the best. However, it is difficult to identify a \( C_{\text{max}} \) so that if \( C \geq C_{\text{max}} \), the model is sufficiently close to \( w_\infty \). We leave more investigations in Section IV.

C. Parameter \( \epsilon \)

We now fix \( C \) and discuss upper and lower bounds for parameter \( \epsilon \).

1) **Lower Bound of \( \epsilon \):** Because \( \epsilon \geq 0 \), a trivial lower bound of \( \epsilon \) is \( \epsilon = 0 \). We argue that this is a meaningful lower bound because \[8\] has shown that \( \epsilon = 0 \) often leads to a good model. That is, for some data sets the \( \epsilon \)-insensitive setting is not needed and regularized least-square regression is as effective as SVR.

2) **Upper Bound of \( \epsilon \):** From the definition of \( \epsilon \)-insensitive loss functions, if \( \epsilon \) is large so that for most data the loss is zero, then the model tends to underfit the training data. Thus an obvious upper bound is

\[ \epsilon_{\text{max}} = \max_i |y_i|. \]

Under this \( \epsilon_{\text{max}} \), \( f(0) = 0 \) implies that \( w = 0 \) is an optimal solution of \[2\] and insufficient information has been learned.

IV. THE SEARCH PROCEDURE

After studying the range of each parameter, we must find an effective search procedure. Under a grid setting, a two-level loop sweeps \( C \) (or \( \epsilon \)) first, and at the inner level, we go through values of the other parameter. Then two issues must be addressed.

- The parameter to be used for the outer loop.
- The search sequence of each parameter.

These two issues are complicated, so our discussion goes from decisions that are easily made to those that are less certain.

We start by checking the search sequence of the parameter \( C \). For the parameter selection of linear classification, an exponential increase of the regularization parameter \( C \) has been commonly considered; see the sequence in \[1\]. The reason is that \( C \in (0, \infty) \) is in a rather large range and we need the exponential increase of the parameter to cover the search space. The same setting should be applied for regression because we still have \( C \in (0, \infty) \). In addition,
we follow (1) to start from \( C_{\text{min}} \) because of two reasons. First, for both classification and regression, \( C_{\text{min}} \) has been specifically derived; see [2] and [4]. In contrast, we do not have a clear way to calculate an upper bound and must rely on techniques discussed later in this section. Second, if we consider a decreasing sequence, solving the first optimization problem may be time-consuming. The reason is that under a large \( C \), the model tries to better fit the training data and the optimization problem is known to be more difficult. Based on these reasons, regardless of whether \( C \) is used in the outer or the inner level of the loop, we always consider an increasing sequence of \( C \) values as in (1).

We now discuss the search sequence of the other parameter \( \epsilon \). An exponential sequence like the setting for \( C \) can be considered. However, it should be a decreasing one starting at \( \epsilon = \epsilon_{\text{max}} \). The reason is that because 0 is the solution when \( \epsilon = \epsilon_{\text{max}} \), the optimization problem is easier when \( \epsilon \) is closed to \( \epsilon_{\text{max}} \). Recall that a similar reason leads us to begin the search of \( C \) at \( C_{\text{min}} \).

Instead of an exponential sequence, we argue that the sequence from a linear segmentation of \([0, \epsilon_{\text{max}}]\) may be more suitable for the parameter \( \epsilon \). An important difference from \( C \) is that \( \epsilon \) is in a bounded interval \([\epsilon_{\text{min}}, \epsilon_{\text{max}}]\), where \( \epsilon_{\text{min}} = 0 \) and \( \epsilon_{\text{max}} < \infty \). Further, while both lower and upper bounds of \( C \) tend to be values not leading to a good model, for \( \epsilon \), [8] has pointed out that the model of using \( \epsilon = 0 \) is often competitive. We also have

**Theorem 4.**

\[
\lim_{\epsilon \to 0} w_{\epsilon} = w_0.
\]

If an exponentially decreasing sequence starting from \( \epsilon_{\text{max}} \) is considered, many problems with \( \epsilon \approx 0 \) are checked, but Theorem 4 shows that their resulting models are similar. In contrast, a linear sequence can clearly avoid this situation. In Section V-A we conduct experiments to compare the two settings (linear and exponential) for the search sequence of \( \epsilon \).

The remaining issue is when to stop increasing the parameter \( C \) in the search procedure because \( C_{\text{max}} \) is the only bound that cannot be explicitly obtained in Section III. We extend the setting in [2] by following Theorem 3 which states that \( \{w_\epsilon\}, C \to \infty \) converge to a point \( w_\infty \). Their idea is to terminate the selection procedure if the approximate solutions of \( t_{\text{stop}} \) consecutive optimization problems are the same. That is, if

\[
\hat{w}_C = \hat{w}_{\Delta C} = \hat{w}_{\Delta^2 C} = \hat{w}_{\Delta^3 C} \cdots = \hat{w}_{\Delta^{t_{\text{stop}}}} C,
\]

then the search process terminates at \( C \). It is easy to check [4] by the stopping condition [4] of the optimization procedure

\[
\|\nabla f(\hat{w}_C; \Delta^t C)\| \leq \tau \|\nabla f(0; \Delta^t C)\|, \quad t = 1, 2, \cdots, t_{\text{stop}}.
\]

In other words, an approximate solution \( \hat{w}_C \) satisfies the above stopping condition with \( t = 0 \), but we check if it is also the returned solution of the next several problems without any optimization iteration. We choose \( t_{\text{stop}} = 5 \) for experiments in Section V though more discussion on its selection is in supplementary materials.

**V. Experiments**

We conduct experiments on some regression sets available at [https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/](https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/). For some data sets, a scaled version is provided at the same site by linearly scaling each attribute to \([-1, 1]\) or \([0, 1]\). We named the scaled version with an extension “-scale.” Our search procedure aims to find a model achieving the best five-fold CV result on the validation MSE (Mean Square Error). Because of space limit, we present only results on some larger sets, while leave detailed experimental settings (including data statistics) and complete experimental results in supplementary materials.

A. Exponential or Linear Search Sequence for the \( \epsilon \) Parameter

In Section IV we discuss the issue of using an exponential or a linear sequence of \( \epsilon \) values in the search procedure. We conduct a comparison by considering \( C \) values in the following set

\[
\{C_{\text{min}}, C_{\text{min}}\Delta, C_{\text{min}}\Delta^2, \cdots, C_{\text{max}}\}
\]

and \( \epsilon \) values in either a linear- or an exponential-spaced sequence:

\[
\{0, \square, 2\square, \cdots, \epsilon_{\text{max}}\} \text{ or } \{2^{-30}, 2^{-30}\Delta, 2^{-30}\Delta^2, \cdots, \epsilon_{\text{max}}\},
\]

where

\[
C_{\text{max}} = 2^{50}, \quad \square = \frac{\epsilon_{\text{max}}}{20} \quad \text{and} \quad \Delta = 2.
\]

In Figure 1 for each data set we show

\[
\left\{ (\log_2 C, \epsilon) \text{ or } (\log_2 C, \log_2 \epsilon) \right\} \text{ versus } \log_2(\text{CV MSE})
\]

depending on the sequence for \( \epsilon \). We observe that if an exponential sequence is used, then CV MSE is almost the same in the entire figure. The reason is that from Theorem 4 after \( \epsilon \) is smaller than a certain value, CV MSE is similar. For the purpose of exploring different CV MSE values, we conclude that a linear sequence should be more suitable in our parameter selection procedure.

B. Evaluation of Various Implementations for the Search Procedure

We compare cross validation MSE of the following settings:

- “Full and independent” (Baseline): By using \((C, \epsilon)\) values in (8) and (9), we solve all linear SVR problems independently. For each SVR problem 0 is the initial point and the stopping condition is (3). This setting aims to show that if without any techniques to reduce the search space and without the warm start implementation, what the resulting MSE is. We compare with this baseline setting to see if our procedures trade the performance for efficiency.
the resulting models are slightly different. From ratios all close
one. However, with approximate solutions satisfying only (3),
problems is solved, theoretically the ratio should be exactly
"Full and independent." For them because the same set of SVR
others go over the same full grid of parameters as the baseline
We observe that all ratios are close to one, indicating that
sequences of
Figure 1: Cross validation MSE (log scaled) by different search
...space-ga

Table I: An MSE comparison with the baseline setting of
running the full grid without warm start; see the ratio defined
in (11). (C, ϵ) and (ϵ, C) indicate that C and ϵ are used in
the outer loop of the parameter grid, respectively. YPMSD is
the abbreviation of YearPredictionMSD. Ratios different from
one are bold-faced.

<table>
<thead>
<tr>
<th>Data set</th>
<th>(ϵ, C)</th>
<th>No criterion</th>
<th>(C, ϵ)</th>
<th>No criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>abalone</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>abalone-scale</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>cadata</td>
<td>1.09</td>
<td>1.09</td>
<td>1.01</td>
<td>1.01</td>
</tr>
<tr>
<td>cpusmall</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>epusmall-scale</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>E2006-train</td>
<td>0.99</td>
<td>0.99</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>housing</td>
<td>1.04</td>
<td>1.04</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>housing-scale</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>log1p-E2006-train</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>mg</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>mg-scale</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>space-ga</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>space-ga-scale</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
<td>1.00</td>
</tr>
<tr>
<td>YPMSD</td>
<td>1.02</td>
<td>1.02</td>
<td>0.99</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Running time by applying warm start
Running time by “Full and independent”. (12)
A smaller ratio indicates a better time reduction by using warm
start. From Table I we have the following observations.
• All the values in Table I are much smaller than one. This
result shows that the warm-start techniques can significantly
reduce the time required to search the parameters.
• For the (ϵ, C) setting, the running time with/without the
early termination of the C sequence is almost the same.
We give the following explanation. The criterion (7) checks
the stopping condition of several consecutive optimization
problems. When (7) holds, the corresponding \( \hat{w}_C \) may
be close enough to \( w_\infty \) by Theorem 3 and the condition (3)
may hold ever since:
\[
\| \nabla f(\hat{w}_{C_{stop}}; C) \| \leq \tau \| \nabla f(0; C) \|, \forall C \geq C_{stop},
\]
where \( C_{stop} \) is the value when (7) is satisfied. Therefore,
if we check more C values all the way up to the specified
\( C_{max} \), at each C the optimization method terminates without
running any iteration. In this situation, the early termination
via (7) is not needed. However, in theory \( C_{stop} \) may not
exist to have (13) because \( \hat{w}_{C_{max}} \) is only an approximate
rather than the optimal solution. Thus it is still possible
that the optimization method takes time at each C and

Running time is estimated by the number of CG steps in the Newton
method for training SVR. See details in supplementary materials.
we expensively run the procedure all the way up to $C_{\text{max}}$. Further, the selection of $C_{\text{max}}$ is a tricky issue; in our experiment we choose $2^{20}$ in (10) without a good reason. Therefore, we can say that (7) is a relaxed condition of (17) to avoid the huge efforts of possibly running up to an extremely large $C_{\text{max}}$.

- Between $(C, \epsilon)$ and $(\epsilon, C)$, we observe that $(C, \epsilon)$ costs more. It seems that warm start is less effective when $C$ is fixed and $\epsilon$ is slightly changed. A reason might be that in our experiments, the number of SVR problems per $\epsilon$ value is often larger than that per $C$ value. Then the time saving by applying warm start for the $(C, \epsilon)$ strategy is less dramatic. Note that in (9) we split $[\epsilon_{\text{min}}, \epsilon_{\text{max}}]$ to 20 intervals, but with a small $C_{\text{min}}$ and a large $C_{\text{max}}$, the number of $C$ values in $[C_{\text{min}}, C_{\text{max}}]$ tends to be larger. More detailed discussion is provided in supplementary materials.

D. Comparison with Other Parameter Selection Methods

We have compared our proposed method with two existing techniques for parameter selection: simulated annealing and particle swarm optimization, which details are in supplementary materials. We find that these alternative approaches, while more sophisticated, are often as competitive. However, they are extremely large $C$.

Table II: Ratio defined in (12) to show the time reduction of using warm start.

<table>
<thead>
<tr>
<th>Data set</th>
<th>$(\epsilon, C)$</th>
<th>No criterion</th>
<th>No criterion</th>
</tr>
</thead>
<tbody>
<tr>
<td>abalone</td>
<td>0.12</td>
<td>0.12</td>
<td>0.54</td>
</tr>
<tr>
<td>abalone-scale</td>
<td>0.11</td>
<td>0.11</td>
<td>0.50</td>
</tr>
<tr>
<td>cadata</td>
<td>0.06</td>
<td>0.06</td>
<td>0.67</td>
</tr>
<tr>
<td>cupsmall</td>
<td>0.06</td>
<td>0.06</td>
<td>0.62</td>
</tr>
<tr>
<td>cupsmall-scale</td>
<td>0.10</td>
<td>0.10</td>
<td>0.69</td>
</tr>
<tr>
<td>E2006-train</td>
<td>0.14</td>
<td>0.14</td>
<td>0.35</td>
</tr>
<tr>
<td>housing</td>
<td>0.07</td>
<td>0.07</td>
<td>0.66</td>
</tr>
<tr>
<td>housing-scale</td>
<td>0.11</td>
<td>0.11</td>
<td>0.59</td>
</tr>
<tr>
<td>log1p-E2006-train</td>
<td>0.08</td>
<td>0.08</td>
<td>0.62</td>
</tr>
<tr>
<td>mg</td>
<td>0.13</td>
<td>0.13</td>
<td>0.61</td>
</tr>
<tr>
<td>mg-scale</td>
<td>0.13</td>
<td>0.13</td>
<td>0.61</td>
</tr>
<tr>
<td>space-ga</td>
<td>0.08</td>
<td>0.08</td>
<td>0.73</td>
</tr>
<tr>
<td>space-ga-scale</td>
<td>0.13</td>
<td>0.13</td>
<td>0.62</td>
</tr>
<tr>
<td>YPMSD</td>
<td>0.04</td>
<td>0.04</td>
<td>0.35</td>
</tr>
</tbody>
</table>

VI. RECOMMENDED PROCEDURE AND CONCLUSIONS

We have shown that the termination criterion (7) works effectively in practice. Because this criterion is applicable when $C$ is in the inner loop and our experiments show that the $(\epsilon, C)$ setting takes less running time, our recommended setting is to have $\epsilon$ in the outer loop and $C$ in the inner, and the criterion (7) is imposed. A detailed algorithm is given in supplementary materials.

We list technical insights from this development and future research issues in supplementary materials. In summary, we have developed an effective parameter-selection procedure based on the warm-start technique for linear SVR.

REFERENCES


