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 ϵ-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 ϵ). 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., [9], [12], [13], [19], [20], [21]). Approaches specific to parameter selection for machine learning have also been available (e.g., [14], [22]). Further, methods specially designed for support vector machines (SVM) have been proposed (e.g., [1], [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 [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

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

where $\Delta > 1$ is a given constant to control the increase of the parameter and $C \le C_{\min}$ 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 ΔC. Although the idea is simple, [2] must solve some issues in order to finish a now widely used parameter-selection tool in the popular package LIBLINEAR [6] for linear classification.

In this work, we aim to extend the procedure in [2] for SVR. However, because of the difference between classification and regression, and the extra parameter ϵ, some modifications must be made. Further, we must solve the following new challenges.

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

We choose to extend the classification work in [2] for linear SVR rather than consider some existing parameter-selection works for kernel SVR (e.g., [23], [26]) because of the following reasons. First, the procedure is simpler because we directly check a grid of (C, ϵ) 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, ϵ) points may be time-consuming, by effective warm-start techniques in [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 III, we discuss the relationship between solutions of optimization problems and SVR parameters. In particular, we identify a suitable range of C and ϵ. In Section IV,

Jui-Yang Hsia and Chih-Jen Lin are with National Taiwan University. (e-mail: hsiajuiyang5174@gmail.com and cjlin@csie.ntu.edu.tw). This work was partially supported by MOST of Taiwan grant 107-2221-E-002-167-MY3.

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 $(y_i, \boldsymbol{x}_i) \in R \times R^n$, for $i = 1, \cdots, l$, where $y_i \in R$ is the target value and $\boldsymbol{x}_i \in R^n$ 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 $\boldsymbol{w}$ such that $\boldsymbol{w}^T\boldsymbol{x}_i$ 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 $\epsilon \geq 0$.

$$\min_{\boldsymbol{w}} f(\boldsymbol{w}; C, \epsilon), \text{ where} \tag{2}$$
$$f(\boldsymbol{w}; C, \epsilon) \equiv \frac{1}{2}\|\boldsymbol{w}\|^2 + CL(\boldsymbol{w}; \epsilon).$$

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

$$L(\boldsymbol{w}; \epsilon) = \begin{cases} \sum_{i=1}^{l} \max(|\boldsymbol{w}^T\boldsymbol{x}_i - y_i| - \epsilon, 0) & \text{L1 loss,} \\ \sum_{i=1}^{l} \max(|\boldsymbol{w}^T\boldsymbol{x}_i - y_i| - \epsilon, 0)^2 & \text{L2 loss.} \end{cases}$$

SVR employs the ϵ-insensitive loss so that small losses of some instances are ignored. That is, the loss is zero if $|\boldsymbol{w}^T\boldsymbol{x}_i - y_i| \leq \epsilon$.

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

$$\boldsymbol{w}_{C,\epsilon} = \arg\min_{\boldsymbol{w}} f(\boldsymbol{w}; C, \epsilon).$$

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 we considered are in [8], [16]. Because differentiability is required, here we consider only L2-loss SVR and investigate its parameter selection.[1]

[1]Note 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 $\tilde{\boldsymbol{w}}_{C,\epsilon}$ of $\boldsymbol{w}_{C,\epsilon}$, 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 $\tilde{\boldsymbol{w}}_{C,\epsilon}$ satisfies

$$\|\nabla f(\tilde{\boldsymbol{w}}_{C,\epsilon}; C, \epsilon)\| \leq \tau \|\nabla f(\boldsymbol{0}; C, \epsilon)\|, \tag{3}$$

where $\tau \in (0, 1)$ is the stopping tolerance. Clearly, (3) is related to the optimal condition $\nabla f(\boldsymbol{w}_{C,\epsilon}; C, \epsilon) = \boldsymbol{0}$, 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 $\boldsymbol{w}_{C_1,\epsilon_1}$ is the optimal solution under $C = C_1$ and $\epsilon = \epsilon_1$. If (C_1, ϵ_1) is slightly changed to (C_2, ϵ_2), we use $\boldsymbol{w}_{C_1,\epsilon_1}$ 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 $\{\boldsymbol{w}_k\}_{k=0}^{\infty}$ 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 $\boldsymbol{0}$ 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 ϵ is fixed, we denote

$$\boldsymbol{w}_C = \boldsymbol{w}_{C,\epsilon},\ \tilde{\boldsymbol{w}}_C = \tilde{\boldsymbol{w}}_{C,\epsilon},$$
$$L(\boldsymbol{w}) = L(\boldsymbol{w}; \epsilon),\ f(\boldsymbol{w}; C) = f(\boldsymbol{w}; C, \epsilon).$$

Similarly, if C is fixed, we have

$$\boldsymbol{w}_\epsilon = \boldsymbol{w}_{C,\epsilon},\ \tilde{\boldsymbol{w}}_\epsilon = \tilde{\boldsymbol{w}}_{C,\epsilon},\ f(\boldsymbol{w}; \epsilon) = f(\boldsymbol{w}; C, \epsilon).$$

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

$$f(\boldsymbol{w}_{C,\epsilon}; C, \epsilon) \le f(\mathbf{0}; C, \epsilon) \text{ and } \|\boldsymbol{w}_{C,\epsilon}\| \ge \|\mathbf{0}\|,$$

we have

$$L(\boldsymbol{w}_{C,\epsilon}, \epsilon) \le L(\mathbf{0}, \epsilon).$$

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

Theorem 1. *If $C_1 > C_0$, then*

$$\|\boldsymbol{w}_{C_1}\| \ge \|\boldsymbol{w}_{C_0}\| \text{ and } L(\boldsymbol{w}_{C_1}) \le L(\boldsymbol{w}_{C_0}).$$

Further,

$$\lim_{C \to 0} \boldsymbol{w}_C = \mathbf{0}.$$

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

B. Parameter C

We fix ϵ 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 $L(\boldsymbol{w}_C)$ is close to $L(\mathbf{0})$ by proving the following theorem.

Theorem 2. *Consider L2 loss. For $0 \le \delta_1 < 1$, we have*

$$L(\boldsymbol{w}_C) \ge (1 - \delta_1) \times L(\mathbf{0}) \quad \forall C \le C_{min},$$

where C_{min} is defined as

$$C_{min} = \begin{cases} \frac{\delta_1^2 L(\mathbf{0})}{8(\sum_{i=1}^{l} |y_i|)^2 (\max_i \|\boldsymbol{x}_i\|)^2} & \text{if } L(\mathbf{0}) > 0,^2 \\ \infty & \text{if } L(\mathbf{0}) = 0. \end{cases} \quad (4)$$

Therefore, by choosing a δ_1 close to 0, $C_{\min}$ can be a lower bound for the parameter C.

[2]We have that $L(\mathbf{0})/0 = \infty$ if this occurs.

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

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

For classification problems, [2] has discussed the convergence property of $\{\boldsymbol{w}_C\}$ 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*

$$\lim_{C \to \infty} \boldsymbol{w}_C = \boldsymbol{w}_\infty, \text{ where } \boldsymbol{w}_\infty = \arg \min_{\boldsymbol{w} \in W_\infty} \|\boldsymbol{w}\|^2. \quad (5)$$

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

Because $\boldsymbol{w}_\infty$ 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 $C_{\max}$ so that if $C \ge C_{\max}$, the model is sufficiently close to $\boldsymbol{w}_\infty$. We leave more investigations in Section IV.

C. Parameter ϵ

We now fix C and discuss upper and lower bounds for parameter ϵ.

1) Lower Bound of ϵ: Because $\epsilon \ge 0$, a trivial lower bound of ϵ 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 ϵ-insensitive setting is not needed and regularized least-square regression is as effective as SVR.

2) Upper Bound of ϵ: From the definition of ϵ-insensitive loss functions, if ϵ 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

$$\epsilon_{\max} = \max_i |y_i|.$$

Under this $\epsilon_{\max}$, $f(\mathbf{0}) = 0$ implies that $\boldsymbol{w} = \mathbf{0}$ 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 ϵ) 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 $C_{\min}$ because of two reasons. First, for both classification and regression, $C_{\min}$ 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.[3] 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 ϵ. An exponential sequence like the setting for C can be considered. However, it should be a decreasing one starting at $\epsilon = \epsilon_{\max}$. The reason is that because $\mathbf{0}$ is the solution when $\epsilon = \epsilon_{\max}$, the optimization problem is easier when ϵ is closed to $\epsilon_{\max}$. Recall that a similar reason leads us to begin the search of C at $C_{\min}$.

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

Theorem 4.

$$\lim_{\epsilon \to 0} \boldsymbol{w}_\epsilon = \boldsymbol{w}_0.$$

If an exponentially decreasing sequence starting from $\epsilon_{\max}$ 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 ϵ.

The remaining issue is when to stop increasing the parameter C in the search procedure because $C_{\max}$ 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 $\{\boldsymbol{w}_C\}$, $C \to \infty$ converge to a point $\boldsymbol{w}_\infty$. Their idea is to terminate the selection procedure if the approximate solutions of t_{stop} consecutive optimization problems are the same. That is, if

$$\tilde{\boldsymbol{w}}_C = \tilde{\boldsymbol{w}}_{\Delta C} = \tilde{\boldsymbol{w}}_{\Delta^2 C} = \tilde{\boldsymbol{w}}_{\Delta^3 C} \cdots = \tilde{\boldsymbol{w}}_{\Delta^{t_{\text{stop}}} C}, \tag{6}$$

then the search process terminates at C. It is easy to check (6) by the stopping condition (3) of the optimization procedure[4]:

$$\|\nabla f(\tilde{\boldsymbol{w}}_C; \Delta^t C)\| \le \tau \|\nabla f(\mathbf{0}; \Delta^t C)\|, \quad t = 1, 2, \cdots, t_{\text{stop}}. \tag{7}$$

In other words, an approximate solution $\tilde{\boldsymbol{w}}_C$ 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.

[3] In fact, some past works think an efficient way to solve a single SVM under a large C is through a warm-start setting on the problems corresponding to an increasing sequence of smaller C values; see, for example, the software BSVM https://www.csie.ntu.edu.tw/~cjlin/bsvm/.

[4] We explain in supplementary materials why on the right-hand side of (7), the $\mathbf{0}$ point is always used.

V. Experiments

We conduct experiments on some regression sets available at 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 ϵ Parameter

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

$$\{C_{\min}, C_{\min}\Delta, C_{\min}\Delta^2, \cdots, C_{\max}\} \tag{8}$$

and ϵ values in either a linear- or an exponential-spaced sequence:

$$\{0, \square, 2\square, \cdots, \epsilon_{\max}\} \text{ or } \{2^{-30}, 2^{-30}\Delta, 2^{-30}\Delta^2, \cdots, \epsilon_{\max}\}, \tag{9}$$

where

$$C_{\max} = 2^{50}, \ \square = \frac{\epsilon_{\max}}{20} \text{ and } \Delta = 2. \tag{10}$$

In Figure 1, for each data set we show

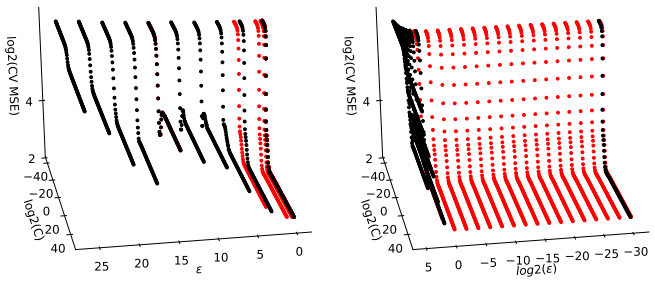
$$\begin{cases} (\log_2 C, \epsilon) \text{ or} \\ (\log_2 C, \log_2 \epsilon) \end{cases} \quad \text{versus } \log_2(\text{CV MSE})$$

depending on the sequence for ϵ. 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 ϵ 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.

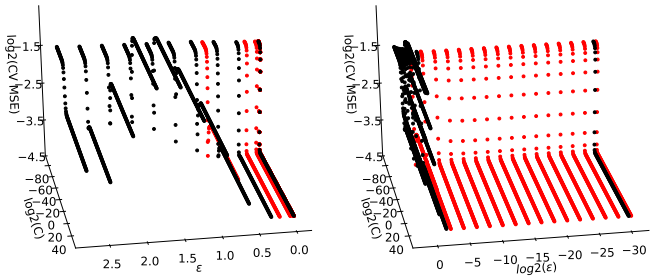
B. Evaluation of Various Implementations for the Search Procedure

We compare cross validation MSE of the following settings:

- "Full and independent" (Baseline): By using (C, ϵ) values in (8) and (9), we solve *all linear SVR problems independently*. For each SVR problem $\mathbf{0}$ 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.



(a) abalone



(b) space-ga

Figure 1: Cross validation MSE (log scaled) by different search sequences of ϵ. Left: linear. Right: exponential.

- (ϵ, C): The warm-start setting is considered. For the search procedure, ϵ is in the outer loop and C is in the inner loop. The search range of ϵ is shown in (9), while for C, we evaluate the following settings:
  - The termination criterion (7) extended from [2] is applied. See details in Section IV.
  - "No termination criterion": We run the full grid up to the large $C_{\max}$ specified in (10). Thus the number of checked (ϵ, C) points is the same as "Full and independent." They differ only in whether the warm-start strategy is applied or not.
- (C, ϵ): We apply the warm-start setting, where for the search procedure, C is in the outer loop and ϵ is in the inner loop. For the termination of the C sequence, the condition (7) is less suitable here. Because C is in the outer loop, to check (7), all models under different ϵ values must be stored. Therefore, we run the full grid up to the specified $C_{\max}$ in (10). The number of checked (C, ϵ) points is thus the same as that of the "Full and independent" setting and that of "No termination criterion" in the (ϵ, C) setting.

To see if any MSE change occurs after applying the warm start technique, in Table I we present the following ratio:

$$\frac{\text{Best CV MSE by applying warm start}}{\text{Best CV MSE by "Full and independent"}}. \tag{11}$$

We observe that all ratios are close to one, indicating that the CV MSE is close to the baseline setting of independently running a full grid without warm start. Note that except the use of (7) to early stop the C sequence in the (ϵ, C) setting, all others go over the same full grid of parameters as the baseline "Full and independent." For them because the same set of SVR problems is solved, theoretically the ratio should be exactly one. However, with approximate solutions satisfying only (3), the resulting models are slightly different. From ratios all close to one in Table I we conclude that equally good approximate solutions of SVR problems are obtained after applying warm start.

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.

Data set	(ϵ, C)		(C, ϵ)
	Criterion in (7)	No criterion	No criterion
abalone	1.00	1.00	1.00
abalone-scale	1.00	1.00	1.00
cadata	**1.09**	**1.09**	**1.01**
cpusmall	1.00	1.00	**1.03**
cpusmall-scale	1.00	1.00	1.00
E2006-train	**0.99**	**0.99**	1.00
housing	**1.04**	**1.04**	1.00
housing-scale	1.00	1.00	1.00
log1p-E2006-train	1.00	1.00	1.00
mg	1.00	1.00	1.00
mg-scale	1.00	1.00	1.00
space-ga	1.00	1.00	1.00
space-ga-scale	1.00	1.00	1.00
YPMSD	**1.02**	**1.02**	**0.99**

More importantly, from Table I the setting via (7) without considering all grid points also has ratios close to one, indicating that it has covered needed parameters without sacrificing the performance.

C. Running-time Reduction of Warm-start Methods

To check the effectiveness of warm-start methods we present in Table II the following ratio.[5]

$$\frac{\text{Running time by applying warm start}}{\text{Running time by "Full and independent"}}. \tag{12}$$

A smaller ratio indicates a better time reduction by using warm start. From Table II, we have the following observations.

- All the values in Table II 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 $\tilde{\boldsymbol{w}}_C$ may be close enough to $\boldsymbol{w}_\infty$ by Theorem 3 and the condition (3) may hold ever since:

$$\|\nabla f(\tilde{\boldsymbol{w}}_{C_{\text{stop}}}; C)\| \le \tau \|\nabla f(\mathbf{0}; C)\|, \forall C \ge C_{\text{stop}}, \tag{13}$$

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 $\tilde{\boldsymbol{w}}_{C_{\text{stop}}}$ is only an approximate rather than the optimal solution. Thus it is still possible that the optimization method takes time at each C and

[5]Running time is estimated by the number of CG steps in the Newton method for training SVR. See details in supplementary materials.

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

Data set	(ϵ, C)		(C, ϵ)
	Criterion in (7)	No criterion	No criterion
abalone	0.12	0.12	0.54
abalone-scale	0.11	0.11	0.50
cadata	0.06	0.06	0.67
cpusmall	0.06	0.06	0.62
cpusmall-scale	0.10	0.10	0.69
E2006-train	0.14	0.14	0.35
housing	0.07	0.07	0.66
housing-scale	0.11	0.11	0.59
log1p-E2006-train	0.08	0.08	0.62
mg	0.13	0.13	0.61
mg-scale	0.13	0.13	0.61
space-ga	0.08	0.08	0.73
space-ga-scale	0.13	0.13	0.62
YPMSD	0.04	0.04	0.35

we expensively run the procedure all the way up to $C_{\max}$. Further, the selection of $C_{\max}$ is a tricky issue; in our experiment we choose 2^{50} in (10) without a good reason. Therefore, we can say that (7) is a relaxed condition of (13) to avoid the huge efforts of possibly running up to an extremely large $C_{\max}$.

- Between (C, ϵ) and (ϵ, C), we observe that (C, ϵ) costs more. It seems that warm start is less effective when C is fixed and ϵ is slightly changed. A reason might be that in our experiments, the number of SVR problems per ϵ value is often larger than that per C value. Then the time saving by applying warm start for the (C, ϵ) strategy is less dramatic. Note that in (9) we split $[\epsilon_{\min}, \epsilon_{\max}]$ to 20 intervals, but with a small $C_{\min}$ and a large $C_{\max}$, the number of C values in $[C_{\min}, C_{\max}]$ 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 not as robust as our search on a grid of parameters. In some situations, they lead to parameters with much worse CV MSE.

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 (ϵ, C) setting takes less running time, our recommended setting is to have ϵ 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

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

[2] B.-Y. Chu, C.-H. Ho, C.-H. Tsai, C.-Y. Lin, and C.-J. Lin. Warm start for parameter selection of linear classifiers. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2015.

[3] 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 Computation*, 15:2643–2681, 2003.

[4] D. DeCoste and K. Wagstaff. Alpha seeding for support vector machines. In *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 345–349, 2000.

[5] K. Duan, S. S. Keerthi, and A. N. Poo. Evaluation of simple performance measures for tuning SVM hyperparameters. *Neurocomputing*, 51:41–59, 2003.

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

[7] T. Hastie, S. Rosset, R. Tibshirani, and J. Zhu. The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 5:1391–1415, 2004.

[8] C.-H. Ho and C.-J. Lin. Large-scale linear support vector regression. *Journal of Machine Learning Research*, 13:3323–3348, 2012.

[9] J. H. Holland. *Adaptation in Natural and Artificial Systems*. MIT Press, 1992.

[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 *Proceedings of the Twenty Fifth International Conference on Machine Learning (ICML)*, 2008.

[11] C.-M. Huang, Y.-J. Lee, D. K. Lin, and S.-Y. Huang. Model selection for support vector machines via uniform design. *Computational Statistics and Data Analysis*, 52(1):335–346, 2007.

[12] J. Kennedy and R. Eberhart. Particle swarm optimization. In *Proceedings of International Conference on Neural Networks (ICNN)*, pages 1942–1948. 1995.

[13] S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[14] R. Kohavi and G. H. John. Automatic parameter selection by minimizing estimated error. In *Proceedings of the Twelfth International Conference on Machine Learning (ICML)*, pages 304–312, 1995.

[15] J.-H. Lee and C.-J. Lin. Automatic model selection for support vector machines. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, 2000.

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

[17] S.-W. Lin, Z.-J. Lee, S.-C. Chen, and T.-Y. Tseng. Parameter determination of support vector machine and feature selection using simulated annealing approach. *Applied Soft Computing*, 8(4):1505–1512, 2008.

[18] O. L. Mangasarian. A finite Newton method for classification. *Optimization Methods and Software*, 17(5):913–929, 2002.

[19] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[20] J. Močkus. On Bayesian methods for seeking the extremum. In *Proceedings of the IFIP Technical Conference*, pages 400–404, 1974.

[21] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, 1965.

[22] J. Snoek, H. Larochelle, and R. P. Adams. Practical Bayesian optimization of machine learning algorithms. In *Advances in Neural Information Processing Systems 25*, pages 2951–2959. 2012.

[23] B. Üstün, W. J. Melssen, M. K. Oudenhuijzen, and L. M. C. Buydens. Determination of optimal support vector regression parameters by genetic algorithms and simplex optimization. *Analytica Chimica Acta*, 544(1-2):292–305, 2005.

[24] V. Vapnik. *Statistical Learning Theory*. Wiley, New York, NY, 1998.

[25] Z. Wen, B. Li, R. Kotagiri, J. Chen, Y. Chen, and R. Zhang. Improving efficiency of svm k-fold cross-validation by alpha seeding. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017.

[26] C.-H. Wu, G.-H. Tzeng, and R.-H. Lin. A novel hybrid genetic algorithm for kernel function and parameter optimization in support vector regression. *Expert Systems with Applications*, 36(3):4725–4735, 2009.

[27] Z.-L. Wu, A. Zhang, C.-H. Li, and A. Sudjianto. Trace solution paths for SVMs via parametric quadratic programming. *KDD Worskshop: Data Mining Using Matrices and Tensors*, 2008.