Supplementary Materials for "Parameter Selection for Linear Support Vector Regression"

Jui-Yang Hsia, Chih-Jen Lin

I. A Review of Past Works

The selection of best parameters can be considered as an optimization problem. To be specific, we usually search for parameters that minimize an estimation of the generalization error (e.g., cross validation error). Parameter selection usually involves solving a complex, two-level problem. That is, for each parameter setting, we solve an optimization problem (a linear SVR here) to attain a model. Then, we can use the model to evaluate a validation loss (second level). Therefore, because of the complexity, many global optimization algorithms have been considered. We give a review of these approaches.

- Particle swam optimization [17]: see Section VI-B.
- Genetic optimization [12]: The process of genetic optimization is an analogy to an evolution process. It considers several iterations (or generations). From one iteration to another, a pool of parameters are mixed or mutated. By selecting those having smaller validation losses, the process can gradually find the best parameters.
- Simplex [23]: A simplex method considers a simplex with $D+1$ vertexes in the solution space with dimension D. At each iteration, a simplex method updates one vertex with worse performance to form a new simplex. To ensure that the newly formed simplex covers the best parameter, the updating process includes reflection, contraction and expansion.
- Bayesian optimization [22]: The concept of Bayesian optimization is to minimize the expected deviation from the optimal value. The current target function is drawn form a pool of functions with some distribution.
- Simulated annealing [18], [21]: see Section VI-A.

For works that have particularly investigated parameter selection for SVM, we briefly review some of them.

- [2] and [5] consider a gradient-based method to minimize a smooth estimation of CV accuracy. Kernel SVC (support vector classification) is considered.
- [3] focuses on linear SVC problems by applying warm-start techniques.
- [7], [19] and [26] consider warm-start techniques to speed up the cross validation procedure for kernel SVC. In the CV procedure, several related optimization problems are solved. The solution of one problem can be adjusted as an initial solution for another. The SVC dual problem is considered with the variable alpha, so the warm-start technique is referred to as alpha seeding.

Jui-Yang Hsia and Chih-Jen Lin are with National Taiwan University. (e-mail: hsiajuiyang5174@gmail.com and cjlin@csie.ntu.edu.tw).

- [8] empirically compare different validation measurements for kernel SVC.
- [10] and [28] investigate the optimality condition of SVC problems to see how the optimal solution changes along the regularization parameter C. They are able to find the entire solution path of all C values, although the process may involve some expensive matrix operations.
- [15] introduces a nested uniform design methodology to select parameters for evaluating CV performances. Both kernel SVC and SVR are considered.
- [20] (see Section VI.)
- [25] uses a genetic algorithm and a simplex optimization algorithm for selecting parameters of kernel SVR.
- [27] proposes a hybrid genetic algorithm for the determination of parameters in kernel SVR.

II. Proofs

A. Proof of Theorem 1

First, we prove that

$$\|\boldsymbol{w}_{C_1}\| \geq \|\boldsymbol{w}_{C_0}\| \text{ and } L(\boldsymbol{w}_{C_1}) \leq L(\boldsymbol{w}_{C_0}). \tag{II.1}$$

Suppose we have

$$f(\boldsymbol{w}_{C_1}; C_1) = f(\boldsymbol{w}_{C_0}; C_1). \tag{II.2}$$

Then (II.1) holds because the uniqueness of the solution implies that

$$\boldsymbol{w}_{C_1} = \boldsymbol{w}_{C_0} \quad \text{and} \quad L(\boldsymbol{w}_{C_1}) = L(\boldsymbol{w}_{C_0}).$$

If (II.2) is not true, we have

$$f(\boldsymbol{w}_{C_1}; C_1) < f(\boldsymbol{w}_{C_0}; C_1) \text{ and } \boldsymbol{w}_{C_1} \neq \boldsymbol{w}_{C_0}. \tag{II.3}$$

Because the solution is unique at C_0, we further have

$$f(\boldsymbol{w}_{C_0}; C_0) < f(\boldsymbol{w}_{C_1}; C_0). \tag{II.4}$$

From (II.3) and (II.4),

$$C_1(L(\boldsymbol{w}_{C_1}) - L(\boldsymbol{w}_{C_0})) < \frac{1}{2}(\|\boldsymbol{w}_{C_0}\|^2 - \|\boldsymbol{w}_{C_1}\|^2), \text{ and}$$

$$C_0(L(\boldsymbol{w}_{C_0}) - L(\boldsymbol{w}_{C_1})) < \frac{1}{2}(\|\boldsymbol{w}_{C_1}\|^2 - \|\boldsymbol{w}_{C_0}\|^2). \tag{II.5}$$

We argue that

$$L(\boldsymbol{w}_{C_1}) < L(\boldsymbol{w}_{C_0}). \tag{II.6}$$

Otherwise, with $C_0 < C_1$,

$$\begin{aligned}
&\frac{1}{2}(\|\boldsymbol{w}_{C_0}\|^2 - \|\boldsymbol{w}_{C_1}\|^2) \\
<&C_0(L(\boldsymbol{w}_{C_1}) - L(\boldsymbol{w}_{C_0})) \\
\leq&C_1(L(\boldsymbol{w}_{C_1}) - L(\boldsymbol{w}_{C_0})) \\
<&\frac{1}{2}(\|\boldsymbol{w}_{C_0}\|^2 - \|\boldsymbol{w}_{C_1}\|^2)
\end{aligned}$$

causes a contradiction. Further, (II.6) and (II.5) imply

$$\|\boldsymbol{w}_{C_0}\| < \|\boldsymbol{w}_{C_1}\|.$$

The first part is complete.

To prove

$$\lim_{C\to 0} \boldsymbol{w}_C = \mathbf{0}, \tag{II.7}$$

we first show from earlier results that for any given $\hat{C}$,

$$\|\boldsymbol{w}_C\| \le \|\boldsymbol{w}_{\hat{C}}\|, \forall C < \hat{C}.$$

Therefore, all $\boldsymbol{w}_C$, $\forall C \le \hat{C}$ are in a compact set. If (II.7) is wrong, there is a convergent sequence $\{\boldsymbol{w}_{C_t}\}$ such that

$$\lim_{t\to\infty} C_t = 0 \quad \text{and} \quad \lim_{t\to\infty} \boldsymbol{w}_{C_t} = \bar{\boldsymbol{w}} \neq \mathbf{0}. \tag{II.8}$$

Because $\boldsymbol{w}_{C_t}$ is the optimal solution when $C = C_t$,

$$\begin{aligned} &C_t L(\mathbf{0};\epsilon) \\ \ge &\frac{1}{2}\boldsymbol{w}_{C_t}^T \boldsymbol{w}_{C_t} + C_t L(\boldsymbol{w}_{C_t};\epsilon). \end{aligned}$$

From (II.8), taking the limit we have

$$0 \ge \frac{1}{2}\bar{\boldsymbol{w}}^T\bar{\boldsymbol{w}},$$

a contradiction to $\bar{\boldsymbol{w}} \neq \mathbf{0}$.

B. Proof of Theorem 2

We begin with defining

$$\psi_i(\boldsymbol{w}) = \max(|y_i - \boldsymbol{w}^T\boldsymbol{x}_i| - \epsilon, 0)$$

and proving the following Lemma.

Lemma II.1. *Consider L2 loss. Assume $L(0) > 0$. Then for all i and $C \le C_{min}$, we have*

$$\xi_\epsilon(\boldsymbol{w}_C;\boldsymbol{x}_i,y_i) \ge \xi_\epsilon(\mathbf{0};\boldsymbol{x}_i,y_i) - 2\psi_i(\mathbf{0})\frac{\delta_1 L(\mathbf{0})}{2\sum_{j=1}^{l}|y_j|}, \tag{II.9}$$

where

$$\xi_\epsilon(\boldsymbol{w};\boldsymbol{x}_i,y_i) = \max(|\boldsymbol{w}^T\boldsymbol{x}_i - y_i| - \epsilon, 0)^2.$$

Proof. First, we show that

$$|{\boldsymbol{w}_C}^T\boldsymbol{x}_i| \le \frac{\delta_1 L(\mathbf{0})}{2\sum_{j=1}^{l}|y_j|}. \tag{II.10}$$

By Theorem 1, we have

$$\|\boldsymbol{w}_C\| \le \|\boldsymbol{w}_{C_{\min}}\|.$$

This implies that

$$|{\boldsymbol{w}_C}^T\boldsymbol{x}_i| \le \|\boldsymbol{w}_C\|\|\boldsymbol{x}_i\| \le \|\boldsymbol{w}_{C_{\min}}\|\|\boldsymbol{x}_i\|. \tag{II.11}$$

Then (4) implies

$$\begin{aligned} |{\boldsymbol{w}_C}^T\boldsymbol{x}_i| \le &\|\boldsymbol{w}_{C_{\min}}\|\|\boldsymbol{x}_i\| \\ \le &\sqrt{2f(\mathbf{0};C_{\min})}\max_j\|\boldsymbol{x}_j\| \qquad \text{(II.12)} \\ = &\sqrt{2C_{\min}L(\mathbf{0})}\max_j\|\boldsymbol{x}_j\| \\ = &\frac{\delta_1 L(\mathbf{0})}{2\sum_{j=1}^{l}|y_j|}, \end{aligned}$$

where (II.12) comes from

$$\frac{1}{2}\|\boldsymbol{w}_{C_{\min}}\|^2 \le f(\boldsymbol{w}_{C_{\min}};C_{\min}) \le f(\mathbf{0};C_{\min})$$

by using the fact $\boldsymbol{w}_{C_{\min}}$ is the solution at $C = C_{\min}$ and the loss function is non-negative. Next, we prove (II.9).

$$\begin{aligned} &\xi_\epsilon(\boldsymbol{w}_C;\boldsymbol{x}_i,y_i) \\ = &\max(|\boldsymbol{w}_C^T\boldsymbol{x}_i - y_i| - \epsilon, 0)^2 \\ \ge &\max((-|\boldsymbol{w}_C^T\boldsymbol{x}_i| + |y_i|) - \epsilon, 0)^2 \\ = &\max(-|\boldsymbol{w}_C^T\boldsymbol{x}_i| + \psi_i(\mathbf{0}), 0)^2 \qquad \text{(II.13)} \\ = &\begin{cases} (-|\boldsymbol{w}_C^T\boldsymbol{x}_i| + \psi_i(\mathbf{0}))^2 \\ = (\boldsymbol{w}_C^T\boldsymbol{x}_i)^2 - 2|\boldsymbol{w}_C^T\boldsymbol{x}_i|\psi_i(\mathbf{0}) + \psi_i(\mathbf{0})^2 \\ \quad \text{if } 0 < -|\boldsymbol{w}_C^T\boldsymbol{x}_i| + \psi_i(\mathbf{0}) \\ 0 = \psi_i(\mathbf{0})^2 - \psi_i(\mathbf{0})^2 \ge \psi_i(\mathbf{0})^2 - 2\psi_i(\mathbf{0})^2 \\ \quad \text{if } 0 \ge -|\boldsymbol{w}_C^T\boldsymbol{x}_i| + \psi_i(\mathbf{0}) \end{cases} \\ \ge &\psi_i(\mathbf{0})^2 - 2\psi_i(\mathbf{0})|\boldsymbol{w}_C^T\boldsymbol{x}_i| \\ \ge &\psi_i(\mathbf{0})^2 - 2\psi_i(\mathbf{0})\frac{\delta_1 L(\mathbf{0})}{2\sum_{j=1}^{l}|y_j|} \qquad \text{(II.14)} \\ = &\xi_\epsilon(\mathbf{0};\boldsymbol{x}_i,y_i) - 2\psi_i(\mathbf{0})\frac{\delta_1 L(\mathbf{0})}{2\sum_{j=1}^{l}|y_j|}, \end{aligned}$$

where (II.14) is from (II.10). We give the following details for deriving (II.13). Let

$$A = -|\boldsymbol{w}_C^T\boldsymbol{x}_i| \le 0 \text{ and } B = |y_i| - \epsilon.$$

We prove that

$$\max(A + B, 0) = \max(A + \max(B, 0), 0). \tag{II.15}$$

If $B \ge 0$, then

$$\max(B, 0) = B$$

and the equality holds. If $B < 0$, then with $A \le 0$, both sides of (II.15) are zero. Therefore, (II.15) holds. With

$$\psi_i(\mathbf{0}) = |y_i| - \epsilon,$$

we have (II.13). □

Proof of Theorem 2. If $L(0) = 0$, it is trivial that

$$L(\boldsymbol{w}_C) \ge (1 - \delta_1)L(0) = 0$$

for every $C < \infty$. If $L(0) > 0$, then for any $C \le C_{\min}$,

$$\begin{aligned} L(\boldsymbol{w}_C) =& \sum_{i=1}^{l} \xi_\epsilon(\boldsymbol{w}_C; \boldsymbol{x}_i, y_i) \\ \ge& \sum_{i=1}^{l} \big(\xi_\epsilon(\mathbf{0}; \boldsymbol{x}_i, y_i) - 2\psi_i(\mathbf{0})\delta_1 \frac{L(\mathbf{0})}{2\sum_{j=1}^{l} |y_j|}\big) \quad \text{(II.16)} \\ =& L(\mathbf{0}) - \delta_1 L(\mathbf{0}) \times \frac{2\sum_{i=1}^{l} \psi_i(\mathbf{0})}{2\sum_{j=1}^{l} |y_j|} \\ \ge& (1-\delta_1) \times L(\mathbf{0}), \end{aligned}$$

where (II.16) is from Lemma II.1. □

In comparison with [3] we see that deriving a $C_{\min}$ for linear classification is simpler: By checking when the model predicts all training data to be in one class, there is no need to involve the zero point. Further, no parameter like δ_1 here is required.

C. Proof of Theorem 3

We omit the proof of (5) because it is the same as that in [3]. Here we prove that if the L2 loss is used, then $W_\infty \neq \emptyset$. The optimization problem $\inf_{\boldsymbol{w}} L(\boldsymbol{w})$ can be rewritten as

$$\begin{aligned} \min_{\boldsymbol{\xi}, \boldsymbol{w}} \quad & \|\boldsymbol{\xi}\|^2 \quad \text{(II.17)} \\ \text{subject to} \quad & \xi_i \ge |y_i - \boldsymbol{w}^T \boldsymbol{x}_i| - \epsilon\,, i = 1, \ldots, l. \end{aligned}$$

Because

$$\boldsymbol{w} = \mathbf{0} \quad \text{and} \quad \xi_i = \max(0, |y_i| - \epsilon), \quad \forall i$$

satisfies the constraints, so problem (II.17) is feasible. Besides, the feasible region of (II.17) is a polyhedral convex set, and the objective function $\|\boldsymbol{\xi}\|^2$ is convex quadratic. Therefore, the infimum value is attained [24, Corollary 27.3.1]. That is, $W_\infty \neq \emptyset$.

D. Proof of Theorem 4

Suppose the result is wrong. Then there exists $\delta > 0$ so that for any $\bar{\epsilon} > 0$, there is $\epsilon \le \bar{\epsilon}$ such that

$$\|\boldsymbol{w}_\epsilon - \boldsymbol{w}_0\| \ge \delta. \quad \text{(II.18)}$$

Next we show that $\boldsymbol{w}_\epsilon, \epsilon \ge 0$ is in a compact set. Because $\boldsymbol{w}_\epsilon$ is optimal at ϵ,

$$\begin{aligned} \frac{1}{2}\|\boldsymbol{w}_\epsilon\|^2 + CL(\boldsymbol{w}_\epsilon; \epsilon) &\le \frac{1}{2}\|\boldsymbol{w}_0\|^2 + CL(\boldsymbol{w}_0; \epsilon) \\ &\le \frac{1}{2}\|\boldsymbol{w}_0\|^2 + CL(\boldsymbol{w}_0; 0), \end{aligned}$$

where the last inequality is from $L(\boldsymbol{w}; \epsilon) \le L(\boldsymbol{w}; 0)$ for any $\boldsymbol{w}$ and $\epsilon \ge 0$. With $L(\boldsymbol{w}_\epsilon; \epsilon) \ge 0$,

$$0 \le \frac{1}{2}\|\boldsymbol{w}_\epsilon\|^2 \le \frac{1}{2}\|\boldsymbol{w}_0\|^2 + CL(\boldsymbol{w}_0; 0)$$

and therefore $\boldsymbol{w}_\epsilon, \epsilon \ge 0$ is in a compact set. This property and (II.18) imply that there is a convergent subsequence $\{\boldsymbol{w}_{\epsilon_t}\}$ such that

$$\lim_{t\to\infty} \epsilon_t = 0 \quad \text{and} \quad \lim_{t\to\infty} \boldsymbol{w}_{\epsilon_t} = \bar{\boldsymbol{w}} \neq \boldsymbol{w}_0. \quad \text{(II.19)}$$

Because $\boldsymbol{w}_{\epsilon_t}$ is optimal at $\epsilon = \epsilon_t$, we have

$$\frac{1}{2}\|\boldsymbol{w}_{\epsilon_t}\|^2 + CL(\boldsymbol{w}_{\epsilon_t}; \epsilon_t) \le \frac{1}{2}\|\boldsymbol{w}_0\|^2 + CL(\boldsymbol{w}_0; \epsilon_t).$$

From (II.19) and the continuity of the loss function, taking the limit leads to

$$\frac{1}{2}\|\bar{\boldsymbol{w}}\|^2 + CL(\bar{\boldsymbol{w}}; 0) \le \frac{1}{2}\|\boldsymbol{w}_0\|^2 + CL(\boldsymbol{w}_0; 0),$$

so $\bar{\boldsymbol{w}}$ is also an optimal solution at $\epsilon = 0$. However, $\boldsymbol{w}_0$ is the unique optimal solution at $\epsilon = 0$, and therefore there is a contradiction.

III. Details of the Proposed Procedure

The procedure is given in Algorithm 1. More explanation about how the CV procedure and the warm-start technique are implemented together in the algorithm can be seen in Section 3.4 of [3].

We give more details about the termination condition (7) and its implementation. Assume that $\tilde{\boldsymbol{w}}_C$ is the obtained approximate solution satisfying

$$\|\nabla f(\tilde{\boldsymbol{w}}_C; C)\| \le \tau \|\nabla f(\mathbf{0}; C)\|.$$

We then use $\tilde{\boldsymbol{w}}_C$ as the initial solution of the Newton method for training SVR with ΔC. If immediately the stopping condition is satisfied

$$\|\nabla f(\tilde{\boldsymbol{w}}_C; \Delta C)\| \le \tau \|\nabla f(\mathbf{0}; \Delta C)\|,$$

then $\tilde{\boldsymbol{w}}_C$ is returned as an approximate solution at ΔC. That is,

$$\tilde{\boldsymbol{w}}_{\Delta C} = \tilde{\boldsymbol{w}}_C.$$

If this occurs for t_{stop} consecutive problems, we terminate the parameter selection procedure.

In [3], they prove a theorem to explain that generally (7) should hold after C is large enough. We restate their theorem and check if it holds in our SVR problem.

Theorem 1. *For L2-loss SVR, we have*

$$\lim_{C\to\infty} \frac{\|\nabla f(\boldsymbol{w}_C; \Delta C)\|}{\|\nabla f(\mathbf{0}; C)\|} = 0. \quad \text{(III.1)}$$

Proof. First, we have

$$\nabla f(\boldsymbol{w}; C) = \boldsymbol{w} + C\nabla L(\boldsymbol{w})$$

and

$$\nabla f(\boldsymbol{w}_C; C) = \boldsymbol{w}_C + C\nabla L(\boldsymbol{w}_C) = \mathbf{0}.$$

Then, the numerator of (III.1) can be written as

$$\begin{aligned} \nabla f(\boldsymbol{w}_C; \Delta C) &= \boldsymbol{w}_C + \Delta C \nabla L(\boldsymbol{w}_C) \\ &= -C\nabla L(\boldsymbol{w}_C) + \Delta C \nabla L(\boldsymbol{w}_C) \\ &= (\Delta - 1) C \nabla L(\boldsymbol{w}_C). \end{aligned}$$

If L2 loss is used, $L(\boldsymbol{w})$ is continuously differentiable. Taking the limit leads to

$$\begin{aligned}\lim_{C\to\infty}\frac{\|\nabla f(\boldsymbol{w}_C;\Delta C)\|}{\|\nabla f(\mathbf{0};C)\|} &= \lim_{C\to\infty}\frac{(\Delta-1)C\|\nabla L(\boldsymbol{w}_C)\|}{C\|\nabla L(\mathbf{0})\|}\\ &= \lim_{C\to\infty}\frac{(\Delta-1)\|\nabla L(\boldsymbol{w}_C)\|}{\|\nabla L(\mathbf{0})\|}\\ &= (\Delta-1)\frac{\|\nabla L(\boldsymbol{w}_\infty)\|}{\|\nabla L(\mathbf{0})\|}\\ &= 0,\end{aligned}$$

where the last equality from the definition of $\boldsymbol{w}_\infty$ in (5). The theorem is complete with that (III.1) is zero when $C\to\infty$. □

A. Some notes on the Stopping Criterion of Solving Each SVR Problem

We discuss why in the stopping condition (7) of solving each SVR problem, we use the zero point to calculate the right side. That is,

$$\|\nabla f(\tilde{\boldsymbol{w}}_{C,\epsilon};C,\epsilon)\|\le\tau\|\nabla f(\mathbf{0};C,\epsilon)\|.$$

We give the explanation offered in [3]. Assume we are solving a problem with parameters

$$\Delta C,\epsilon.$$

The initial $\boldsymbol{w}_{\text{init}}$, by the warm-start setting, is an approximate solution at

$$C,\epsilon.$$

From Theorem 3, when C is large, $\boldsymbol{w}_{\text{init}}=\tilde{\boldsymbol{w}}_{C,\epsilon}$ is close to a solution at ΔC. Therefore,

$$\|\nabla f(\boldsymbol{w}_{\text{init}};\Delta C,\epsilon)\|\approx 0.$$

The stopping condition becomes very tight. The same situation occurs if $\epsilon\approx 0$ because of Theorem 4. Therefore, it is more suitable to have $\|\nabla f(\mathbf{0};C,\epsilon)\|$ on the right side so criteria across parameters are more consistent.

IV. Details of Experimental Settings

We mentioned that CV MSE is used to select the best parameters. The MSE to validate one fold is

$$\frac{\sum_i\{(y_i-\tilde{\boldsymbol{w}}^T\boldsymbol{x}_i)^2\mid \boldsymbol{x}_i \text{ in the validation fold}\}}{\text{size of the validation fold}}.$$

In the end results of all folds are averaged as the CV MSE. Our implementation is extended from LIBLINEAR [9], and we apply a Newton method for solving each SVR optimization problem. Parameters used in our procedure are $\tau=10^{-4}$ in (3) and $\delta_1=0.1$ in (4). The data statistics are presented in Table A.

If a running-time comparison is needed, we check the total number of CG steps in the Newton method for training a linear SVR. Note that each Newton iteration involves some inner CG (conjugate gradient) steps, each of which takes the same amount of operations. Because CG steps are the main computational cost, it is well known [13], [14] that comparing the number of CG steps gives an accurate timing comparison.

Algorithm 1 The proposed procedure for SVR parameter selection.

1: **Given**
2: K as number of CV folds.
3: τ as stopping tolerance in (3).
4: Δ as C increment.
5: S as number of ϵ steps and $\square=\epsilon_{\text{max}}/S$.
6: $\delta_1\in(0,1)$ as the parameter to calculate C_{min}.
7: $C_{\text{max}}=$ a large constant
8: $t_{\text{stop}}=5$.
9: **End Given**
10: Initialize $\epsilon_{\text{min}}=0$, $\epsilon_{\text{max}}=\max_i|y_i|$.
11: Initialize best CV score $\text{MSE}_{\text{best}}\leftarrow\infty$.
12: **for** $\epsilon=\epsilon_{\text{max}},\epsilon_{\text{max}}-\square,\ldots,\epsilon_{\text{min}}$ **do**
13: **for** CV fold $k=1,\ldots,K$ **do**
14: Initialize solution $\bar{\boldsymbol{w}}^k\leftarrow\mathbf{0}$.
15: **end for**
16: Calculate C_{min} by (4).
17: Initialize $t=-1$.
18: **for** $C=C_{\text{min}},\Delta C_{\text{min}},\Delta^2C_{\text{min}},\ldots,C_{\text{max}}$ **do**
19: **for** CV fold $k=1,\ldots,K$ **do**
20: Apply warm start with the initial solution $\bar{\boldsymbol{w}}^k$.
21: Use all data except fold k for training.
22: Obtain an approximate solution $\tilde{\boldsymbol{w}}^k_{C,\epsilon}$ satisfying (3) with the stopping tolerance τ.
23: Predict fold k by $\tilde{\boldsymbol{w}}^k_{C,\epsilon}$.
24: **if** $\bar{\boldsymbol{w}}^k\neq\tilde{\boldsymbol{w}}^k_{C,\epsilon}$ **then**
25: $t=-1$.
26: **end if**
27: $\bar{\boldsymbol{w}}^k\leftarrow\tilde{\boldsymbol{w}}^k_{C,\epsilon}$.
28: **end for**
29: Calculate CV MSE from stored predicted values of each fold in line 23.
30: **if** $\text{MSE}<\text{MSE}_{\text{best}}$ **then**
31: $\text{MSE}_{\text{best}}\leftarrow\text{MSE}$.
32: $C_{\text{best}}\leftarrow C$.
33: $\epsilon_{\text{best}}\leftarrow\epsilon$.
34: **end if**
35: $t\leftarrow t+1$.
36: **if** $t=t_{\text{stop}}$ **then**
37: **break**
38: **end if**
39: **end for**
40: **end for**
41: **return** C_{best} and ϵ_{best}

V. Additional Experiments of the Proposed Procedure

We conduct more experiments to analyze the τ value in (3), the cost between two types of loops (C,ϵ) and (ϵ,C), and the selection of t_{stop} in the termination condition (7) for the C sequence.

A. Selected (ϵ,C) Values

In Table B, we list the selected (ϵ,C) values by the three settings considered in Tables I and II. Besides, we also list the

Table A: Data statistics.

	l: # instances	n: # features
abalone	4,177	8
bodyfat	252	14
cadata	20,640	8
cpusmall	8,192	12
E2006-train	16,087	150,360
eunite2001	336	16
housing	506	13
log1p-E2006-train	16,087	4,272,227
mg	1,385	6
mpg	392	7
pyrim	74	27
space-ga	3,107	6
triazines	186	60
YearPredictionMSD	463,715	90

selected parameters by the two alternative parameter selection methods investigated later in Section VI.

We observe that two alternative parameter selection methods usually end up with a large C value. This result can be explained by the combination of two factors.

- The initial C value considered by the two methods may be large because it is uniformly drawn from a large interval. In contrast, our grid search starts from a small value $C_{\min}$.
- Once a large initial C value is considered, the search procedure may stay in a nearby region because the CV MSE between two large C values is indifferent. Note that from Theorem 3, when C value is large enough, the solution is close to $\boldsymbol{w}_\infty$.

B. The Selection of τ in the Stopping Condition of Training Each SVR

The choose of τ is a trade-off between the cost of time and the approximation of $\tilde{\boldsymbol{w}}_{C,\epsilon}$ to $\boldsymbol{w}_{C,\epsilon}$. The experiment considers two settings (C, ϵ) and (ϵ, C) without imposing a termination condition on the C sequence.

Results of using various τ values are in Table C, where we check both CV MSE and running time. The CV MSE is stable when τ is smaller than 10^{-4}. For the running time, from Table Cb, the cost can significant increase with smaller τ. If τ is very small, the cost may be hundred times more than that of using a large τ.

Because CV MSE has stabilized after $\tau = 10^{-4}$ and the cost with $\tau = 10^{-5}$ and 10^{-6} is significantly higher, we decide to set $\tau = 10^{-4}$ in our implementation.

In Table Ca, for the bodyfat and pyrim sets the CV MSE is worse under a smaller τ. We give further investigation in Section V-E.

C. Loop Selection

Two settings (ϵ, C) and (C, ϵ) under different number intervals of $[\epsilon_{\min}, \epsilon_{\max}]$ are compared. We consider 20, 40, 60, 80 and 100 intervals. We do not impose a termination criterion on the C sequence so both (ϵ, C) and (C, ϵ) settings solve the same number of SVR problems. The CV MSE and time comparison is shown in Tables Da and Db.

From Table Da, the best CV MSE is not significantly smaller when the number of ϵ intervals increases. This result indicates that the 20 intervals have an enough coverage over the search space. A finer grid of the $[\epsilon_{\min}, \epsilon_{\max}]$ space may not be required.

For the running time, if ϵ is in the outer loop, we observe that the total running time of (ϵ, C) is almost proportional to the number of ϵ intervals. This result indicates that the training of the C sequence under a given ϵ is similar to that under nearby ϵ values. In contrast, the cost of (C, ϵ) is not increased much if the number of ϵ intervals in enlarged. The reason is apparently that warm start on the ϵ sequence is effective.

In conclusion, a finer grid of a parameter makes warm start more effective if that parameter is used in the inner loop. Because CV MSE has stabilized by using 20 ϵ values, and in general the number of C values checked is higher, we decide to consider (ϵ, C) by letting ϵ be in the outer loop.

D. Selection of t_{stop} in the Termination Condition (7)

To ensure that a proper range of C is covered by applying the stopping criterion (7), the selection of t_{stop} is crucial. A larger t_{stop} indicates that the current $\tilde{\boldsymbol{w}}_{C,\epsilon}$ is required to satisfy the stopping condition of more subsequent SVR problems. We consider the (ϵ, C) setting with the criterion (7) by trying

$$t_{\text{stop}} = 3, 4, 5, 6. \tag{V.1}$$

The results of comparing with the setting of not applying (7) are presented in Tables Ea and Eb.

From Table Ea, if $t_{\text{stop}} \geq 5$, then the CV MSE is the same as if (7) is not applied (i.e., no termination condition). Further, from Table Eb the cost under various t_{stop} is similar. Therefore, we consider that $t_{\text{stop}} = 5$ is a proper choice.

E. Further Analysis on the τ value for Data Sets Bodyfat and Pyrim

In Section V-B to check the stopping condition (3) in training an SVR, we generally observe that as the tolerance τ decreases the CV MSE slightly improves and converges to that of using the optimal $\boldsymbol{w}_{C,\epsilon}$. However, the opposite occurs for problems Bodyfat and Pyrim. It is known that in some situations an approximate solution leads to a better model than that by the optimal solution. We suspect that this situation occurs for the two data sets. We confirm this suspicion by conducting the parameter selection without the warm start strategy. In Table F we report CV MSE under various τ values. Results show that as a stricter tolerance τ is used, the CV MSE slightly deteriorates.

VI. Comparison with Other Approaches for SVR Parameter Selection

In this section we compare our approach (searching a grid of parameters with the warm-start trick) with some other parameter selection methods.

Table B: Selected best parameters are showed in the pair $(\epsilon, \log_2 C)$.

Data set \ Method	(ϵ, C)		(C, ϵ)	SA	PSO
	Criterion in (7)	No criterion	No criterion		
abalone	(0.00, 0.00)	(0.00, 0.00)	(0.00, 0.00)	(0.04, 48.64)	(0.10, 49.59)
abalone-scale	(0.00, 3.00)	(0.00, 3.00)	(0.00, 3.00)	(0.01, 48.59)	(0.00, 49.66)
bodyfat	(0.00, -12.00)	(0.00, -12.00)	(0.00, -12.00)	(0.00, 48.63)	(0.00, 47.50)
bodyfat-scale	(0.00, 6.00)	(0.00, 6.00)	(0.00, 6.00)	(0.00, 48.64)	(0.00, 46.45)
cadata	(50000.10, -9.00)	(49999.98, -9.00)	(49999.98, -9.00)	(10778.74, 46.01)	(10911.24, 48.44)
cpusmall	(0.00, -32.00)	(0.00, -32.00)	(0.00, -32.00)	(0.09, 48.64)	(0.00, 49.42)
cpusmall-scale	(0.00, -1.00)	(0.00, -1.00)	(0.00, -1.00)	(0.01, 48.60)	(0.15, 49.31)
E2006.train	(0.00, -1.00)	(0.00, -1.00)	(0.00, -1.00)	(0.00, 45.85)	(0.09, 49.77)
eunite2001	(0.00, 2.00)	(0.00, 2.00)	(0.00, 2.00)	(0.01, 48.64)	(0.00, 47.16)
housing	(0.00, -7.00)	(0.00, -7.00)	(0.00, -7.00)	(0.74, 47.33)	(0.27, 49.70)
housing-scale	(0.00, -2.00)	(0.00, -2.00)	(0.00, -2.00)	(0.00, 48.64)	(0.00, 48.55)
log1p.E2006.train	(0.00, -12.00)	(0.00, -12.00)	(0.00, -12.00)	(0.17, 33.26)	(0.18, 49.29)
mg	(0.00, 0.00)	(0.00, 0.00)	(0.00, 0.00)	(0.00, 48.65)	(0.00, 48.77)
mg-scale	(0.07, 1.00)	(0.07, 1.00)	(0.07, 1.00)	(0.05, -0.25)	(0.04, 49.48)
mpg	(0.00, -9.00)	(0.00, -9.00)	(0.00, -9.00)	(0.59, 17.97)	(0.05, 49.81)
mpg-scale	(0.00, 0.00)	(0.00, 0.00)	(0.00, 0.00)	(0.00, 48.64)	(0.00, 49.63)
pyrim	(0.00, 2.00)	(0.00, 2.00)	(0.00, 2.00)	(0.23, 49.93)	(0.26, 49.30)
pyrim-scale	(0.09, 2.00)	(0.09, 2.00)	(0.09, 2.00)	(0.10, 47.24)	(0.10, 47.95)
space-ga	(0.00, -56.00)	(0.00, -56.00)	(0.00, -56.00)	(0.00, 48.64)	(0.02, 49.90)
space-ga-scale	(0.00, 8.00)	(0.00, 8.00)	(0.00, 8.00)	(0.00, 48.64)	(0.00, 49.67)
triazines	(0.00, -1.00)	(0.00, -1.00)	(0.00, -1.00)	(0.30, 48.60)	(0.30, 48.98)
triazines-scale	(0.00, -4.00)	(0.00, -4.00)	(0.00, -4.00)	(0.00, 48.64)	(0.00, 49.77)
YearPredictionMSD	(100.55, -18.00)	(100.55, -18.00)	(100.55, -18.00)	(9.90, 47.86)	(11.84, 50.00)

A. Simulated Annealing Approach

We consider simulated annealing (SA) approach in [20]. In [20], the SA approach has successfully determined parameters and selected features for support vector machine (SVM). SA approach is a global search algorithm inspired from the cooling process of metal: the particle in metal will converge to the lowest-energy state when the initial heat is high enough.

1) Implementation Details: Our implementations are based on [20] and details are in Algorithm 2. A search range

$$[C_{\min}, C_{\max}] \times [\epsilon_{\min}, \epsilon_{\max}] \tag{VI.1}$$

must be pre-specified. In the paper, we have derived all bounds except $C_{\max}$. We follow the setting in Section V-A to set $C_{\max} = 2^{50}$.

However, following [20], because the range of $[C_{\min}, C_{\max}]$ is usually much greater than the one of $[\epsilon_{\min}, \epsilon_{\max}]$, the domination of one parameter may occur. Therefore, an equally scaled space is considered to be the SA search range

$$[0, 1] \times [0, 1] \tag{VI.2}$$

instead of directly using the range in (VI.1). For each point in (VI.2) we must map it back to (VI.1) for training SVR. Thus, we define the following mapping between (VI.2) and (VI.1).

$$\Theta(z_0, z_1) = (z_0 \times C_{\max} + C_{\min}, z_1 \times \epsilon_{\max} + \epsilon_{\min}),$$

where

$$(z_0, z_1) \in [0, 1] \times [0, 1].$$

As a result, we can evaluate CV MSE with parameter pair $(C, \epsilon) = \Theta(z_0, z_1)$. For the selection of initial starting point $\boldsymbol{z}^0$, we follow [20] to randomly choose some values from (VI.2). We set the maximal number of iterations to be 300.

B. Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a global optimization method [6], [17]. PSO is a population-base approach that has a swarm of particles to search for an optimal solution (or position). In the process, each particle considers information from itself and its neighbors to update the position. Among the various PSO implementations, we consider the one in [29]. It is one of the most popular PSO variants and possesses a strong ability for searching an optimal solution [16].

For implementation details, we consider the same search space (VI.2) as in the annealing approach and use Θ to retrieve (C, ϵ). We set the maximal number of iterations to be 10 and each iteration involves 40 CV MSE evaluations.

C. Results and Analysis

We compare SA, PSO and our methods in this section. Results of showing the ratio

$$\frac{\text{Best CV MSE by a method}}{\text{Best CV MSE by "Full and independent"}} \tag{VI.3}$$

in comparison with the baseline "Full and independent" approach are in Table G. From this table we have have following observations.

- PSO and SA methods can achieve competitive CV MSE values because most of the ratios are close to 1.
- When stability is considered, SA and PSO may not be a suitable choice. For example, in log1p.E2006.train, pyrim, traizines and traizines-scale, the CV MSE is 10% worse than the baseline. This result is expected because for an optimization technique such as SA or PSO, it is possible that the procedure reaches a local minimum. In contrast, a grid search, usually not practically feasible, guarantees to find a point close to a global minimum. Because linear SVR involves only two parameters, we can afford to conduct a grid search.

Table C: The comparison of using $\tau = 10^{-2}, 10^{-3}, 10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}$ is presented. The (ϵ, C) setting does not impose a termination condition on the C sequence, so both (ϵ, C) and (C, ϵ) run the full grid. All values are normalized by the first column.

(a) CV MSE

Data set	(C, ϵ)					(ϵ, C)				
	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
abalone	1.00	0.90	0.90	0.90	0.90	1.05	0.90	0.90	0.90	0.90
abalone-scale	1.00	0.96	0.96	0.96	0.96	1.07	0.97	0.96	0.96	0.96
bodyfat	1.00	0.37	0.39	0.42	0.42	6.25	0.44	0.42	0.42	0.42
bodyfat-scale	1.00	0.97	0.96	0.96	0.96	1.08	0.96	0.96	0.96	0.96
cadata	1.00	0.89	0.50	0.49	0.49	1.01	0.94	0.54	0.49	0.49
cpusmall	1.00	1.00	0.83	0.81	0.79	1.00	1.00	0.81	0.81	0.80
cpusmall-scale	1.00	0.96	0.96	0.96	0.96	1.21	0.98	0.96	0.96	0.96
E2006-train	1.00	1.00	1.00	0.97	0.97	1.00	1.00	0.99	0.97	0.97
eunite2001	1.00	0.93	0.89	0.89	0.89	1.02	0.89	0.89	0.89	0.89
housing	1.00	0.72	0.42	0.42	0.42	1.18	0.77	0.44	0.42	0.42
housing-scale	1.00	1.00	1.00	1.00	1.00	1.01	1.00	1.00	1.00	1.00
log1p-E2006-train	1.00	0.95	0.95	0.95	0.95	2.06	0.98	0.95	0.95	0.95
mg	1.00	1.00	1.00	1.00	1.00	1.02	1.00	1.00	1.00	1.00
mg-scale	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
mpg	1.00	0.61	0.52	0.50	0.49	6.24	0.82	0.52	0.50	0.49
mpg-scale	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
pyrim	1.00	1.19	1.19	1.19	1.19	1.32	1.18	1.19	1.19	1.19
pyrim-scale	1.00	0.98	0.99	1.06	1.06	1.06	1.05	1.06	1.06	1.06
space-ga	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
space-ga-scale	1.00	1.00	1.00	1.00	1.00	1.01	1.00	1.00	1.00	1.00
triazines	1.00	1.02	1.02	1.02	1.02	1.02	1.02	1.02	1.02	1.02
triazines-scale	1.00	1.00	0.99	0.99	0.99	1.01	1.00	0.99	0.99	0.99
YearPredictionMSD	1.00	0.43	0.33	0.33	0.33	6.74	1.41	0.34	0.33	0.33

(b) Running time

Data set	(C, ϵ)					(ϵ, C)				
	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}	10^{-2}	10^{-3}	10^{-4}	10^{-5}	10^{-6}
abalone	1.00	2.00	3.49	5.45	7.81	0.33	0.50	0.75	1.09	1.49
abalone-scale	1.00	1.96	3.46	4.89	6.27	0.33	0.49	0.78	1.10	1.46
bodyfat	1.00	1.54	2.13	3.15	3.76	0.15	0.27	0.40	0.56	0.74
bodyfat-scale	1.00	1.96	4.31	6.91	10.38	0.16	0.37	0.73	1.15	1.63
cadata	1.00	3.08	7.16	14.06	23.92	0.18	0.35	0.61	1.03	1.59
cpusmall	1.00	1.68	3.82	6.79	10.12	0.15	0.25	0.39	0.60	0.89
cpusmall-scale	1.00	1.85	3.58	5.37	7.14	0.18	0.33	0.51	0.86	1.29
E2006-train	1.00	1.29	1.67	4.80	19.71	0.41	0.53	0.68	0.94	1.74
eunite2001	1.00	1.62	3.14	4.53	5.89	0.18	0.33	0.55	0.85	1.24
housing	1.00	2.26	5.68	10.15	16.61	0.19	0.33	0.59	1.08	1.61
housing-scale	1.00	2.03	3.45	4.96	6.90	0.21	0.37	0.63	0.97	1.40
log1p-E2006-train	1.00	2.18	5.55	11.81	20.87	0.26	0.39	0.68	1.26	2.53
mg	1.00	1.78	2.84	3.98	5.07	0.23	0.40	0.59	0.89	1.23
mg-scale	1.00	1.69	2.56	3.16	3.77	0.20	0.35	0.54	0.75	0.99
mpg	1.00	2.16	3.74	6.04	8.39	0.23	0.37	0.58	0.83	1.14
mpg-scale	1.00	1.96	3.30	4.15	4.96	0.22	0.38	0.62	0.86	1.13
pyrim	1.00	3.10	10.43	28.23	46.86	0.19	0.37	0.87	1.63	2.28
pyrim-scale	1.00	2.42	6.04	12.14	22.06	0.20	0.38	0.70	1.18	1.75
space-ga	1.00	1.59	2.43	2.88	3.08	0.16	0.21	0.28	0.34	0.40
space-ga-scale	1.00	1.49	2.04	2.54	3.09	0.19	0.30	0.43	0.55	0.67
triazines	1.00	3.83	13.22	43.53	126.28	0.19	0.37	0.93	2.44	6.55
triazines-scale	1.00	2.84	10.74	29.33	78.30	0.18	0.36	0.86	2.09	5.61
YearPredictionMSD	1.00	2.42	7.12	20.72	63.21	0.13	0.27	0.81	2.04	4.47

The above results are obtained by running SA or PSO up to a pre-specified number of iterations. The termination of these methods is certainly a practical issue.

Next we compare the running time. Because no clear termination condition is available for SA or PSO, we consider the following setting. We split the entire process of running SA or PSO to several stages. At each stage we respectively present the CV MSE and running time up to the current stage in comparison with the final result of the proposed Algorithm 1. Specifically, the following two ratios are calculated.

$$\frac{\text{CV MSE at the current stage}}{\text{Final CV MSE by Algorithm 1}} \tag{VI.4}$$

and

$$\frac{\text{cumulative running time}}{\text{Total running time by Algorithm 1}}. \tag{VI.5}$$

Results for SA and PSO are respectively given in Tables H and I. Clearly, we see that CV MSE ratio gradually decreases, while the running-time ratio increases. When the CV MSE ratio reaches 1 or a smaller value, a running-time ratio smaller

Table D: The comparison of using different number of ϵ values in the parameter search is presented. We consider 20, 40, 60, 80 and 100. The (ϵ, C) setting does not impose a termination condition on the C sequence, so both (ϵ, C) and (C, ϵ) run the full grid. All values are normalized by the first column.

(a) CV MSE

Data set	(C, ϵ)					(ϵ, C)				
	20	40	60	80	100	20	40	60	80	100
abalone	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
abalone-scale	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
bodyfat	1.00	1.05	1.05	1.07	1.10	1.07	1.07	1.07	1.07	1.07
bodyfat-scale	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
cadata	1.00	0.99	1.00	0.99	0.99	1.07	1.07	1.07	1.07	1.07
cpusmall	1.00	1.01	1.02	0.98	1.01	0.98	0.98	0.98	0.98	0.98
cpusmall-scale	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
E2006-train	1.00	0.99	0.99	0.99	0.98	0.99	0.99	0.99	0.99	0.99
eunite2001	1.00	1.00	1.00	1.01	1.00	1.00	1.00	1.00	1.00	1.00
housing	1.00	0.98	0.99	1.00	0.99	1.04	1.04	1.04	1.04	1.04
housing-scale	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
log1p-E2006-train	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
mg	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
mg-scale	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
mpg	1.00	1.00	1.00	1.01	1.00	1.00	1.00	1.00	1.00	1.00
mpg-scale	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
pyrim	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
pyrim-scale	1.00	1.03	1.02	1.04	1.03	1.07	1.03	1.03	1.03	1.03
space-ga	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
space-ga-scale	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
triazines	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
triazines-scale	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
YearPredictionMSD	1.00	1.00	1.00	1.01	1.01	1.03	1.03	1.03	1.03	1.03

(b) Running time

Data set	(C, ϵ)					(ϵ, C)				
	20	40	60	80	100	20	40	60	80	100
abalone	1.00	1.47	1.86	2.26	2.67	0.22	0.43	0.64	0.86	1.07
abalone-scale	1.00	1.37	1.76	2.20	2.58	0.22	0.45	0.66	0.89	1.11
bodyfat	1.00	1.26	1.62	1.83	2.09	0.19	0.37	0.56	0.75	0.93
bodyfat-scale	1.00	1.52	1.87	2.08	2.33	0.17	0.34	0.50	0.67	0.84
cadata	1.00	1.85	2.02	2.26	2.41	0.08	0.17	0.25	0.34	0.42
cpusmall	1.00	1.31	1.52	1.77	1.87	0.10	0.21	0.31	0.41	0.51
cpusmall-scale	1.00	1.48	1.82	2.05	2.21	0.14	0.28	0.42	0.57	0.71
E2006-train	1.00	1.76	2.51	3.08	3.79	0.41	0.82	1.24	1.65	2.07
eunite2001	1.00	1.48	1.68	1.88	2.02	0.17	0.35	0.52	0.69	0.86
housing	1.00	1.45	1.75	2.01	2.20	0.10	0.21	0.31	0.41	0.52
housing-scale	1.00	1.66	2.19	2.77	3.22	0.18	0.36	0.54	0.72	0.90
log1p-E2006-train	1.00	1.38	1.48	1.86	2.21	0.12	0.24	0.35	0.47	0.59
mg	1.00	1.53	1.92	2.31	2.66	0.21	0.42	0.63	0.84	1.04
mg-scale	1.00	1.67	2.26	2.84	3.42	0.21	0.43	0.64	0.86	1.07
mpg	1.00	1.46	1.73	1.99	2.16	0.15	0.31	0.46	0.61	0.77
mpg-scale	1.00	1.50	1.94	2.38	2.77	0.19	0.37	0.56	0.74	0.92
pyrim	1.00	1.52	1.75	2.02	2.22	0.08	0.17	0.25	0.33	0.41
pyrim-scale	1.00	1.57	2.02	2.47	2.73	0.12	0.23	0.34	0.46	0.57
space-ga	1.00	1.58	2.20	2.60	3.09	0.12	0.23	0.34	0.46	0.57
space-ga-scale	1.00	1.66	2.22	2.78	3.31	0.21	0.41	0.62	0.83	1.03
triazines	1.00	1.40	1.62	1.76	1.79	0.07	0.14	0.21	0.27	0.34
triazines-scale	1.00	1.48	1.84	1.99	2.14	0.08	0.16	0.24	0.32	0.39
YearPredictionMSD	1.00	1.16	1.27	1.33	1.41	0.11	0.22	0.33	0.43	0.54

than 1 indicates that if the approach can rightly stop at that point, then it is more efficient than the proposed Algorithm 1. From Tables H and I, we can see that the situation significantly varies. For example, in Table H for problem mg-scale in 1% of the running time by Algorithm 1, SA achieves the same CV MSE. However, for problem pyrim-scale, SA needs four folds of running time to achieve the same CV MSE by Algorithm 1. Further, in some situations CV MSE by SA or PSO is always bigger than that by the proposed Algorithm 1.

In the above discussion, we somewhat assume that SA or PSO can terminate right after achieving a satisfactory CV MSE, but as noted earlier, deciding when to stop the search procedure is not an easy task.

D. Discussion on Complexity and Running Time

To solve a linear SVR problem under parameter by the Newton method, from [11], [14], the complexity is

$$\#\text{ of Newton iterations} \times \#\text{ of CG steps} \times \mathcal{O}(nl),$$

where n is the feature size and l is the number of instances in the data set. The number of Newton iterations is often small ($\leq$ 100), though the precise number depends on the data,

Table E: The comparison of using $t_{\text{stop}} = 3, 4, 5, 6$ in the termination condition of the C sequence of the (ϵ, C) setting is presented. All values are normalized by the first column.

(a) CV MSE

Data set	t_{stop}				
	No criterion	3	4	5	6
abalone	1.00	1.00	1.00	1.00	1.00
abalone-scale	1.00	1.00	1.00	1.00	1.00
bodyfat	1.00	1.00	1.00	1.00	1.00
bodyfat-scale	1.00	1.00	1.00	1.00	1.00
cadata	1.00	1.04	1.04	1.00	1.00
cpusmall	1.00	1.09	1.09	1.00	1.00
cpusmall-scale	1.00	1.00	1.00	1.00	1.00
E2006-train	1.00	1.01	1.00	1.00	1.00
eunite2001	1.00	1.00	1.00	1.00	1.00
housing	1.00	1.00	1.00	1.00	1.00
housing-scale	1.00	1.00	1.00	1.00	1.00
log1p-E2006-train	1.00	1.00	1.00	1.00	1.00
mg	1.00	1.00	1.00	1.00	1.00
mg-scale	1.00	1.00	1.00	1.00	1.00
mpg	1.00	1.03	1.00	1.00	1.00
mpg-scale	1.00	1.00	1.00	1.00	1.00
pyrim	1.00	1.00	1.00	1.00	1.00
pyrim-scale	1.00	1.00	1.00	1.00	1.00
space-ga	1.00	1.00	1.00	1.00	1.00
space-ga-scale	1.00	1.00	1.00	1.00	1.00
triazines	1.00	1.00	1.00	1.00	1.00
triazines-scale	1.00	1.00	1.00	1.00	1.00
YearPredictionMSD	1.00	1.05	1.00	1.00	1.00

(b) Running time

Data set	t_{stop}				
	No criterion	3	4	5	6
abalone	1.00	1.00	1.00	1.00	1.00
abalone-scale	1.00	1.00	1.00	1.00	1.00
bodyfat	1.00	1.00	1.00	1.00	1.00
bodyfat-scale	1.00	1.00	1.00	1.00	1.00
cadata	1.00	0.99	1.00	1.00	1.00
cpusmall	1.00	1.00	1.00	1.00	1.00
cpusmall-scale	1.00	1.00	1.00	1.00	1.00
E2006-train	1.00	1.00	1.00	1.00	1.00
eunite2001	1.00	1.00	1.00	1.00	1.00
housing	1.00	0.99	1.00	1.00	1.00
housing-scale	1.00	1.00	1.00	1.00	1.00
log1p-E2006-train	1.00	1.00	1.00	1.00	1.00
mg	1.00	1.00	1.00	1.00	1.00
mg-scale	1.00	1.00	1.00	1.00	1.00
mpg	1.00	1.00	1.00	1.00	1.00
mpg-scale	1.00	1.00	1.00	1.00	1.00
pyrim	1.00	0.99	0.99	0.99	1.00
pyrim-scale	1.00	1.00	1.00	1.00	1.00
space-ga	1.00	1.00	1.00	1.00	1.00
space-ga-scale	1.00	1.00	1.00	1.00	1.00
triazines	1.00	1.00	1.00	1.00	1.00
triazines-scale	1.00	1.00	1.00	1.00	1.00
YearPredictionMSD	1.00	0.97	1.00	1.00	1.00

Table F: The best MSE by using different τ values in the stopping condition (3) for training each SVR. Warm start is not applied so all SVR problems are run independently. All values are normalized by the first column.

	τ values			
Data set	10^{-3}	10^{-4}	10^{-5}	10^{-6}
bodyfat	1.00	1.00	1.18	1.18
pyrim	1.00	1.02	1.03	1.03

the parameters, and the stopping condition. See, for example, the analysis in Chapter 9 of [1]. Regarding the number of CG steps, a theoretical upper bound is n, though under most practically used inner CG stopping conditions, the number is in general no more than 50. We use the table J to illustrate that in the "full and independent" approach, the running time for SVR problems under different parameters can significantly vary. In table J we show the following ratio:

$$\frac{\text{running time under a } (\epsilon, C)}{\text{smallest running time of all parameters}}.$$

Algorithm 2 Simulated annealing approach.

Given
  K as number of CV folds.
  τ as stopping tolerance in (3).
  $\epsilon_{\min} = 0$, $\epsilon_{\max} = \max_i |y_i|$.
  $\delta_1 \in (0, 1)$ as the parameter to calculate $C_{\min}$.
  $C_{\max}$ = a large constant.
  T_0 = a large constant, $T = T_0$.
  Select a random initial solution $\boldsymbol{z}^0 \in [0, 1] \times [0, 1]$.
  Set max iteration $I = 300$.
  $D = 2$.
  demon $= 1.0/0.9^{D/2.0} - 1.0$.
  chi2 = the 99 percentile point of χ^2 distribution with D degree of freedom.
End Given
Evaluate K-fold CV MSE^0 at $\boldsymbol{z}^0$.
MSE$_{\text{best}}$ = MSE^0.
$i \leftarrow 1$.
while $i < I$ **do**
  Select a random direction.
  Select a random point $\boldsymbol{z}^i \in [0, 1] \times [0, 1]$ on the direction.
  Evaluate K-folds CV MSE^i at $\Theta(\boldsymbol{z}^i)$.
  $\Delta E \leftarrow$ MSE^0 − MSE^i.
  if $\Delta E < 0$ **then**
    accp $= e^{\Delta E/T}$.
  else
    accp $= 1$.
  end if
  Select a random value μ from the uniform distribution in $(0, 1)$.
  if $\mu \leq$ accp **then**
    $\boldsymbol{z}^0 = \boldsymbol{z}^i$.
    MSE^0 = MSE^i.
  end if
  if MSE^0 < MSE$_{\text{best}}$ **then**
    MSE$_{\text{best}}$ = MSE^0.
    $\boldsymbol{z}_{\text{best}} = \boldsymbol{z}^0$.
    $T = 2 \times \frac{\text{MSE}_{\text{best}} - \text{MSE}^0}{\text{demon} \times \text{chi2}}$.
  end if
  $i \leftarrow i + 1$.
end while
Return $\boldsymbol{z}_{\text{best}}$.

Algorithm 3 Standard particle swarm optimization approach.

Given
  K as number of CV folds.
  τ as stopping tolerance in (3).
  $\epsilon_{\min} = 0$, $\epsilon_{\max} = \max_i |y_i|$.
  $\delta_1 \in (0, 1)$ as the parameter to calculate $C_{\min}$.
  $C_{\max}$ = a large constant.
  Choose $N = 40$ as the swarm size.
  Choose max iteration $I = 10$.
  Choose neighborhood size $b = 3$.
  Choose velocity update rate $\omega = \frac{1}{2\ln(2)}$ and $c = \frac{1}{2} + \ln(2)$.
End Given
$\min_0 = 0$, $\max_0 = 1$, $\min_1 = 0$, $\max_1 = 1$
for $i = 1, \cdots, N$ **do**
  Initialize particle i's position $\boldsymbol{z}^i$ uniformly from $[\min_0, \max_0] \times [\min_1, \max_1]$.
  Initialize particle i's velocity $\boldsymbol{v}^i$ uniformly from $[\min_0 - z_0^i, \max_0 - z_0^i] \times [\min_1 - z_1^i, \max_1 - z_1^i]$.
  Evaluate particle performance MSE^i with $\Theta(\boldsymbol{z}^i)$.
  Initialize particle i's previous best position $\boldsymbol{p}^i = \boldsymbol{z}^i$ and local best position $\boldsymbol{l}^i = \boldsymbol{z}^i$.
  if MSE^i < MSE$_{\text{best}}$ **then**
    MSE$_{\text{best}}$ = MSE^i.
    $\boldsymbol{z}_{\text{best}} = \boldsymbol{z}^i$.
  end if
  MSE$_{\text{best}}^{i,\text{pre}} \leftarrow$ MSE^i.
  MSE$_{\text{best}}^{i,\text{loc}} \leftarrow$ MSE^i.
  for $l = 1, \cdots, b$ **do**
    Select a random particle with index j (accept repeat selection)
    if MSE^i < MSE^j **then**
      $\boldsymbol{l}^j \leftarrow \boldsymbol{z}^i$
      MSE$_{\text{best}}^{j,\text{loc}} \leftarrow$ MSE^i
    end if
  end for
end for
$t = 0$
while $t < I$ **do**
  for $i = 1, \cdots, N$ **do**
    if $\boldsymbol{l}^i \neq \boldsymbol{p}^i$ **then**
      $\boldsymbol{g}^i \leftarrow \boldsymbol{z}^i + c\frac{\boldsymbol{p}^i + \boldsymbol{l}^i - 2\boldsymbol{z}^i}{3}$.
    else
      $\boldsymbol{g}^i \leftarrow \boldsymbol{z}^i + c\frac{\boldsymbol{p}^i - \boldsymbol{z}^i}{2}$.
    end if
    Select a random point $\boldsymbol{z}'$ uniformly from sphere with center $\boldsymbol{z}^i$ and radius $\|\boldsymbol{z}^i - \boldsymbol{g}^i\|$.
    Update velocity $\boldsymbol{v}^i \leftarrow \omega\boldsymbol{v}^i + (\boldsymbol{z}' - \boldsymbol{z}^i)$.
    Update position $\boldsymbol{z}^i \leftarrow \boldsymbol{z}^i + \boldsymbol{v}^i$.

Note that we obtain the above ratio by using the total number of CG steps in solving an SVR problem because we have indicated that the running time is roughly proportional to it.

Based on the above discussion about the complexity of solving each individual SVR problem, for parameter selection approaches considered in the paper, we can roughly summarize their complexity as follows.

- Full and independent:

$$\#\text{ of } \epsilon \text{ values in the outer loop} \times \tag{VI.6}$$
$$\#\text{ of } C \text{ values in the inner loop} \times \tag{VI.7}$$
$$\#\text{ of average Newton iterations} \times$$
$$\#\text{ of average CG steps} \times \mathcal{O}(nl).$$

if $\boldsymbol{z}^i \notin [\max_0, \max_0] \times [\min_1, \max_1]$ **then**
  for $d = 1, 2$ **do**
    if $z_d^i > \max_d$ **then**
      $z_d^i \leftarrow \max_d$
      $v_d^i \leftarrow -0.5 v_d^i$
    else if $z_d^i < \min_d$ **then**
      $z_d^i \leftarrow \min_d$
      $v_d^i \leftarrow -0.5 v_d^i$
    end if
  end for
end if
Evaluate MSE^i with $\Theta(\boldsymbol{z}^i)$.
if $\text{MSE}^i < \text{MSE}_{\text{best}}$ **then**
  $\text{MSE}_{\text{best}} = \text{MSE}^i$.
  $\boldsymbol{z}_{\text{best}} = \boldsymbol{z}^i$.
end if
if $\text{MSE}^i < \text{MSE}_{\text{best}}^{i,\text{pre}}$ **then**
  $\boldsymbol{p}^i \leftarrow \boldsymbol{z}^i$.
  $\text{MSE}_{\text{best}}^{i,\text{pre}} \leftarrow \text{MSE}^i$.
end if
for $l = 1, \cdots, b$ **do**
  Select a random particle with index j
  if $\text{MSE}^i < \text{MSE}^j$ **then**
    $\boldsymbol{l}^j \leftarrow \boldsymbol{z}^i$
    $\text{MSE}_{\text{best}}^{j,\text{loc}} \leftarrow \text{MSE}^i$
  end if
end for
$t \leftarrow t + 1$.
end for
end while
Return $\boldsymbol{z}_{\text{best}}$.

Note that in our grid setting, the number of ϵ values in the outer loop is a constant 10, while the number of C values in the inner loop is another constant $log_2(C_{\text{max}}) - log_2(C_{\text{min}})$. The C_{min} depends on each data set.

- Our proposed warm-start setting:

  # of ϵ values in the outer loop$\times$
  average # of C values in the inner loop$\times$
  # of average Newton iterations
  $\times$ # of average CG steps $\times\ \mathcal{O}(nl)$.

  The main difference from (VI.6) is that the following two values are reduced:

  $$\text{\# of } C \text{ values and \# of Newton iterations,} \qquad \text{(VI.8)}$$

  where the former comes from the termination condition (6), while the latter is from the warm-start strategy. Note that by using a better initial solution, the number of Newton iterations in solving a single SVR problem can be significantly reduced.
- Partial swarm approach:

  # of epochs $\times$ swarm size $\times$ # of average Newton iterations
  $\times$ # of average CG steps $\times\ \mathcal{O}(nl)$,

  where the swarm size is 40.

Table G: An MSE comparison with the baseline setting of running the full grid without warm start; see the ratio defined in (11). We compare simulated annealing (SA), particle swarm optimization (PSO) and our method. Ratios larger than one are boldfaced.

	SA	PSO	Criterion in (7)
abalone	**1.01**	**1.01**	1
abalone-scale	1	1	1
bodyfat	1	1	**1.18**
bodyfat-scale	1	1	1
cadata	0.99	0.99	**1.09**
cpusmall	1	1	1
cpusmall-scale	1	1	1
E2006.train	1	1	0.99
eunite2001	1	**1.01**	1
housing	0.99	1	**1.04**
housing-scale	1	1	1
log1p.E2006.train	**1.11**	**1.11**	1
mg	1	1	1
mg-scale	1	1	1
mpg	0.99	0.99	0.99
mpg-scale	1	1	1
pyrim	**1.59**	**1.57**	1
pyrim-scale	0.95	0.96	1
space-ga	1	1	1
space-ga-scale	1	1	1
triazines	**1.09**	**1.10**	1
triazines-scale	**1.23**	**1.23**	1
YearPredictionMSD	0.99	0.99	**1.02**

- Simulated annealing:

  # of epochs $\times$ # of average Newton iterations
  $\times$ # of average CG steps $\times\ \mathcal{O}(nl)$.

The above summary gives a good guideline on the computational cost, but does not accurately reflect the practical running time. Therefore, in the main paper as well as the supplementary materials we show direct time comparisons in various places. We check the running time reduction of the warm start technique in Table II in the main paper. Also, we have shown Tables H and I in supplementary materials to demonstrate the overall time cost comparison between our method, the particle swarm approach and the simulated annealing approach.

VII. Summary and Future Issues

We begin with summarizing technical insights gained in this work.

- Upper and lower bounds of C and ϵ except C_{max} are derived.
- With two parameters, we thoroughly investigate which one should be in the outer level of the search procedure, while the other is in the inner.
- We investigate conditions for terminating the parameter search. Results show that a setting proposed in [4] can be extended here.
- We compare the proposed method with two alternating approaches for parameter selection and demonstrate the robustness of our method.

Regarding future research, we plan to study the following issues.

Table H: CV MSE and cumulative running time of SA in comparison with the final results of Algorithm 1. We present ratios along SA's iterations; see (VI.5) and (VI.4). The first iteration achieving the ratio of CV MSE ≤ 1 is boldfaced.

iteration		30	60	90	120	150	180	210	240	270	300
abalone	CV MSE	1.41	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01
abalone	time	0.02	0.33	0.61	0.90	1.17	1.47	1.74	1.99	2.26	2.50
abalone-scale	CV MSE	1.43	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
abalone-scale	time	0.02	**0.29**	0.53	0.79	1.05	1.31	1.55	1.79	2.06	2.30
bodyfat	CV MSE	9.08	**0.85**	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
bodyfat	time	0.01	**0.24**	0.44	0.65	0.85	1.06	1.25	1.44	1.64	1.82
bodyfat-scale	CV MSE	1.71	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
bodyfat-scale	time	0.03	**0.32**	0.57	0.84	1.08	1.35	1.59	1.85	2.13	2.37
cadata	CV MSE	**1.00**	0.92	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91
cadata	time	**0.03**	0.40	0.76	1.16	1.54	1.93	2.29	2.68	3.07	3.45
cpusmall	CV MSE	1.14	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
cpusmall	time	0.02	**0.39**	0.75	1.15	1.66	2.10	2.57	3.01	3.44	3.89
cpusmall-scale	CV MSE	1.48	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
cpusmall-scale	time	0.02	**0.36**	0.69	1.04	1.41	1.76	2.09	2.43	2.76	3.09
E2006.train	CV MSE	1.59	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
E2006.train	time	0.01	**0.12**	0.21	0.29	0.37	0.44	0.51	0.58	0.66	0.73
eunite2001	CV MSE	1.34	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01
eunite2001	time	0.01	0.35	0.65	0.99	1.37	1.72	2.07	2.41	2.73	3.06
housing	CV MSE	1.25	**0.96**	0.95	0.95	0.95	0.95	0.95	0.95	0.95	0.95
housing	time	0.03	**0.52**	1.02	1.60	2.17	2.74	3.28	3.80	4.36	4.91
housing-scale	CV MSE	1.19	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
housing-scale	time	0.02	**0.27**	0.52	0.77	1.02	1.26	1.50	1.73	2.00	2.24
log1p.E2006.train	CV MSE	1.84	1.12	1.11	1.11	1.11	1.11	1.11	1.11	1.11	1.11
log1p.E2006.train	time	0.02	1.09	2.26	3.27	4.58	5.79	6.96	8.20	9.30	10.39
mg	CV MSE	1.04	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
mg	time	0.01	**0.22**	0.41	0.62	0.82	1.02	1.21	1.40	1.59	1.78
mg-scale	CV MSE	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
mg-scale	time	**0.01**	0.14	0.27	0.39	0.51	0.63	0.75	0.87	0.99	1.12
mpg	CV MSE	1.35	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
mpg	time	0.03	**0.32**	0.60	0.92	1.27	1.61	1.94	2.26	2.61	2.94
mpg-scale	CV MSE	1.05	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
mpg-scale	time	0.03	**0.24**	0.45	0.64	0.81	1.02	1.19	1.41	1.60	1.77
pyrim	CV MSE	2.46	1.62	1.61	1.61	1.59	1.59	1.59	1.59	1.59	1.59
pyrim	time	0.07	1.22	2.26	3.09	4.17	4.93	5.81	6.75	7.63	8.48
pyrim-scale	CV MSE	1.09	1.02	1.02	1.02	**1.00**	0.95	0.95	0.95	0.95	0.95
pyrim-scale	time	0.02	1.00	2.13	2.95	**4.05**	4.87	5.79	6.88	7.85	8.70
space-ga	CV MSE	1.04	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
space-ga	time	0.04	**0.35**	0.58	0.81	1.03	1.27	1.48	1.70	1.97	2.17
space-ga-scale	CV MSE	1.15	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
space-ga-scale	time	0.02	**0.27**	0.50	0.73	0.96	1.20	1.41	1.65	1.88	2.09
triazines	CV MSE	1.30	1.30	1.21	1.12	1.09	1.09	1.09	1.09	1.09	1.09
triazines	time	0.09	2.09	3.18	3.80	4.94	5.80	6.90	7.93	8.90	9.73
triazines-scale	CV MSE	1.36	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.22
triazines-scale	time	0.07	0.95	1.60	2.46	3.19	4.00	4.68	5.34	6.03	6.74
YearPredictionMSD	CV MSE	1.30	**0.97**	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97
YearPredictionMSD	time	0.05	**0.88**	1.69	2.52	3.36	4.22	5.04	5.82	6.62	7.38

- From users' experiences in running the proposed procedure, we plan to refine the settings. For example, some constants such as δ_1 in (4), t_{stop} in (7), τ in (7), and $\square$, Δ in (10) may be adjusted. In particular, if feature or target values are extremely large or small, the effectiveness of the proposed procedure might be affected.
- It is important to investigate the relation between C and ϵ. A good understanding may lead us to reduce the search space of parameters.

REFERENCES

[1] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004.

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

[3] 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.

[4] 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 *KDD*, 2015.

[5] 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.

[6] M. Clerc. Standard particle swarm optimisation. 2012.

[7] 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.

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

[9] 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.

[10] 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.

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

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

[13] C.-Y. Hsia, W.-L. Chiang, and C.-J. Lin. Preconditioned conjugate

Table I: CV MSE and cumulative running time of PSO in comparison with the final results of Algorithm 1. We present ratios along PSO's iterations; see (VI.5) and (VI.4). The first iteration achieving the ratio of CV MSE ≤ 1 is boldfaced.

iteration		1	2	3	4	5	6	7	8	9	10
abalone	CV MSE	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01
abalone	time	0.60	0.95	1.35	1.77	2.21	2.60	3.02	3.45	3.86	4.27
abalone-scale	CV MSE	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
abalone-scale	time	**0.64**	0.96	1.38	1.79	2.21	2.59	2.97	3.34	3.74	4.12
bodyfat	CV MSE	**0.85**	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85	0.85
bodyfat	time	**0.52**	0.76	1.03	1.32	1.63	1.89	2.16	2.43	2.73	3.03
bodyfat-scale	CV MSE	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
bodyfat-scale	time	**0.89**	1.35	1.80	2.25	2.71	3.07	3.45	3.82	4.18	4.54
cadata	CV MSE	**0.91**	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91	0.91
cadata	time	**0.91**	1.36	1.85	2.36	2.91	3.38	3.89	4.44	4.92	5.43
cpusmall	CV MSE	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
cpusmall	time	**0.64**	1.09	1.57	2.08	2.61	3.21	3.80	4.38	4.99	5.58
cpusmall-scale	CV MSE	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
cpusmall-scale	time	**0.67**	1.06	1.51	2.01	2.46	2.94	3.41	3.90	4.39	4.88
E2006.train	CV MSE	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	**1.00**
E2006.train	time	0.42	0.58	0.75	0.92	1.10	1.25	1.40	1.55	1.68	**1.80**
eunite2001	CV MSE	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01	1.01
eunite2001	time	0.55	0.89	1.28	1.64	2.09	2.56	2.99	3.40	3.81	4.27
housing	CV MSE	**0.96**	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.96
housing	time	**0.92**	1.48	2.12	2.88	3.60	4.32	5.02	5.83	6.58	7.34
housing-scale	CV MSE	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
housing-scale	time	**0.63**	0.96	1.34	1.76	2.17	2.57	2.96	3.37	3.78	4.14
log1p.E2006.train	CV MSE	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.12	1.11
log1p.E2006.train	time	0.84	1.59	2.48	3.47	4.60	5.62	6.71	7.76	8.79	9.94
mg	CV MSE	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
mg	time	**0.48**	0.72	1.01	1.33	1.62	1.89	2.18	2.49	2.79	3.08
mg-scale	CV MSE	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
mg-scale	time	**0.52**	0.69	0.87	1.04	1.22	1.36	1.52	1.67	1.83	1.99
mpg	CV MSE	1.01	1.01	1.01	1.01	**1.00**	1.00	1.00	1.00	1.00	1.00
mpg	time	0.73	1.15	1.52	1.95	**2.43**	2.86	3.26	3.70	4.12	4.55
mpg-scale	CV MSE	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
mpg-scale	time	**0.78**	1.16	1.43	1.75	2.08	2.35	2.65	2.98	3.22	3.51
pyrim	CV MSE	1.63	1.63	1.63	1.62	1.62	1.62	1.62	1.56	1.56	1.56
pyrim	time	1.51	2.11	3.24	4.16	5.01	5.69	6.69	7.41	8.03	8.97
pyrim-scale	CV MSE	**0.96**	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.96	0.96
pyrim-scale	time	**0.86**	1.24	1.86	2.52	3.19	3.76	4.25	4.86	5.37	5.98
space-ga	CV MSE	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
space-ga	time	**1.09**	1.55	2.00	2.53	3.03	3.32	3.72	4.15	4.58	4.95
space-ga-scale	CV MSE	**1.00**	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
space-ga-scale	time	**0.62**	0.98	1.32	1.75	2.14	2.47	2.87	3.28	3.66	4.00
triazines	CV MSE	1.13	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10	1.10
triazines	time	2.10	2.99	4.35	5.55	6.77	7.62	8.53	9.28	10.14	11.51
triazines-scale	CV MSE	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23	1.23
triazines-scale	time	1.58	2.69	3.70	5.08	6.38	7.76	9.01	10.35	11.60	12.92
YearPredictionMSD	CV MSE	**0.99**	0.98	0.97	0.97	0.97	0.97	0.97	0.97	0.97	0.97
YearPredictionMSD	time	**1.37**	2.19	3.15	4.20	5.32	6.39	7.44	8.57	9.66	10.71

gradient methods in truncated Newton frameworks for large-scale linear classification. In *Proceedings of the Asian Conference on Machine Learning (ACML)*, 2018.

[14] C.-Y. Hsia, Y. Zhu, and C.-J. Lin. A study on trust region update rules in Newton methods for large-scale linear classification. In *Proceedings of the Asian Conference on Machine Learning (ACML)*, 2017.

[15] 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.

[16] K. Jin'no, T. Sasaki, and H. Nakano. Search property of nonlinear map optimization. *IEEE Congress on Evolutionary Computation*, pages 3213–3220, 2019.

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

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

[19] 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.

[20] 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.

[21] 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.

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

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

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

[25] 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.

[26] 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.

[27] 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.

[28] 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.

Table J: The time comparison under different parameter pairs in cadata.

		C												
		2^{-30}	2^{-26}	2^{-22}	2^{-18}	2^{-14}	2^{-10}	2^{-6}	2^{-2}	2^{2}	2^{6}	2^{10}	2^{14}	2^{18}
	0	1.50	2.00	2.83	2.67	2.50	3.00	3.17	3.17	3.17	3.17	3.17	3.17	3.17
	47500.09	1.67	2.17	3.33	4.17	3.17	4.50	4.67	4.67	4.50	4.67	4.50	4.50	4.67
	95000.19	1.50	2.50	3.50	3.67	4.50	6.00	6.50	6.00	6.17	6.17	6.17	6.00	6.00
	142500.28	1.67	2.67	3.67	4.33	5.00	6.33	6.00	6.00	6.33	6.33	6.17	6.17	6.00
ϵ	190000.38	1.67	2.17	3.50	4.17	5.50	5.33	5.50	5.83	5.83	5.83	5.83	5.83	5.83
	237500.47	1.67	2.00	3.00	4.33	3.00	4.67	2.00	2.50	2.50	2.50	2.50	2.50	2.50
	285000.57	1.67	2.00	2.50	3.17	3.50	4.67	2.50	2.50	2.50	2.50	2.50	2.50	2.50
	332500.66	1.50	1.83	2.67	3.00	3.67	4.67	3.17	2.33	2.33	2.33	2.33	2.33	2.33
	380000.76	1.50	2.17	2.50	4.50	3.83	4.67	3.00	2.50	2.33	2.33	2.33	2.33	2.33
	427500.85	1.17	1.83	2.50	7.00	4.17	4.00	3.17	2.50	2.50	2.50	2.50	2.50	2.50
	475000.95	1.00	1.83	3.83	6.83	2.67	4.17	3.00	2.67	2.50	2.50	2.50	2.50	2.50

[29] M. Zambrano-Bigiarini, M. Clerc, and R. Rojas-Mujica. Standard particle swarm optimisation 2011 at cec-2013: A baseline for future pso improvements. In *IEEE Congress on Evolutionary Computation*, pages 2337–2344, 2013.