Neural Networks 67 (2015) 140–150

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

# Incremental learning for $\nu$-Support Vector Regression

Bin Gu [a,b,c,d,*], Victor S. Sheng [e], Zhijie Wang [f], Derek Ho [g], Said Osman [h], Shuo Li [f,d]

[a] Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing, PR China
[b] Jiangsu Collaborative Innovation Center on Atmospheric Environment and Equipment Technology, PR China
[c] School of Computer & Software, Nanjing University of Information Science & Technology, Nanjing, PR China
[d] Department of Medical Biophysics, University of Western Ontario, London, Ontario, Canada
[e] Department of Computer Science, University of Central Arkansas, Conway, AR, USA
[f] GE Health Care, London, Ontario, Canada
[g] Victoria Hospital, London Health Science Center, London, Ontario, Canada
[h] St. Joseph's Health Care, London, Ontario, Canada

ARTICLE INFO

*Article history:*
Received 19 May 2014
Received in revised form 28 December 2014
Accepted 20 March 2015
Available online 6 April 2015

*Keywords:*
Incremental learning
Online learning
$\nu$-Support Vector Regression
Support vector machine

ABSTRACT

The $\nu$-Support Vector Regression ($\nu$-SVR) is an effective regression learning algorithm, which has the advantage of using a parameter $\nu$ on controlling the number of support vectors and adjusting the width of the tube automatically. However, compared to $\nu$-Support Vector Classification ($\nu$-SVC) (Schölkopf et al., 2000), $\nu$-SVR introduces an additional linear term into its objective function. Thus, directly applying the accurate on-line $\nu$-SVC algorithm (AONSVM) to $\nu$-SVR will not generate an effective initial solution. It is the main challenge to design an incremental $\nu$-SVR learning algorithm. To overcome this challenge, we propose a special procedure called *initial adjustments* in this paper. This procedure adjusts the weights of $\nu$-SVC based on the Karush–Kuhn–Tucker (KKT) conditions to prepare an initial solution for the incremental learning. Combining the *initial adjustments* with the two steps of AONSVM produces an exact and effective incremental $\nu$-SVR learning algorithm (INSVR). Theoretical analysis has proven the existence of the three key inverse matrices, which are the cornerstones of the three steps of INSVR (including the *initial adjustments*), respectively. The experiments on benchmark datasets demonstrate that INSVR can avoid the infeasible updating paths as far as possible, and successfully converges to the optimal solution. The results also show that INSVR is faster than batch $\nu$-SVR algorithms with both cold and warm starts.



## 1. Introduction

In real-world regression tasks, such as time-series prediction (e.g. Cao and Tay (2003); Lu, Lee, and Chiu (2009)), training data is usually provided sequentially, in the extreme case, one example at a time, which is an online scenario (Murata, 1998). Batch algorithms seems computationally wasteful as they retrain a learning model from scratch. Incremental learning algorithms are more capable in this case, because the advantage of the incremental learning algorithms is that they incorporate additional training data without re-training the learning model from scratch (Laskov et al., 2006).

$\nu$-Support Vector Regression ($\nu$-SVR) (Schölkopf, Smola, Williamson, & Bartlett, 2000) is an interesting Support Vector Regression (SVR) algorithm, which can automatically adjust the parameter $\epsilon$ of the $\epsilon$-insensitive loss function.[1] Given a training sample set $T = \{(x_1, y_1), \ldots, (x_l, y_l)\}$ with $x_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$,

* Corresponding author at: Jiangsu Engineering Center of Network Monitoring, Nanjing University of Information Science & Technology, Nanjing, PR China, and School of Computer and Software, Nanjing University of Information Science and Technology, Nanjing, PR China.
*E-mail addresses:* jsgubin@nuist.edu.cn (B. Gu), ssheng@uca.edu (V.S. Sheng), zhijie@ualberta.ca (Z. Wang), derek.ho@lhsc.on.ca (D. Ho), sidosman@hotmail.com (S. Osman), Shuo.Li@ge.com (S. Li).

[1] The $\epsilon$-insensitive loss function used in SVR is defined as $|y - f(x)|_\epsilon = \max\{0, |y - f(x)| - \epsilon\}$ for a predicted value $f(x)$ and a true output $y$, which does not penalize errors below some $\epsilon > 0$, chose a priori. Thus, the region of all $(x, y)$ with $|\{y - f(x)| \leq \epsilon\}$ is called $\epsilon$-tube (see Fig. 1).

http://dx.doi.org/10.1016/j.neunet.2015.03.013
0893-6080/

Schölkopf et al. (2000) considered the following primal problem:

$$\begin{aligned} \min_{w,\epsilon,b,\xi_i^{(*)}} \quad & \frac{1}{2}\langle w, w\rangle + C \cdot \left( \nu\epsilon + \frac{1}{l}\sum_{i=1}^{l}\left(\xi_i + \xi_i^*\right)\right) \\ s.t. \quad & (\langle w, \phi(x_i)\rangle + b) - y_i \le \epsilon + \xi_i, \\ & y_i - (\langle w, \phi(x_i)\rangle + b) \le \epsilon + \xi_i^*, \\ & \xi_i^{(*)} \ge 0, \qquad \epsilon \ge 0,\ i = 1, \dots, l. \end{aligned} \tag{1}$$

The corresponding dual is:

$$\begin{aligned} \min_{\alpha,\alpha^*} \quad & \frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)K(x_i, x_j) - \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)y_i \\ s.t. \quad & \sum_{i=1}^{l}\left(\alpha_i^* - \alpha_i\right) = 0, \qquad \sum_{i=1}^{l}\left(\alpha_i^* + \alpha_i\right) \le C\nu, \\ & 0 \le \alpha_i^{(*)} \le \frac{C}{l},\ i = 1, \dots, l \end{aligned} \tag{2}$$

where, following Schölkopf et al. (2000), training samples $x_i$ are mapped into a high dimensional reproducing kernel Hilbert space (RKHS) (Schölkopf & Smola, 2001) by the transformation function $\phi$. $K(x_i, x_j) = \langle\phi(x_i), \phi(x_j)\rangle$, $\langle\cdot,\cdot\rangle$ denotes inner product in RKHS. $^{(*)}$ is a shorthand implying both the variables with and without asterisks. $C$ is the regularization constant, and $\nu$ is the introduced proportion parameter with $0 \le \nu \le 1$, which lets one control the number of support vectors and errors. To be more precise, they proved that $\nu$ is an upper bound on the fraction of margin errors, and a lower bound on the fraction of support vectors. In addition, with probability 1, asymptotically, $\nu$ equals both fractions.

Compared with $\epsilon$-Support Vector Regression ($\epsilon$-SVR) (Smola & Schölkopf, 2003), $\nu$-SVR introduces two complications: the first one is that the box constraints are related to the size of the training sample set, and the second one is that one more inequality constraint is introduced in the formulation. Compared with $\nu$-Support Vector Classification ($\nu$-SVC) (Schölkopf et al., 2000), $\nu$-SVR introduces an additional linear term into the objective function of (2). To sum up, the formulation of $\nu$-SVR is more complicated than the formulations of $\epsilon$-SVR and $\nu$-SVC.

Early studies about SVR mostly focus on solving large-scale problems. For example, Chang and Lin (2001, 2002) gave SMO algorithm and implementation for training $\epsilon$-SVR. Tsang, Kwok, and Zurada (2006) proposed core vector regression for training very large regression problems. Shalev-Shwartz, Singer, Srebro, and Cotter (2011) proposed stochastic sub-gradient descent algorithm with explicit feature mapping for training $\epsilon$-SVR. Ho and Lin (2012) and Wang and Lin (2014) proposed coordinate descent algorithm for linear L1 and L2 SVR. Due to the complications in the formulation of $\nu$-SVR as mentioned above, there are still no effective methods proposed for solving incremental $\nu$-SVR learning.

Let us pay our attention to the exact incremental and decremental SVM algorithm (Cauwenberghs & Poggio, 2001) (hereinafter referred to as the C&P algorithm). Since the C&P algorithm was proposed by Cauwenberghs and Poggio in 2001, further studies mainly focus on two aspects. One is focusing on the C&P algorithm itself. For example, Gu, Wang, and Chen (2008) and Laskov et al. (2006) provided more detailed theoretical analysis for it. Gâlmeanu and Andonie (2008) addressed some implementation issues. Karasuyama and Takeuchi (2010) proposed an extension version which can update multiple samples simultaneously. The other applies the C&P algorithm to solve other problems. For example, Gretton and Desobry (2003) and Laskov et al. (2006) applied it to implementing an incremental one-class SVM algorithm. Martin (2002) and Ma, Theiler, and Perkins (2003) introduced it to $\epsilon$-SVR (Vapnik, 1998) and developed an accurate on-line support vector regression (AOSVR). Recently, Gu et al. (2012) introduced the C&P algorithm to $\nu$-SVC and proposed an effective accurate on-line $\nu$-SVC algorithm (AONSVM), which includes the *relaxed adiabatic incremental adjustments* and the *strict restoration adjustments*. Further, Gu and Sheng (2013) proved the feasibility and finite convergence of AONSVM. Because great resemblance exists in $\nu$-SVR and $\nu$-SVC, in this paper, we wish to design an exact and effective incremental $\nu$-SVR algorithm based on AONSVM.

As $\nu$-SVR has an additional linear term in the objective function compared with $\nu$-SVC, directly applying AONSVM to $\nu$-SVR will not generate an effective initial solution for the incremental $\nu$-SVR learning. To address this issue, we propose a new incremental $\nu$-SVR algorithm (collectively called INSVR) based on AONSVM. In addition to the basic steps of AONSVM (i.e., the relaxed adiabatic incremental adjustments and the strict restoration adjustments), INSVR has an especial adjusting process (i.e. *initial adjustments*), which is used to address the complications of the $\nu$-SVR formulation and to prepare the initial solution before the incremental learning. Through theoretical analysis, we can show the existence of the three key inverse matrices, which are the cornerstone of the *initial adjustments*, the *relaxed adiabatic incremental adjustments*, and the *strict restoration adjustments*, respectively. The experiments on benchmark datasets demonstrate that INSVR can avoid the infeasible updating path as far as possible, and successfully converge to the optimal solution. The results also show that INSVR is faster than batch $\nu$-SVR algorithms with both cold and warm starts.

The rest of this paper is organized as follows. In Section 2, we modify the formulation of $\nu$-SVR and give its KKT conditions. The INSVR algorithm is presented in Section 3. The experimental setup, results and discussions are presented in Section 4. The last section gives some concluding remarks.

*Notation*: To make the notations easier to follow, we give a summary of the notations in the following list.

| | |
|---|---|
| $\alpha_i, g_i$ | The $i$th element of the vector $\alpha$ and $g$. |
| $\alpha_c, y_c, z_c$ | The weight, output, and label of the candidate extended sample $(x_c, y_c, z_c)$. |
| $\Delta$ | The amount of the change of each variable. |
| $\boxed{\epsilon'}, \boxed{\Delta\epsilon'}$ | If $\lvert\sum_{i\in S_S} z_i\rvert = \lvert S_S\rvert$, $\boxed{\epsilon'}$ and $\boxed{\Delta\epsilon'}$ stands for $\epsilon'$ and $\Delta\epsilon'$, respectively. Otherwise, they will be ignored. |
| $Q_{S_S S_S}$ | The submatrix of $Q$ with the rows and columns indexed by $S_S$. |
| $\widetilde{Q}_{\setminus \widetilde{M}^2}$ | The submatrix of $\widetilde{Q}$ with deleting the rows and columns indexed by $\widetilde{M}$. |
| $\check{R}_{t*}, \check{R}_{*t}$ | The row and the column of a matrix $\check{R}$ corresponding to the sample $(x_t, y_t, z_t)$, respectively. |
| $\mathbf{0}, \mathbf{1}$ | The vectors having all the elements equal to 0 and 1, respectively, with proper dimension. |
| $\mathbf{z}_{S_S}, \mathbf{u}_{S_S}$ | A $\lvert S_S\rvert$-dimensional column vector with all equal to $z_i$, and $z_i y_i$ respectively. |
| $\det(\cdot)$ | The determinant of a square matrix. |
| $\mathrm{cols}(\cdot)$ | The number of columns of a matrix. |
| $\mathrm{rank}(\cdot)$ | The rank of a matrix. |

## 2. Modified formulation of $\nu$-SVR

Obviously, the correlation between the box constraints and the size of the training sample set makes it difficult to design an incremental $\nu$-SVR learning algorithm. To obtain an equivalent formulation, whose box constraints are independent to the size of the training sample set, we multiply the objective function of (1) by the size of the training sample set. Thus, we consider the following primal problem:

$$\begin{aligned}
\min_{w,\epsilon,b,\xi_i^{(*)}} \quad & \frac{l}{2}\langle w, w\rangle + C\cdot\left(\nu\epsilon l + \sum_{i=1}^{l}\left(\xi_i+\xi_i^*\right)\right) \\
s.t. \quad & (\langle w, \phi(x_i)\rangle + b) - y_i \le \epsilon + \xi_i, \qquad (3) \\
& y_i - (\langle w, \phi(x_i)\rangle + b) \le \epsilon + \xi_i^*, \\
& \xi_i^{(*)} \ge 0, \qquad \epsilon \ge 0,\ i = 1, \ldots, l.
\end{aligned}$$

It is easy to verify that the primal problem (3) is equivalent to the primal problem (1), and $\nu$ is also an upper bound on the fraction of margin errors and a lower bound of the fraction of support vectors. The dual problem of (3) is:

$$\begin{aligned}
\min_{\alpha,\alpha^*} \quad & \frac{1}{2l}\sum_{i,j=1}^{l}(\alpha_i^*-\alpha_i)(\alpha_j^*-\alpha_j)K(x_i,x_j) - \sum_{i=1}^{l}(\alpha_i^*-\alpha_i)y_i \qquad (4) \\
s.t. \quad & \sum_{i=1}^{l}\left(\alpha_i^*-\alpha_i\right) = 0, \qquad \sum_{i=1}^{l}\left(\alpha_i^*+\alpha_i\right) \le C\nu l, \\
& 0 \le \alpha_i^{(*)} \le C,\ i = 1, \ldots, l.
\end{aligned}$$

Furthermore, we introduce Theorem 1, which concludes that for any given $\nu$ in (4), there are always optimal solutions which happen at the equality $\sum_{i=1}^{l}\left(\alpha_i^*+\alpha_i\right) = C\nu l$. The original version of this theorem is proved in Chang and Lin (2002).

**Theorem 1** (*Chang & Lin, 2002*). *For dual problem (4), $0 \le \nu \le 1$, there are always optimal solutions which happen at $\sum_{i=1}^{l}\left(\alpha_i^*+\alpha_i\right) = C\nu l$.*

According to Theorem 1, the inequality constraint $\sum_{i=1}^{l}(\alpha_i^* + \alpha_i) \le C\nu l$ in (4) can be treated as the equality $\sum_{i=1}^{l}\left(\alpha_i^*+\alpha_i\right) = C\nu l$, which means that we can consider the following minimization problem instead of the dual (4):

$$\begin{aligned}
\min_{\alpha,\alpha^*} \quad & \frac{1}{2l}\sum_{i,j=1}^{l}(\alpha_i^*-\alpha_i)(\alpha_j^*-\alpha_j)K(x_i,x_j) - \sum_{i=1}^{l}(\alpha_i^*-\alpha_i)y_i \qquad (5) \\
s.t. \quad & \sum_{i=1}^{l}\left(\alpha_i^*-\alpha_i\right) = 0, \qquad \sum_{i=1}^{l}\left(\alpha_i^*+\alpha_i\right) = C\nu l, \\
& 0 \le \alpha_i^{(*)} \le C,\ i = 1, \ldots, l.
\end{aligned}$$

Furthermore, to present the minimization problem in a more compact form, we introduce the extended training sample set $S$, which is defined as $S = S^- \cup S^+$, where $S^- = \{(x_i, y_i, z_i = -1)\}_{i=1}^{l}$, $S^+ = \{(x_i, y_i, z_i = +1)\}_{i=1}^{l}$, and $z_i$ is the label of the training sample $(x_i, y_i)$. Thus, the minimization problem (5) can be further rewritten as:

$$\begin{aligned}
\min_{\alpha} \quad & \frac{1}{2}\sum_{i,j=1}^{2l}\alpha_i\alpha_j Q_{ij} - \sum_{i=1}^{2l} z_i y_i \alpha_i \qquad (6) \\
s.t. \quad & \sum_{i=1}^{2l} z_i\alpha_i = 0, \qquad \sum_{i=1}^{2l}\alpha_i = C\nu l, \\
& 0 \le \alpha_i \le C,\ i = 1, \ldots, 2l
\end{aligned}$$

where $Q$ is a positive semidefinite matrix with $Q_{ij} = \frac{1}{l}z_i z_j K(x_i, x_j)$.

According to convex optimization theory (Boyd & Vandenberghe, 2004), the solution of the minimization problem (6) can also be obtained by minimizing the following convex quadratic objective function under constraints:

$$\begin{aligned}
\min_{0\le\alpha_i\le C} W = {} & \frac{1}{2}\sum_{i,j=1}^{2l}\alpha_i\alpha_j Q_{ij} - \sum_{i=1}^{2l} z_i y_i \alpha_i \\
& + b'\left(\sum_{i=1}^{2l} z_i\alpha_i\right) + \epsilon'\left(\sum_{i=1}^{2l}\alpha_i - C\nu l\right) \qquad (7)
\end{aligned}$$

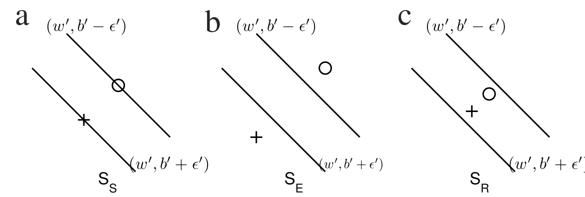


**Fig. 1.** The partition of the training samples $S$ into three independent sets by KKT-conditions. (a) $S_S$. (b) $S_E$. (c) $S_R$.

**Table 1**
Two cases of conflicts between $\sum_{i\in S}\Delta\alpha_i + \Delta\alpha_c = 0$ and $\sum_{i\in S} z_i\Delta\alpha_i + z_c\Delta\alpha_c = 0$ when $\left|\sum_{i\in S_S} z_i\right| = |S_S|$ with a small increment $\Delta\alpha_c$.

| Label of margin support vectors | | Label of the candidate sample | | Conflict |
|---|---|---|---|---|
| +1 | −1 | +1 | −1 | Yes/no |
| √ | | √ | | No |
| √ | | | √ | Yes |
| | √ | √ | | Yes |
| | √ | | √ | No |

where $b'$ and $\epsilon'$ are Lagrangian multipliers.

Then by the KKT theorem (Karush, 1939), the first-order derivative of $W$ leads to the following KKT conditions:

$$\frac{\partial W}{\partial b'} = \sum_{i=1}^{2l} z_i\alpha_i = 0 \qquad (8)$$

$$\frac{\partial W}{\partial \epsilon'} = \sum_{i=1}^{2l} \alpha_i = C\nu l \qquad (9)$$

$$\forall i \in S : g_i = \frac{\partial W}{\partial \alpha_i} = \sum_{j=1}^{2l}\alpha_j Q_{ij} - z_i y_i + z_i b' + \epsilon' \begin{cases} \ge 0 & \text{for } \alpha_i = 0 \\ = 0 & \text{for } 0 < \alpha_i < C \\ \le 0 & \text{for } \alpha_i = C. \end{cases} \qquad (10)$$

According to the value of the function $g_i$, the extended training sample set $S$ is partitioned into three independent sets (see Fig. 1):

(i) $S_S = \{i : g_i = 0,\ 0 < \alpha_i < C\}$, the set $S_S$ includes *margin support vectors* strictly on the $\epsilon$-tube;
(ii) $S_E = \{i : g_i \le 0,\ \alpha_i = C\}$, the set $S_E$ includes *error support vectors* exceeding the $\epsilon$-tube;
(iii) $S_R = \{i : g_i \ge 0,\ \alpha_i = 0\}$, the set $S_R$ includes *the remaining vectors* covered by the $\epsilon$-tube.

## 3. Incremental $\nu$-SVR algorithm

In this section, we focus on the incremental $\nu$-SVR learning algorithm especially for the minimization problem (6). If a new sample $(x_{new}, y_{new})$ is added into the training sample set $T$, there will exist an increment in the extended training sample set $S$, which can be defined as $S_{new} = \{(x_{new}, y_{new}, +1), (x_{new}, y_{new}, -1)\}$. Initially, the weights of the samples in $S_{new}$ are set zero. If this assignment violates the KKT conditions, the adjustments to the weights will become necessary. As stated in Gu et al. (2012), the incremental $\nu$-SVR algorithm is actually to find an effective method for updating the weights without re-training from scratch, when any constraint of the KKT conditions is not held.

The classical C&P algorithm is gradually increasing the weight $\alpha_c$ of the added sample $(x_c, y_c)$, while rigorously ensuring all the samples satisfying the KKT conditions. As proved in Gu and Sheng (2013) and Gu et al. (2008), there exists a feasible updating path achieving the optimal solution for the enlarged training sample set. Unfortunately, Gu et al. (2012) pointed out that this idea cannot hold for $\nu$-SVC and $\nu$-SVR. In this paper, we provide an in-depth analysis for $\nu$-SVR. As listed in Table 1, if $\left|\sum_{i\in S_S} z_i\right| = |S_S|$, and the label of the candidate sample $(x_c, y_c, z_c)$ is different from those of the margin support vectors in $S_S$, there exists a conflict (referred to as Conflict-1) between Eqs. (8) and (9) with a small increment of $\alpha_c$. To address this issue, AONSVM introduced two special steps (i.e., the *relaxed adiabatic incremental adjustments* and the *strict restoration adjustments*). However, directly applying AONSVM to $\nu$-SVR will not generate an efficient initial solution for the incremental $\nu$-SVR learning. Specifically, the objective of the initial solution is to retrieve the optimal solution of the minimization problem (6) when $Q \leftarrow \frac{l}{l+1}Q$. It is obvious that the initial strategy of AONSVM (i.e., setting $g \leftarrow \frac{l}{l+1}g$, $b' \leftarrow \frac{l}{l+1}b'$, $\epsilon' \leftarrow \frac{l}{l+1}\epsilon'$) does not apply to the incremental $\nu$-SVR learning due to an additional linear term in the objective function (2). Thus, a new procedure should be designed especially for tackling this issue.

---

**Algorithm 1** Incremental $\nu$-SVR algorithm INSVR

---

1: Read a new sample $(x_{new}, y_{new})$, and let $S_{new} = \{(x_{new}, y_{new}, +1), (x_{new}, y_{new}, -1)\}$.
2: Update $g \leftarrow \frac{l}{l+1}g$, $b' \leftarrow \frac{l}{l+1}b'$, $\epsilon' \leftarrow \frac{l}{l+1}\epsilon'$, $\eta \leftarrow \frac{l}{l+1}$.
3: **while** $\eta \neq 1$ **do**
4: Compute $\check{\beta}$ and $\check{\gamma}$ according to (16)-(17).
5: Compute the maximal increment $\Delta\eta^{\max}$ according to (19).
6: Update $\eta, \alpha, g, b', \epsilon', S_S, S_E$ and $S_R$.
7: Update the inverse matrix $\check{R}$ according to (20)-(22).
8: **end while**
9: Compute the inverse matrix $R$ based on $\check{R}$.
10: Update $S \leftarrow S \cup S_{new}$.
11: Initial the weights of the samples in $S_{new}$ as 0, and compute their values of the function $g_i$.
12: **if** $\exists (x_c, y_c, z_c) \in S_{new}$ such that $g_c < 0$ **then**
13: Using the *relaxed adiabatic incremental adjustments*.
14: **end if**
15: Using the *strict restoration adjustments*.
16: Compute the inverse matrix $\check{R}$ based on $\widehat{R}$. {See Section 3.1.3.}

---

To address this issue, we propose the *initial adjustments*. The objective of this step is to initial the solution of the minimization problem (6) before adding a new sample $(x_{new}, y_{new})$ into $T$, i.e. retrieving the optimal solution of the minimization problem (6) when $Q \leftarrow \frac{l}{l+1}Q$. Our idea is first setting $g \leftarrow \frac{l}{l+1}g$, $b' \leftarrow \frac{l}{l+1}b'$, $\epsilon' \leftarrow \frac{l}{l+1}\epsilon'$, next imposing a shrinkage $\eta = \frac{l}{l+1}$ on $y_i$, $1 \leq i \leq 2l$, then gradually increasing $\eta$ under the condition of rigorously keeping all samples satisfying the KKT conditions. This is repeated until $\eta = 1$. This procedure is described with pseudo code in lines 2–8 of Algorithm 1, and the details are expounded in Section 3.1.

Before presenting the whole incremental $\nu$-SVR algorithm, we first introduce Theorem 2, which tells us $\epsilon' \geq 0$ after the *initial adjustments*.

**Theorem 2.** *After the initial adjustments, it can be concluded that the Lagrangian multiplier $\epsilon'$ must be greater than or equal to 0.*

The detailed proof to Theorem 2 is proved in Appendix A.1.

Initially, we set the weights of two samples in $S_{new}$ to be zero. According to Theorem 2, it can be concluded that there exists at most one sample $(x_c, y_c, z_c)$ from $S_{new}$ violating the KKT conditions (i.e., $g_c < 0$) after the *initial adjustments*. If existing a sample $(x_c, y_c, z_c)$ in $S_{new}$ with $g_c < 0$, the *relaxed adiabatic incremental adjustments* will be used to make all samples satisfying the KKT condition except the equality restriction (9). Finally, the *strict restoration adjustments* is used to restore the equality restriction (9). The three steps constitute the incremental $\nu$-SVR algorithm INSVR (see Algorithm 1), which can find the optimal solution for the enlarged training sample set without re-training from scratch. To make this paper self-contained, we give a brief review of the *strict restoration adjustments* and the *strict restoration adjustments* in Sections 3.2 and 3.3, respectively. Their details can be found in Gu et al. (2012). Section 3.4 proves the existence of the three key inverse matrices in INSVR.

### 3.1. Initial adjustments

To prepare the initial solution of the minimization problem (6) before adding a new sample $(x_{new}, y_{new})$ into $T$, our strategy is first setting $g \leftarrow \frac{l}{l+1}g$, $b' \leftarrow \frac{l}{l+1}b'$, $\epsilon' \leftarrow \frac{l}{l+1}\epsilon'$, next imposing a shrinkage $\eta = \frac{l}{l+1}$ on $y_i$, $1 \leq i \leq 2l$, then gradually increasing $\eta$ under the condition of rigorously keeping all samples satisfying the KKT conditions, until $\eta = 1$.

During the initial adjustments, in order to keep all the samples satisfying the KKT conditions, we can have the following linear system:

$$\sum_{j\in S_S} z_j \Delta\alpha_j = 0 \tag{11}$$

$$\sum_{j\in S_S} \Delta\alpha_j = 0 \tag{12}$$

$$\Delta g_i = \sum_{j\in S_S} \Delta\alpha_j Q_{ij} + z_i \Delta b' + \Delta\epsilon' - \Delta\eta z_i y_i = 0, \quad \forall i \in S_S \tag{13}$$

where $\Delta\eta$, $\Delta\alpha_j$, $\Delta b'$, $\Delta\epsilon'$ and $\Delta g_i$ denote the corresponding variations.

If we define $\mathbf{e}_{S_S} = [1, \ldots, 1]^T$ as the $|S_S|$-dimensional column vector with all ones, and let $\mathbf{z}_{S_S} = [z_1, \ldots, z_{|S_S|}]^T$, $\mathbf{u}_{S_S} = [z_1 y_1, \ldots, z_{|S_S|} y_{|S_S|}]^T$, then the linear system (11)–(13) can be further rewritten as:

$$\underbrace{\begin{bmatrix} 0 & 0 & \mathbf{z}_{S_S}^T \\ 0 & 0 & \mathbf{1}^T \\ \mathbf{z}_{S_S} & \mathbf{1} & Q_{S_S S_S} \end{bmatrix}}_{\widetilde{Q}} \underbrace{\begin{bmatrix} \Delta b' \\ \Delta\epsilon' \\ \Delta\alpha_{S_S} \end{bmatrix}}_{\Delta h} = \begin{bmatrix} 0 \\ 0 \\ \mathbf{u}_{S_S} \end{bmatrix} \Delta\eta. \tag{14}$$

Supposing $\widetilde{Q}$ has the inverse matrix $\check{R}$ (the detailed discussion about the reversibility of $\widetilde{Q}$ is provided in Section 3.1.1), the linear relationship between $\Delta h$ and $\Delta\eta$ can be easily solved as follows:

$$\Delta h = \begin{bmatrix} \Delta b' \\ \Delta\epsilon' \\ \Delta\alpha_{S_S} \end{bmatrix} = \check{R} \begin{bmatrix} 0 \\ 0 \\ \mathbf{u}_{S_S} \end{bmatrix} \Delta\eta \overset{\text{def}}{=} \begin{bmatrix} \check{\beta}_{b'} \\ \check{\beta}_{\epsilon'} \\ \check{\beta}_{S_S} \end{bmatrix} \Delta\eta. \tag{15}$$

Substituting (15) into (13), we can get the linear relationship between $\Delta g_i$ and $\Delta\eta$ as follows:

$$\Delta g_i = \left( \sum_{j\in S_S} \check{\beta}_j Q_{ij} + z_i \check{\beta}_{b'} + \check{\beta}_{\epsilon'} - z_i y_i \right) \Delta\eta$$
$$\overset{\text{def}}{=} \check{\gamma}_i \Delta\eta, \quad \forall i \in S. \tag{16}$$

And obviously, $\forall i \in S_S$, we have $\check{\gamma}_i = 0$.

#### 3.1.1. Special cases in initial adjustments

Under the condition that $\widetilde{Q}$ has the inverse matrix $\check{R}$, the *initial adjustments* can easily obtain the linear relationships between $\Delta h$ and $\Delta\eta$, and between $\Delta g$ and $\Delta\eta$ according to (15)–(16). In this section, we will discuss how to determine the linear relationships between $\Delta h$ and $\Delta\eta$, and between $\Delta g$ and $\Delta\eta$, when $\widetilde{Q}$ becomes singular.

We will show that $\widetilde{Q}$ becomes a singular matrix in the following two situations[2]:

(i) The first one is that $\left|\sum_{i\in S_S} z_i\right| = |S_S|$, i.e., the samples of $S_S$ just have one kind of labels. For example:
 (a) When $\forall i \in S_S, z_i = +1$, we have $\mathbf{e}_{S_S} - \mathbf{z}_{S_S} = \mathbf{0}$.
 (b) When $\forall i \in S_S, z_i = -1$, we have $\mathbf{e}_{S_S} + \mathbf{z}_{S_S} = \mathbf{0}$.
 In these two cases, $\widetilde{Q}$ is clearly a singular matrix.

(ii) The second one is that $|M^+| > 1$, where $M^+$ is defined as $M^+ = \{(x_i, y_i, -1) \in S_S : (x_i, y_i, +1) \in S_S\}$, which implies that $\epsilon$-tube becomes 0-tube. Specifically, there exist four samples indexed by $i_1, i_2, k_1$ and $k_2$, respectively, where $(x_{i_1}, y_{i_1}) = (x_{i_2}, y_{i_2}), z_{i_1} = -z_{i_2}$, and $(x_{k_1}, y_{k_1}) = (x_{k_2}, y_{k_2}), z_{k_1} = -z_{k_2}$. Then according to (13), we have $\Delta g_{i_1} + \Delta g_{i_2} = \Delta g_{k_1} + \Delta g_{k_2}$, which means $\widetilde{Q}_{i_1*} + \widetilde{Q}_{i_2*} = \widetilde{Q}_{k_1*} + \widetilde{Q}_{k_2*}$. In this case, it is easy to verify that $\widetilde{Q}$ is a singular matrix.
 And if $M^+ \neq \emptyset$, we define $M$ as the contracted set which is obtained by deleting any one sample from $M^+$; if $M^+ = \emptyset$, we define $M$ as an empty set. Finally we let $S'_S = S_S - M$. Then if $M \neq \emptyset$, $\widetilde{Q}$ is clearly a singular matrix.

If $|\sum_{i\in S_S} z_i| \neq |S_S|$, $\widetilde{M} = M \cup \{i_{\epsilon'}\}$, and otherwise $\widetilde{M} = M$. Thus, we have the contracted matrix $\widetilde{Q}_{\setminus \widetilde{M}^2}$. Theorem 4 shows that $\widetilde{Q}_{\setminus \widetilde{M}^2}$ has the inverse matrix $\check{R}$ under Assumption 1 (the details of Theorem 4 will be discussed in Section 3.4). Furthermore, we let $\boxed{\Delta\epsilon'} = 0$, $\Delta\alpha_M = \mathbf{0}$, then the linear relationship between $\Delta h_{\setminus \widetilde{M}}$ and $\Delta\alpha_c$ can be obtained similarly as follows:

$$\Delta h_{\setminus \widetilde{M}} = \begin{bmatrix} \Delta b' \\ \Delta\epsilon' \\ \Delta\alpha_{S_S} \end{bmatrix}_{\setminus \widetilde{M}} = \check{R} \begin{bmatrix} 0 \\ 0 \\ \mathbf{u}_{S_S} \end{bmatrix}_{\setminus \widetilde{M}} \Delta\eta \stackrel{\text{def}}{=} \begin{bmatrix} \check{\beta}_{b'} \\ \check{\beta}_{\epsilon'} \\ \check{\beta}_{S_S} \end{bmatrix}_{\setminus \widetilde{M}} \Delta\eta. \tag{17}$$

Finally, letting $\boxed{\check{\beta}_{\epsilon'}} = 0$, and $\check{\beta}_M = \mathbf{0}$, substituting (17) into (13), we can get the linear relationship between $\Delta g_i$ and $\Delta\eta$ as (16).

### 3.1.2. Computing maximal increment $\Delta\eta^{\max}$

The principles of *initial adjustments* cannot be used directly to obtain the new state, such that all the samples satisfy the KKT conditions. To handle this problem, the main strategy is to compute the maximal increment $\Delta\eta^{\max}$ for each adjustment, such that a certain sample migrates among the sets $S_S$, $S_R$ and $S_E$. Three cases must be considered to account for such structural changes:

(i) A certain $\alpha_i$ in $S_S$ reaches a bound (an upper or a lower bound). Compute the sets: $I_+^{S_S} = \{i \in S_S : \check{\beta}_i > 0\}$, $I_-^{S_S} = \{i \in S_S : \check{\beta}_i < 0\}$, where the samples with $\check{\beta}_i = 0$ are ignored due to their insensitivity to $\Delta\eta$. Thus the maximum possible weight updates are

$$\Delta\alpha_i^{\max} = \begin{cases} C - \alpha_i, & \text{if } i \in I_+^{S_S} \\ -\alpha_i, & \text{if } i \in I_-^{S_S} \end{cases} \tag{18}$$

and the maximal possible $\Delta\eta^{S_S}$, before a certain sample in $S_S$ moves to $S_R$ or $S_E$, is: $\Delta\eta^{S_S} = \min_{i\in I_+^{S_S} \cup I_-^{S_S}} \frac{\Delta\alpha_i^{\max}}{\check{\beta}_i}$.

(ii) A certain $g_i$ corresponding to a sample in $S_R$ or $S_E$ reaches zero. Compute the sets: $I_+^{S_E} = \{i \in S_E : \check{\gamma}_i > 0\}$, $I_-^{S_R} = \{i \in S_R : \check{\gamma}_i < 0\}$, where samples with $\check{\gamma}_i = 0$ are again ignored because of their insensitivity to $\Delta\eta$. Thus the maximal possible $\Delta\eta^{S_R,S_E}$, before a certain sample in $S_R$ or $S_E$ migrates to $S_S$, is: $\Delta\eta^{S_R,S_E} = \min_{i\in I_+^{S_E} \cup I_-^{S_R}} \frac{-g_i}{\check{\gamma}_i}$.

(iii) $\eta$ reaches the upper bound. The maximal possible $\Delta\eta^{\eta}$, before reaching the upper bound 1, is: $\Delta\eta^{\eta} = 1 - \eta$.

Finally, the three smallest values constitute the maximal increment of $\Delta\eta$. That is

$$\Delta\eta^{\max} = \min\left\{\Delta\eta^{S_S}, \Delta\eta^{S_R,S_E}, \Delta\eta^{\eta}\right\}. \tag{19}$$

After the critical adjustment quantity $\Delta\eta^{\max}$ is determined, we can update $\eta, \alpha, g, S_S, S_E$ and $S_R$, similarly to the approaches in Diehl and Cauwenberghs (2003).

### 3.1.3. Updating the inverse matrix $\check{R}$

Once the components of the set $S_S$ are changed, the set $S'_S$ and the state of $\left|\sum_{i\in S_S} z_i\right|$ may also change. That is a sample is either added to or removed from the set $S'_S$, and the state of $\left|\sum_{i\in S_S} z_i\right|$ is transformed from $\left|\sum_{i\in S_S} z_i\right| = |S_S|$ to $\left|\sum_{i\in S_S} z_i\right| \neq |S_S|$.[3] Accordingly, there exist changes in $\widetilde{Q}_{\setminus \widetilde{M}^2}$ and $\check{R}$. In this section, we describe the following rules for updating the inverse matrix $\check{R}$.

(i) If a sample $(x_t, y_t, z_t)$ is added into $S'_S$, then the inverse matrix $\check{R}$ can be expanded as follows:

$$\check{R} \leftarrow \begin{bmatrix} \check{R} & & 0 \\ & & \vdots \\ 0 & \cdots & 0 \end{bmatrix} + \frac{1}{\widetilde{\gamma}_t^t} \begin{bmatrix} \widetilde{\beta}_{b'}^t \\ \widetilde{\beta}_{\epsilon'}^t \\ \widetilde{\beta}_{S_S}^t \\ 1 \end{bmatrix}_{\setminus \widetilde{M}} \cdot \begin{bmatrix} \widetilde{\beta}_{b'}^t \\ \widetilde{\beta}_{\epsilon'}^t \\ \widetilde{\beta}_{S_S}^t \\ 1 \end{bmatrix}_{\setminus \widetilde{M}}^T \tag{20}$$

where $\begin{bmatrix} \widetilde{\beta}_{b'}^t \\ \widetilde{\beta}_{\epsilon'}^t \\ \widetilde{\beta}_{S_S}^t \end{bmatrix}_{\setminus \widetilde{M}} = -\check{R} \cdot \begin{bmatrix} z_t \\ 1 \\ Q_{S_S t} \end{bmatrix}_{\setminus \widetilde{M}}$, $\boxed{\widetilde{\beta}_{\epsilon'}^t} = 0$, $\widetilde{\beta}_M^t = \mathbf{0}$, and $\widetilde{\gamma}_t^t = \sum_{j\in S_S} \widetilde{\beta}_j^t Q_{tj} + z_t \widetilde{\beta}_{b'}^t + \widetilde{\beta}_{\epsilon'}^t + Q_{tt}$.

(ii) If the state of $\left|\sum_{i\in S_S} z_i\right|$ transforms into $\left|\sum_{i\in S_S} z_i\right| \neq |S_S|$ from $\left|\sum_{i\in S_S} z_i\right| = |S_S|$, then the inverse matrix $\check{R}$ can be expanded as follows:

$$\check{R} \leftarrow \begin{bmatrix} \check{R} + ac^T c & ac^T \\ ac & a \end{bmatrix} \tag{21}$$

where $c = -\begin{bmatrix} 0 & \mathbf{e}_{S'_S}^T \end{bmatrix} \cdot \check{R} \stackrel{\text{def}}{=} -\bar{c} \cdot \check{R}$, and $a = \frac{-1}{\bar{c} \cdot \check{R} \cdot \bar{c}^T}$.

(iii) If a sample $(x_t, y_t, z_t)$ is removed from $S'_S$, then the inverse matrix $\check{R}$ can be contracted as follows:

$$\check{R} \leftarrow \check{R}_{\setminus tt} - \frac{1}{R_{tt}} \left(\check{R}_{*t} \check{R}_{t*}\right)_{\setminus tt}. \tag{22}$$

In summary, during the *initial adjustments*, the inverse matrix $\check{R}$ can be updated as described above. In addition, after the *strict restoration adjustments*, we need to recompute the inverse matrix

[2] In this paper, we do not consider the training sample set $T$ having duplicate training samples. A more general assumption of the dataset is that the margin support vectors are all linearly independent in RKHS (collectively called Assumption 1). The more detailed explanation of Assumption 1 can be found in Gu and Sheng (2013).

[3] When the set of *margin support vectors* with the label $z_o$ becomes $S_S^{z_o} = \{(x_t, y_t, z_o)\}$, after removing a sample with the label $z_o$ from the set $S_S$, we can have that $\check{\beta}_t = 0$ according to the definition of inverse matrix. This means that removing a sample from $S'_S$ will not lead to the transformation from $|\sum_{i\in S_S} z_i| \neq |S_S|$ to $|\sum_{i\in S_S} z_i| = |S_S|$.

$\check{R}$ for the next round of *initial adjustments*, which can be obtained from $\widehat{R}$ by the following steps:

(i) Compute the inverse matrix of $Q_{S_S S_S}$ based on $\widehat{R}$ using the contracted rule, similar to (22).
(ii) Update the inverse matrix of $\frac{l}{l+1}Q_{S_S S_S}$ by the rule $R \leftarrow \frac{l+1}{l}R$.
(iii) Calculate the inverse matrix $\check{R}$ for the next round of the *initial adjustments* by the expanded rule, similar to (21).

### 3.2. Relaxed adiabatic incremental adjustments

Because there may exist Conflict-1 between Eqs. (8) and (9) during the adjustments for $\alpha_c$, the limitation imposed by Eq. (9) is removed in this step. Thus, during the incremental adjustment for $\alpha_c$, in order to keep all the samples satisfying the KKT conditions except the restriction (9), we have the following linear system under the condition that $\Delta\epsilon' = 0$:

$$\sum_{j\in S_S} z_j \Delta\alpha_j + z_c \Delta\alpha_c = 0 \tag{23}$$

$$\Delta g_i = \sum_{j\in S_S} \Delta\alpha_j Q_{ij} + z_i \Delta b' + \Delta\alpha_c Q_{ic} = 0, \quad \forall i \in S_S. \tag{24}$$

The linear system (23)–(24) can be written as:

$$\begin{bmatrix} 0 & \mathbf{z}_{S_S}^T \\ \mathbf{z}_{S_S} & Q_{S_S S_S} \end{bmatrix} \begin{bmatrix} \Delta b' \\ \Delta\alpha_{S_S} \end{bmatrix} = - \begin{bmatrix} z_c \\ Q_{S_S c} \end{bmatrix} \Delta\alpha_c. \tag{25}$$

Like the *initial adjustments*, we define $\widehat{M} = M \cup \{i_{\epsilon'}\}$. Thus, we have the contracted matrix $\widetilde{Q}_{\backslash \widehat{M}^2}$. Corollary 5 shows that $\widetilde{Q}_{\backslash \widehat{M}^2}$ has an inverse (the details of the corollary will be discussed in Section 3.4). Let $R = \widetilde{Q}_{\backslash \widehat{M}^2}^{-1}$, the linear relationship between $\Delta b'$, $\Delta\alpha_{S'_S}$, and $\Delta\alpha_c$ can be solved as follows:

$$\begin{bmatrix} \Delta b' \\ \Delta\alpha_{S'_S} \end{bmatrix} = -R \begin{bmatrix} z_c \\ Q_{S'_S c} \end{bmatrix} \Delta\alpha_c \stackrel{\text{def}}{=} \begin{bmatrix} \beta_{b'}^c \\ \beta_{S'_S}^c \end{bmatrix} \Delta\alpha_c. \tag{26}$$

Let $\beta_M^c = \mathbf{0}$, and substitute (26) into (23), we can get the linear relationship between $\Delta g_i$ and $\Delta\alpha_c$ as follows:

$$\begin{aligned} \Delta g_i &= \left( \sum_{j\in S_S} \beta_j^c Q_{ij} + z_i \beta_{b'}^c + Q_{ic} \right) \Delta\alpha_c \\ &\stackrel{\text{def}}{=} \gamma_i^c \Delta\alpha_c, \quad \forall i \in S. \end{aligned} \tag{27}$$

Obviously, $\forall i \in S_S$, we have $\gamma_i^c = 0$.

### 3.3. Strict restoration adjustments

After the *relaxed adiabatic incremental adjustments*, we need to adjust $\sum_{i\in S}\alpha_i$ to restore the equality $\sum_{i\in S}\alpha_i = C\nu(l+1)$. For each adjustment of $\sum_{i\in S}\alpha_i$, in order to keep all the samples satisfying the KKT conditions and to prevent the reoccurrence of the conflict (referred to as Conflict-2) between Eqs. (8) and (9) efficiently, we have the following linear system:

$$\sum_{j\in S_S} z_j \Delta\alpha_j = 0 \tag{28}$$

$$\sum_{j\in S_S} \Delta\alpha_j + \varepsilon\Delta\epsilon' + \Delta\zeta = 0 \tag{29}$$

$$\Delta g_i = \sum_{j\in S_S} \Delta\alpha_j Q_{ij} + z_i \Delta b' + \Delta\epsilon' = 0, \quad \forall i \in S_S \tag{30}$$

where $\Delta\zeta$ is the introduced variable for adjusting $\sum_{i\in S}\alpha_i$. $\varepsilon$ is any negative number. $\varepsilon\Delta\epsilon'$ is incorporated in (29) as an extra term. Gu et al. (2012) wish to prevent the reoccurrence of Conflict-2 between Eqs. (8) and (9) efficiently using this extra term.

The linear system (30)–(29) can be further rewritten as:

$$\underbrace{\begin{bmatrix} 0 & 0 & \mathbf{z}_{S_S}^T \\ 0 & \varepsilon & \mathbf{1}^T \\ \mathbf{z}_{S_S} & \mathbf{1} & Q_{S_S S_S} \end{bmatrix}}_{\widehat{Q}} \begin{bmatrix} \Delta b' \\ \Delta\epsilon' \\ \Delta\alpha_{S_S} \end{bmatrix} = - \begin{bmatrix} 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \Delta\zeta. \tag{31}$$

Let $\widehat{Q}_{\backslash M^2}$ be the contracted matrix of $\widehat{Q}$. Theorem 6 shows that $\widehat{Q}_{\backslash M^2}$ has an inverse (the details of Theorem 6 will be discussed in Section 3.4). Let $\widehat{R} = \widehat{Q}_{\backslash M^2}^{-1}$, the linear relationship between $\Delta b'$, $\Delta\epsilon'$, $\Delta\alpha_{S'_S}$ and $\Delta\zeta$ can be obtained as follows:

$$\begin{bmatrix} \Delta b' \\ \Delta\epsilon' \\ \Delta\alpha_{S'_S} \end{bmatrix} = -\widehat{R} \begin{bmatrix} 0 \\ 1 \\ \mathbf{0} \end{bmatrix} \Delta\zeta \stackrel{\text{def}}{=} \begin{bmatrix} \widehat{\beta}_{b'} \\ \widehat{\beta}_{\epsilon'} \\ \widehat{\beta}_{S'_S} \end{bmatrix} \Delta\zeta. \tag{32}$$

From (32), we have $\sum_{i\in S}\Delta\alpha_i = -(1+\varepsilon\widehat{\beta}_{\epsilon'})\Delta\zeta$, which implies that the control of the adjustment of $\sum_{i\in S}\alpha_i$ is achieved by $\Delta\zeta$.

Finally, letting $\widehat{\beta}_M = \mathbf{0}$, and substituting (32) into (30), we get the linear relationship between $\Delta g_i$ and $\Delta\zeta$ as follows:

$$\begin{aligned} \Delta g_i &= \left( \sum_{j\in S_S} \widehat{\beta}_j Q_{ij} + z_i \widehat{\beta}_{b'} + \widehat{\beta}_{\epsilon'} \right) \Delta\zeta \\ &\stackrel{\text{def}}{=} \widehat{\gamma}_i \Delta\zeta, \quad \forall i \in S. \end{aligned} \tag{33}$$

Obviously, $\forall i \in S_S$, we also have $\widehat{\gamma}_i = 0$.

### 3.4. Do the three key inverse matrices exist?

As stated above, the inverses of $\widetilde{Q}_{\backslash \widetilde{M}^2}$, $\widetilde{Q}_{\backslash \widehat{M}^2}$, and $\widehat{Q}_{\backslash M^2}$ are the cornerstone of the *initial adjustments*, the *relaxed adiabatic incremental adjustments*, and the *strict restoration adjustments*, respectively. In this section, we prove their existence under Assumption 1 through Theorem 4, Corollary 5, and Theorem 6.

**Lemma 3.** *If $A$ is a $k \times n$ matrix with a rank $k$, $B$ is an $n \times n$ positive definite matrix, then $ABA^T$ will also be a positive definite matrix.*

Lemma 3 can be easily proved by Cholesky decomposition (Householder, 1974) and Sylvester's rank inequality (Householder, 1974).

**Theorem 4.** *During the initial adjustments, if $\left|\sum_{i\in S_S} z_i\right| = |S_S|$, the determinant of $\widetilde{Q}_{\backslash \widetilde{M}^2}$ is always less than 0; otherwise, it is always greater than 0.*

We prove Theorem 4 detailedly in Appendix A.2.

**Corollary 5.** *During the relaxed adiabatic incremental adjustments, the determinant of $\widetilde{Q}_{\backslash \widehat{M}^2}$ is always less than 0.*

Corollary 5 can be obtained easily according to Theorem 4.

**Theorem 6.** *During the strict restoration adjustments, if $\varepsilon < 0$, then the determinant of $\widehat{Q}_{\backslash M^2}$ is always more than 0.*

We prove Theorem 6 detailedly in Appendix A.3.

In addition, it is not difficult to find that the time complexities of updating/downdating $\check{R}$, $R$, and $\widehat{R}$ are all $O(|S_S|^2)$, based on the rules in Section 3.1.3 and the inverse updating/downdating rules in Gu et al. (2012). A lot of references (Gunter & Zhu, 2007; Hastie, Rosset, Tibshirani, & Zhu, 2004; Wang, Yeung, & Lochovsky, 2008) reported that the average size of $S_S$ does not increase with the size of training set. Our experimental results also verified this point, which means that INSVR can efficiently handle large scale problems.

## 4. Experiments

### 4.1. Design of experiments

In order to demonstrate the effectiveness of INSVR, and to show the advantage of INSVR in terms of computation efficiency, we conduct a detailed experimental study.

To demonstrate the effectiveness of INSVR (i.e., to show that INSVR is a workable and meaningful algorithm under Assumption 1), we investigate the existence of the two kinds of conflicts, the singularity of $\widetilde{Q}$, and the convergence of INSVR, respectively. To validate the existence of the two kinds of conflicts, we count the two kinds of conflicts (i.e., Conflict-1 and Conflict-2, the details can be found in Gu et al. (2012)) during the *relaxed adiabatic incremental adjustments* and the *strict restoration adjustments*, respectively, over 500 trials. To investigate the singularity of $\widetilde{Q}$, we count the two special cases (c.f. Section 3.1.1, denoted as SC-1 and SC-2) during the *initial adjustments*, and all the three steps of INSVR, respectively, over 500 trials. To illustrate the fast convergence of INSVR empirically, we investigate the average numbers of iterations of the *initial adjustments* (IA), the *relaxed adiabatic incremental adjustments* (RAIA), and the *strict restoration adjustments* (SRA), respectively, over 20 trials.

To show that INSVR has the computational superiority over the batch learning algorithm (i.e., the Sequential Minimal Optimization (SMO) algorithm of $\nu$-SVR) with both cold start and warm start, we provide the empirical analysis of them in terms of scaling of run-time efficiency. It should be noted that INSVR and the SMO algorithm have the same generalization performance, because our INSVR obtains the exact solution of $\nu$-SVR, and the SMO algorithm also does.

### 4.2. Implementation

We implement our proposed INSVR in MATLAB. Chang and Lin (2002) proposed a recognized SMO-type algorithm specially designed for batch $\nu$-SVR training, which is implemented in C++ as a part of the LIBSVM software package (Chang & Lin, 2001). To compare the run-time in the same platform, we implement the $\nu$-SVR part of the LIBSVM with both cold start and warm start in MATLAB (Chen, Lin, & Schölkopf, 2005).

All experiments were performed on a 2.5 GHz Intel Core i5 machine with 8GB RAM and MATLAB 7.10 platform. For kernels, the linear kernel $K(x_1, x_2) = x_1 \cdot x_2$, polynomial kernel $K(x_1, x_2) = (x_1 \cdot x_2 + 1)^d$ with $d = 2$, and Gaussian kernel $K(x_1, x_2) = \exp(-\|x_1 - x_2\|^2/2\sigma^2)$ with $\sigma = 0.707$ are used in all the experiments. The parameter, $\varepsilon$ of the *strict restoration adjustments* is fixed at $-1$, because any negative value of $\varepsilon$ does not change the updating path.[4] The values of $\nu$ and $C$ are fixed at 0.3 and 100, respectively, in all the experiments.

### 4.3. Datasets

Table 2 presents the nine benchmark datasets used in our experiments. These datasets are divided into two parts: the first five are small datasets, and the last four are larger datasets.

**Table 2**
Benchmark datasets used in the experiments.

| Dataset | Max #training set | #attributes |
|---|---|---|
| Housing | 506 | 13 |
| Forest Fires | 517 | 12 |
| Auto MPG | 392 | 7 |
| Triazines | 186 | 60 |
| Concrete Compressive Strength | 1,030 | 8 |
| Friedman | 15,000 | 10 |
| Cpusmall | 8,192 | 12 |
| Cadata | 20,640 | 8 |
| YearPredictionMSD | 51,630 | 90 |

The first five in Table 2 are Housing, Forest Fires, Auto MPG, Triazines, and Concrete Compressive Strength. They are from the UCI machine learning repository (Frank & Asuncion, 2010). Their sizes vary from 186 to 1030.

The sizes of the last four datasets vary from 8192 to 51,630. Cpusmall, Cadata, and YearPredictionMSD are available at http://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/regression.html. Friedman is an artificial dataset (Friedman, 1991). The input attributes $(x_1, \ldots, x_{10})$ are generated independently, each of which uniformly distributed over [0, 1]. The target is defined by

$$y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \sigma(0, 1) \quad (34)$$

where $\sigma(0, 1)$ is the noise term which is normally distributed with mean 0 and variance 1. Note that $x_1, \ldots, x_5$ only are used in (34), while $x_6, \ldots, x_{10}$ are noisy irrelevant input attributes.

### 4.4. Experimental results and discussion

When the training data sizes of the first six benchmark dataset are 10, 15, 20, 25, and 30, respectively, Table 3 presents the corresponding numbers of occurrences of Conflict-1 and Conflict-2. From this table, we find that the two kinds of conflicts happen with a high probability, especially Conflict-1. Thus, it is essential to handle the conflicts within the incremental $\nu$-SVR learning algorithm. Our INSVR can avoid these conflicts successfully.

Table 3 also presents the numbers of occurrences of SC-1 and SC-2 on the first six benchmark datasets, where the training data size of each dataset is also set as 10, 15, 20, 25, and 30, respectively. From this table, we find that SC-1 happens with a higher probability than SC-2 does. Although SC-2 happens with a low probability, the possibility of the occurrences still cannot be excluded. Thus, it is very significant that INSVR handles these two special cases.

Fig. 2 presents the average numbers of iterations of IA, RAIA, and SRA, respectively, on the different benchmark datasets and different kernels. It is obvious that these three steps exhibit quick convergence for all benchmark datasets and kernels, especially IA. Combined with the results in Table 3, we can conclude that INSVR avoids the infeasible updating paths as far as possible, and successfully converges to the optimal solution with a faster convergence speed.

Fig. 3 compares the run-time of our INSVR and LibSVM with both cold start and warm start, on the different benchmark datasets and different kernels. The results demonstrate that our INSVR is generally much faster than the batch implementations using both cold start and warm start.

## 5. Concluding remarks

To design an exact incremental $\nu$-SVR algorithm based on AONSVM, we propose a special procedure called *initial adjustments*

[4] Like AONSVM, it is easy to verify that $\varepsilon$ can determine $\Delta\zeta^*$, but is independent with the structural changes of the sets $S_S$, $S_R$ and $S_E$.

**Table 3**
The number of occurrences of Conflict-1, Conflict-2, SC-1 and SC-2 on the six benchmark datasets over 500 trials. Note that L, P, and G are the abbreviations of linear, polynomial and Gaussian kernels, respectively.

| Dataset | Size | Conflict-1 | | | Conflict-2 | | | SC-1 | | | SC-2 | | | Dataset | Size | Conflict-1 | | | Conflict-2 | | | SC-1 | | | SC-2 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | L | P | G | L | P | G | L | P | G | L | P | G | | | L | P | G | L | P | G | L | P | G | L | P | G |
| Housing | 10 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 0 | 1 | Triazines | 10 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 27 |
| | 15 | 0 | 1 | 4 | 0 | 1 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | | 15 | 0 | 8 | 0 | 0 | 1 | 0 | 0 | 2 | 0 | 0 | 0 | 0 |
| | 20 | 0 | 2 | 186 | 0 | 2 | 55 | 0 | 5 | 125 | 0 | 0 | 0 | | 20 | 6 | 29 | 0 | 0 | 6 | 0 | 1 | 20 | 0 | 0 | 0 | 0 |
| | 25 | 0 | 0 | 8 | 0 | 0 | 4 | 0 | 0 | 3 | 0 | 0 | 0 | | 25 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| | 30 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 1 | 0 | 0 | 0 | | 30 | 0 | 2 | 0 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| Forest fires | 10 | 5 | 3 | 0 | 2 | 3 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | Concrete compressive strength | 10 | 13 | 27 | 14 | 4 | 4 | 4 | 0 | 0 | 0 | 0 | 0 | 2 |
| | 15 | 11 | 12 | 0 | 1 | 4 | 0 | 3 | 0 | 0 | 0 | 0 | 0 | | 15 | 20 | 24 | 66 | 5 | 4 | 17 | 2 | 12 | 0 | 0 | 0 | 1 |
| | 20 | 164 | 289 | 250 | 29 | 91 | 178 | 46 | 123 | 218 | 0 | 0 | 0 | | 20 | 57 | 120 | 284 | 9 | 22 | 63 | 25 | 96 | 219 | 0 | 0 | 0 |
| | 25 | 60 | 22 | 0 | 12 | 8 | 0 | 12 | 3 | 0 | 0 | 0 | 0 | | 25 | 32 | 31 | 73 | 6 | 11 | 41 | 0 | 13 | 24 | 0 | 0 | 0 |
| | 30 | 62 | 10 | 0 | 8 | 2 | 0 | 5 | 0 | 0 | 0 | 0 | 0 | | 30 | 38 | 27 | 16 | 6 | 6 | 12 | 3 | 0 | 8 | 0 | 0 | 0 |
| Auto MPG | 10 | 21 | 21 | 0 | 3 | 6 | 0 | 1 | 0 | 0 | 0 | 0 | 2 | Friedman | 10 | 0 | 0 | 156 | 0 | 0 | 61 | 0 | 0 | 0 | 0 | 45 | 4 |
| | 15 | 38 | 8 | 2 | 4 | 1 | 0 | 2 | 1 | 0 | 0 | 0 | 1 | | 15 | 0 | 0 | 279 | 0 | 0 | 64 | 0 | 2 | 5 | 0 | 3 | 0 |
| | 20 | 95 | 49 | 14 | 7 | 6 | 3 | 28 | 26 | 11 | 0 | 0 | 0 | | 20 | 2 | 0 | 537 | 0 | 0 | 125 | 1 | 0 | 368 | 0 | 0 | 0 |
| | 25 | 34 | 9 | 0 | 4 | 0 | 0 | 6 | 1 | 0 | 0 | 0 | 0 | | 25 | 0 | 0 | 213 | 0 | 0 | 98 | 2 | 4 | 24 | 0 | 0 | 0 |
| | 30 | 18 | 8 | 0 | 2 | 1 | 0 | 2 | 0 | 0 | 0 | 0 | 0 | | 30 | 1 | 0 | 210 | 2 | 0 | 0 | 83 | 0 | 2 | 0 | 0 | 0 |

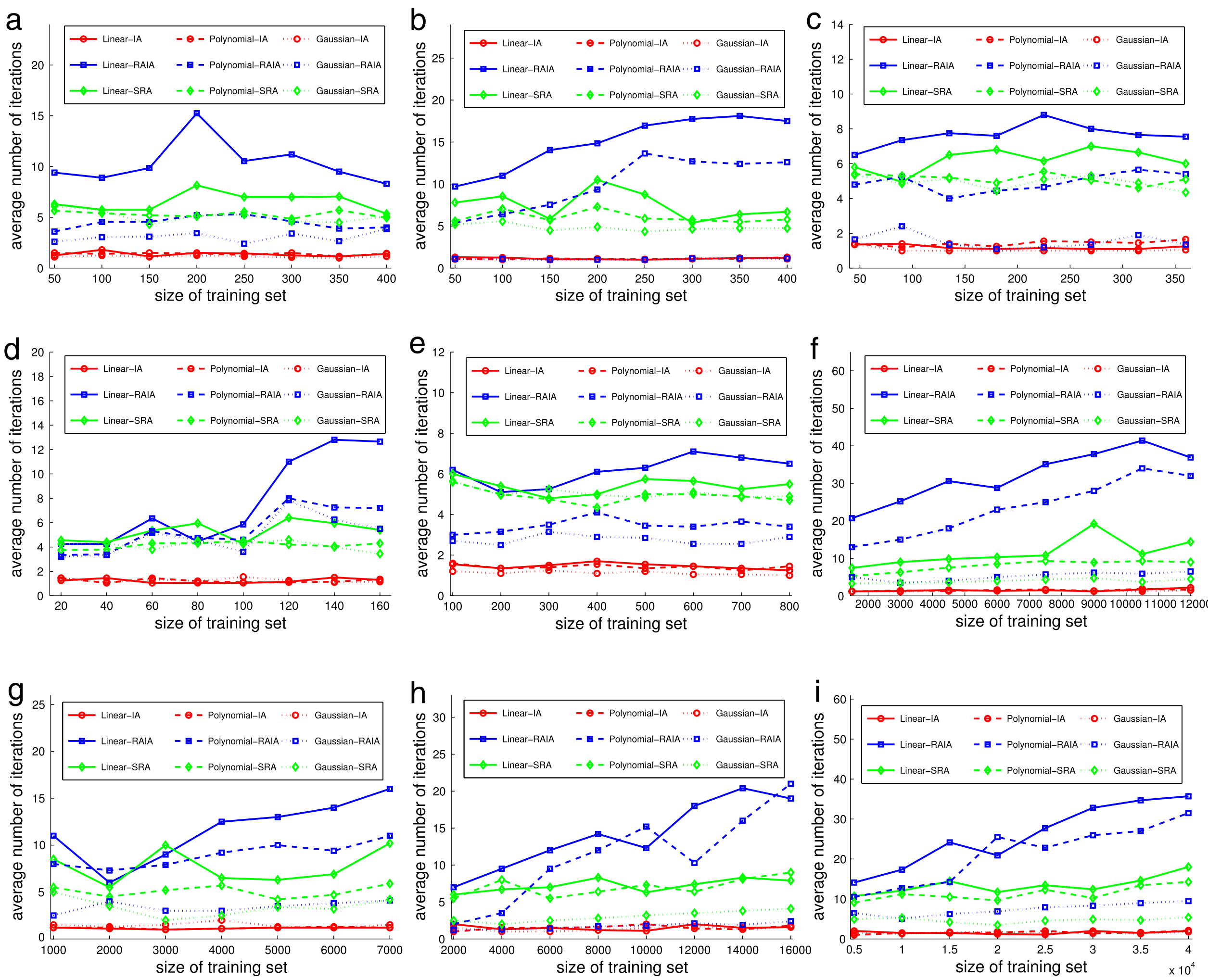


**Fig. 2.** Average numbers of iterations of IA, RAIA, and SRA on the different benchmark datasets. (a) Housing. (b) Forest Fires. (c) Auto MPG. (d) Triazines. (e) Concrete Compressive Strength. (f) Friedman. (g) Cpusmall. (h) Cadata. (i) YearPredictionMSD.

for preparing the initial solution before the incremental learning. The *initial adjustments* and the two steps of AONSVM constitute INSVR. We also prove the existence of the three key inverse matrices, which are the cornerstone of INSVR. The experimental results demonstrate that INSVR can successfully converge to the exact optimal solution in a finite number of steps by avoiding

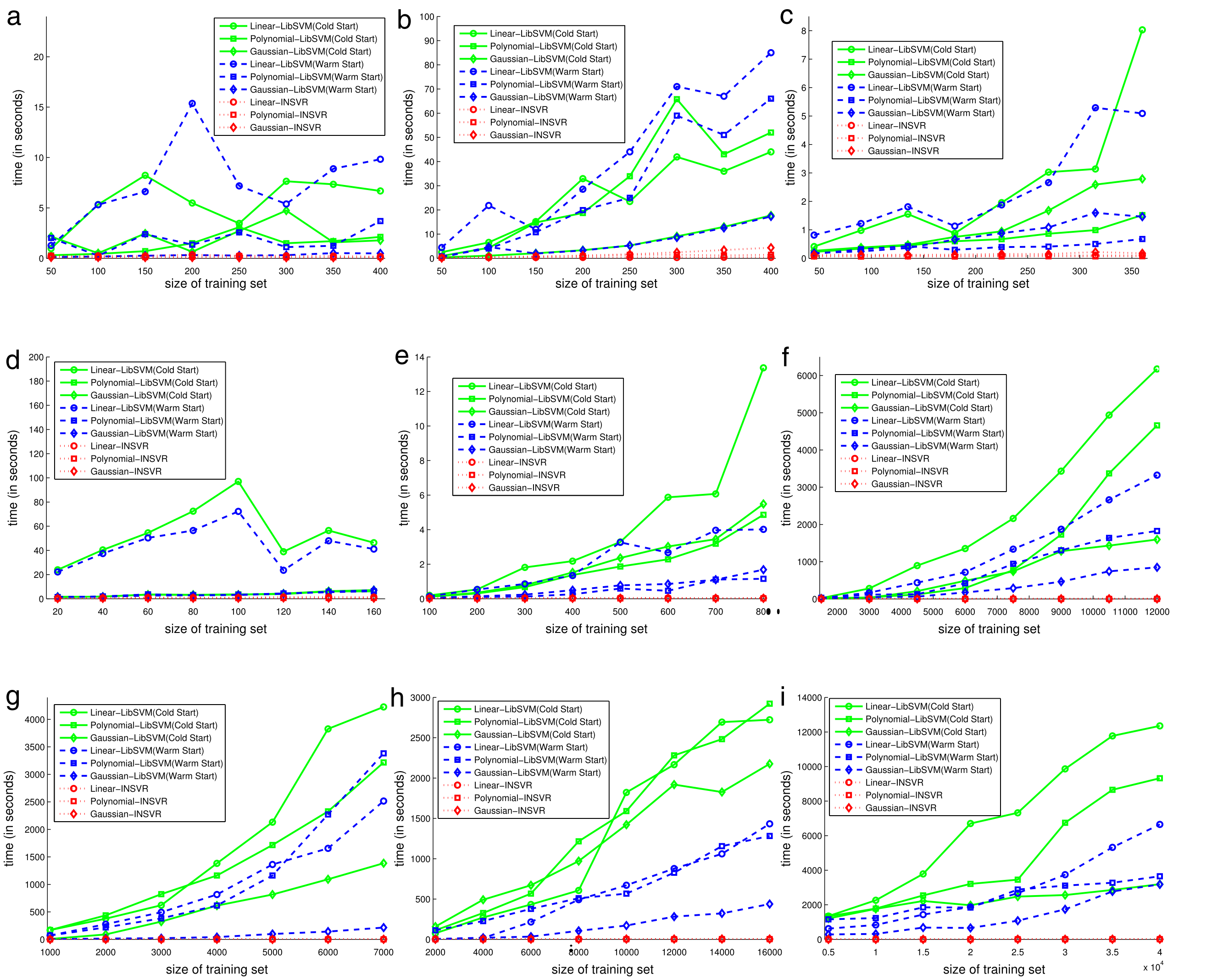


**Fig. 3.** Run-time of LibSVM(Cold Start), LibSVM(Warm Start) and INSVR (in seconds) on the different benchmark datasets. (a) Housing. (b) Forest Fires. (c) Auto MPG. (d) Triazines. (e) Concrete Compressive Strength. (f) Friedman. (g) Cpusmall. (h) Cadata. (i) YearPredictionMSD.

Conflict-1 and Conflict-2, and is faster than batch $\nu$-SVR algorithms with both cold and warm start.

Theoretically, the decremental $\nu$-SVR learning can also be designed in a similar manner. Based on the incremental and decremental $\nu$-SVR algorithms, we can implement leave-one-out cross validation (Weston, 1999) and the learning with limited memory (Laskov et al., 2006) efficiently. In the future, we also plan to implement approximate on-line $\nu$-SVR learning based on the large-scale SVR training algorithms, such as stochastic sub-gradient descent algorithm (Shalev-Shwartz et al., 2011), and coordinate descent algorithm (Ho & Lin, 2012; Wang & Lin, 2014), and use the method to analyze the images of synthetic aperture radar (Zhang, Wu, Nguyen, & Sun, 2014) and vehicle (Wen, Shao, Fang, & Xue, 2015).

## Acknowledgments

This work was supported by the Project Funded by the Priority Academic Program Development (PAPD) of Jiangsu Higher Education Institutions, the U.S. National Science Foundation (IIS-1115417), and the National Natural Science Foundation of China (Nos: 61232016 and 61202137).

## Appendix. Proofs to theoretical works

### A.1. Proof to Theorem 2

The *initial adjustments* is to retrieve the optimal solution of the following problem:

$$\min_{\alpha} \quad \frac{1}{2(l+1)} \sum_{i,j=1}^{2l} \alpha_i \alpha_j z_i z_j K(x_i, x_j) - \sum_{i=1}^{2l} z_i y_i \alpha_i$$
$$s.t. \quad \sum_{i=1}^{2l} z_i \alpha_i = 0, \qquad \sum_{i=1}^{2l} \alpha_i = C\nu l, \qquad (35)$$
$$0 \le \alpha_i \le C, \quad i = 1, \ldots, 2l.$$

It is easy to verify that (35) is the dual of the following problem.

$$\min_{w,\epsilon,b,\xi_i^{(*)}} \quad \frac{l+1}{2}\langle w, w\rangle + C \cdot \left(\nu\epsilon l + \sum_{i=1}^{l}(\xi_i + \xi_i^*)\right)$$
$$s.t. \quad (\langle w, \phi(x_i)\rangle + b) - y_i \le \epsilon + \xi_i, \qquad (36)$$
$$y_i - (\langle w, \phi(x_i)\rangle + b) \le \epsilon + \xi_i^*,$$
$$\xi_i^{(*)} \ge 0, \quad i = 1, \ldots, l.$$

As stated in (7), $b'$ and $\epsilon'$ are Lagrangian multipliers of (35). Furthermore, according to the KKT conditions, we have the relationships between the optimal solution of (36) and the one of (35) as: $w = \frac{1}{l+1}\sum_{i=1}^{2l}\alpha_i z_i \phi(x_i)$, $b' = b$, and $\epsilon' = \epsilon$. According to the strong duality theorem (Boyd & Vandenberghe, 2004), the duality gap between (36) and (35) is equal to zero. Assume $\epsilon' < 0$ after the *initial adjustments*, we can find a smaller value for the objective function in (36) by setting $\epsilon = 0$, which implies a contradiction. This completes the proof.

### *A.2. Proof to Theorem 4*

To prove this theorem, the special case $M^+ = \emptyset$ is considered first.

According to (17) and the definition of an inverse matrix, if $S_S = \{(x_t, y_t, z_t)\}$, we have $\check{\beta}_t = \check{R}_{t*}\,[0\; z_t y_t]^T = \frac{1}{\det(\widetilde{Q}_{\setminus \widetilde{M}^2})}[-y_t\; 0]\,[0\; z_t y_t]^T = 0$, which implies that $S_S$ is always nonempty during the *initial adjustments*. Because $M^+ = \emptyset$, it is easy to verify that: $\forall i, j \in S_S$, if $i \neq j$, then $x_i \neq \pm x_j$. According to Assumption 1, $Q_{S_S S_S}$ is a positive definite matrix, so there must exist the inverse matrix $Q_{S_S S_S}^{-1}$, and $Q_{S_S S_S}^{-1}$ is also a positive definite matrix.

If $|\sum_{i\in S_S} z_i| \neq |S_S|$, let $P = \left[\mathbf{z}_{S'_S}, \mathbf{1}_{S'_S}\right]$, and otherwise $P = \mathbf{z}_{S'_S}$. It is easy to verify that $\operatorname{rank}(P) = 2$, if $|\sum_{i\in S_S} z_i| \neq |S_S|$, otherwise $P = \mathbf{z}_{S'_S}$ and $\operatorname{rank}(P) = 1$. Then from Lemma 3 in Gu and Sheng (2013), we have

$$\begin{aligned}\det\left(\widetilde{Q}_{\setminus \widetilde{M}^2}\right) &= \det(Q_{S_S S_S})\det(\mathbf{0} - P^T Q_{S_S S_S}^{-1} P)\\ &= (-1)^{\operatorname{rank}(P)}\det(Q_{S_S S_S})\det(P^T Q_{S_S S_S}^{-1} P).\end{aligned}$$

Because $Q_{S_S S_S}$ is a positive definite matrix, $\det(Q_{S_S S_S}) > 0$. From Lemma 3, we can also show that $P^T Q_{S_S S_S}^{-1} P$ is a positive definite matrix, so $\det(P^T Q_{S_S S_S}^{-1} P) > 0$. This completes the proof under the condition $M^+ = \emptyset$.

Next, we consider the case $M^+ \neq \emptyset$. First, let $\widetilde{Q}_{\setminus \widetilde{M}^2} = \widetilde{Q}''$. We can construct the elementary transformation matrix $\widetilde{Q}'$ as follows:

$$\underbrace{\begin{bmatrix} 0 & 0 & \mathbf{z}_{S_S}^T \\ 0 & 0 & \mathbf{1}_{S_S}^T \\ \mathbf{z}_{S_S} & \mathbf{1}_{S_S} & Q_{S_S S_S}\end{bmatrix}_{\setminus \widetilde{M}^2}}_{\widetilde{Q}''} \xRightarrow{\substack{\widetilde{Q}''_{i_1*} + \widetilde{Q}''_{i_2*} \to \widetilde{Q}''_{i_1*} \\ \widetilde{Q}''_{*i_1} + \widetilde{Q}''_{*i_2} \to \widetilde{Q}''_{*i_1}}}$$

$$\begin{bmatrix} 0 & 0 & z'_{i_1} & \mathbf{z}_{S''_S}^T \\ 0 & 0 & e'_{i_1} & \mathbf{1}_{S''_S}^T \\ z'_{i_1} & e'_{i_1} & 0 & \mathbf{0} \\ \mathbf{z}_{S''_S} & \mathbf{1}_{S''_S} & \mathbf{0} & Q_{S''_S S''_S}\end{bmatrix}_{\setminus (\widetilde{M}^-)^2} \stackrel{\text{def}}{=} \widetilde{Q}'$$

where $M' = M^+ - M$, $S''_S = S_S - M^+$, $\widetilde{M}^- = \widetilde{M} - M'$, and $\widetilde{Q}'$ is obtained from $\widetilde{Q}''$ by adding the $i_2$th row and column to the $i_1$-th row and column for all $i_1 \in M'$, among them $(x_{i_2}, y_{i_2}, z_{i_2}) \in S''_S$ with $(x_{i_1}, y_{i_1}) = (x_{i_2}, y_{i_2})$ and $z_{i_1} = -z_{i_2}$. Obviously, $\widetilde{Q}'$ has the same determinant as $\widetilde{Q}''$. Then we can compute the determinant of $\widetilde{Q}''$ from $\widetilde{Q}'$. $\forall i, j \in S''_S$, if $i \neq j$, we have $x_i \neq \pm x_j$. So from Assumption 1, $Q_{S''_S S''_S}$ is a positive definite matrix. If $|\sum_{i\in S_S} z_i| \neq |S_S|$, let $P' = \left[\mathbf{z}_{S''_S}, \mathbf{1}_{S''_S}\right]$, $P'' = \left[z'_{i_1}, e'_{i_1}\right]$, otherwise $P' = \mathbf{z}_{S''_S}$, $P'' = z'_{i_1}$. It is easy to verify that $\operatorname{rank}(P'') = 1$.

If $\operatorname{rank}(P') = \operatorname{cols}(P')$, from Lemma 3 in Gu and Sheng (2013), we have

$$\begin{aligned}\det\left(\widetilde{Q}'\right) &= \det(Q_{S''_S S''_S}) \cdot \det\left(\begin{bmatrix}\mathbf{0} & P''^T \\ P'' & 0\end{bmatrix} - \begin{bmatrix}\overbrace{P'^T Q_{S''_S S''_S}^{-1} P'}^{\widetilde{P}} & \mathbf{0} \\ \mathbf{0} & 0\end{bmatrix}\right)\\ &= (-1)^{\operatorname{rank}(P')}\det\left(Q_{S''_S S''_S}\right)\det\left(\widetilde{P}\right)\det \cdot \left(P''^T \widetilde{P}^{-1} P''\right).\end{aligned}$$

Both $\widetilde{P}$ and $P''^T \widetilde{P}^{-1} P''$ are positive definite, because $\operatorname{rank}(P') = \operatorname{cols}(P')$ and $\operatorname{rank}(P'') = 1$. This completes the proof under the condition that $M^+ \neq \emptyset$ and $\operatorname{rank}(P') = \operatorname{cols}(P')$.

If $\operatorname{rank}(P') \neq \operatorname{cols}(P')$, i.e., $\mathbf{z}_{S''_S} = \pm \mathbf{1}_{S''_S}$, we can construct another elementary transformation matrix $\widetilde{Q}^{Tprime}$ based on $\widetilde{Q}'$ as follows:

$$\underbrace{\begin{bmatrix} 0 & 0 & z'_{i_1} & \mathbf{z}_{S''_S}^T \\ 0 & 0 & e'_{i_1} & \mathbf{1}_{S''_S}^T \\ z'_{i_1} & e'_{i_1} & 0 & \mathbf{0} \\ \mathbf{z}_{S''_S} & \mathbf{1}_{S''_S} & \mathbf{0} & Q_{S''_S S''_S}\end{bmatrix}}_{\widetilde{Q}'} \xRightarrow{\substack{\widetilde{Q}'_{\epsilon'*} \pm \widetilde{Q}'_{b'*} \to \widetilde{Q}'_{\epsilon'*} \\ \widetilde{Q}'_{*\epsilon'} \pm \widetilde{Q}'_{*b'} \to \widetilde{Q}'_{*\epsilon'}}}$$

$$\begin{bmatrix} 0 & 0 & z'_{i_1} & \mathbf{z}_{S''_S}^T \\ 0 & 0 & e''_{i_1} & \mathbf{0} \\ z'_{i_1} & e''_{i_1} & 0 & \mathbf{0} \\ \mathbf{z}_{S''_S} & \mathbf{0} & \mathbf{0} & Q_{S''_S S''_S}\end{bmatrix} \stackrel{\text{def}}{=} \widetilde{Q}'''$$

where $\widetilde{Q}'''$ is obtained from $\widetilde{Q}'$ by adding (deleting) the row and column indexed by $b'$ to the row and column indexed by $\epsilon'$. It is easy to verify that $e''_{i_1} \neq 0$. Obviously, $\widetilde{Q}'''$ has the same determinant as $\widetilde{Q}'$. We can compute the determinant of $\widetilde{Q}''$ from $\widetilde{Q}'''$. Then from Lemma 3 in Gu and Sheng (2013), we have

$$\begin{aligned}\det\left(\widetilde{Q}'''\right) &= \det\left(Q_{S''_S S''_S}\right)\det\left(\begin{bmatrix} 0 & 0 & z'_{i_1} \\ 0 & 0 & e''_{i_1} \\ z'_{i_1} & e''_{i_1} & 0\end{bmatrix}\right.\\ &\qquad \left. - \begin{bmatrix}\overbrace{\mathbf{z}_{S''_S}^T Q_{S''_S S''_S}^{-1}\mathbf{z}_{S''_S}}^{\widetilde{P}'} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0\end{bmatrix}\right)\\ &= -\det\left(Q_{S''_S S''_S}\right)\det\left(\widetilde{P}'\right)\det\left(\begin{bmatrix} 0 & e''_{i_1} \\ e''_{i_1} & \underbrace{z'_{i_1}\widetilde{P}'^{-1} z'_{i_1}}_{\widetilde{P}''}\end{bmatrix}\right)\\ &= \det\left(Q_{S''_S S''_S}\right)\det\left(\widetilde{P}'\right)\det\left(\widetilde{P}''\right)\det\left(e''_{i_1}\widetilde{P}''^{-1} e''_{i_1}\right)\end{aligned}$$

$\widetilde{P}'$, $\widetilde{P}''$ and $e''_{i_1}\widetilde{P}''^{-1} e''_{i_1}$ are positive definite because $\operatorname{rank}(\mathbf{z}_{S''_S}) = 1$, $z'_{i_1} \neq 0$, and $e''_{i_1} \neq 0$. This completes the proof.

### *A.3. Proof to Theorem 6*

According to the Laplace expansion of the determinant of $\widehat{Q}$ (Householder, 1974), we can have

$$\det\left(\widehat{Q}_{\setminus M^2}\right) = \det\left(\widetilde{Q}_{\setminus M^2}\right) + \varepsilon \det\left(\widetilde{Q}_{\setminus \widetilde{M}^2}\right).$$

Based on the conclusions of Theorem 4 and Corollary 5, and the premise of $\varepsilon < 0$, we have that $\det\left(\widehat{Q}_{\setminus M^2}\right) > 0$. This completes the proof.

## References


Boyd, Stephen, & Vandenberghe, Lieven (2004). *Convex optimization*. Stanford University, Department of Electrical Engineering: Cambridge University Press.

Cao, L. J., & Tay, Francis E. H. (2003). Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on Neural Networks*, *14*(6), 1506–1518.

Cauwenberghs, Gert, & Poggio, Tomaso (2001). Incremental and decremental support vector machine learning. In *Advances in neural information processing systems 13* (pp. 409–415). MIT Press.

Chang, Chih-Chung, & Lin, Chih-Jen (2001). LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Chang, Chih-Chung, & Lin, Chih-Jen (2002). Training $\nu$-support vector regression: Theory and algorithms. *Neural Computation*, *14*(8), 1959–1977.

Chen, Pai-Hsuen, Lin, Chih-Jen, & Schölkopf, Bernhard (2005). A tutorial on $\nu$-support vector machines. *Applied Stochastic Models in Business and Industry*, *21*(2), 111–136.

Diehl, Christopher P., & Cauwenberghs, Gert (2003). Svm incremental learning, adaptation and optimization. In *Proceedings of the 2003 international joint conference on neural networks* (pp. 2685–2690).

Frank, A., & Asuncion, A. (2010). UCI machine learning repository. URL: http://archive.ics.uci.edu/ml.

Friedman, Jerome H. (1991). Multivariate adaptive regression splines (with discussion). *The Annals of Statistics*, *19*(1), 1–141.

Gâlmeanu, Honorius, & Andonie, Răzvan (2008). Implementation issues of an incremental and decremental svm. In *Proceedings of the 18th international conference on artificial neural networks, part I* (pp. 325–335). Berlin, Heidelberg: Springer-Verlag.

Gretton, Arthur, & Desobry, Frédéric (2003). On-line one-class support vector machines. An application to signal segmentation. In *Proceedings of the 2003 IEEE international conference on acoustics, speech, and signal processing*. Vol. 2 (pp. 709–712).

Gu, B., & Sheng, V. S. (2013). Feasibility and finite convergence analysis for accurate on-line $\nu$-support vector machine. *IEEE Transactions on Neural Networks and Learning Systems*, *24*(8), 1304–1315.

Gu, B., Wang, J. D., & Chen, H. Y. (2008). On-line off-line ranking support vector machine and analysis. In *Proceedings of international joint conference on neural networks* (pp. 1365–1370). IEEE Press.

Gu, Bin, Wang, Jian-Dong, Yu, Yue-Cheng, Zheng, Guan-Sheng, Huang, Yu-Fan, & Xu, Tao (2012). Accurate on-line $\nu$-support vector learning. *Neural Networks*, *27*(0), 51–59.

Gunter, Lacey, & Zhu, Ji (2007). Efficient computation and model selection for the support vector regression. *Neural Computation*, *19*(6), 1633–1655.

Hastie, Trevor, Rosset, Saharon, Tibshirani, Robert, & Zhu, Ji. (2004). The entire regularization path for the support vector machine. *Journal of Machine Learning Research*, 1391–1415.

Ho, Chia-Hua, & Lin, Chih-Jen (2012). Large-scale linear support vector regression. *The Journal of Machine Learning Research*, *13*(1), 3323–3348.

Householder, A. S. (1974). *The theory of matrices in numerical analysis*. New York: Dover.

Karasuyama, Masayuki, & Takeuchi, Ichiro (2010). Multiple incremental decremental learning of support vector machines. *IEEE Transactions on Neural Networks*, *21*(7), 1048–1059.

Karush, W. (1939). *Minima of functions of several variables with inequalities as side constraints*. (M.Sc. dissertation). Chicago, Illinois: Dept. of Mathematics, Univ. of Chicago.

Laskov, Pavel, Gehl, Christian, Krüger, Stefan, robert Müller, Klaus, Bennett, Kristin, & Parrado-hern, Emilio (2006). Incremental support vector learning: Analysis, implementation and applications. *Journal of Machine Learning Research*, *7*, 1909–1936.

Lu, Chi-Jie, Lee, Tian-Shyug, & Chiu, Chih-Chou (2009). Financial time series forecasting using independent component analysis and support vector regression. *Decision Support Systems*, [ISSN: 0167-9236] *47*(2), 115–125.

Ma, Junshui, Theiler, James, & Perkins, Simon (2003). Accurate on-line support vector regression. *Neural Computation*, [ISSN: 0899-7667] *15*(11), 2683–2703.

Martin, Mario (2002). On-line support vector machine regression. In *Proceedings of the 13th European conference on machine learning* (pp. 282–294). London, UK: Springer-Verlag.

Murata, Noboru (1998). A statistical study of on-line learning (pp. 63–92).

Schölkopf, Bernhard, & Smola, Alexander J. (2001). *Learning with kernels: Support vector machines, regularization, optimization, and beyond*. Cambridge, MA, USA: MIT Press.

Schölkopf, Bernhard, Smola, Alex J., Williamson, Robert C., & Bartlett, Peter L. (2000). New support vector algorithms. *Neural Computation*, *12*(5), 1207–1245.

Shalev-Shwartz, Shai, Singer, Yoram, Srebro, Nathan, & Cotter, Andrew (2011). Pegasos: Primal estimated sub-gradient solver for svm. *Mathematical Programming*, *127*(1), 3–30.

Smola, Alex J., & Schölkopf, Bernhard (2003). *A tutorial on support vector regression. Technical report, statistics and computing*.

Tsang, Ivor W., Kwok, J. T.-Y., & Zurada, Jacek M. (2006). Generalized core vector machines. *IEEE Transactions on Neural Networks*, *17*(5), 1126–1140.

Vapnik, V. (1998). *Statistical learning theory*. New York, NY: John Wiley and Sons, Inc.

Wang, Po-Wei, & Lin, Chih-Jen (2014). Iteration complexity of feasible descent methods for convex optimization. *Journal of Machine Learning Research*, *15*, 1523–1548. URL: http://jmlr.org/papers/v15/wang14a.html.

Wang, Gang, Yeung, Dit-Yan, & Lochovsky, Frederick H. (2008). A new solution path algorithm in support vector regression. *IEEE Transactions on Neural Networks*, *19*(10), 1753–1767.

Wen, Xuezhi, Shao, Ling, Fang, Wei, & Xue, Yu (2015). Efficient feature selection and classification for vehicle detection. *IEEE Transactions on Circuits and Systems for Video Technology*, *25*(3), 508–517.

Weston, Jason (1999). Leave-one-out support vector machines. In *IJCAI* (pp. 727–733).

Zhang, Hui, Wu, Q. M. Jonathan, Nguyen, Thanh Minh, & Sun, Xingmin (2014). Synthetic aperture radar image segmentation by modified student's t-mixture model. *IEEE Transaction on Geoscience and Remote Sensing*, *52*(7), 4391–4403.