Manuscript Number: 2403

Training ν-Support Vector Regression: Theory and Algorithms

Chih-Chung Chang and Chih-Jen Lin
Department of Computer Science and
Information Engineering
National Taiwan University
Taipei 106, Taiwan (cjlin@csie.ntu.edu.tw)

Abstract

We discuss the relation between ϵ-Support Vector Regression (ϵ-SVR) and ν-Support Vector Regression (ν-SVR). In particular we focus on properties which are different from those of C-Support Vector Classification (C-SVC) and ν-Support Vector Classification (ν-SVC). We then discuss some issues which do not occur in the case of classification: the possible range of ϵ and the scaling of target values. A practical decomposition method for ν-SVR is implemented and computational experiments are conducted. We show some interesting numerical observations specific to regression.

1 Introduction

The ν-support vector machine (Schölkopf et al. 2000; Schölkopf et al. 1999) is a new class of support vector machines (SVM). It can handle both classification and regression. Properties on training ν-support vector classifiers (ν-SVC) have been discussed in (Chang and Lin 2001b). In this paper we focus on ν-support vector regression (ν-SVR). Given a set of data points, $\{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_l, y_l)\}$, such that $\mathbf{x}_i \in R^n$ is an input and $y_i \in R^1$ is a target output, the primal problem of ν-SVR is as follows:

$$(P_\nu) \qquad \min \frac{1}{2}\mathbf{w}^T\mathbf{w} + C(\nu\epsilon + \frac{1}{l}\sum_{i=1}^{l}(\xi_i + \xi_i^*)) \qquad (1.1)$$

$$(\mathbf{w}^T\phi(\mathbf{x}_i) + b) - y_i \le \epsilon + \xi_i,$$

$$y_i - (\mathbf{w}^T\phi(\mathbf{x}_i) + b) \le \epsilon + \xi_i^*,$$

$$\xi_i, \xi_i^* \ge 0, i = 1, \ldots, l,\ \epsilon \ge 0.$$

Here $0 \leq \nu \leq 1$, C is the regularization parameter, and training vectors $\mathbf{x}_i$ are mapped into a higher (maybe infinite) dimensional space by the function ϕ. The ϵ-insensitive loss function means that if $\mathbf{w}^T\phi(\mathbf{x})$ is in the range of $y \pm \epsilon$, no loss is considered. This formulation is different from the original ϵ-SVR (Vapnik 1998):

$$(P_\epsilon) \qquad \begin{aligned} &\min \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{C}{l}\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\ &(\mathbf{w}^T\phi(\mathbf{x}_i) + b) - y_i \leq \epsilon + \xi_i, \\ &y_i - (\mathbf{w}^T\phi(\mathbf{x}_i) + b) \leq \epsilon + \xi_i^*, \\ &\xi_i, \xi_i^* \geq 0, i = 1, \ldots, l. \end{aligned} \tag{1.2}$$

As it is difficult to select an appropriate ϵ, Schölkopf et al. introduced a new parameter ν which lets one control the number of support vectors and training errors. To be more precise, they proved that ν is an upper bound on the fraction of margin errors and a lower bound of the fraction of support vectors. In addition, with probability 1, asymptotically, ν equals to both fractions.

Then there are two different dual formulations for (P_ν) and (P_ϵ):

$$\begin{aligned} &\min \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)^T\mathbf{Q}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \mathbf{y}^T(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) \\ &\mathbf{e}^T(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) = 0,\ \mathbf{e}^T(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*) \leq C\nu, \\ &0 \leq \alpha_i, \alpha_i^* \leq C/l, \qquad i = 1, \ldots, l. \end{aligned} \tag{1.3}$$

$$\begin{aligned} &\min \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)^T\mathbf{Q}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \mathbf{y}^T(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + \epsilon\mathbf{e}^T(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*) \\ &\mathbf{e}^T(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) = 0, \\ &0 \leq \alpha_i, \alpha_i^* \leq C/l, \qquad i = 1, \ldots, l, \end{aligned} \tag{1.4}$$

where $Q_{ij} \equiv \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ is the kernel and $\mathbf{e}$ is the vector of all ones. Then the approximating function is

$$f(\mathbf{x}) = \sum_{i=1}^{l}(\alpha_i^* - \alpha_i)\phi(\mathbf{x}_i)^T\phi(\mathbf{x}) + b.$$

For regression, the parameter ν replaces ϵ while in the case of classification, ν replaces C. In (Chang and Lin 2001b) we have discussed the relation between ν-SVC and C-SVC as well as how to solve ν-SVC in detail. Here we are interested in different properties for regression. For example, the relation between ν-SVR and ϵ-SVR is not the same as that between ν-SVC and C-SVC. In addition, similar to the situation of C-SVC, we make sure that the inequality $\mathbf{e}^T(\boldsymbol{\alpha}+\boldsymbol{\alpha}^*) \leq C\nu$ can be replaced by an equality so algorithms for ν-SVC can be applied to ν-SVR. They will be the main topics of Sections 2 and 3.

In Section 4 we discuss the possible range of ϵ and show that it might be easier to use ν-SVM. We also demonstrate some situations where the scaling of the target values $\mathbf{y}$ is needed. Note that these issues do not occur for classification. Finally Section 5 presents computational experiments. We discuss some interesting numerical observations which are specific to support vector regression.

2 The Relation Between ν-SVR and ϵ-SVR

In this section we will derive a relationship between the solution set of ϵ-SVR and ν-SVR which allows us to conclude that the inequality constraint (1.3) can be replaced by an equality.

In the dual formulations mentioned earlier, $\mathbf{e}^T(\boldsymbol{\alpha}+\boldsymbol{\alpha}^*)$ is related to $C\nu$. Similar to (Chang and Lin 2001b), we scale them to the following formulations so $\mathbf{e}^T(\boldsymbol{\alpha}+\boldsymbol{\alpha}^*)$ is related to ν:

$$(D_\nu) \qquad \begin{aligned} &\min \frac{1}{2}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)^T\mathbf{Q}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*) + (\mathbf{y}/C)^T(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*) \\ &\mathbf{e}^T(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*) = 0,\ \mathbf{e}^T(\boldsymbol{\alpha}+\boldsymbol{\alpha}^*) \leq \nu, \\ &0 \leq \alpha_i, \alpha_i^* \leq 1/l, \qquad i = 1, \ldots, l. \end{aligned} \tag{2.1}$$

$$(D_\epsilon) \qquad \begin{aligned} &\min \frac{1}{2}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*)^T\mathbf{Q}(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*) + (\mathbf{y}/C)^T(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*) + (\epsilon/C)\mathbf{e}^T(\boldsymbol{\alpha}+\boldsymbol{\alpha}^*) \\ &\mathbf{e}^T(\boldsymbol{\alpha}-\boldsymbol{\alpha}^*) = 0, \\ &0 \leq \alpha_i, \alpha_i^* \leq 1/l, \qquad i = 1, \ldots, l. \end{aligned}$$

For convenience, following (Schölkopf et al. 2000), we represent $\begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\alpha}^* \end{bmatrix}$ as $\boldsymbol{\alpha}^{(*)}$.

Remember that for ν-SVC, not all $0 \leq \nu \leq 1$ lead to meaningful problems of (D_ν). Here the situation is similar so in the following we define a ν^* which will be the upper bound of the interesting interval of ν.

Definition 1 *Define* $\nu^* \equiv \min_{\boldsymbol{\alpha}^{(*)}} \mathbf{e}^T(\boldsymbol{\alpha}+\boldsymbol{\alpha}^*)$*, where* $\boldsymbol{\alpha}^{(*)}$ *is any optimal solution of* $(D_\epsilon), \epsilon = 0$.

Note that $0 \leq \alpha_i, \alpha_i^* \leq 1/l$ implies that the optimal solution set of (D_ϵ) or (D_ν) is bounded. As their objective and constraint functions are all continuous, any limit point of a sequence in the optimal solution set is in it as well. Hence we have that the optimal solution set of (D_ϵ) or (D_ν) is close and bounded (i.e. compact). Using this property, if $\epsilon = 0$, there is at least one optimal solution which satisfies $\mathbf{e}^T(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*) = \nu^*$.

The following lemma shows that for $(D_\epsilon), \epsilon > 0$, at an optimal solution one of α_i and α_i^* must be zero:

Lemma 1 *If* $\epsilon > 0$*, all optimal solutions of* (D_ϵ) *satisfy* $\alpha_i\alpha_i^* = 0$.

Proof. If the result is wrong, then we can reduce values of two nonzero α_i and α_i^* such that $\boldsymbol{\alpha}-\boldsymbol{\alpha}^*$ is still the same but the term $\mathbf{e}^T(\boldsymbol{\alpha}+\boldsymbol{\alpha}^*)$ of the objective function is decreased. Hence $\boldsymbol{\alpha}^{(*)}$ is not an optimal solution so there is a contradiction. □

The following lemma is similar to (Chang and Lin 2001b, Lemma 4):

Lemma 2 *If* $\boldsymbol{\alpha}_1^{(*)}$ *is any optimal solution of* (D_{ϵ_1})*,* $\boldsymbol{\alpha}_2^{(*)}$ *is any optimal solution of* (D_{ϵ_2})*, and* $0 \leq \epsilon_1 < \epsilon_2$*, then*

$$\mathbf{e}^T(\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_1^*) \geq \mathbf{e}^T(\boldsymbol{\alpha}_2 + \boldsymbol{\alpha}_2^*). \tag{2.2}$$

Therefore, for any optimal solution $\boldsymbol{\alpha}_\epsilon^{(*)}$ *of* $(D_\epsilon), \epsilon > 0$*,* $\mathbf{e}^T(\boldsymbol{\alpha}_\epsilon + \boldsymbol{\alpha}_\epsilon^*) \leq \nu^*$.

Unlike the case of classification where $\mathbf{e}^T\boldsymbol{\alpha}_C$ is a well-defined function of C if $\boldsymbol{\alpha}_C$ is an optimal solution of the C-SVC problem, here for ϵ-SVR, for the same (D_ϵ) there may have different $\mathbf{e}^T(\boldsymbol{\alpha}_\epsilon + \boldsymbol{\alpha}_\epsilon^*)$. The main reason is that $\mathbf{e}^T\boldsymbol{\alpha}$ is the only linear term of the objective function of C-SVC but for (D_ϵ), the linear term becomes $(\mathbf{y}/C)^T(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + (\epsilon/C)\mathbf{e}^T(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*)$. We will elaborate more on this in Lemma 4 where we prove that $\mathbf{e}^T(\boldsymbol{\alpha}_\epsilon+\boldsymbol{\alpha}_\epsilon^*)$ can be a function of ϵ if $\mathbf{Q}$ is a positive definite matrix.

The following lemma shows the relation between (D_ν) and (D_ϵ). In particular, we show that for $0 \leq \nu < \nu^*$, any optimal solution of (D_ν) satisfies $\mathbf{e}^T(\boldsymbol{\alpha}+\boldsymbol{\alpha}^*) = \nu$.

Lemma 3 *For any $(D_\nu), 0 \leq \nu < \nu^*$, one of the following two situations must happen:*

1. *(D_ν)'s optimal solution set is part of the solution set of an $(D_\epsilon), \epsilon > 0$,*
2. *(D_ν)'s optimal solution set is the same as that of (D_ϵ), where $\epsilon > 0$ is any one element in a unique open interval.*

In addition, any optimal solution of (D_ν) satisfies $\mathbf{e}^T(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*) = \nu$ and $\alpha_i \alpha_i^* = 0$.

Proof. The Karush-Kuhn-Tucker (KKT) condition of (D_ν) shows that there exist $\rho \geq 0$ and b such that

$$\begin{bmatrix} \mathbf{Q} & -\mathbf{Q} \\ -\mathbf{Q} & \mathbf{Q} \end{bmatrix} \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\alpha}^* \end{bmatrix} + \begin{bmatrix} \mathbf{y}/C \\ -\mathbf{y}/C \end{bmatrix} + \frac{\rho}{C} \begin{bmatrix} \mathbf{e} \\ \mathbf{e} \end{bmatrix} - b \begin{bmatrix} \mathbf{e} \\ -\mathbf{e} \end{bmatrix} = \begin{bmatrix} \boldsymbol{\lambda} - \boldsymbol{\xi} \\ \boldsymbol{\lambda}^* - \boldsymbol{\xi}^* \end{bmatrix}. \tag{2.3}$$

If $\rho = 0$, $\boldsymbol{\alpha}^{(*)}$ is an optimal solution of $(D_\epsilon), \epsilon = 0$. Then $\mathbf{e}^T(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*) \geq \nu^* > \nu$ causes a contradiction.

Therefore $\rho > 0$ so the KKT condition implies that $\mathbf{e}^T(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*) = \nu$. Then there are two possible situations:

Case 1: ρ is unique: By assigning $\epsilon = \rho$, all optimal solutions of (D_ν) are KKT points of (D_ϵ). Hence (D_ν)'s optimal solution set is part of that of a (D_ϵ). This (D_ϵ) is unique as otherwise we can find another ρ which satisfies (2.3).

Case 2: ρ is not unique. That is, there are two $\rho_1 < \rho_2$. Suppose ρ_1 and ρ_2 are the smallest and largest one satisfying the KKT. Again the existence of ρ_1 and ρ_2 are based on the compactness of the optimal solution set. Then for any $\rho_1 < \rho < \rho_2$, we consider the problem $(D_\epsilon), \epsilon = \rho$. Define $\epsilon_1 \equiv \rho_1$ and $\epsilon_2 \equiv \rho_2$. From Lemma 2, since $\epsilon_1 < \epsilon < \epsilon_2$,

$$\nu = \mathbf{e}^T(\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_1^*) \geq \mathbf{e}^T(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*) \geq \mathbf{e}^T(\boldsymbol{\alpha}_2 + \boldsymbol{\alpha}_2^*) = \nu,$$

where $\boldsymbol{\alpha}^{(*)}$ is any optimal solution of (D_ϵ), and $\boldsymbol{\alpha}_1^{(*)}$ and $\boldsymbol{\alpha}_2^{(*)}$ are optimal solutions of (D_{ϵ_1}) and (D_{ϵ_2}), respectively. Hence all optimal solutions of (D_ϵ) satisfy $\mathbf{e}^T(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*) = \nu$ and all KKT conditions of (D_ν) so (D_ϵ)'s optimal solution set is in that of (D_ν).

Hence (D_ν) and (D_ϵ) share at least one optimal solution $\boldsymbol{\alpha}^{(*)}$. For any other optimal solution $\bar{\boldsymbol{\alpha}}^{(*)}$ of (D_ν), it is feasible for (D_ϵ). Since

$$\begin{aligned} & \frac{1}{2}(\bar{\boldsymbol{\alpha}} - \bar{\boldsymbol{\alpha}}^*)^T \mathbf{Q} (\bar{\boldsymbol{\alpha}} - \bar{\boldsymbol{\alpha}}^*) + (\mathbf{y}/C)^T (\bar{\boldsymbol{\alpha}} - \bar{\boldsymbol{\alpha}}^*) \\ = \; & \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)^T \mathbf{Q} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + (\mathbf{y}/C)^T (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*), \end{aligned}$$

and $\mathbf{e}^T(\bar{\boldsymbol{\alpha}} + \bar{\boldsymbol{\alpha}}^*) \leq \nu$, we have

$$\begin{aligned} & \frac{1}{2}(\bar{\boldsymbol{\alpha}} - \bar{\boldsymbol{\alpha}}^*)^T \mathbf{Q} (\bar{\boldsymbol{\alpha}} - \bar{\boldsymbol{\alpha}}^*) + (\mathbf{y}/C)^T (\bar{\boldsymbol{\alpha}} - \bar{\boldsymbol{\alpha}}^*) + (\epsilon/C)\mathbf{e}^T(\bar{\boldsymbol{\alpha}} + \bar{\boldsymbol{\alpha}}^*) \\ \leq \; & \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)^T \mathbf{Q} (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + (\mathbf{y}/C)^T (\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + (\epsilon/C)\mathbf{e}^T(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*). \end{aligned}$$

Therefore, all optimal solutions of (D_ν) are also optimal for (D_ϵ). Hence (D_ν)'s optimal solution set is the same as that of (D_ϵ), where $\epsilon > 0$ is any one element in a unique open interval (ρ_1, ρ_2).

Finally, as (D_ν)'s optimal solution set is the same or part of a (D_ϵ), from Lemma 1, we have $\alpha_i \alpha_i^* = 0$. □

Using the above results, we now summarize a main theorem:

Theorem 1 *We have*

1. $\nu^* \leq 1$.
2. *For any* $\nu \in [\nu^*, 1]$, (D_ν) *has the same optimal objective value as* $(D_\epsilon), \epsilon = 0$.
3. *For any* $\nu \in [0, \nu^*)$, *Lemma 3 holds. That is, one of the following two situations must happen:*
   (a) (D_ν)*'s optimal solution set is part of the solution set of an* $(D_\epsilon), \epsilon > 0$,
   (b) (D_ν)*'s optimal solution set is the same as that of* (D_ϵ), *where* $\epsilon > 0$ *is any one element in a unique open interval.*
4. *For all* $(D_\nu), 0 \leq \nu \leq 1$, *there are always optimal solutions which happen at the equality* $\mathbf{e}^T(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*) = \nu$.

Proof.

From the explanation after Definition 1, there exists an optimal solution of $(D_\epsilon), \epsilon = 0$ which satisfies $\mathbf{e}^T(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*) = \nu^*$. Then this $\boldsymbol{\alpha}^{(*)}$ must satisfy $\alpha_i \alpha_i^* =$

$0, i = 1, \ldots, l$, so $\nu^* \leq 1$. In addition, this $\boldsymbol{\alpha}^{(*)}$ is also feasible to (D_ν), $\nu \geq \nu^*$. Since (D_ν) has the same objective function as $(D_\epsilon), \epsilon = 0$ but has one more constraint, this solution of $(D_\epsilon), \epsilon = 0$, is also optimal for (D_ν). Hence (D_ν) and (D_{ν^*}) have the same optimal objective value.

For $0 \leq \nu < \nu^*$, we already know from Theorem 3 that the optimal solution happens only at the equality. For $1 \geq \nu \geq \nu^*$, first we know that $(D_\epsilon), \epsilon = 0$ has an optimal solution $\boldsymbol{\alpha}^{(*)}$ which satisfies $\mathbf{e}^T(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*) = \nu^*$. Then we can increase some elements of $\boldsymbol{\alpha}$ and $\boldsymbol{\alpha}^*$ such that new vectors $\hat{\boldsymbol{\alpha}}^{(*)}$ satisfy $\mathbf{e}^T(\hat{\boldsymbol{\alpha}} + \hat{\boldsymbol{\alpha}}^*) = \nu$ but $\hat{\boldsymbol{\alpha}} - \hat{\boldsymbol{\alpha}}^* = \boldsymbol{\alpha} - \boldsymbol{\alpha}^*$. Hence $\hat{\boldsymbol{\alpha}}^{(*)}$ is an optimal solution of (D_ν) which satisfies the equality constraint. □

Therefore, the above results make sure that it is safe to solve the following problem instead of (D_ν):

$$
\begin{aligned}
(\bar{D}_\nu) \qquad & \min \frac{1}{2}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*)^T \mathbf{Q}(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) + (\mathbf{y}/C)^T(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) \\
& \mathbf{e}^T(\boldsymbol{\alpha} - \boldsymbol{\alpha}^*) = 0, \ \mathbf{e}^T(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*) = \nu, \\
& 0 \leq \alpha_i, \alpha_i^* \leq 1/l, \qquad i = 1, \ldots, l.
\end{aligned}
$$

This is important as for existing SVM algorithms, it is easier to handle equalities than inequalities.

Note that for ν-SVC, there is also a ν^* where for $\nu \in (\nu^*, 1]$, (D_ν) is infeasible. At that time $\nu^* = 2\min(\#\text{positive data}, \#\text{negative data})/l$ can be easily calculated (Crisp and Burges 2000). Now for ν-SVR, it is difficult to know ν^* in priori. However, we do not have to worry about this. If a $(\bar{D}_\nu), \nu > \nu^*$ is solved, a solution with its objective value equal to that of $(D_\epsilon), \epsilon = 0$ is obtained. Then some α_i and α_i^* may both be nonzero.

Since there are always optimal solutions of the dual problem which satisfy $\mathbf{e}^T(\boldsymbol{\alpha}+\boldsymbol{\alpha}^*) = \nu$, this also implies that the $\epsilon \geq 0$ constraint of (P_ν) is not necessary. In the following theorem, we derive the same result directly from the primal side:

Theorem 2 *Consider a problem which is the same as (P_ν) but without the inequality constraint $\epsilon \geq 0$. We have that for any $0 < \nu < 1$, any optimal solution of (P_ν) must satisfy $\epsilon \geq 0$.*

Proof.

Assume $(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*, \epsilon)$ is an optimal solution with $\epsilon < 0$. Then for each i,

$$-\epsilon - \xi_i^* \leq \mathbf{w}^T\phi(\mathbf{x}_i) + b - y_i \leq \epsilon + \xi_i \tag{2.4}$$

implies

$$\xi_i + \xi_i^* + 2\epsilon \geq 0. \tag{2.5}$$

With (2.4),

$$\begin{aligned} -0 - \max(0, \xi_i^* + \epsilon) \leq -\epsilon - \xi_i^* &\leq \mathbf{w}^T\phi(\mathbf{x}_i) + b - y_i \\ &\leq 0 + \epsilon + \xi_i \leq 0 + \max(0, \xi_i + \epsilon). \end{aligned}$$

Hence $(\mathbf{w}, b, \max(0, \boldsymbol{\xi} + \epsilon\mathbf{e}), \max(0, \boldsymbol{\xi}^* + \epsilon\mathbf{e}), 0)$ is a feasible solution of (P_ν). From (2.5),

$$\max(0, \xi_i + \epsilon) + \max(0, \xi_i^* + \epsilon) \leq \xi_i + \xi_i^* + \epsilon.$$

Therefore, with $\epsilon < 0$ and $0 < \nu < 1$,

$$\begin{aligned} & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C(\nu\epsilon + \frac{1}{l}\sum_{i=1}^{l}(\xi_i + \xi_i^*)) \\ > \; & \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{C}{l}(l\epsilon + \sum_{i=1}^{l}(\xi_i + \xi_i^*)) \\ \geq \; & \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{C}{l}\sum_{i=1}^{l}(\max(0, \xi_i + \epsilon) + \max(0, \xi_i^* + \epsilon)) \end{aligned}$$

implies that $(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*)$ is not an optimal solution. Therefore, any optimal solution of (P_ν) must satisfy $\epsilon \geq 0$. □

Next we demonstrate an example where one (D_ϵ) corresponds to many (D_ν). Given two training points $\mathbf{x}_1 = 0, \mathbf{x}_2 = 0$, and target values $y_1 = -\Delta < 0$ and $y_2 = \Delta > 0$. When $\epsilon = \Delta$, if the linear kernel is used and $C = 1$, (D_ϵ) becomes

$$\begin{aligned} \min \quad & 2\Delta(\alpha_1^* + \alpha_2) \\ & 0 \leq \alpha_1, \alpha_1^*, \alpha_2, \alpha_2^* \leq 1/l, \\ & \alpha_1 - \alpha_1^* + \alpha_2 - \alpha_2^* = 0. \end{aligned}$$

Thus $\alpha_1^* = \alpha_2 = 0$ so any $0 \leq \alpha_1 = \alpha_2^* \leq 1/l$ is an optimal solution. Therefore, for this ϵ, the possible $\mathbf{e}^T(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*)$ ranges from 0 to 1. The relation between ν and ϵ is illustrated in Figure 1.

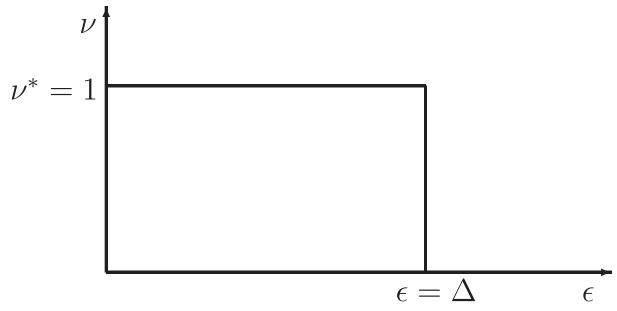


Figure 1: An example where one (D_ϵ) corresponds to different (D_ν)

3 When the Kernel Matrix Q is Positive Definite

In the previous section we have shown that for ϵ-SVR, $\mathbf{e}^T(\boldsymbol{\alpha}_\epsilon + \boldsymbol{\alpha}_\epsilon^*)$ may not be a well-defined function of ϵ, where $\boldsymbol{\alpha}_\epsilon^{(*)}$ is any optimal solution of (D_ϵ). Because of this difficulty, we cannot exactly apply results on the relation between C-SVC and ν-SVC to ϵ-SVR and ν-SVR. In this section we show that if $\mathbf{Q}$ is positive definite, then $\mathbf{e}^T(\boldsymbol{\alpha}_\epsilon + \boldsymbol{\alpha}_\epsilon^*)$ is a function of ϵ and all results discussed in (Chang and Lin 2001b) hold.

Assumption 1 $\mathbf{Q}$ *is positive definite.*

Lemma 4 *If $\epsilon > 0$, then (D_ϵ) has a unique optimal solution. Therefore, we can define a function $\mathbf{e}^T(\boldsymbol{\alpha}_\epsilon + \boldsymbol{\alpha}_\epsilon^*)$ on ϵ, where $\boldsymbol{\alpha}_\epsilon^{(*)}$ is the optimal solution of (D_ϵ).*

Proof. Since (D_ϵ) is a convex problem, if $\boldsymbol{\alpha}_1^{(*)}$ and $\boldsymbol{\alpha}_2^{(*)}$ are both optimal solutions, for all $0 \leq \lambda \leq 1$,

$$\begin{aligned}
& \frac{1}{2}(\lambda(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_1^*) + (1-\lambda)(\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_2^*))^T \mathbf{Q}(\lambda(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_1^*) + (1-\lambda)(\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_2^*)) \\
& +(\mathbf{y}/C)^T(\lambda(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_1^*) + (1-\lambda)(\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_2^*)) \\
& +(\epsilon/C)\mathbf{e}^T(\lambda(\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_1^*) + (1-\lambda)(\boldsymbol{\alpha}_2 + \boldsymbol{\alpha}_2^*)) \\
= \; & \lambda\Big(\frac{1}{2}(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_1^*)^T\mathbf{Q}(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_1^*) + (\mathbf{y}/C)^T(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_1^*) + (\epsilon/C)\mathbf{e}^T(\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_1^*)\Big) \\
& +(1-\lambda)\Big(\frac{1}{2}(\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_2^*)^T\mathbf{Q}(\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_2^*) + (\mathbf{y}/C)^T(\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_2^*) + (\epsilon/C)\mathbf{e}^T(\boldsymbol{\alpha}_2 + \boldsymbol{\alpha}_2^*)\Big).
\end{aligned}$$

This implies

$$(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_1^*)^T\mathbf{Q}(\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_2^*) = \frac{1}{2}(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_1^*)^T\mathbf{Q}(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_1^*) + \frac{1}{2}(\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_2^*)^T\mathbf{Q}(\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_2^*). \quad (3.1)$$

Since $\mathbf{Q}$ is positive semidefinite, $\mathbf{Q} = \mathbf{L}^T\mathbf{L}$ so (3.1) implies $\|\mathbf{L}(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_1^*) - \mathbf{L}(\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_2^*)\| = 0$. Hence $\mathbf{L}(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_1^*) = \mathbf{L}(\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_2^*)$. Since $\mathbf{Q}$ is positive definite, $\mathbf{L}$ is invertible so $\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_1^* = \boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_2^*$. Since $\epsilon > 0$, from Lemma 1, $(\alpha_1)_i(\alpha_1^*)_i = 0$ and $(\alpha_2)_i(\alpha_2^*)_i = 0$. Thus we have $\boldsymbol{\alpha}_1 = \boldsymbol{\alpha}_2$ and $\boldsymbol{\alpha}_1^* = \boldsymbol{\alpha}_2^*$. $\square$

For convex optimization problems, if the Hessian is positive definite, there is a unique optimal solution. Unfortunately here the Hessian is $\begin{bmatrix} \mathbf{Q} & -\mathbf{Q} \\ -\mathbf{Q} & \mathbf{Q} \end{bmatrix}$ which is only positive semi-definite. Hence special efforts are needed for proving the property of the unique optimal solution.

Note that in the above proof, $\mathbf{L}(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_1^*) = \mathbf{L}(\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_2^*)$ implies $(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_1^*)^T\mathbf{Q}(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_1^*) = (\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_2^*)^T\mathbf{Q}(\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_2^*)$. Since $\boldsymbol{\alpha}_1^{(*)}$ and $\boldsymbol{\alpha}_2^{(*)}$ are both optimal solutions, they have the same objective value so $-\mathbf{y}^T(\boldsymbol{\alpha}_1 - \boldsymbol{\alpha}_1^*) + \epsilon\mathbf{e}^T(\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_1^*) = -\mathbf{y}^T(\boldsymbol{\alpha}_2 - \boldsymbol{\alpha}_2^*) + \epsilon\mathbf{e}^T(\boldsymbol{\alpha}_2 + \boldsymbol{\alpha}_2^*)$. This is not enough for proving that $\mathbf{e}^T(\boldsymbol{\alpha}_1 + \boldsymbol{\alpha}_1^*) = \mathbf{e}^T(\boldsymbol{\alpha}_2 + \boldsymbol{\alpha}_2^*)$. On the contrary, for ν-SVC, the objective function is $\frac{1}{2}\boldsymbol{\alpha}^T\mathbf{Q}\boldsymbol{\alpha} - \mathbf{e}^T\boldsymbol{\alpha}$ so without the positive definite assumption, $\boldsymbol{\alpha}_1^T\mathbf{Q}\boldsymbol{\alpha}_1 = \boldsymbol{\alpha}_2^T\mathbf{Q}\boldsymbol{\alpha}_2$ already implies $\mathbf{e}^T\boldsymbol{\alpha}_1 = \mathbf{e}^T\boldsymbol{\alpha}_2$. Thus $\mathbf{e}^T\boldsymbol{\alpha}_C$ is a function of C.

We then state some parallel results in (Chang and Lin 2001b) without proofs:

Theorem 3 *If Q is positive definite, then the relation between (D_ν) and (D_ϵ) is summarized as follows:*

1. *(a) For any $1 \geq \nu \geq \nu^*$, (D_ν) has the same optimal objective value as $(D_\epsilon), \epsilon = 0$.*

   (b) For any $\nu \in [0, \nu^*)$, (D_ν) has a unique solution which is the same as that of either one (D_ϵ), $\epsilon > 0$, or some (D_ϵ), where ϵ is any number in an interval.

2. *If α_ϵ^* is the optimal solution of $(D_\epsilon), \epsilon > 0$, the relation between ν and ϵ is as follows:*

   There are $0 < \epsilon_1 < \cdots < \epsilon_s$ and $A_i, B_i, i = 1, \ldots, s$ such that

$$\mathbf{e}^T(\boldsymbol{\alpha}_\epsilon + \boldsymbol{\alpha}_\epsilon^*) = \begin{cases} \nu^* & 0 < \epsilon \leq \epsilon_1, \\ A_i + B_i\epsilon & \epsilon_i \leq \epsilon \leq \epsilon_{i+1}, i = 1, \ldots, s-1, \\ 0 & \epsilon_s \leq \epsilon, \end{cases}$$

where $\boldsymbol{\alpha}_\epsilon^{(*)}$ *is the optimal solution of* (D_ϵ)*. We also have*

$$A_i + B_i\epsilon_{i+1} = A_{i+1} + B_{i+1}\epsilon_{i+1}, i = 1, \ldots, s-2, \tag{3.2}$$

and

$$A_{s-1} + B_{s-1}\epsilon_s = 0.$$

In addition, $B_i \leq 0, i = 1, \ldots, s-1$.

The second result of Theorem 3 shows that ν is a piece-wise linear function of ϵ. In addition, it is always decreasing.

4 Some Issues Specific to Regression

The motivation of ν-SVR is that it may not be easy to decide the parameter ϵ. Hence here we are interested in the possible range of ϵ. As expected, results show that ϵ is related to the target values $\mathbf{y}$.

Theorem 4 *The zero vector is an optimal solution of* (D_ϵ) *if and only if*

$$\epsilon \geq \frac{\max_{i=1,\ldots,l} y_i - \min_{i=1,\ldots,l} y_i}{2}. \tag{4.1}$$

Proof. If the zero vector is an optimal solution of (D_ϵ), the KKT condition implies that there is a b such that

$$\begin{bmatrix} \mathbf{y} \\ -\mathbf{y} \end{bmatrix} + \epsilon \begin{bmatrix} \mathbf{e} \\ \mathbf{e} \end{bmatrix} - b \begin{bmatrix} \mathbf{e} \\ -\mathbf{e} \end{bmatrix} \geq 0.$$

Hence $\epsilon - b \geq -y_i$ and $\epsilon + b \geq y_i$, for all i. Therefore,

$$\epsilon - b \geq -\min_{i=1,\ldots,l} y_i \text{ and } \epsilon + b \geq \max_{i=1,\ldots,l} y_i$$

so

$$\epsilon \geq \frac{\max_{i=1,\ldots,l} y_i - \min_{i=1,\ldots,l} y_i}{2}.$$

On the other hand, if (4.1) is true, we can easily check that $\boldsymbol{\alpha} = \boldsymbol{\alpha}^* = 0$ satisfy the KKT condition so the zero vector is an optimal solution of (D_ϵ). □

Therefore, when using ϵ-SVR, the largest value of ϵ to try is $(\max_{i=1,\ldots,l} y_i - \min_{i=1,\ldots,l} y_i)/2$.

On the other hand, ϵ should not be too small as if $\epsilon \to 0$, most data are support vectors and overfitting tends to happen. Unfortunately we have not been able to find an effective lower bound on ϵ. However, intuitively we would think that it is also related to the target values $\mathbf{y}$.

As the effective range of ϵ is affected by the target values y, a way to solve this difficulty for ϵ-SVM is by scaling the target valves before training the data. For example, if all target values are scaled to $[-1, +1]$, then the effective range of ϵ will be $[0, 1]$, the same as that of ν. Then it may be easier to choose ϵ.

There are other reasons to scale the target values. For example, we encountered some situations where if the target values $\mathbf{y}$ are not properly scaled, it is difficult to adjust the value of C. In particular, if $y_i, i = 1, \ldots, l$ are large numbers and C is chosen to be a small number, the approximating function is nearly a constant.

5 Algorithms

The algorithm will be considered here for ν-SVR is similar to the decomposition method in (Chang and Lin 2001b) for ν-SVC. The implementation is part of the software LIBSVM (Chang and Lin 2001a). Another SVM software which has also implemented ν-SVR is mySVM (Rüping 2000).

The basic idea of the decomposition method is that in each iteration, the indices $\{1, \ldots, l\}$ of the training set are separated to two sets B and N, where B is the working set and $N = \{1, \ldots, l\} \backslash B$. The vector $\boldsymbol{\alpha}_N$ is fixed and then a sub-problem with the variable $\boldsymbol{\alpha}_B$ is solved.

The decomposition method was first proposed for SVM classification (Osuna et al. 1997; Joachims 1998; Platt 1998). Extensions to ϵ-SVR are in, for example, (Keerthi et al. 2000; Laskov 2002). The main difference on these methods is their working set selections which may significantly affect the number of iterations. Due to the additional equality (1.3) in the ν-SVM, more considerations on the working set selection are needed. Discussions for classification are in (Keerthi and Gilbert 2002; Chang and Lin 2001b).

For the consistency with other SVM formulations in LIBSVM, we consider (D_ν)

as the following scaled form:

$$\begin{aligned} \min_{\bar{\boldsymbol{\alpha}}} \quad & \frac{1}{2}\bar{\boldsymbol{\alpha}}^T\bar{\mathbf{Q}}\bar{\boldsymbol{\alpha}} + \bar{\mathbf{p}}^T\bar{\boldsymbol{\alpha}} \\ & \bar{\mathbf{y}}^T\bar{\boldsymbol{\alpha}} = \Delta_1, \\ & \bar{\mathbf{e}}^T\bar{\boldsymbol{\alpha}} = \Delta_2, \\ & 0 \le \bar{\alpha}_t \le C, t = 1, \ldots, 2l, \end{aligned} \tag{5.1}$$

where

$$\bar{\mathbf{Q}} = \begin{bmatrix} \mathbf{Q} & -\mathbf{Q} \\ -\mathbf{Q} & \mathbf{Q} \end{bmatrix}, \bar{\boldsymbol{\alpha}} = \begin{bmatrix} \boldsymbol{\alpha} \\ \boldsymbol{\alpha}^* \end{bmatrix}, \bar{\mathbf{p}} = \begin{bmatrix} \mathbf{y} \\ \mathbf{y} \end{bmatrix}, \bar{\mathbf{y}} = \begin{bmatrix} \mathbf{e} \\ -\mathbf{e} \end{bmatrix}, \bar{\mathbf{e}} = \begin{bmatrix} \mathbf{e} \\ \mathbf{e} \end{bmatrix}, \Delta_1 = 0, \Delta_2 = Cl\nu.$$

That is, we replace C/l by C. Note that because of the result in Theorem 1, we are safe to use an equality constraint here in (5.1).

Then the sub-problem is as follows:

$$\begin{aligned} \min_{\bar{\boldsymbol{\alpha}}_B} \quad & \frac{1}{2}\bar{\boldsymbol{\alpha}}_B^T\bar{\mathbf{Q}}_{BB}\bar{\boldsymbol{\alpha}}_B + (\bar{\mathbf{p}}_B + \bar{\mathbf{Q}}_{BN}\bar{\boldsymbol{\alpha}}_N^k)^T\bar{\boldsymbol{\alpha}}_B \\ & \bar{\mathbf{y}}_B^T\bar{\boldsymbol{\alpha}}_B = \Delta_1 - \bar{\mathbf{y}}_N^T\bar{\boldsymbol{\alpha}}_N, \\ & \bar{\mathbf{e}}_B^T\bar{\boldsymbol{\alpha}}_B = \Delta_2 - \bar{\mathbf{e}}_N^T\bar{\boldsymbol{\alpha}}_N, \\ & 0 \le (\bar{\boldsymbol{\alpha}}_B)_t \le C, t = 1, \ldots, q, \end{aligned} \tag{5.2}$$

where q is the size of the working set.

Following the idea of Sequential Minimal Optimization (SMO) by Platt (1998), we use only two elements as the working set in each iteration. The main advantage is that an analytic solution of (5.2) can be obtained so there is no need to use an optimization software.

Our working set selection follows from (Chang and Lin 2001b) which is a modification of the selection in the software SVM^{light} (Joachims 1998). Since they dealt with the case of more general selections where the size is not restricted to two, here we have a simpler derivation directly using the KKT condition. It is similar to that in (Keerthi and Gilbert 2002, Section 5).

Now if only two elements i and j are selected but $\bar{y}_i \neq \bar{y}_j$, then $\bar{\mathbf{y}}_B^T\bar{\boldsymbol{\alpha}}_B = \Delta_1 - \bar{\mathbf{y}}_N^T\bar{\boldsymbol{\alpha}}_N$ and $\bar{\mathbf{e}}_B^T\bar{\boldsymbol{\alpha}}_B = \Delta_2 - \bar{\mathbf{e}}_N^T\bar{\boldsymbol{\alpha}}_N$ imply that there are two equations with two variables so in general (5.2) has only one feasible point. Therefore, from $\bar{\boldsymbol{\alpha}}_k$, the solution of the kth iteration, it cannot be moved any more. On the other hand, if

$\bar{y}_i = \bar{y}_j$, $\bar{\mathbf{y}}_B^T \bar{\boldsymbol{\alpha}}_B = \Delta_1 - \bar{\mathbf{y}}_N^T \bar{\boldsymbol{\alpha}}_N$ and $\bar{\mathbf{e}}_B^T \bar{\boldsymbol{\alpha}}_B = \Delta_2 - \bar{\mathbf{e}}_N^T \bar{\boldsymbol{\alpha}}_N$ become the same equality so there are multiple feasible solutions. Therefore, we have to keep $\bar{y}_i = \bar{y}_j$ while selecting the working set.

The KKT condition of (5.1) shows that there are ρ and b such that

$$\begin{aligned} \nabla f(\bar{\boldsymbol{\alpha}})_i - \rho + b\bar{y}_i &= 0 \text{ if } 0 < \bar{\alpha}_i < C, \\ &\geq 0 \text{ if } \bar{\alpha}_i = 0, \\ &\leq 0 \text{ if } \bar{\alpha}_i = C. \end{aligned}$$

Define

$$r_1 \equiv \rho - b, \; r_2 \equiv \rho + b.$$

If $\bar{y}_i = 1$, the KKT condition becomes

$$\begin{aligned} \nabla f(\bar{\boldsymbol{\alpha}})_i - r_1 &\geq 0 \text{ if } \bar{\alpha}_i < C, \\ &\leq 0 \text{ if } \bar{\alpha}_i > 0. \end{aligned} \tag{5.3}$$

On the other hand, if $\bar{y}_i = -1$, it is

$$\begin{aligned} \nabla f(\bar{\boldsymbol{\alpha}})_i - r_2 &\geq 0 \text{ if } \bar{\alpha}_i < C, \\ &\leq 0 \text{ if } \bar{\alpha}_i > 0. \end{aligned} \tag{5.4}$$

Hence, indices i and j are selected from either

$$\begin{aligned} i &= \text{argmin}_t \{\nabla f(\bar{\boldsymbol{\alpha}})_t | \bar{y}_t = 1, \bar{\alpha}_t < C\}, \\ j &= \text{argmax}_t \{\nabla f(\bar{\boldsymbol{\alpha}})_t | \bar{y}_t = 1, \bar{\alpha}_t > 0\}, \end{aligned} \tag{5.5}$$

or

$$\begin{aligned} i &= \text{argmin}_t \{\nabla f(\bar{\boldsymbol{\alpha}})_t | \bar{y}_t = -1, \bar{\alpha}_t < C\}, \\ j &= \text{argmax}_t \{\nabla f(\bar{\boldsymbol{\alpha}})_t | \bar{y}_t = -1, \bar{\alpha}_t > 0\}, \end{aligned} \tag{5.6}$$

depending on which one gives a larger $\nabla f(\bar{\boldsymbol{\alpha}})_j - \nabla f(\bar{\boldsymbol{\alpha}})_i$ (i.e. larger KKT violations). If the selected $\nabla f(\bar{\boldsymbol{\alpha}})_j - \nabla f(\bar{\boldsymbol{\alpha}})_i$ is smaller than a given ϵ (10^{-3} in our experiments), the algorithm stops.

Similar to the case of ν-SVC, here the zero vector cannot be the initial solution. This is due to the additional equality constraint $\bar{\mathbf{e}}^T \bar{\boldsymbol{\alpha}} = \Delta_2$ of (5.1). Here we assign both initial $\bar{\boldsymbol{\alpha}}$ and $\bar{\boldsymbol{\alpha}}^*$ with the same values. The first $\lceil \nu l/2 \rceil$ elements are $[C, \ldots, C, C(\nu l/2 - \lfloor \nu l/2 \rfloor)]^T$ while others are zero.

Table 1: Solving ν-SVR and ϵ-SVR: $C = 1$ (time in seconds)

Problem	l	ν	ϵ	ν Iter.	ϵ Iter.	ν Time	ϵ Time	ν^*
pyrimidines	74	0.2	0.135131	181	145	0.03	0.02	0.817868
		0.4	0.064666	175	156	0.03	0.04	
		0.6	0.028517	365	331	0.04	0.03	
		0.8	0.002164	695	460	0.05	0.05	
mpg	392	0.2	0.152014	988	862	0.19	0.16	0.961858
		0.4	0.090124	1753	1444	0.32	0.27	
		0.6	0.048543	2115	1847	0.40	0.34	
		0.8	0.020783	3046	2595	0.56	0.51	
bodyfat	252	0.2	0.012700	1112	1047	0.14	0.13	0.899957
		0.4	0.006332	2318	2117	0.25	0.23	
		0.6	0.002898	3553	2857	0.37	0.31	
		0.8	0.001088	4966	3819	0.48	0.42	
housing	506	0.2	0.161529	799	1231	0.30	0.34	0.946593
		0.4	0.089703	1693	1650	0.53	0.45	
		0.6	0.046269	1759	2002	0.63	0.60	
		0.8	0.018860	2700	2082	0.85	0.65	
triazines	186	0.2	0.380308	175	116	0.13	0.10	0.900243
		0.4	0.194967	483	325	0.18	0.15	
		0.6	0.096720	422	427	0.20	0.18	
		0.8	0.033753	532	513	0.23	0.23	
mg	1385	0.2	0.366606	1928	1542	1.58	1.18	0.992017
		0.4	0.216329	3268	3294	2.75	2.35	
		0.6	0.124792	3400	3300	3.36	2.76	
		0.8	0.059115	4516	4296	4.24	3.65	
abalone	4177	0.2	0.168812	4189	3713	15.68	11.69	0.994775
		0.4	0.094959	8257	7113	30.38	22.88	
		0.6	0.055966	12483	12984	42.74	37.41	
		0.8	0.026165	18302	18277	65.98	54.04	
space_ga	3107	0.2	0.087070	5020	4403	10.47	7.56	0.990468
		0.4	0.053287	8969	7731	18.70	14.44	
		0.6	0.032080	12261	10704	26.27	20.72	
		0.8	0.014410	16311	13852	32.71	27.19	
cpusmall	8192	0.2	0.086285	8028	7422	82.66	59.14	0.990877
		0.4	0.054095	16585	15240	203.20	120.48	
		0.6	0.031285	22376	19126	283.71	163.96	
		0.8	0.013842	28262	24840	355.59	213.25	
cadata	20640	0.2	0.294803	12153	10961	575.11	294.53	0.997099
		0.4	0.168370	24614	20968	1096.87	574.77	
		0.6	0.097434	35161	30477	1530.01	851.91	
		0.8	0.044636	42709	40652	1883.35	1142.27	

Table 2: Solving ν-SVR and ϵ-SVR: $C = 100$ (time in seconds)

Problem	l	ν	ϵ	ν Iter.	ϵ Iter.	ν Time	ϵ Time	ν^*
pyrimidines	74	0.2*	0.000554	29758	11978	0.63	0.27	0.191361
		0.4*	0.000317	30772	11724	0.65	0.27	
		0.6*	0.000240	27270	11802	0.58	0.27	
		0.8*	0.000146	20251	12014	0.44	0.28	
mpg	392	0.2	0.121366	85120	74878	9.53	8.26	0.876646
		0.4	0.069775	210710	167719	24.50	19.32	
		0.6	0.032716	347777	292426	42.08	34.82	
		0.8	0.007953	383164	332725	47.61	40.86	
bodyfat	252	0.2	0.001848	238927	164218	16.80	11.58	0.368736
		0.4*	0.000486	711157	323016	50.77	23.24	
		0.6*	0.000291	644602	339569	46.23	24.33	
		0.8*	0.000131	517370	356316	37.28	25.55	
housing	506	0.2	0.092998	154565	108220	24.21	16.87	0.815085
		0.4	0.051726	186136	182889	30.49	29.51	
		0.6	0.026340	285354	271278	48.62	45.64	
		0.8	0.002161	397115	284253	69.16	49.12	
triazines	186	0.2	0.193718	16607	22651	0.94	1.20	0.582147
		0.4	0.074474	34034	47205	1.89	2.52	
		0.6*	0.000381	106621	51175	5.69	2.84	
		0.8*	0.000139	68553	50786	3.73	2.81	
mg	1385	0.2	0.325659	190065	195519	87.99	89.20	0.966793
		0.4	0.189377	291315	299541	139.10	141.73	
		0.6	0.107324	397449	407159	194.81	196.14	
		0.8	0.043439	486656	543520	241.20	265.27	
abalone	4177	0.2	0.162593	465922	343594	797.48	588.92	0.988298
		0.4	0.091815	901275	829951	1577.83	1449.37	
		0.6	0.053244	1212669	1356556	2193.97	2506.52	
		0.8	0.024670	1680704	1632597	2970.98	2987.30	
space_ga	3107	0.2	0.078294	510035	444455	595.42	508.41	0.984568
		0.4	0.048643	846873	738805	1011.82	867.32	
		0.6	0.028933	1097732	1054464	1362.67	1268.40	
		0.8	0.013855	1374987	1393044	1778.38	1751.39	
cpusmall	8192	0.2	0.070568	977374	863579	4304.42	3606.35	0.978351
		0.4	0.041640	1783725	1652396	8291.12	7014.32	
		0.6	0.022280	2553150	2363251	11673.62	10691.95	
		0.8	0.009616	3085005	2912838	14784.05	12737.35	
cadata	20640	0.2	0.263428	1085719	1081038	16003.55	15475.36	0.995602
		0.4	0.151341	2135097	2167643	31936.05	31474.21	
		0.6	0.087921	2813070	2614179	42983.89	38580.61	
		0.8	0.039595	3599953	3379580	54917.10	49754.27	

*: experiments where $\nu \geq \nu^*$

It has been proved that if the decomposition method of LIBSVM is used for solving $(D_\epsilon), \epsilon > 0$, during iterations $\alpha_i \alpha_i^* = 0$ always holds (Lin 2001, Theorem 4.1). Now for ν-SVR we do not have this property as α_i and α_i^* may both be nonzero during iterations.

Next we discuss how to find ν^*. We claim that if $\mathbf{Q}$ is positive definite and $(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*)$ is any optimal solution of (D_ϵ), $\epsilon = 0$, then

$$\nu^* = \sum_{i=1}^{l} |\alpha_i - \alpha_i^*|.$$

Note that by defining $\boldsymbol{\beta} \equiv \alpha - \alpha^*$, $(D_\epsilon), \epsilon = 0$ is equivalent to

$$\begin{aligned} \min \quad & \frac{1}{2}\boldsymbol{\beta}^T \mathbf{Q} \boldsymbol{\beta} + (\mathbf{y}/C)^T \boldsymbol{\beta} \\ & \mathbf{e}^T \boldsymbol{\beta} = 0, \\ & -1/l \leq \beta_i \leq 1/l, \qquad i = 1, \ldots, l. \end{aligned}$$

When $\mathbf{Q}$ is positive definite, it becomes a strictly convex programming problem so there is a unique optimal solution $\boldsymbol{\beta}$. That is, we have a unique $\boldsymbol{\alpha} - \boldsymbol{\alpha}^*$ but may have multiple optimal $(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*)$. With conditions $0 \leq \alpha_i, \alpha_i^* \leq 1/l$, the calculation of $|\alpha_i - \alpha_i^*|$ is like to equally reduce α_i and α_i^* until one becomes zero. Then $|\alpha_i - \alpha_i^*|$ is the smallest possible $\alpha_i + \alpha_i^*$ with a fixed $\alpha_i - \alpha_i^*$. In the next section we will use the RBF kernel so if no data points are the same, $\mathbf{Q}$ is positive definite.

6 Experiments

In this section we demonstrate some numerical comparisons between ν-SVR and ϵ-SVR. We test the RBF kernel with $Q_{ij} = e^{-\|\mathbf{x}_i - \mathbf{x}_j\|^2/n}$, where n is the number of attributes of a training data.

The computational experiments for this section were done on a Pentium III-500 with 256MB RAM using the gcc compiler. Our implementation is part of the software LIBSVM which includes both ν-SVR and ϵ-SVR using the decomposition method. We used 100MB as the cache size of LIBSVM for storing recently used Q_{ij}. The shrinking heuristics in LIBSVM is turned off for an easier comparison.

We test problems from various collections. Problems housing, abalone, mpg, pyrimidines, and triazines are from the Statlog collection (Michie et al. 1994). From

StatLib (`http://lib.stat.cmu.edu/datasets`) we select bodyfat, space_ga, and cadata. Problem cpusmall is from the Delve archive which collects data for evaluating learning in valid experiments (`http://www.cs.toronto.edu/~delve`). Problem mg is a Mackey-Glass time series where we use the same settings as the experiments in (Flake and Lawrence 2002). Thus we predict 85 times steps in the future with six inputs. For these problems, some data entries have missing attributes so we remove them before conducting experiments. Both the target and attribute values of these problems are scaled to $[-1,+1]$. Hence the effective range of ϵ is $[0,1]$.

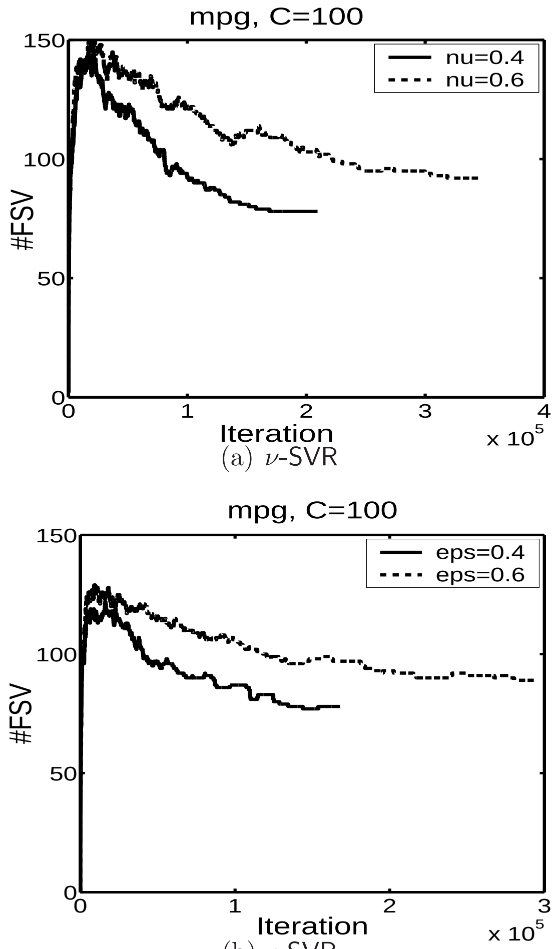
For each problem, we solve its (D_ν) form using $\nu = 0.2, 0.4, 0.6$, and 0.8 first. Then we solve (D_ϵ) with $\epsilon = \rho$ for comparison. Tables 1 and 2 present the number of training data ("l"), the number of iterations ("ν Iter." and "ϵ Iter."), and the training time ("ν Time" and "ϵ Time") by using $C = 1$ and $C = 100$, respectively. In the last column we also list the number ν^* of each problem.

From both tables we have the following observations:

1. Following theoretical results, we really see that as ν increases, its corresponding ϵ decreases.

2. If $\nu \leq \nu^*$, as ν increases, the number of iterations of ν-SVR and its corresponding ϵ-SVR is increasing. Note that the case of $\nu \leq \nu^*$ covers all results in Table 1 and most of Table 2. Our explanation is as follows: When ν is larger, there are more support vectors so during iterations the number of non-zero variables is also larger. In (Hsu and Lin 2002) it has been pointed out that if during iterations there are more non-zero variables than those at the optimum, the decomposition method will take many iterations to reach the final face. Here a face means the sub-space by considering only free variables. An example is in Figure 2 where we plot the number of free variables during iterations against the number of iterations. To be more precise, the y-axis is the number of $0 < \alpha_i < C$ and $0 < \alpha_i^* < C$ where $(\boldsymbol{\alpha}, \boldsymbol{\alpha}^*)$ is the solution at one iteration. We can see that no matter for solving ϵ-SVR or ν-SVR, it takes a lot of iteration to identify the optimal face.

   From the aspect of ϵ-SVR we can consider $\epsilon \mathbf{e}^T(\boldsymbol{\alpha} + \boldsymbol{\alpha}^*)$ as a penalty term



(a) ν-SVR

(b) ϵ-SVR

Figure 2: Iterations and Number of free variables ($\nu \leq \nu^*$)

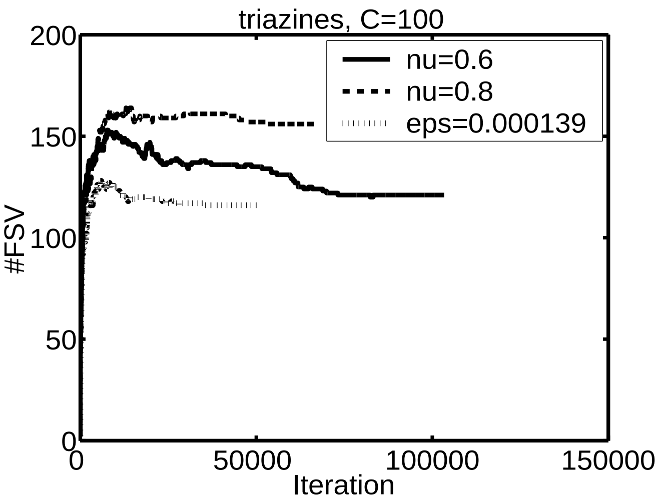


Figure 3: Iterations and Number of free variables ($\nu \geq \nu^*$)

in the objective function of (D_ϵ). Hence when ϵ is larger, fewer α_i, α_i^* are non-zero. That is, the number of support vectors is less.

3. There are few problems (e.g. pyrimidines, bodyfat, and triazines) where $\nu \geq \nu^*$ is encountered. When this happens, their ϵ should be zero but due to numerical inaccuracy, the output ϵ are only small positive numbers. Then for different $\nu \geq \nu^*$, when solving their corresponding (D_ϵ), the number of iterations is about the same as essentially we solve the same problem: (D_ϵ) with $\epsilon = 0$.

   On the other hand, surprisingly we see that at this time as ν increases, it is easier to solve (D_ν) with fewer iterations. Now its solution is optimal for $(D_\epsilon), \epsilon = 0$ but the larger ν is, the more (α_i, α_i^*) are both non-zeros. Therefore, contrary to the general case $\nu \leq \nu^*$ where it is difficult to identify and move free variables during iterations back to bounds at the optimum, now there are no strong needs to do so. To be more precise, in the beginning of the decomposition method, many variables become nonzero as we try to modify them for minimizing the objective function. If finally most of these variables are still nonzero, we do not need the efforts to put them back to bounds. In Figure 3 again we plot the number of free variables against the number of iterations using problem triazines with $\nu = 0.6, 0.8$, and $\epsilon = 0.000139 \approx 0$. It can be clearly seen that for large ν, the decomposition method identifies the optimal face more quickly so the total number of iterations is less.

4. When $\nu \leq \nu^*$, we observe that there are minor differences on the number of iterations for ϵ-SVR and ν-SVR. In Table 1, for nearly all problems ν-SVR takes a little more iterations than ϵ-SVR. However, in Table 2, for problems triazines and mg, ν-SVR is slightly faster. Note that there are several dissimilarities between algorithms for ν-SVR and ϵ-SVR. For example, ϵ-SVR generally starts from the zero vector but ν-SVR has to use a nonzero initial solution. For the working set selection, the two indices selected for ϵ-SVR can be any α_i or α_i^* but the two equality constraints lead to the selection (5.5) and (5.6) for ν-SVR where the set is either from $\{\alpha_1, \ldots, \alpha_l\}$

or $\{\alpha_1^*, \ldots, \alpha_l^*\}$. Furthermore, as the stopping tolerance 10^{-3} might be too loose in some cases, the ϵ obtained after solving (D_ν) may be a little different from the theoretical value. Hence we actually solve two problems with slightly different optimal solution sets. All these factors may contribute to the distinction on iterations.

5. We also see that it is much harder to solve problems using $C = 100$ than using $C = 1$. The difference is even more dramatic than the case of classification. We do not have a good explanation of this observation.

7 Conclusions and Discussions

In this paper we have shown that the inequality in the ν-SVR formulation can be treated as an equality. Hence algorithms similar to those for ν-SVC can be applied for ν-SVR. In addition, in Section 6 we have shown similarities and dissimilarities on numerical properties of ϵ-SVR and ν-SVR. We think that in the future, the relation between C and ν (or C and ϵ) should be investigated in more detail. The model selection on these parameters is also an important issue.

References

Chang, C.-C. and C.-J. Lin (2001a). *LIBSVM: a library for support vector machines*. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

Chang, C.-C. and C.-J. Lin (2001b). Training ν-support vector classifiers: Theory and algorithms. *Neural Computation 13*(9), 2119–2147.

Crisp, D. J. and C. J. C. Burges (2000). A geometric interpretation of ν-SVM classifiers. In S. Solla, T. Leen, and K.-R. Müller (Eds.), *Advances in Neural Information Processing Systems*, Volume 12, Cambridge, MA. MIT Press.

Flake, G. W. and S. Lawrence (2002). Efficient SVM regression training with SMO. *Machine Learning 46*, 271–290.

Hsu, C.-W. and C.-J. Lin (2002). A simple decomposition method for support vector machines. *Machine Learning 46*, 291–314.

Joachims, T. (1998). Making large-scale SVM learning practical. In B. Schölkopf, C. J. C. Burges, and A. J. Smola (Eds.), *Advances in Kernel Methods - Support Vector Learning*, Cambridge, MA. MIT Press.

Keerthi, S. S. and E. G. Gilbert (2002). Convergence of a generalized SMO algorithm for SVM classifier design. *Machine Learning 46*, 351–360.

Keerthi, S. S., S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy (2000). Improvements to SMO algorithm for SVM regression. *IEEE Transactions on Neural Networks 11*(5), 1188–1193.

Laskov, P. (2002). An improved decomposition algorithm for regression support vector machines. *Machine Learning 46*, 315–350.

Lin, C.-J. (2001). On the convergence of the decomposition method for support vector machines. *IEEE Transactions on Neural Networks 12*(6), 1288–1298.

Michie, D., D. J. Spiegelhalter, C. C. Taylor, and J. Campbell (Eds.) (1994). *Machine learning, neural and statistical classification*. Upper Saddle River, NJ, USA: Ellis Horwood. Data available at `http://archive.ics.uci.edu/ml/machine-learning-databases/statlog/`.

Osuna, E., R. Freund, and F. Girosi (1997). Training support vector machines: An application to face detection. In *Proceedings of CVPR'97*, New York, NY, pp. 130–136. IEEE.

Platt, J. C. (1998). Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola (Eds.), *Advances in Kernel Methods - Support Vector Learning*, Cambridge, MA. MIT Press.

Rüping, S. (2000). mySVM - another one of those support vector machines. Software available at `http://www-ai.cs.uni-dortmund.de/SOFTWARE/MYSVM/`.

Schölkopf, B., A. Smola, R. C. Williamson, and P. L. Bartlett (2000). New support vector algorithms. *Neural Computation 12*, 1207–1245.

Schölkopf, B., A. J. Smola, and R. Williamson (1999). Shrinking the tube: A new support vector regression algorithm. In M. S. Kearns, S. A. Solla, and

D. A. Cohn (Eds.), *Advances in Neural Information Processing Systems*, Volume 11, Cambridge, MA. MIT Press.

Vapnik, V. (1998). *Statistical Learning Theory*. New York, NY: Wiley.

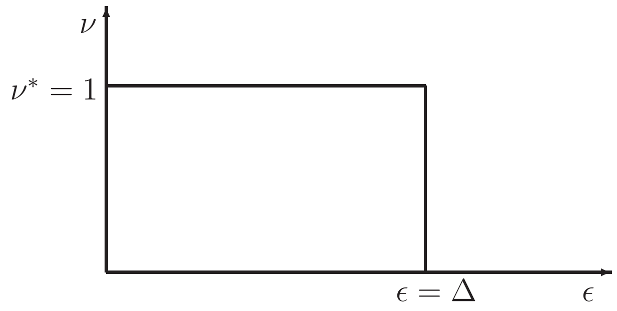
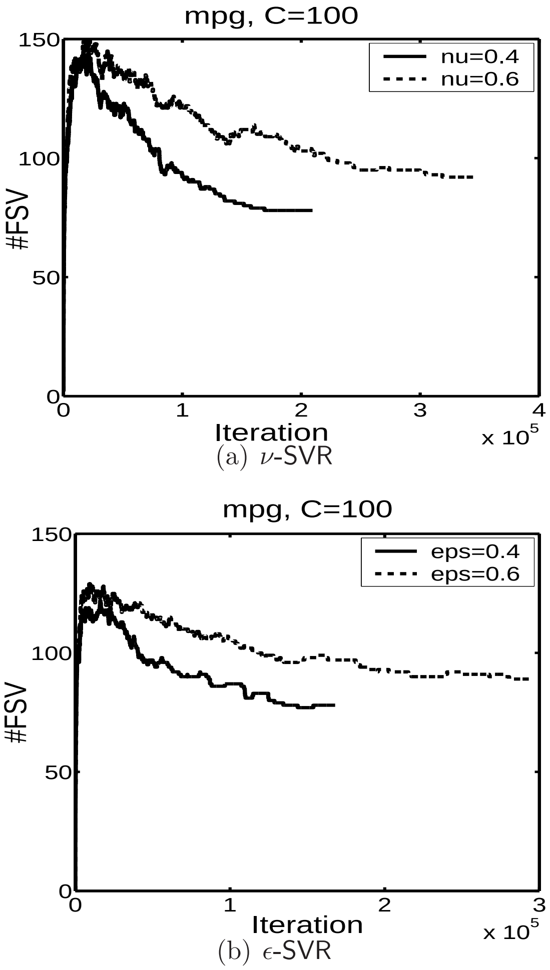


Figure 1: An example where one (D_ϵ) corresponds to different (D_ν)

Author: Chang, Manuscript Number: 2403



(a) ν-SVR

(b) ϵ-SVR

Figure 2: Iterations and Number of free variables ($\nu \leq \nu^*$)

Author: Chang, Manuscript Number: 2403

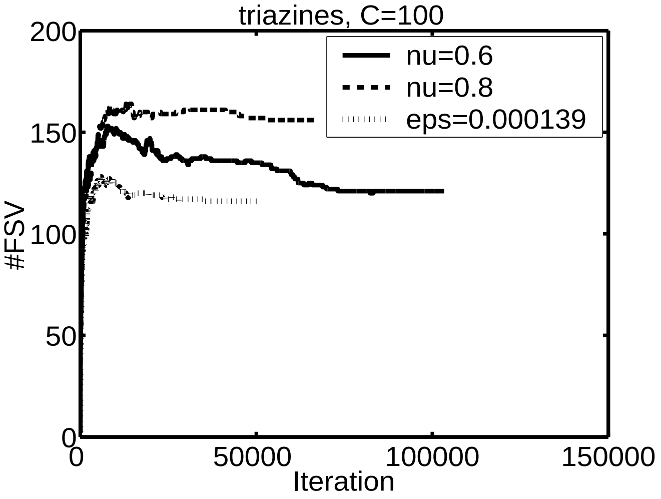


Figure 3: Iterations and Number of free variables ($\nu \geq \nu^*$)

Author: Chang, Manuscript Number: 2403