Manuscript Number: 2370

A Note on the Decomposition Methods for Support Vector Regression†

Shuo-Peng Liao, Hsuan-Tien Lin, and Chih-Jen Lin*

Abstract

The dual formulation of support vector regression involves with two closely related sets of variables. When the decomposition method is used, many existing approaches use pairs of indices from these two sets as the working set. Basically they select a base set first and then expand it so all indices are pairs. This makes the implementation different from that for support vector classification. In addition, a larger optimization sub-problem has to be solved in each iteration. In this paper we provide theoretical proofs and conduct experiments to show that directly using the base set as the working set leads to similar convergence (number of iterations). Therefore, by using a smaller working set while keeping similar number of iterations, the program can be simpler and more efficient.

1 Introduction

Given a set of data points, $\{(x_1, z_1), \ldots, (x_l, z_l)\}$, such that $x_i \in R^n$ is an input and $z_i \in R^1$ is a target output. A major form for solving support vector regression (SVR) is the following optimization problem (Vapnik 1998):

$$\min \frac{1}{2}(\alpha - \alpha^*)^T Q (\alpha - \alpha^*) + \epsilon \sum_{i=1}^{l} (\alpha_i + \alpha_i^*) + \sum_{i=1}^{l} z_i (\alpha_i - \alpha_i^*)$$
$$\sum_{i=1}^{l} (\alpha_i - \alpha_i^*) = 0, 0 \le \alpha_i, \alpha_i^* \le C, i = 1, \ldots, l, \tag{1.1}$$

where C is the upper bound, $Q_{ij} \equiv \phi(x_i)^T \phi(x_j)$, α_i and α_i^* are Lagrange multipliers associated with the ith data x_i, and ϵ is the error that users can tolerate. Note that training vectors x_i are mapped into a higher dimensional space by the function ϕ. An important property is that for any optimal solution, $\alpha_i \alpha_i^* = 0, i = 1, \ldots, l$.

Due to the density of Q, currently the decomposition method is the major method to solve (1.1) (Smola and Schölkopf 1998; Keerthi et al. 2000; Laskov 2001). It is an iterative process where in each iteration the index set of variables are separated to two sets B and N, where B is the working set. Then in that iteration variables corresponding to N are fixed while a sub-problem on variables corresponding to B is minimized.

†This work was supported in part by the National Science Council of Taiwan via the grant NSC 89-2213-E-002-106.

*Department of Computer Science and Information Engineering, National Taiwan University, Taipei 106, Taiwan (cjlin@csie.ntu.edu.tw).

Following approaches for support vector classification, there are some methods for selecting the working set. For many existing approaches for regression, they first use these methods to find a set of variables, called the "base set" here, then they expand the base set so all elements are pairs. Here we define the expanded set as the "pair set." For example, if $\{\alpha_i, \alpha_j^*\}$ are chosen first, they include $\{\alpha_i^*, \alpha_j\}$ into the working set. Then the following sub-problem of four variables $(\alpha_i, \alpha_i^*, \alpha_j, \alpha_j^*)$ is solved:

$$
\begin{aligned}
\min \quad & \frac{1}{2}\begin{bmatrix}\alpha_i - \alpha_i^* \\ \alpha_j - \alpha_j^*\end{bmatrix}^T \begin{bmatrix}Q_{ii} & Q_{ij} \\ Q_{ji} & Q_{jj}\end{bmatrix} \begin{bmatrix}\alpha_i - \alpha_i^* \\ \alpha_j - \alpha_j^*\end{bmatrix} + (Q_{i,N}(\alpha_N - \alpha_N^*) + z_i)(\alpha_i - \alpha_i^*) \\
& +(Q_{j,N}(\alpha_N - \alpha_N^*) + z_j)(\alpha_j - \alpha_j^*) + \epsilon(\alpha_i + \alpha_i^* + \alpha_j + \alpha_j^*) \\
& (\alpha_i - \alpha_i^*) + (\alpha_j - \alpha_j^*) = -\sum_{t \in N}(\alpha_t - \alpha_t^*) \qquad (1.2) \\
& 0 \le \alpha_i, \alpha_j, \alpha_i^*, \alpha_j^* \le C.
\end{aligned}
$$

Note that α_N and α_N^* are fixed elements corresponding to $N = \{t | 1 \le t \le l, t \ne i, t \ne j\}$.

A reason of doing so is to maintain $\alpha_i \alpha_i^* = 0, i = 1, \ldots, l$ throughout all iterations. Hence the number of nonzero variables during iterations can be kept small. However, from (Lin 2001a, Theorem 4.1), it has been shown that for some existing work (e.g. (Keerthi et al. 2000; Laskov 2001)), if they only use the base set as the working set, the property $\alpha_i \alpha_i^* = 0, i = 1, \ldots, l$ still holds. In Section 2 we will discuss this in more detail.

Recently there have been implementations without using pairs of indices. For example, LIBSVM (Chang and Lin 2001), SVMTorch (Collobert and Bengio 2001), and mySVM (Rüping 2000). A question immediately raised is on the performance of these two approaches, called the "base approach" and the "pair approach." On one hand, the pair approach solves a larger sub-problem in each iteration so the number of iterations may be less. On the other hand, a larger sub-problem takes more time so the cost of each iteration is higher.

It has been stated in (Collobert and Bengio 2001) that working with pairs of variables would force the algorithm to do many computations with null variables until the end of the optimization process. In Section 3 we elaborate on this in a detailed proof. First we consider approaches with the smallest size of working set (i.e. two and four elements for both approaches) where the analytic solution of the sub-problem is handily available from the Sequential Minimal Optimization (SMO) (Joachims 1998). From mathematical explanations we show that while solving the sub-problems of the pair set containing four variables, in most cases, only those two variables in the base set are updated. Therefore, the number of iterations of both the base and the pair approaches are nearly the same. In addition, for larger working sets, we prove that after some finite number of iterations, the sub-problem using only the base set are already optimal for the sub-problem using the pair set. These give us theoretical justifications that it is not necessary to use pairs of variables.

Next in Section 4 we conduct experiments to demonstrate the validity of our analysis. Then in Section 5 we give some conclusions and discussions.

There are other decomposition approaches for support vector regression (for example, (Flake and Lawrence 2001)). They deal with different situations which will not be discussed here.

2 Working Set Selection

Here we consider the working set selection from (Joachims 1998; Keerthi et al. 2001) which were originally designed for classification. We then apply them for SVR. To make SVR similar to the form of classification, we define the following $2l$ by 1 vectors:

$$\alpha^{(*)} \equiv \begin{bmatrix} \alpha \\ \alpha^* \end{bmatrix}, \text{ and } y_i \equiv \begin{cases} +1 & i = 1, \ldots, l, \\ -1, & i = l+1, \ldots, 2l. \end{cases} \tag{2.1}$$

Then the regression problem (1.1) can be reformulated as

$$\begin{aligned} \min \quad f(\alpha^{(*)}) = \; & \frac{1}{2}(\alpha^{(*)})^T \begin{bmatrix} Q & -Q \\ -Q & Q \end{bmatrix} \alpha^{(*)} + \left[\epsilon e^T + z^T, \epsilon e^T - z^T\right] \alpha^{(*)} \\ & 0 \le \alpha_i^{(*)} \le C, \quad i = 1, \ldots, 2l, \\ & y^T \alpha^{(*)} = 0. \end{aligned} \tag{2.2}$$

Now f is the objective function of (2.2).

Practically we can use the Karush-Kuhn-Tucker (KKT) condition to test if a given $\alpha^{(*)}$ is an optimal solution of (2.2). If there exists a number b such that for all $i = 1, 2, \cdots, 2l$,

$$\nabla f(\alpha^{(*)})_i + b y_i \ge 0 \qquad \text{if } \alpha_i^{(*)} = 0, \tag{2.3a}$$

$$\nabla f(\alpha^{(*)})_i + b y_i \le 0 \qquad \text{if } \alpha_i^{(*)} = C, \tag{2.3b}$$

$$\nabla f(\alpha^{(*)})_i + b y_i = 0 \qquad \text{if } 0 < \alpha_i^{(*)} < C, \tag{2.3c}$$

a feasible $\alpha^{(*)}$ is optimal for (2.2). Note that the range of b can be determined by

$$m(\alpha^{(*)}) \equiv \max_{1 \le t \le 2l} \Big(\max_{\alpha_t^{(*)} < C, y_t = 1} -y_t \nabla f(\alpha^{(*)})_t, \max_{\alpha_t^{(*)} > 0, y_t = -1} -y_t \nabla f(\alpha^{(*)})_t \Big), \tag{2.4}$$

$$M(\alpha^{(*)}) \equiv \min_{1 \le t \le 2l} \Big(\min_{\alpha_t^{(*)} > 0, y_t = 1} -y_t \nabla f(\alpha^{(*)})_t, \min_{\alpha_t^{(*)} < C, y_t = -1} -y_t \nabla f(\alpha^{(*)})_t \Big). \tag{2.5}$$

That is, a feasible $\alpha^{(*)}$ is an optimal solution if and only if $m(\alpha^{(*)}) \le b \le M(\alpha^{(*)})$, or equivalently,

$$M(\alpha^{(*)}) - m(\alpha^{(*)}) \ge 0. \tag{2.6}$$

For convenience, we define the candidates of $m(\alpha^{(*)})$ as the set of all indices t which satisfy $\alpha_t^{(*),k} < C, y_t = 1$ or $\alpha_t^{(*),k} > 0, y_t = -1$ where $1 \leq t \leq 2l$. Similarly we can define the candidates of $M(\alpha^{(*)})$.

At the beginning of iteration k, let $\alpha^{(*),k} = [\alpha^k, (\alpha^*)^k]^T$ be the vector that we are working on. Then we denote $m_k \equiv m(\alpha^{(*),k})$ and $M_k \equiv M(\alpha^{(*),k})$. Also, let $\arg m_k$ be the subset of indices t in the candidates of m_k such that $-y_t \nabla f(\alpha^{(*),k})_t = m(\alpha^{(*),k})$. Similarly we define $\arg M_k$.

Thus during iterations of the decomposition method, $\alpha^{(*),k}$ is not optimal yet so

$$m_k > M_k, \text{ for all } k. \tag{2.7}$$

If we would like to select two elements as the working set, intuitively we tend to choose indices i and j which satisfy

$$i \in \arg m_k \text{ and } j \in \arg M_k, \tag{2.8}$$

since they cause the maximal violation of the KKT condition.

A systematic way to select a larger working set in each iteration is as follows. If q, an even number, is the size of the working set, $q/2$ indices are sequentially selected from the largest $-y_i \nabla f(\alpha^{(*)})_i$ values to the smaller ones in the candidates of m_k. That is,

$$-y_{i_1} \nabla f(\alpha^{(*),k})_{i_1} \geq \cdots \geq -y_{i_{q/2}} \nabla f(\alpha^{(*),k})_{i_{q/2}},$$

where $i_1 \in \arg m_k$. The other $q/2$ indices are sequentially selected from the smallest $-y_i \nabla f(\alpha^{(*)})_i$ values to the larger ones in the candidates of M_k. That is,

$$-y_{j_{q/2}} \nabla f(\alpha^{(*),k})_{j_{q/2}} \geq \cdots \geq -y_{j_1} \nabla f(\alpha^{(*),k})_{j_1},$$

where $j_1 \in \arg M_k$. Also, we have

$$-y_{j_{q/2}} \nabla f(\alpha^{(*),k})_{j_{q/2}} < -y_{i_{q/2}} \nabla f(\alpha^{(*),k})_{i_{q/2}}. \tag{2.9}$$

to ensure that the intersection of these two groups is empty. Thus if q is large, sometimes the actual number of selected indices may be less than q.

Note that this is the same as the working set selection in (Joachims 1998). However, the original derivation in (Joachims 1998) was from the concept of feasible directions for constrained optimization problems but not from the violation of the KKT condition.

After the base set of q indices is selected, earlier approaches (Keerthi et al. 2000; Laskov 2001) expand the set so all elements in it are pairs. The reason is to keep the property that $\alpha_i^k (\alpha^*)_i^k = 0, i = 1, \ldots, l,$ for all k. However, if directly using elements in the base set, the following theorem has been proved in (Lin 2001a, Theorem 4.1):

Theorem 2.1 *If the initial solution is zero, then* $\alpha_i^k (\alpha^*)_i^k = 0, i = 1, \ldots, l$ *for all* k.

Hence we know that $\alpha_i^k(\alpha^*)_i^k = 0$ is not a particular advantage of using pairs of indices. Another important issue for the decomposition method is the stopping criterion. From (2.6), a natural choice of the stopping criterion is

$$M_k - m_k \geq -\delta, \tag{2.10}$$

where δ, the stopping tolerance, is a small positive number. For $q = 2$, (2.10) is the same as

$$-y_j \nabla f(\alpha^{(*),k})_j - (-y_i \nabla f(\alpha^{(*),k})_i) \geq -\delta, \tag{2.11}$$

where i, j are selected by (2.8).

Note that the convergence of the decomposition method under some conditions of the kernel matrix Q is shown in (Lin 2001a) for the base approach. And some theoretical justification on the use the stopping criteria (2.10) for the decomposition method is in (Lin 2001b). There are, however, no particular convergence proof which has been made for the pair approach, but we will assume it for our analyses.

3 Number of Iterations

In this section we discuss the the relationship between the solutions of sub-problems using the base and the pair approaches. The discussion is divided into two parts. First we consider approaches with the smallest size of working set (i.e. two and four elements for both approaches). We show that for most iterations, the optimal solution of the sub-problem using the base set is already optimal for the sub-problem using the pair set. So the difference between the number of iterations of the base and pair approaches should not be large. Then we consider larger working sets. Though we do not get the result as elegant as the first part, we can still show that after enough iterations, the optimal solution of the sub-problem using the base set is the same as the optimal solution of the sub-problem using the pair set.

To start our proof, first we state an important property on the difference between the ith and $(i+l)$th gradient elements. Consider α_i and α_i^*, $1 \leq i \leq l$. We have

$$\begin{aligned} \nabla f(\alpha^{(*)})_{i+l} &= -(Q(\alpha - \alpha^*))_i + \epsilon - z_i \\ &= -\nabla f(\alpha^{(*)})_i + 2\epsilon. \end{aligned}$$

Note that $y_i = 1$ and $y_{i+l} = -1$ as defined in (2.1), so

$$-y_{i+l} \nabla f(\alpha^{(*)})_{i+l} = -y_i \nabla f(\alpha^{(*)})_i + 2\epsilon. \tag{3.1}$$

We will use this frequently in later analyses.

When $q = 2$, the base set is selected from (2.8). It is easy to see that indices i and $i+l$ where $1 \le i \le l$ cannot both be chosen at the same time. For example, if indices i and $i+l$ are both selected from (2.8), by (3.1) and (2.7), we must have $i+l \in \arg m_k$ and $i \in \arg M_k$. By (2.4) and (2.5), this means $\alpha_{i+l}^{(*),k} > 0$ and $\alpha_i^{(*),k} > 0$, which violates Theorem 2.1. Therefore, $(i, i+l)$ cannot be chosen together.

Also, if one of $\alpha_i^{(*),k}$ and $\alpha_{i+l}^{(*),k}$ is selected, the other must be zero. We can proof this by contradiction. Without loss of generality, say if $\alpha_i^{(*),k}$ is selected and $\alpha_{i+l}^{(*),k}$ is nonzero, then by Theorem 2.1, $\alpha_i^{(*),k}$ must be zero. So by (2.4), (2.5) and (2.8), $i \in \arg m_k$. Moreover, by (2.4), both i and $i+l$ are in the candidates of m_k, and we must have $-y_i \nabla f(\alpha^{(*),k})_i \ge -y_{i+l} \nabla f(\alpha^{(*),k})_{i+l}$ since $i \in \arg m_k$. But this contradicts (3.1). Therefore, if one of $\alpha_i^{(*),k}$ and $\alpha_{i+l}^{(*),k}$ is selected, the other must be zero.

If any one of $(i,j), (i,j+l), (i+l,j), (i+l,j+l)$ where $1 \le i,j \le l$ is chosen from (2.8), our goal is to see the difference on solving the two-variable sub-problem and the four-variable sub-problem of $(i, j, i+l, j+l)$.

Without loss of generality, we consider the case where (i,j) is chosen by (2.8). Then $(\alpha_i^{(*),k}, \alpha_j^{(*),k}, \alpha_{i+l}^{(*),k}, \alpha_{j+l}^{(*),k})$ are the corresponding variables at iteration k, from earlier discussions, we know $\alpha_{i+l}^{(*),k} = \alpha_{j+l}^{(*),k} = 0$. After a two-variable sub-problem on $\alpha_i^{(*)}$ and $\alpha_j^{(*)}$ is solved, we assume the new values are $(\bar{\alpha}_i^{(*),k}, \bar{\alpha}_j^{(*),k}, 0, 0)$. From (3.1) and the KKT condition, it is easy to see that if $\bar{\alpha}_i^{(*),k} > 0$ and $\bar{\alpha}_j^{(*),k} > 0$, $(\bar{\alpha}_i^{(*),k}, \bar{\alpha}_j^{(*),k}, 0, 0)$ is already an optimal solution of the four-variable problem (1.2).

Therefore the only difference happens when there is a "jump" from the (i,j) plane to another plane of two variables and the objective value of (1.2) can be further decreased. We illustrate this in Figure 1.

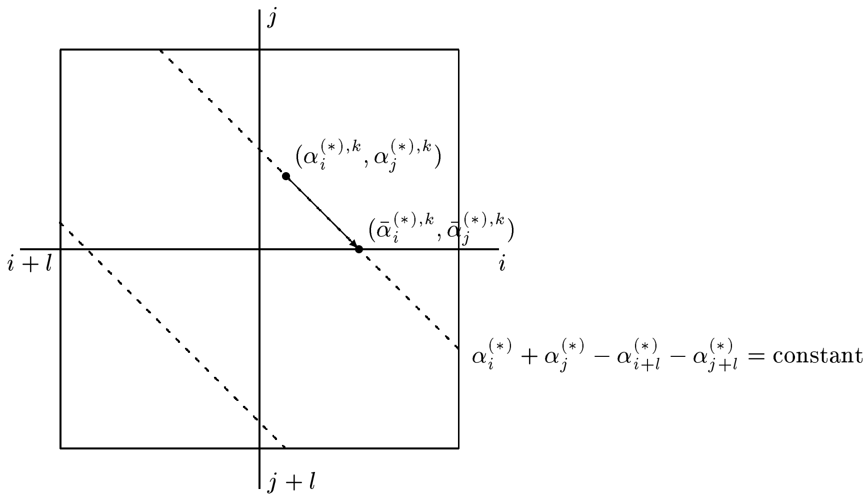

Figure 1: Possible situation of plane changes

In this figure each square represents a plane of two nonzero variables. From the linear constraint

$$\alpha_i^{(*)} + \alpha_j^{(*)} = -\sum_{t \neq i,j} y_t \alpha_t^{(*)}, \tag{3.2}$$

the two dashed parallel lines in Figure 1 show how the solution plane possibly changes. For example, after $\bar{\alpha}_j^{(*),k}$ becomes zero, if $(\bar{\alpha}_i^{(*),k}, \bar{\alpha}_j^{(*),k}, 0, 0)$ is not an optimal solution of (1.2), we may further reduce its objective value by entering the (i, j^*) plane. We will check under what conditions, $(\bar{\alpha}_i^{(*),k}, \bar{\alpha}_j^{(*),k}, 0, 0)$ is not an optimal solution of (1.2).

Since $\alpha_i^{(*)}$ and $\alpha_j^{(*)}$ are adjusted on the line (3.2), we consider the objective value of the sub-problem on the (i, j) plane as the following function of a single variable v, where $N = \{t | 1 \leq t \leq l, t \neq i, t \neq j\}$ are indices of the fixed variables:

$$\begin{aligned}
& g(v) \\
\equiv\ & \frac{1}{2} \begin{bmatrix} \alpha_i^{(*),k} + v & \alpha_j^{(*),k} - v \end{bmatrix} \begin{bmatrix} Q_{ii} & Q_{ij} \\ Q_{ji} & Q_{jj} \end{bmatrix} \begin{bmatrix} \alpha_i^{(*),k} + v \\ \alpha_j^{(*),k} - v \end{bmatrix} \\
& + (Q_{i,N}(\alpha_N^k - (\alpha^*)_N^k) + \epsilon + z_i)(\alpha_i^{(*),k} + v) \\
& + (Q_{j,N}(\alpha_N^k - (\alpha^*)_N^k) + \epsilon + z_j)(\alpha_j^{(*),k} - v) \\
=\ & \frac{1}{2}(Q_{ii} - 2Q_{ij} + Q_{jj})v^2 + (\nabla f(\alpha^{(*),k})_i - \nabla f(\alpha^{(*),k})_j)v + \text{ constant.}
\end{aligned}$$

Since $\alpha_i^{(*)}$ is increased from $\alpha_i^{(*),k}$ to $\bar{\alpha}_i^{(*),k}$, we know

$$g'(0) = \nabla f(\alpha^{(*),k})_i - \nabla f(\alpha^{(*),k})_j < 0.$$

Now if $(\bar{\alpha}_i^{(*),k}, \bar{\alpha}_j^{(*),k}, 0, 0)$ is not an optimal solution of (1.2), we can define a new function $\bar{g}(v)$ similar to $g(v)$ at $(\bar{\alpha}_i^{(*),k}, 0)$ of the $(i, j + l)$ plane. If $\bar{\alpha}_i^{(*),k}$ can be further increased,

$$\bar{g}'(0) = \nabla f(\bar{\alpha}^{(*),k})_i + \nabla f(\bar{\alpha}^{(*),k})_{j+l} < 0. \tag{3.3}$$

However, from (3.1) and $\bar{\alpha}_i^{(*),k} - \alpha_i^{(*),k} = -(\bar{\alpha}_j^{(*),k} - \alpha_j^{(*),k})$,

$$\begin{aligned}
& \nabla f(\bar{\alpha}^{(*),k})_i + \nabla f(\bar{\alpha}^{(*),k})_{j+l} \\
=\ & \nabla f(\bar{\alpha}^{(*),k})_i - \nabla f(\bar{\alpha}^{(*),k})_j + 2\epsilon \\
=\ & \nabla f(\alpha^{(*),k})_i + Q_{ii}(\bar{\alpha}_i^{(*),k} - \alpha_i^{(*),k}) + Q_{ij}(\bar{\alpha}_j^{(*),k} - \alpha_j^{(*),k}) - \\
& (\nabla f(\alpha^{(*),k})_j + Q_{ji}(\bar{\alpha}_i^{(*),k} - \alpha_i^{(*),k}) + Q_{jj}(\bar{\alpha}_j^{(*),k} - \alpha_j^{(*),k})) + 2\epsilon \\
=\ & (\nabla f(\alpha^{(*),k})_i - \nabla f(\alpha^{(*),k})_j) + (\bar{\alpha}_i^{(*),k} - \alpha_i^{(*),k})(Q_{ii} - 2Q_{ij} + Q_{jj}) + 2\epsilon. \quad (3.4)
\end{aligned}$$

Since Q is positive semidefinite, $Q_{ii}Q_{jj} - Q_{ij}^2 \geq 0$ implies $Q_{ii} - 2Q_{ij} + Q_{jj} \geq 0$. With $\bar{\alpha}_i^{(*),k} - \alpha_i^{(*),k} \geq 0$ and (3.3), we know if

$$(\nabla f(\alpha^{(*),k})_i - \nabla f(\alpha^{(*),k})_j) + (\bar{\alpha}_i^{(*),k} - \alpha_i^{(*),k})(Q_{ii} - 2Q_{ij} + Q_{jj}) + 2\epsilon \geq 0, \tag{3.5}$$

it is impossible to move $(\bar{\alpha}_i^{(*),k}, 0)$ on $(i, j+l)$ plane further. That is, $(\bar{\alpha}_i^{(*),k}, \bar{\alpha}_j^{(*),k}, 0, 0)$ is already an optimal solution of (1.2). For other cases, i.e. $(i, j+l), (i+l, j)$, and $(i+l, j+l)$, results are the same. Note that now $1 \leq i, j \leq l$ so $\nabla f(\alpha^{(*),k})_i - \nabla f(\alpha^{(*),k})_j$ is actually the value obtained in (2.11). That is, it is the number used for checking the stopping criterion. Therefore, we have the following theorem:

Theorem 3.2 *For all iterations with the violation on the stopping criterion (2.11) no more than 2ϵ, an optimal solution of the two-variable sub-problem is already an optimal solution of the corresponding four-variable sub-problem.*

If ϵ is not small, in most iterations the stopping tolerance is smaller than 2ϵ. In addition, as most decomposition iterations are spent in the final stage due to slow convergence, this theorem has shown a conclusive result that no matter using two-variable or four-variable approaches, the difference on the number of iterations should not be much.

For larger working set (i.e. $q > 2$), we may not be able to get results as elegant as Theorem 3.2. When $q = 2$, for example, we exactly know the relation on the changes of $\alpha_i^{(*),k}$ and $\alpha_j^{(*),k}$ in one iteration, as $\bar{\alpha}_i^{(*),k} - \alpha_i^{(*),k} = -(\bar{\alpha}_j^{(*),k} - \alpha_j^{(*),k})$. However, when $q > 2$, the change on each variable can be different. In the following we will show that if $\{\alpha^{(*),k}\}$ is an infinite sequence, during final iterations, i.e. after k is large enough, solving the sub-problem of the base set is the same as solving the larger sub-problem of the pair set. Next we describe some properties which will be used for the proof.

Assume that the sequence $\{\alpha^{(*),k}\}$ of the base approach converges to an optimal solution $\hat{\alpha}^{(*)}$. Then we can define

$$\hat{M} \equiv M(\hat{\alpha}^{(*)}) \text{ and } \hat{m} \equiv m(\hat{\alpha}^{(*)}). \tag{3.6}$$

We also note that (2.9) implies that for any index $1 \leq i \leq 2l$ in the working set of the kth iteration,

$$M_k \leq -y_i \nabla f(\alpha^{(*),k})_i \leq m_k. \tag{3.7}$$

Now we describe two theorems from (Lin 2001b) which are needed for the main proof. These theorems deal with a general framework of decomposition methods for different SVM formulations. We can easily check that the base approach satisfies the required conditions of these two theorems so they can be applied:

Theorem 3.3

$$\lim_{k \to \infty} m_k - M_k = 0. \tag{3.8}$$

Theorem 3.4 *For any $\hat{\alpha}_i^{(*)}$, $1 \leq i \leq 2l$, whose corresponding $-y_i \nabla f(\hat{\alpha}^{(*)})_i$ is neither $\hat{m}$ nor $\hat{M}$, after k is large enough, $\alpha_i^{(*),k}$ is at a bound and is equal to $\hat{\alpha}_i^{(*)}$.*

Immediately we have a corollary of Theorem 3.3 which is specific to SVR:

Corollary 3.5 *After k is large enough, for all $i = 1, 2, \cdots, l$, $\alpha_i^{(*),k}$ and $\alpha_{i+l}^{(*),k}$ would not be both selected in the base working set.*

Proof. By the convergence of $m_k - M_k$ to 0, after k is large enough, $m_k - M_k < \epsilon$. If there exists $1 \leq i \leq l$ such that $\alpha_i^{(*),k}$ and $\alpha_{i+l}^{(*),k}$ are both selected in the base working set, from (3.7), $M_k \leq -\nabla f(\alpha^{(*),k})_i \leq m_k$ and $M_k \leq \nabla f(\alpha^{(*),k})_{i+l} \leq m_k$. However, (3.1) shows $\nabla f(\alpha^{(*),k})_{i+l} = -\nabla f(\alpha^{(*),k})_i + 2\epsilon$ so $m_k - M_k \geq 2\epsilon$ and there is a contradiction.

Next we describe the main proof of this section, which is an analysis on the infinite sequence $\{\alpha^{(*),k}\}$.

Theorem 3.6 *We assume that $\hat{M} \neq \hat{m} + 2\epsilon$. After k is large enough, any optimization sub-problem of the base set is already optimal for the larger sub-problem of the pair set.*

Proof. If the result is wrong, there is an index $1 \leq i \leq l$ and an infinite set $\mathcal{K}$ such that for all $k \in \mathcal{K}$, $\alpha_i^{(*),k}$ (or $\alpha_{i+l}^{(*),k}$) is selected in the working set but then $\alpha_{i+l}^{(*),k}$ (or $\alpha_i^{(*),k}$) is also modified. Without loss of generality, we assume that $\alpha_i^{(*),k}$ is selected in the working set but $\alpha_{i+l}^{(*),k}$ is modified infinite times. So by Theorem 3.4,

$$\nabla f(\hat{\alpha}^{(*)})_{i+l} = \hat{m}, \text{ or } \nabla f(\hat{\alpha}^{(*)})_{i+l} = \hat{M}.$$

By (3.1),

$$-\nabla f(\hat{\alpha}^{(*)})_i = \hat{m} - 2\epsilon < \hat{m}, \text{ or } -\nabla f(\hat{\alpha}^{(*)})_i = \hat{M} - 2\epsilon < \hat{M}.$$

For the second case, by the assumption that $\hat{m} \neq \hat{M} - 2\epsilon$, we have $\hat{m} < -\nabla f(\hat{\alpha}^{(*)})_i$ or $\hat{m} > -\nabla f(\hat{\alpha}^{(*)})_i$. But if $\hat{m} < -\nabla f(\hat{\alpha}^{(*)})_i$, we have $\hat{m} < -\nabla f(\hat{\alpha}^{(*)})_i < \hat{M}$ which is impossible for an optimal solution. Hence

$$-y_i \nabla f(\hat{\alpha}^{(*)})_i < \hat{m} \tag{3.9}$$

holds for both cases. Therefore, we can define

$$\Delta \equiv \min(\epsilon/2, (\hat{m} - (-y_i \nabla f(\hat{\alpha}^{(*)})_i))/3) > 0.$$

By the convergence of the sequence $\{-y_j \nabla f(\alpha^{(*),k})_j\}$ to $-y_j \nabla f(\hat{\alpha}^{(*)})_j$, for all $j = 1, \ldots, 2l$, after k is large enough,

$$|y_j \nabla f(\alpha^{(*),k})_j - y_j \nabla f(\alpha^{(*),k+1})_j| \leq \Delta \text{ and} \tag{3.10}$$

$$|y_j \nabla f(\alpha^{(*),k})_j - y_j \nabla f(\hat{\alpha}^{(*)})_j| \leq \Delta. \tag{3.11}$$

Suppose that at the kth iteration $j \in \arg M_k$ is selected in the working set and

$$-y_j \nabla f(\alpha^{(*),k})_j = M_k.$$

By (3.7), (3.10), (3.11), and (3.9),

$$\begin{aligned}
&- y_j \nabla f(\hat{\alpha}^{(*)})_j \\
&\leq - y_j \nabla f(\alpha^{(*),k})_j + \Delta \ = \ M_k + \Delta \\
&\leq - y_i \nabla f(\alpha^{(*),k})_i + \Delta \ \leq \ -y_i \nabla f(\hat{\alpha}^{(*)})_i + 2\Delta \\
&\leq - y_i \nabla f(\hat{\alpha}^{(*)})_i + 2(\hat{m} - (-y_i \nabla f(\hat{\alpha}^{(*)})_i))/3 \\
&< \hat{m} \leq \hat{M}.
\end{aligned} \tag{3.12}$$

From Theorem 3.4 and (3.12), after k is large enough, $\alpha_j^{(*),k}$ is at a bound and is equal to $\hat{\alpha}_j^{(*)}$. That is, $\alpha_j^{(*),k} = \alpha_j^{(*),k+1} = \hat{\alpha}_j^{(*)}$. Since $\alpha_j^{(*),k+1} = \alpha_j^{(*),k}$ and $\alpha_j^{(*),k}$ is in the candidates of M_k, by (3.7), (3.10), and (3.11)

$$\begin{aligned}
M_{k+1} &\leq -y_j \nabla f(\alpha^{(*),k+1})_j \\
&\leq -y_j \nabla f(\alpha^{(*),k})_j + \Delta = M_k + \Delta \\
&\leq -y_i \nabla f(\alpha^{(*),k})_i + \Delta \\
&\leq -y_i \nabla f(\alpha^{(*),k+1})_i + 2\Delta.
\end{aligned}$$

Hence we get

$$M_{k+1} \leq -y_i \nabla f(\alpha^{(*),k+1})_i + \epsilon. \tag{3.13}$$

On the other hand, $\alpha_i^{(*),k}$ is modified to $\alpha_i^{(*),k+1}$ so at least one of them is strictly positive. By the definition of m_k,

$$\nabla f(\alpha^{(*),k})_{i+l} \leq m_k, \text{ or } \nabla f(\alpha^{(*),k+1})_{i+l} \leq m_{k+1}. \tag{3.14}$$

From (3.1), (3.7), (3.13), and (3.14), for all large enough $k \in \mathcal{K}$,

$$M_k \leq m_k - 2\epsilon, \text{ or } M_{k+1} \leq m_{k+1} - \epsilon.$$

Therefore,

$$\lim_{k \to \infty} m_k - M_k \neq 0$$

which contradicts Theorem 3.3.

4 Experiments

Table 4.1: Problem abalone

Parameters	Iter.(2-var.)	Iter.(4-var.)	SV	Candidates*	Jumps+
$C = 10, \epsilon = 0.1$	18930	18790	3967	8438	14
$C = 10, \epsilon = 1$	10173	10173	2183	1705	0
$C = 10, \epsilon = 10$	17	17	7	0	0
$C = 100, \epsilon = 0.1$	142190	140686	3938	63057	207
$C = 100, \epsilon = 1$	122038	119981	2147	10187	48
$C = 100, \epsilon = 10$	261	261	9	0	0

* The term "Candidates" indicates the number of iterations which violate conditions of Theorem 3.2.

+ The term "Jumps" is the number of plane changes as illustrated in Figure 1.

Table 4.2: Problem add10

Parameters	Iter.(2-var.)	Iter.(4-var.)	SV	Candidates	Jumps
$C = 10, \epsilon = 0.1$	28625	28955	9254	13918	1
$C = 10, \epsilon = 1$	22067	21913	4997	2795	1
$C = 10, \epsilon = 10$	116	116	14	0	0
$C = 100, \epsilon = 0.1$	350344	350325	9158	109644	12
$C = 100, \epsilon = 1$	227604	227604	4265	14628	0
$C = 100, \epsilon = 10$	105	105	11	0	0

We consider two regression problems abalone (4177 data) and add10 (9792 data) from (Blake and Merz 1998) and (Friedman 1988), respectively. The RBF kernel is used:

$$Q_{ij} \equiv e^{-\gamma\|x_i - x_j\|^2}.$$

Since our purpose is not on the quality of the solutions, we do not perform model selection on the value of γ. Instead, we fix it to $1/n$, where n is the number of attributes in each data. For these two problems, n is eight and ten, respectively. Based on our past experience, we think that it is an appropriate value when data are scaled to $[-1, 1]$.

Tables 4.1 and 4.2 present results using different ϵ and C on two problems. We consider ϵ below to 0.1 because for smaller ϵ the number of support vectors approaches the number of training data. On the other hand, we consider ϵ up to 10 where the number of support vectors is close to zero. For each parameter set, we present the number of iterations by both two-variable and four-variable approaches, number of support vectors, number of iterations which violate conditions of Theorem 3.2 (i.e. possible candidates of jumps), and the number of real jumps as illustrated in Figure 1 when using the four-variable approach.

The solution of the four-variable approach is obtained as follows: First a two-variable problem obtained from (2.8) is solved. If there is at least one variable which goes to zero, another two-variable problem has to be solved. As indicated in Figure 1, at most three two-variable problems are needed. For our experiments, both versions of the code are directly modified from LIBSVM (version 2.03).

It can be clearly seen that both approaches take nearly the same number of iterations. In addition, the number of jumps while using the four-variable approach is very small, especially when ϵ is larger. Furthermore, the number of "Candidates" is much larger than the number of real jumps. This means $(\bar{\alpha}_i^{(*),k} - \alpha_i^{(*),k})(Q_{ii} - 2Q_{ij} + Q_{jj})$ of (3.4) is large enough so (3.5) is usually satisfied.

Next we experiment with using larger working sets. We use a simple implementation written in MATLAB so only small problems are tested. We consider the first 200 data points of abalone and add10. Results are in Tables 4.3 and 4.4 where we show the number of iterations and the "number of added variables that are modified"(NAVM) for the pair approach. Note that we use "number of added variables that are modified" instead of "number of jumps" since for larger sub-problems we cannot model the change of variables as jumps between planes. The NAVM column is defined as follows:

If the pair approach are used, the working set in the kth iteration is the union of two sets: B_k, which is the base set and its extension $\bar{B}_k$. We check the value of variables in $\alpha_{\bar{B}_k}^{(*)}$ before and after solving the sub-problem. NAVM is the sum of the count for those modified variables in $\alpha_{\bar{B}_k}^{(*)}$ throughout all iterations. So it is at most the sum of $|\bar{B}_k|$ throughout all iterations, which is roughly the number of iterations multiplied by the maximal size of the base set, q.

In Tables 4.3 and 4.4, the column NAVM is relatively small compared to the number of iterations multiplied by q. That means in nearly all the optimization steps, only variables corresponding to the base set rather than the extended set are changed. In addition, from Tables 4.1-4.4, we found that the pair approach, however, may not lead to fewer iterations. So it is not necessary to use the pair approach for solving SVR.

Table 4.3: Problem abalone (first 200 data)

	$q = 10$			$q = 20$		
Parameters	Iter.(base)	Iter.(pairs)	NAVM*	Iter.(base)	Iter.(pairs)	NAVM
$C = 10, \epsilon = 0.1$	132	173	77	100	114	111
$C = 10, \epsilon = 1$	49	46	3	42	40	10
$C = 10, \epsilon = 10$	1	1	0	1	1	0
$C = 100, \epsilon = 0.1$	1641	1983	184	1168	1086	250
$C = 100, \epsilon = 1$	807	552	29	258	310	46
$C = 100, \epsilon = 10$	1	1	0	1	1	0

* NAVM stands for the "number of added variables that are modified" for the pair approach.

Table 4.4: Problem `add10` (first 200 data)

	$q = 10$			$q = 20$		
Parameters	Iter.(base)	Iter.(pairs)	NAVM	Iter.(base)	Iter.(pairs)	NAVM
$C = 10, \epsilon = 0.1$	59	50	13	27	26	33
$C = 10, \epsilon = 1$	55	60	0	18	18	0
$C = 10, \epsilon = 10$	2	2	0	2	2	0
$C = 100, \epsilon = 0.1$	2519	2094	112	1317	1216	135
$C = 100, \epsilon = 1$	944	956	12	236	278	28
$C = 100, \epsilon = 10$	2	2	0	2	2	0

5 Conclusions and Discussions

From our theoretical proofs, we show that in the final iterations of the decomposition methods, the solution of the sub-problem for the base approach is the same as that for the pair approach. This means extending the base working set to pair set will not benefit much so it is not necessary to use the pair method.

We also made experiments to confirm our analysis. The difference between numbers of iterations of the two approaches is negligible. Moreover, the pair approach solves a larger optimization sub-problem in each iteration, which costs more time, so the program using the base approach is more efficient.

Remember that we mentioned in (2.2) that the regression problem can be reformulated to have the same structure as the classification problem. So if we solve SVR using the base approach, it is possible to use the same program for classification with little modifications. For example, LIBSVM used this strategy. However, if (2.2) is directly applied without using as many regression properties as possible, our experience shows that the performance may be a little worse than a software specially designed for regression.

References

Blake, C. L. and C. J. Merz (1998). UCI repository of machine learning databases. Technical report, University of California, Department of Information and Computer Science, Irvine, CA. Available at `http://www.ics.uci.edu/~mlearn/MLRepository.html`.

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

Collobert, R. and S. Bengio (2001). SVMTorch: A support vector machine for large-scale regression and classification problems. *Journal of Machine Learning Research*, 143–160. Available at `http://www.idiap.ch/learning/SVMTorch.html`.

Flake, G. W. and S. Lawrence (2001). Efficient SVM regression training with SMO. *Machine Learning*. To appear.

Friedman, J. (1988). Multivariate adaptive regression splines. Technical Report No. 102, Laboratory for Computational Statistics, Department of Statistics, Stanford University.

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

Keerthi, S. S., S. Shevade, C. Bhattacharyya, and K. Murthy (2001). Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation 13*, 637–649.

Laskov, P. (2001). An improved decomposition algorithm for regression support vector machines. *Machine Learning*. To appear.

Lin, C.-J. (2001a). On the convergence of the decomposition method for support vector machines. *IEEE Transactions on Neural Networks*. To appear.

Lin, C.-J. (2001b). Stopping criteria of decomposition methods for support vector machines: a theoretical justification. Technical report, Department of Computer Science and Information Engineering, National Taiwan University, Taipei, Taiwan.

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

Smola, A. J. and B. Schölkopf (1998). A tutorial on support vector regression. Neuro COLT Technical Report TR-1998-030, Royal Holloway College.

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