2004 International Symposium on Nonlinear
Theory and its Applications (NOLTA2004)
Fukuoka, Japan, Nov. 29 - Dec. 3, 2004

# An Efficient Decomposition Learning Method for Support Vector Regression

Masashi Kuranoshita, Norikazu Takahashi and Tetsuo Nishi

Department of Computer Science and Communication Engineering, Kyushu University
6-10-1 Hakozaki, Higashi-ku, Fukuoka 812-8581 Japan
Email:masashi@kairo.csce.kyushu-u.ac.jp, {norikazu,nishi}@csce.kyushu-u.ac.jp

**Abstract**—A decomposition learning method for support vector regression is proposed. This method is based on the algorithm proposed by Flake and Lawrence but distinguished from it by the following properties: 1) the size of subproblems is not restricted to two but can be any even number, 2) working set is determined in a systematic way, 3) subproblems are in the form of QP problems and therefore can be solved efficiently. Efficiency of the proposed method is shown via computer simulations.

## 1. Introduction

Recently support vector machines (SVMs) have attracted great attention in the field of pattern classification due to their high generalization property. Learning of an SVM leads to a convex quadratic programming (QP) problem with $l$ variables where $l$ is the number of training samples. Hence the computation time increases rapidly as $l$ glows. To overcome this difficulty, some algorithms known as decomposition methods have been proposed [1, 2]. Basic strategy common to all decomposition methods is to try to find an optimal solution of the original QP problem by solving relatively small QP subproblems iteratively. Efficiency of the decomposition methods has been verified by many computer simulations.

The basic idea of SVMs can also be applied to nonlinear regression problems. This technique is known as support vector regression (SVR). Learning of an SVM in SVR is formulated as a QP problem as in the pattern classification case, but the number of variables in the QP problem is twice the number of training samples. Therefore more efficient learning algorithm than the usual decomposition methods is required for SVR.

Flake and Lawrence [3] have recently proposed an SVM learning algorithm for regression. Many numerical results have shown that their method can yield dramatic runtime improvements. However, since it is based on the sequential minimal optimization (SMO) algorithm of Platt [1], the size of subproblems is restricted to two.

In this paper, a new decomposition learning algorithm for SVR is presented. Our algorithm is a generalization of the SMO algorithm by Flake and Lawrence [3] and has the following properties: 1) the size of subproblems is not restricted to two but can be any even number, 2) working set is determined in a systematic way, 3) subproblems are in the form of QP problems and therefore can be solved efficiently. In what follows, we first explain the problem formulation of SVR. We next introduce the SMO algorithm of Flake and Lawrence. We then describe our method in detail and finally show some experimental results.

## 2. Support Vector Regression

Let us consider a pair of variables $\boldsymbol{x}$ and $y$ such that the dependence of $y$ on $\boldsymbol{x}$ is expressed as

$$y = f(\boldsymbol{x}) + \nu$$

where $f(\cdot)$ is an unknown nonlinear function and $\nu$ is a noise independent of $\boldsymbol{x}$ whose value and distribution are unknown. Regression is to estimate the unknown function $f(\cdot)$ from the given samples $(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_l, y_l)$ where $\boldsymbol{x}_i$ is the $i$th sample of the variable $\boldsymbol{x}$ and $y_i$ the observed value of $y$ corresponding to $\boldsymbol{x}_i$. These $l$ samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{l}$ are referred to as the training samples.

SVR is a technique to solve regression problems by means of SVMs. For the given training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{l}$, SVR leads to the following quadratic programming (QP) problem (for more details on derivation of Problem 1 see [4] for example).

**Problem 1** *Find* $\{\alpha_i\}_{i=1}^{l}$ *and* $\{\alpha'_i\}_{i=1}^{l}$ *that minimize the objective function*

$$Q(\boldsymbol{\alpha}, \boldsymbol{\alpha}') = -\sum_{i=1}^{l} y_i(\alpha_i - \alpha'_i) + \epsilon \sum_{i=1}^{l}(\alpha_i + \alpha'_i) + \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i - \alpha'_i)(\alpha_j - \alpha'_j)K(\boldsymbol{x}_i, \boldsymbol{x}_j) \quad (1)$$

*subject to the constraints:*

$$\sum_{i=1}^{l}(\alpha_i - \alpha'_i) = 0 \quad (2)$$

$$0 \le \alpha_i \le C, \quad i = 1, 2, \cdots, l \quad (3)$$

$$0 \le \alpha'_i \le C, \quad i = 1, 2, \cdots, l \quad (4)$$

*where* $\epsilon$ *and* $C$ *are user-specified parameters.*

Function $K(\cdot, \cdot)$ in Eq.(1) is called a kernel. Any function satisfying Mercer's condition [4] can be used as a kernel for SVMs, but the Gaussian kernel described by

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{|\boldsymbol{x} - \boldsymbol{x}'|^2}{2\sigma^2}\right)$$

where $\sigma$ is a positive number specified by users and the polynomial kernel described by

$$K(\boldsymbol{x}, \boldsymbol{x}') = (1 + \boldsymbol{x}^T \boldsymbol{x}')^p$$

where $p$ is a positive integer specified by users are most widely used. For any kernel $K(\cdot, \cdot)$ satisfying Mercer's condition, the matrix $K = [k_{ij}]$ with $k_{ij} = K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is positive semi-definite. This means that Problem 1 is a convex QP problem and therefore does not have local minima.

Let $(\hat{\alpha}_1, \ldots, \hat{\alpha}_l, \hat{\alpha}'_1, \ldots, \hat{\alpha}'_l)$ be any optimal solution of Problem 1. Then the approximating function $F(\boldsymbol{x})$ derived by SVR can be expressed as

$$F(\boldsymbol{x}) = \sum_{i=1}^{l} (\hat{\alpha}_i - \hat{\alpha}'_i) K(\boldsymbol{x}_i, \boldsymbol{x}) + \hat{b}. \quad (5)$$

## 3. SMO algorithm for SVR

As shown in Problem 1, in order to solve a regression problem with $l$ training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{l}$ by means of SVR, a QP problem with $2l$ variables have to be solved. Since $l$ is very large in general, it is required to develop efficient methods for solving large QP problems in the form of Problem 1. Flake and Lawrence [3] have recently proposed an efficient method of solving Problem 1 based on the SMO algorithm [1]. A key idea of their method is to transform Problem 1 into a convex optimization problem with $l$ variables in the following way: First it follows from the KKT condition that any optimal solution $\{\hat{\alpha}_i\}_{i=1}^{l}$ and $\{\hat{\alpha}'_i\}_{i=1}^{l}$ of Problem 1 satisfies $\hat{\alpha}_i \hat{\alpha}'_i = 0$ for $i = 1, 2, \ldots, l$. Hence if we put $\hat{\beta}_i = \hat{\alpha}_i - \hat{\alpha}'_i$ then $\hat{\alpha}_i + \hat{\alpha}'_i$ can be expressed as $|\hat{\beta}_i|$. Moreover $\hat{\beta}_i$ satisfies $-C \leq \hat{\beta}_i \leq C$. Therefore $\{\hat{\beta}_i\}_{i=1}^{l}$ is an optimal solution of the following optimization problem with $l$ variables $\beta_1, \beta_2, \ldots, \beta_l$.

**Problem 2** *Find $\{\beta_i\}_{i=1}^{l}$ that minimize the objective function*

$$W(\boldsymbol{\beta}) = -\sum_{i=1}^{l} y_i \beta_i + \epsilon \sum_{i=1}^{l} |\beta_i| + \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \beta_i \beta_j K(\boldsymbol{x_i}, \boldsymbol{x_j}) \quad (6)$$

*subject to the constraints:*

$$\sum_{i=1}^{l} \beta_i = 0 \quad (7)$$

$$-C \leq \beta_i \leq C, \quad i = 1, 2, \cdots, l \quad (8)$$

Note that the objective function $W(\boldsymbol{\beta})$ is convex because the second term $\epsilon \sum_{i=1}^{l} |\beta_i|$ is convex with respect $\boldsymbol{\beta}$. This means Problem 2 has no local minima. Note also that the approximating function (5) can be expressed with the optimal solution $\{\hat{\beta}_i\}_{i=1}^{l}$ or Problem 2 as follows:

$$F(\boldsymbol{x}) = \sum_{i=1}^{l} \hat{\beta}_i K(\boldsymbol{x}_i, \boldsymbol{x}) + \hat{b}.$$

## 4. Proposed Method

In this section, we present a decomposition method for solving Problem 2. As mentioned before, decomposition methods are iterative algorithms. In each iteration, $q (\leq l)$ variables are chosen for updating and then the subproblem with respect to these $q$ variables are solved. This process is repeated until a certain criterion is satisfied. Therefore any decomposition method is characterized by 1) how to choose $q$ variables for updating, 2) how to solve subproblems with $q$ variables and 3) how to set the termination criterion. In the following, we will show details of our method according to this characterization.

### 4.1. Working Set Selection

Convergence rate of a decomposition method strongly depends on how to choose $q$ variables for the working set in each iteration. The set of $q$ selected variables are usually called the working set. We want to choose $q$ variables so that the objective function $W(\boldsymbol{\beta})$ decreases most rapidly by updating those variables. For this purpose, we make use of Taylor expansion of the objective function $W(\boldsymbol{\beta})$. Let $\boldsymbol{e}_i$ be the unit vector such that the $i$th element is one and others are all zero. Then for any sufficiently small $l$ dimensional vector $\boldsymbol{d} = [d_1, d_2, \ldots, d_l]^T$ we have from the first order Taylor expansion that

$$W(\boldsymbol{\beta} + \boldsymbol{d}) \approx W(\boldsymbol{\beta}) + \boldsymbol{g_d}(\boldsymbol{\beta})^T \boldsymbol{d}$$

where $\boldsymbol{g_d}(\boldsymbol{\beta})$ is an $l$ dimensional vector of which the $i$th element is defined by

$$(\boldsymbol{g_d}(\boldsymbol{\beta}))_i = \begin{cases} \lim_{\delta \to 0+} \frac{W(\boldsymbol{\beta} + \delta \boldsymbol{e}_i) - W(\boldsymbol{\beta})}{\delta}, & \text{if } d_i > 0 \\ \lim_{\delta \to 0-} \frac{W(\boldsymbol{\beta} + \delta \boldsymbol{e}_i) - W(\boldsymbol{\beta})}{\delta}, & \text{if } d_i < 0 \\ 0, & \text{if } d_i = 0 \end{cases} \quad (9)$$

Substituting (6) into the right-hand side of (9), we have

$$(\boldsymbol{g_d}(\boldsymbol{\beta}))_i = \begin{cases} h_i^+(\boldsymbol{\beta}), & \text{if } \beta_i > 0 \text{ or } (\beta_i = 0 \text{ and } d_i > 0) \\ h_i^-(\boldsymbol{\beta}), & \text{if } \beta_i < 0 \text{ or } (\beta_i = 0 \text{ and } d_i < 0) \\ 0, & \text{if } d_i = 0 \end{cases}$$

where

$$h_i^{\pm}(\boldsymbol{\beta}) = -y_i \pm \epsilon + \sum_{j=1}^{l} K(\boldsymbol{x}_i, \boldsymbol{x}_j) \beta_j. \quad (10)$$

Our strategy is to find the vector $\boldsymbol{d}$ such that $\boldsymbol{g_d}(\boldsymbol{\beta})^T \boldsymbol{d}$ is minimized under the constraints: 1) the number of nonzero elements in $\boldsymbol{d}$ is $q$, and 2) $\boldsymbol{\beta} + \boldsymbol{d}$ belongs to the feasible region of Problem 2. This is based on the working set selection method employed in SVM$^{\text{light}}$ [2] and formulated as the following optimization problem.

**Problem 3** *Find $\boldsymbol{d}$ that minimizes the objective function*

$$\boldsymbol{g_d}(\boldsymbol{\beta})^T \boldsymbol{d}$$

*under the constraints:*

$$\sum_{i=1}^{l} d_i = 0$$
$$d_i \geq 0 \;\; \text{if } \beta_i = -C, \;\; d_i \leq 0 \;\; \text{if } \beta_i = C, \quad i = 1, 2, \ldots, l$$
$$|\{d_i | d_i \neq 0\}| = q, \quad i = 1, 2, \ldots, l$$
$$-1 \leq d_i \leq 1, \quad i = 1, 2, \ldots, l$$

As an systematic way of solving Problem 3, we propose the following algorithm.

*Algorithm 1:* Given a vector $\boldsymbol{\beta}$ belonging to the feasible region of Problem 2 and an even number $q$, execute the following procedures.

1. Set $\boldsymbol{d} = \boldsymbol{0}$, $k = 0$, $m = 0$ and $n = l + 1$.
2. Sort the elements of $\boldsymbol{g}_{\boldsymbol{1}}(\boldsymbol{\beta})$ and $\boldsymbol{g}_{-\boldsymbol{1}}(\boldsymbol{\beta})$ in the decreasing order as follows:
$$(\boldsymbol{g}_{\boldsymbol{1}}(\boldsymbol{\beta}))_{i_1} \geq (\boldsymbol{g}_{\boldsymbol{1}}(\boldsymbol{\beta}))_{i_2} \geq \cdots \geq (\boldsymbol{g}_{\boldsymbol{1}}(\boldsymbol{\beta}))_{i_l}$$
$$(\boldsymbol{g}_{-\boldsymbol{1}}(\boldsymbol{\beta}))_{j_1} \geq (\boldsymbol{g}_{-\boldsymbol{1}}(\boldsymbol{\beta}))_{j_2} \geq \cdots \geq (\boldsymbol{g}_{-\boldsymbol{1}}(\boldsymbol{\beta}))_{j_l}$$
where $\boldsymbol{1} = [1, 1, \ldots, 1]^T$.
3. Increase $m$ one by one until $\beta_{i_m} > -C$ or $m = l + 1$ is satisfied. Decrease $n$ one by one until $\beta_{j_n} < C$ or $n = 0$ is satisfied. If either $m = l+1$ or $n = 0$ is satisfied then stop, otherwise goto Step 4.
4. If $-(\boldsymbol{g}_{-\boldsymbol{1}}(\boldsymbol{\beta}))_{i_m} + (\boldsymbol{g}_{\boldsymbol{1}}(\boldsymbol{\beta}))_{j_n} < 0$ then add 2 to $k$ and set $d_{i_m} = -1$ and $d_{j_n} = 1$.
5. If $k = q$ then stop, otherwise goto Step 3.

Unfortunately we have not proved yet the vector $\boldsymbol{d}$ obtained by Algorithm 1 is an optimal solution of Problem 3. However, since our algorithm gives us good results in most cases, we choose $q$ variables corresponding to nonzero elements of $\boldsymbol{d}$ obtained by Algorithm 1 for the working set.

### 4.2. Subproblem

Let $\boldsymbol{\beta}^{\text{old}} = [\beta_1^{\text{old}}, \beta_2^{\text{old}}, \ldots, \beta_l^{\text{old}}]^T$ be the present value of $\boldsymbol{\beta}$, $\hat{\boldsymbol{d}}$ the vector obtained by Algorithm 1 for $\boldsymbol{\beta} = \boldsymbol{\beta}^{\text{old}}$, and $B = \{i \,|\, \hat{d}_i \neq 0\}$ the working set. Then the problem to minimize $W(\boldsymbol{\beta})$ under the constraints (7), (8) and $\beta_i = \beta_i^{\text{old}}$ for all $i \notin B$ is formulated as follows:

**Problem 4** *Find $\{\beta_i\}_{i\in B}$ that minimizes*

$$W(\boldsymbol{\beta}_B) = -\sum_{i\in B} y_i\beta_i + \epsilon \sum_{i\in B} |\beta_i| + \frac{1}{2}\sum_{i\in B}\sum_{j\in B}\beta_i\beta_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{i\in B}\sum_{j\notin B}\beta_i\beta_j^{\text{old}} K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

*under the constraints:*

$$\sum_{i\in B}\beta_i = \sum_{i\in B}\beta_i^{\text{old}}$$

$$-1 \leq \beta_i \leq 1, \quad \forall i \in B$$

Since Problem 4 has only $q$ variables, it can be solved much faster than Problem 2. However, it is still not easy to solve Problem 4 due to the absolute value function contained in $W(\boldsymbol{\beta}_B)$. In order to avoid this difficulty, we employ the following optimization problem as a subproblem instead of Problem 4.

**Problem 5** *Find $\{\beta_i\}_{i\in B}$ that minimizes*

$$\tilde{W}(\boldsymbol{\beta}_B) = -\sum_{i\in B} y_i\beta_i + \epsilon\sum_{i\in B} s_i\beta_i + \frac{1}{2}\sum_{i\in B}\sum_{j\in B}\beta_i\beta_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) + \sum_{i\in B}\sum_{j\notin B}\beta_i\beta_j^{\text{old}} K(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

*under the constraints:*

$$\sum_{i\in B}\beta_i = \sum_{i\in B}\beta_i^{\text{old}}$$

$$L_i \leq \beta_i \leq U_i, \quad \forall i \in B$$

*where $s_i$, $L_i$ and $U_i$ are constants determined by*

$$(s_i, L_i, U_i) = \begin{cases} (1, 0, C), & \beta_i^{\text{old}} > 0 \ \ \textit{or} \ \ (\beta_i^{\text{old}} = 0 \ \ \textit{and} \ d_i = 1) \\ (-1, -C, 0), & \beta_i^{\text{old}} < 0 \ \ \textit{or} \ \ (\beta_i^{\text{old}} = 0 \ \ \textit{and} \ \ d_i = -1) \end{cases}$$

In Problem 5, the range of each variable $\beta_i$ is restricted to either $[-C, 0]$ or $[0, C]$ depending on the values of $\beta_i$ and $\hat{d}_i$. This means the search region for $\{\beta_i\}_{i\in B}$ in Problem 5 is much smaller than that in Problem 4. However, due to this restriction, Problem 5 can be expressed as a QP problem. Since QP problems can be solved very efficiently with many software such as MATLAB, Scilab and so on, computation time required for solving Problem 5 is expected to be much shorter than Problem 4.

### 4.3. Termination Criterion

By exploring the KKT condition for Problem 1, we can easily derive the optimality condition for Problem 2. It follows from the KKT condition that $(\alpha_1, \ldots, \alpha_l, \alpha'_1, \ldots, \alpha'_l)$ is an optimal solution of Problem 1 if and only if there exist $\lambda^{\text{eq}}$, $\{\lambda_i^{\text{low}}\}_{i=1}^l$, $\{\lambda_i^{\text{up}}\}_{i=1}^l$, $\{\lambda_i'^{\text{low}}\}_{i=1}^l$ and $\{\lambda_i'^{\text{up}}\}_{i=1}^l$ such that the following conditions hold.

$$-y_i + \epsilon + \textstyle\sum_{j=1}^l(\alpha_j - \alpha'_j)K_{ij} + \lambda^{\text{eq}} - \lambda_i^{\text{low}} + \lambda_i^{\text{up}} = 0$$
$$y_i + \epsilon - \textstyle\sum_{j=1}^l(\alpha_j - \alpha'_j)K_{ij} - \lambda^{\text{eq}} - \lambda_i'^{\text{low}} + \lambda_i'^{\text{up}} = 0$$
$$\lambda_i^{\text{low}}, \lambda_i^{\text{up}}, \lambda_i'^{\text{low}}, \lambda_i'^{\text{up}} \geq 0$$
$$\lambda_i^{\text{low}}(-\alpha_i) = 0, \quad \lambda_i^{\text{up}}(C - \alpha_i) = 0$$
$$\lambda_i'^{\text{low}}(-\alpha'_i) = 0, \quad \lambda_i'^{\text{up}}(C - \alpha'_i) = 0$$

where $K(\boldsymbol{x}_i, \boldsymbol{x}_j)$ is denoted by $K_{ij}$ for simplicity. The above conditions can be expressed more simply as follows:

1) $\lambda_i^{\text{low}} = \lambda_i^{\text{up}} = 0$ and $y_i - \epsilon - \sum_{j=1}^l(\alpha_j - \alpha'_j)K_{ij} = \lambda^{\text{eq}}$ if $0 < \alpha_i < C$ and $\alpha'_i = 0$

2) $\lambda_i'^{\text{low}} = \lambda_i'^{\text{up}} = 0$ and $y_i + \epsilon - \sum_{j=1}^l(\alpha_j - \alpha'_j)K_{ij} = \lambda^{\text{eq}}$ if $\alpha_i = 0$ and $0 < \alpha'_i < C$.

3) $\lambda_i^{\text{low}} = 0$ and $y_i - \epsilon - \sum_{j=1}^l(\alpha_j - \alpha'_j)K_{ij} \geq \lambda^{\text{eq}}$ if $\alpha_i = C$ and $\alpha'_i = 0$.

4) $\lambda_i'^{\text{low}} = 0$ and $y_i + \epsilon - \sum_{j=1}^l(\alpha_j - \alpha'_j)K_{ij} \leq \lambda^{\text{eq}}$ if $\alpha_i = 0$ and $\alpha'_i = C$.

5) $\lambda^{\rm low} = \lambda^{\rm up} = 0$, $y_i - \epsilon - \sum_{j=1}^{l}(\alpha_j - \alpha'_j)K_{ij} \le \lambda^{\rm eq}$ and $y_i + \epsilon - \sum_{j=1}^{l}(\alpha_j - \alpha'_j)K_{ij} \ge \lambda^{\rm eq}$ if $\alpha_i = \alpha'_i = 0$.

Substituting $\beta_i = \alpha_i - \alpha'_i$ into the above equations and reorganizing the conditions, we derive the optimality condition for Problem 2 as follows:

$$\begin{cases} h_i^+(\boldsymbol{\beta}) \le -\lambda^{\rm eq}, & \text{if } \beta_i = C \\ h_i^+(\boldsymbol{\beta}) = -\lambda^{\rm eq}, & \text{if } 0 < \beta_i < C \\ h_i^-(\boldsymbol{\beta}) \le -\lambda^{\rm eq} \le h_i^+(\boldsymbol{\beta}), & \text{if } \beta_i = 0 \\ h_i^-(\boldsymbol{\beta}) = -\lambda^{\rm eq}, & \text{if } -C < \beta_i < 0 \\ h_i^-(\boldsymbol{\beta}) \ge -\lambda^{\rm eq}, & \text{if } \beta_i = -C \end{cases}$$

where $h_i^{\pm}(\boldsymbol{\beta})$ is defined by (10).

## 5. Simulations

To test the efficiency of the proposed method we have considered a regression problem such that the training samples $\{(x_i, y_i)\}_{i=1}^{l}$ are generated by the following equation:

$$y_i = \frac{\sin x_i}{x_i} + \nu \quad (-4\pi \le x_i \le 4\pi), \quad i = 1, 2, \ldots, l$$

where the noise $\nu$ is Gaussian with zero mean and variance 0.01. For many values of $l$ and $q$, we have measured the CPU time required for the proposed method and compared it with the conventional one where Problem 1 is solved by using SVM$^{\rm light}$ algorithm [2]. Both methods were implemented in Scilab [5] and executed on a PC with 1.2GHz Pentium III processor and 256MB RAM. Kernel function and parameters used in our experiments are as follows:

$$K(x, y) = \exp(-|x - y|^2), \quad \epsilon = 0.05, \quad C = 1$$

Figure 1 shows the CPU time spent by the proposed and conventional methods for various values of $q$ under the condition that $l$ is fixed to 500. Results were obtained by averaging over 10 trials for each value of $q$. It is easily seen from Fig.1 that the proposed method is much faster than the conventional method for all values of $q$. It is also seen that the minimum is achieved at $q = 40$ for the comventional method and $q = 10$ for the proposed method. On the other hand, Fig.2 shows the CPU time spent by the proposed and conventional method for various values of $l$ under the condition that $q$ is fixed to 40. It is seen from this figure that the proposed method is much faster than the conventional method for all values of $l$. Moreover, we confirmed that both algorithms converged to the solution of the same quality.

## 6. Conclusion

We have proposed a new decomposition learning algorithm for SVR. Experimental results show that the proposed method is much faster than the conventional method. Theoretical analysis of the convergence property of the proposed method is one of the future problems.

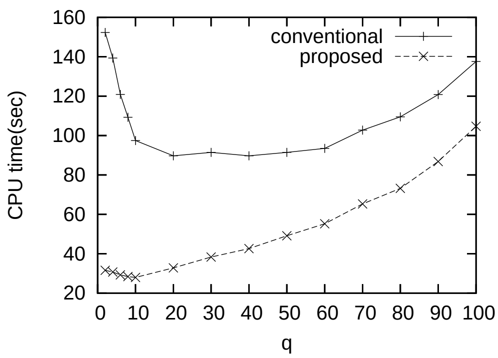


Figure 1: CPU time spent by the proposed and conventional methods for $l = 500$.

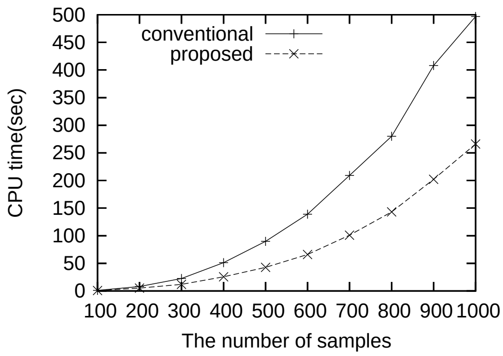


Figure 2: CPU time spent by the proposed and conventional methods for $q = 40$.


## Acknowledgements

This research was partly supported by Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Encouragement of Young Scientists, 15760268 and the 21st Century COE Program 'Reconstruction of Social Infrastructure Related to Information Science and Electrical Engineering'.


## References


[1] J.C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods: Support Vector Machines*, B. Schölkopf, C. Burges and A. Smola, Eds. MIT Press, Cambridge, MA, 1998.

[2] T. Joachims, "Making large-scale SVM learning pratical," in *Advances in Kernel Methods: Support Vector Machines*, B. Schölkopf, C. Burges and A. Smola, Eds. MIT Press, Cambridge, MA, 1998.

[3] G.W. Flake and S. Lawrence, "Efficient SVM regression training with SMO," *Machine Learning*, vol.46, pp.271–290, 2002.

[4] S. Haykin, *Neural Networks*, Prentice Hall, Upper Saddle River, NJ, 1999.

[5] http://scilabsoft.inria.fr/