Optimizing Working Sets for Training Support Vector Regressors by Newton's Method

Shigeo Abe
Kobe University
Kobe, Japan
Email:abe@kobe-u.ac.jp

***Abstract*—In this paper, we train support vector regressors (SVRs) fusing sequential minimal optimization (SMO) and Newton's method. We use the SVR formulation that includes the absolute variables. A partial derivative of the absolute variable with respect to the associated variable is indefinite when the variable takes on zero. We determine the derivative value according to whether the optimal solution exits in the positive region, negative region, or at zero. In selecting working set, we use the method that we have developed for the SVM, namely, in addition to the pair of variables selected by SMO, loop variables that repeatedly appear in training, are added to the working set. By this method the working set size is automatically determined. We demonstrate the validity of our method over SMO using several benchmark data sets.**

I. Introduction

Support vector regressors (SVRs) are one of the most frequently used regressors because of their high generalization ability for a wide range of applications.

Support vector regressors are extended from support vector machines (SVMs) by introducing the epsilon tube that confines the training data near the boundary of the decision hyperplane. This leads to increasing the number of variables twice as large as that of SVMs. This problem is solved by combining the two slack variables associated with an inequality constraint pair into one [1].

One of the widely used training methods is sequential minimal optimization (SMO) [2], [3], which optimizes two variables at a time. The objective function discussed in [1] includes absolute variables. Therefore, the partial derivatives of the objective function with respect to the absolute variables are indefinite when the variables take on zero values. This problem is solved in [4], [5]. In their methods, they assume the change of signs of the variables during variable corrections, i.e., variables with positive signs may change their signs to negative and vice versa.

The exact Karush-Kuhn-Tucker (KKT) conditions [6], which exclude the bias term included in the original KKT conditions, work to speed up SMO training. However, slow SMO training still occurs when a large margin parameter value is set. The use of quadratic information [7] works to improve convergence for a margin parameter value around 1000, but for a larger value, training slows down significantly. To cope with this situation, in [8] if a loop, in which the same variable appears in a sequence of selected violating variables, is detected, corrections are made combining the descent directions of variables in the loop. This idea is extended to the introduction of the momentum term [9].

To improve convergence, more than two variables are optimized at a time [10], [11], [12]. In [12] SMO-NM was proposed, in which SMO and Newton's method are fused. In SMO-NM, in addition to the variables that are selected by SMO, if a loop is detected, loop variables that are in the loop are added to the working set.

In this paper, we extend SMO-NM to function approximation. In solving the optimization problem given in [1], we assume that the signs of the variables do not change in a single correction to allow support vectors to be non-support vectors. By this assumption, for SMO we derive the partial derivative of the objective function with respect to a variable around the zero value, considering the conditions that the optimum solution exists in a positive region, negative region, and at zero point. Using the derived derivatives, monotonic convergence of the solution by SMO is guaranteed.

For the working set size more than two, we calculate the derivative based on SMO. By this method, monotonic convergence may be violated if the variables are corrected opposite to the directions calculated by SMO. But according to the computer experiments, there was no convergence problem.

In Section II, we briefly summarize SVRs and the KKT conditions, and in Section III we discuss the proposed training method. In Section IV, we discuss characteristics of the solution and in Section V we compare SMO-NM with SMO using several benchmark data sets.

II. Support Vector Regressors

We discuss three types of support vector regressor: L1 SVRs, L2 SVRs, and LS (least squares) SVRs [13].

A. L1 SVRs

Using the M training input-output pairs $(\mathbf{x}_i, y_i)$ ($i = 1, \ldots, M$), where $\mathbf{x}_i$ is the ith training input and y_i is the associated output, we consider determining the regression function $f(\mathbf{x})$:

$$y = f(\mathbf{x}) = \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}) + b, \tag{1}$$

where $\boldsymbol{\phi}(\mathbf{x})$ is the mapping function to the feature space, $\mathbf{w}$ is the coefficient vector of the hyperplane in the feature space and b is its bias term.

The L1 and L2 SVRs are given by

$$\min \quad Q(\mathbf{w}, b, \boldsymbol{\xi}, \boldsymbol{\xi}^*) = \frac{1}{2}\|\mathbf{w}\|^2 + \frac{C}{p}\sum_{i=1}^{M}(\xi_i^p + \xi_i^{*p}) \tag{2}$$

$$\text{s.t.} \quad y_i - f(\mathbf{x}_i) \le \varepsilon + \xi_i \text{ for } i = 1, \ldots, M, \tag{3}$$

$$f(\mathbf{x}_i) - y_i \le \varepsilon + \xi_i^* \text{ for } i = 1, \ldots, M, \tag{4}$$

$$\xi_i \ge 0, \quad \xi_i^* \ge 0 \text{ for } i = 1, \ldots, M, \tag{5}$$

where $p = 1$ for the L1 SVR and $p = 2$ for the L2 SVR, ε is the parameter to define the epsilon tube, ξ_i and ξ_i^* are slack variables, and C is the margin parameter that determines the trade-off between the magnitude of the margin and the approximation error of the training data.

The above optimization problem can be converted into the dual form introducing nonnegative slack variables α_i and α_i^* associated with the inequality constraints (3) and (4), respectively. Then the number of variables of the support vector regressor in the dual form is twice the number of the training data. But because nonnegative dual variables α_i and α_i^* appear only in the forms of $\alpha_i - \alpha_i^*$ and $\alpha_i + \alpha_i^*$ and both α_i and α_i^* are not positive at the same time, we can reduce the number of variables to half by replacing $\alpha_i - \alpha_i^*$ with α_i, which take negative values as well as nonnegative values, and $\alpha_i + \alpha_i^*$ with $|\alpha_i|$ [1]. Then, we obtain the following dual problem for the L1 SVR:

$$\max \quad Q(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i,j=1}^{M} \alpha_i\, \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) - \varepsilon \sum_{i=1}^{M} |\alpha_i| + \sum_{i=1}^{M} y_i\, \alpha_i \tag{6}$$

$$\text{s.t.} \quad \sum_{i=1}^{M} \alpha_i = 0, \tag{7}$$

$$C \ge |\alpha_i| \quad \text{for} \quad i = 1, \ldots, M, \tag{8}$$

where α_i are dual variables associated with $\mathbf{x}_i$ and take negative values as well as nonnegative values, $K(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}^\top(\mathbf{x})\, \boldsymbol{\phi}(\mathbf{x})$ is the kernel.

The KKT complementarity conditions are

$$\alpha_i \left(\varepsilon + \xi_i - y_i + \sum_{j=1}^{M} \alpha_j K_{ij} + b\right) = 0 \text{ for } \alpha_i \ge 0, \tag{9}$$

$$\alpha_i \left(\varepsilon + \xi_i + y_i - \sum_{j=1}^{M} \alpha_j K_{ij} - b\right) = 0 \text{ for } \alpha_i < 0, \tag{10}$$

$$\eta_i\, \xi_i = (C - |\alpha_i|)\xi_i = 0 \text{ for } i = 1, \ldots, M, \tag{11}$$

where $K_{ij} = K(\mathbf{x}_i, \mathbf{x}_j)$.

To avoid estimating b in the above KKT conditions during training, we use the exact KKT conditions [6], [14].

We define F_i by

$$F_i = y_i - \sum_{j=1}^{M} \alpha_j K_{ij}. \tag{12}$$

We can classify the KKT conditions into the following five cases:

$$\text{Case 1.} \quad 0 < \alpha_i < C, \quad F_i - b = \varepsilon, \tag{13}$$

$$\text{Case 2.} \quad -C < \alpha_i < 0, \quad F_i - b = -\varepsilon, \tag{14}$$

$$\text{Case 3.} \quad \alpha_i = 0, \quad -\varepsilon \le F_i - b \le \varepsilon, \tag{15}$$

$$\text{Case 4.} \quad \alpha_i = -C, \quad F_i - b \le -\varepsilon, \tag{16}$$

$$\text{Case 5.} \quad \alpha_i = C, \quad F_i - b \ge \varepsilon. \tag{17}$$

Then the KKT conditions are simplified as follows:

$$\bar{F}_i \ge b \ge \tilde{F}_i \quad \text{for} \quad i = 1, \ldots, M, \tag{18}$$

where

$$\tilde{F}_i = \begin{cases} F_i - \varepsilon & \text{if } 0 \le \alpha_i < C, \\ F_i + \varepsilon & \text{if } -C \le \alpha_i < 0, \end{cases} \tag{19}$$

$$\bar{F}_i = \begin{cases} F_i - \varepsilon & \text{if } 0 < \alpha_i \le C, \\ F_i + \varepsilon & \text{if } -C < \alpha_i \le 0. \end{cases} \tag{20}$$

To detect the violating variables, we define $b_{\text{low}}, b_{\text{up}}$ as follows:

$$b_{\text{low}} = \max_i \tilde{F}_i, \quad b_{\text{up}} = \min_i \bar{F}_i. \tag{21}$$

Then if the KKT conditions are not satisfied, $b_{\text{up}} < b_{\text{low}}$ and the data sample i that satisfies

$$b_{\text{up}} < \tilde{F}_i - \tau \quad \text{or} \quad b_{\text{low}} > \bar{F}_i + \tau \quad \text{for} \quad i \in \{1, \ldots, M\} \tag{22}$$

violates the KKT conditions, where τ is a positive parameter to loosen the KKT conditions.

As training proceeds, b_{up} and b_{low} approach each other and at the optimal solution, $b_{\text{up}} = b_{\text{low}}$ if the solution is unique. If not, $b_{\text{up}} > b_{\text{low}}$. In this case, we set $b = (b_{\text{up}} + b_{\text{low}})/2$.

B. L2 SVRs

Setting $p = 2$ in (2) we obtain the L2 SVR. Its dual form is given by

$$\max \quad Q(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i,j=1}^{M} \alpha_i\, \alpha_j \left(K_{ij} + \frac{\delta_{ij}}{C}\right) - \varepsilon \sum_{i=1}^{M} |\alpha_i| + \sum_{i=1}^{M} y_i\, \alpha_i \tag{23}$$

$$\text{s.t.} \quad \sum_{i=1}^{M} \alpha_i = 0, \tag{24}$$

where α_i are dual variables associated with $\mathbf{x}_i$ and take negative values as well as nonnegative values and $\delta_{ij} = 1$ for $i = j$ and 0, otherwise.

The KKT complementarity conditions are

$$\alpha_i\left(\varepsilon+\xi_i-y_i+\sum_{j=1}^{M}\alpha_j K_{ij}+b\right)=0 \text{ for } \alpha_i\geq 0, \quad (25)$$

$$\alpha_i\left(\varepsilon+\xi_i+y_i-\sum_{j=1}^{M}\alpha_j K_{ij}-b\right)=0 \text{ for } \alpha_i< 0, \quad (26)$$

$$C\,\xi_i=|\alpha_i| \text{ for } i=1,\dots,M. \quad (27)$$

For the L2 SVR, we define $\tilde{F}_i$ and $\bar{F}_i$ as follows:

$$\tilde{F}_i = \begin{cases} F_i-\varepsilon & \text{if} \quad \alpha_i=0,\\ F_i-\varepsilon-\dfrac{\alpha_i}{C} & \text{if} \quad \alpha_i>0,\\ F_i+\varepsilon-\dfrac{\alpha_i}{C} & \text{if} \quad \alpha_i<0,\end{cases} \quad (28)$$

$$\bar{F}_i = \begin{cases} F_i+\varepsilon & \text{if} \quad \alpha_i=0,\\ F_i-\varepsilon-\dfrac{\alpha_i}{C} & \text{if} \quad \alpha_i>0,\\ F_i+\varepsilon-\dfrac{\alpha_i}{C} & \text{if} \quad \alpha_i<0.\end{cases} \quad (29)$$

The remaining procedure is the same as that of the L1 SVR.

C. LS SVRs

In the LS SVR, the constraints (3) to (5) are replaced with the equality constraints

$$y_i-f(\mathbf{x}_i)=\varepsilon+\xi_i \quad \text{for } i=1,\dots,M \quad (30)$$

and ξ_i^{*p} in (2) is deleted. The obtained LS SVR is the same as the LS SVM and can be trained by solving a set of linear equations. But because it is slow for a large data set, SMO is extended to training LS SVMs [15].

The dual form of the LS SVR is as follows:

$$\max \quad Q(\boldsymbol{\alpha})=-\frac{1}{2}\sum_{i,j=1}^{M}\alpha_i\,\alpha_j\left(K_{ij}+\frac{\delta_{ij}}{C}\right)+\sum_{i=1}^{M}\alpha_i\,y_i \quad (31)$$

$$\text{s.t.} \quad \sum_{i=1}^{M}\alpha_i=0. \quad (32)$$

The KKT conditions of the above problem is given by

$$\alpha_i\left(\sum_{j=1}^{M}\alpha_j\,K_{ij}+\alpha_i/C+b-y_i\right)=0 \quad \text{for} \quad i=1,\dots,M. \quad (33)$$

We define

$$F_i=y_i-\sum_{j=1}^{M}\alpha_j\,K_{ij}. \quad (34)$$

Then, (33) becomes

$$\alpha_i\,(b-F_i+\alpha_i/C)=0 \quad \text{for} \quad i=1,\dots,M. \quad (35)$$

Because of the equality constraints in the primal form, we can assume that irrespective of α_i the following conditions are satisfied for the optimal solution:

$$b-F_i+\alpha_i/C=0 \quad \text{for} \quad i=1,\dots,M. \quad (36)$$

Then the KKT conditions are satisfied when

$$b_{\mathrm{up}}\geq b_{\mathrm{low}}, \quad (37)$$

where

$$b_{\mathrm{low}} = \max_{i=1,\dots,M}(F_i-\alpha_i/C), \quad (38)$$

$$b_{\mathrm{up}} = \min_{i=1,\dots,M}(F_i-\alpha_i/C). \quad (39)$$

In training, we use the stopping condition (22).

III. Training Methods

In this section we discuss SMO-NM for SVRs: corrections of variables by Newton's method including the derivation of derivatives of absolute variables, working set selection, and calculating corrections by the Cholesky factorization.

A. Calculating Corrections by Newton's Method

First, we discuss corrections of variables for the L1 SVR and then for the L2 and LS SVRs.

1) L1 SVRs: We optimize the variables $\alpha_i\,(i\in W)$ fixing $\alpha_i\,(i\in N)$, where $W\cup N=\{1,\dots,M\}$ and $W\cap N=\phi$, by

$$\max\ Q(\boldsymbol{\alpha}_W)=-\frac{1}{2}\sum_{i,j\in W}\alpha_i\,\alpha_j\,K_{ij}+\sum_{i\in W}y_i\,\alpha_i-\sum_{\substack{i\in W,\\ j\in N}}\alpha_i\,\alpha_j\,K_{ij}-\varepsilon\sum_{i\in W}|\alpha_i| \quad (40)$$

$$\text{s.t.}\ \sum_{i\in W}\alpha_i=-\sum_{i\in N}\alpha_i,\ 0\leq|\alpha_i|\leq C \text{ for } i\in W. \quad (41)$$

Here $\boldsymbol{\alpha}_W=(\dots,\alpha_i,\dots)^\top$, $i\in W$.

Solving the equality in (41) for $\alpha_s\,(s\in W)$, we obtain

$$\alpha_s=-\sum_{i\neq s,i=1}^{M}\alpha_i. \quad (42)$$

Substituting (42) into (40), we eliminate the equality constraint. Let $\boldsymbol{\alpha}_{W'}=(\dots,\alpha_i,\dots)^\top$ $(i\neq s, i\in W)$. Now because $Q(\boldsymbol{\alpha}_{W'})$ is quadratic, we can express the change of $Q(\boldsymbol{\alpha}_{W'})$, $\Delta Q(\boldsymbol{\alpha}_{W'})$, as a function of the change of $\boldsymbol{\alpha}_{W'}$, $\Delta\boldsymbol{\alpha}_{W'}$, by

$$\Delta Q(\boldsymbol{\alpha}_{W'})=\frac{1}{2}\Delta\boldsymbol{\alpha}_{W'}^\top\frac{\partial^2 Q(\boldsymbol{\alpha}_{W'})}{\partial\boldsymbol{\alpha}_{W'}^2}\Delta\boldsymbol{\alpha}_{W'}+\frac{\partial Q(\boldsymbol{\alpha}_{W'})}{\partial\boldsymbol{\alpha}_{W'}}\Delta\boldsymbol{\alpha}_{W'}. \quad (43)$$

Then, neglecting the bounds, $\Delta Q(\boldsymbol{\alpha}_{W'})$ has the maximum at

$$\Delta\boldsymbol{\alpha}_{W'}=-\left(\frac{\partial^2 Q(\boldsymbol{\alpha})}{\partial\boldsymbol{\alpha}_{W'}^2}\right)^{-1}\frac{\partial Q(\boldsymbol{\alpha}_{W'})}{\partial\boldsymbol{\alpha}_{W'}}, \quad (44)$$

where

$$\frac{\partial Q(\boldsymbol{\alpha}_{W'})}{\partial \alpha_i} = F_i - F_s - \varepsilon\,(\mathrm{sign}(\alpha_i) - \mathrm{sign}(\alpha_s)) \quad \text{for } i \in W', \tag{45}$$

$$\frac{\partial^2 Q(\boldsymbol{\alpha}_{W'})}{\partial \alpha_i \partial \alpha_j} = -K_{ij} + K_{is} + K_{sj} - K_{ss} \quad \text{for } i, j \in W'. \tag{46}$$

Here, $\mathrm{sign}(x) = 1$ for $x > 0$ and $\mathrm{sign}(x) = -1$ for $x < 0$. We will discuss the derivative value for $x = 0$ in Section III-A3. We assume that $-\partial^2 Q(\boldsymbol{\alpha})/\partial \boldsymbol{\alpha}_{W'}^2$ is positive definite. The procedure when the matrix is positive semi-definite is discussed in Section III-C.

Then from (41) and (44), we obtain the correction of α_s:

$$\Delta\alpha_s = -\sum_{i \in W'} \Delta\alpha_i. \tag{47}$$

For $\alpha_i\ (i \in W)$, if

$$\alpha_i = C, \quad \Delta\alpha_i > 0 \quad \text{or} \quad \alpha_i = -C, \quad \Delta\alpha_i < 0, \tag{48}$$

we delete these variables from the working set and repeat the procedure for the reduced working set. Let $\Delta\alpha'_i$ be the maximum or minimum correction of α_i that is within the bounds. Here, if α_i changes signs by the correction, we reduce correction so that α_i reaches zero to guarantee monotonic convergence of the objective function value. Then,

1) if $\alpha_i > 0$ and $\alpha_i + \Delta\alpha_i < 0$, then $\Delta\alpha'_i = -\alpha_i$;
2) if $\alpha_i < 0$ and $\alpha_i + \Delta\alpha_i > 0$, then $\Delta\alpha'_i = -\alpha_i$;
3) if $\alpha_i > 0$ and $\alpha_i + \Delta\alpha_i > C$, then $\Delta\alpha'_i = C - \alpha_i$;
4) if $\alpha_i < 0$ and $\alpha_i + \Delta\alpha_i < -C$, then $\Delta\alpha'_i = -C - \alpha_i$;
5) otherwise $\Delta\alpha'_i = \Delta\alpha_i$.

Then we calculate

$$r = \min_{i \in W} \frac{\Delta\alpha'_i}{\Delta\alpha_i}, \tag{49}$$

where $r\ (0 < r \leq 1)$ is the scaling factor.

The corrections of the variables in the working set are given by

$$\boldsymbol{\alpha}_W^{\mathrm{new}} = \boldsymbol{\alpha}_W^{\mathrm{old}} + r\,\Delta\boldsymbol{\alpha}_W. \tag{50}$$

2) L2 SVRs: The training method for the L2 SVR is similar to that for L1 SVR.

We replace (45) and (46), respectively, with

$$\frac{\partial Q(\boldsymbol{\alpha}_{W'})}{\partial \alpha_i} = F_i - F_s - \alpha_i/C + \alpha_s/C - \varepsilon\,(\mathrm{sign}(\alpha_i) - \mathrm{sign}(\alpha_s)) \quad \text{for } i \in W', \tag{51}$$

$$\frac{\partial^2 Q(\boldsymbol{\alpha}_{W'})}{\partial \alpha_i \partial \alpha_j} = -K_{ij} + K_{is} + K_{sj} - K_{ij} - 2\,\delta_{ij}/C \quad \text{for } i, j \in W'. \tag{52}$$

Because α_i are not upper or lower bounded, (48) is not necessary.

We do not allow α_i to change signs by corrections. Thus the change $\Delta\alpha'_i$ in (49) is given as follows: If $\alpha_i > 0$ and $\alpha_i + \Delta\alpha_i < 0$, or $\alpha_i < 0$ and $\alpha_i + \Delta\alpha_i > 0$, then $\Delta\alpha'_i = -\alpha_i$. Otherwise $\Delta\alpha'_i = \Delta\alpha_i$.

In the L2 SVR, because $1/C$ is added to the diagonal elements of the kernel matrix, $-\partial^2 Q(\boldsymbol{\alpha})/\partial \boldsymbol{\alpha}_{W'}^2$ is positive definite.

3) Derivative of $|\alpha_i|$: Because $|\alpha_i|$ is not differentiable at $\alpha_i = 0$, we need to determine the derivative according to whether the correction of α_i is positive, negative, or zero. This is possible for SMO. We consider the following function, which is a simplified SMO version of (40) and (41):

$$\max \ Q(\alpha) = -K\alpha^2 - \varepsilon|\alpha| - \varepsilon|A - \alpha| + F\,\alpha, \tag{53}$$

where $\alpha = \alpha_1$, $\alpha_2 = A$, A is a constant, $K = (K_{11} - 2K_{12} + K_{22})/2 > 0$, $F = F_1 - F_2$ for the L1 SVR and $F_1 - F_2 - \alpha_1/C + \alpha_2/C$ for the L2 SVR. If $A = 0$, both α_1 and α_2 are zero, and otherwise, $\alpha_1 = 0$ and $\alpha_2 \neq 0$.

According to the value of A, the objective function of (53) becomes

1) For $A = 0$,

$$Q(\alpha) = \begin{cases} -K\alpha^2 - 2\varepsilon\,\alpha + F\,\alpha & \text{for } \alpha > 0, \\ -K\alpha^2 + 2\varepsilon\,\alpha + F\,\alpha & \text{for } \alpha < 0. \end{cases} \tag{54}$$

2) For $A > 0$,

$$Q(\alpha) = \begin{cases} -K\alpha^2 + F\,\alpha & \text{for } A \geq \alpha \geq 0, \\ -K\alpha^2 + 2\varepsilon\,\alpha + F\,\alpha & \text{for } \alpha < 0. \end{cases} \tag{55}$$

Here, we exclude the constant terms and because α_2 changes signs for $\alpha > A$, we exclude this case.

3) For $A < 0$,

$$Q(\alpha) = \begin{cases} -K\alpha^2 - 2\varepsilon\,\alpha + F\,\alpha & \text{for } \alpha > 0, \\ -K\alpha^2 + F\,\alpha & \text{for } 0 \geq \alpha \geq A, \end{cases} \tag{56}$$

where the constant terms and the case for $\alpha > A$ are excluded.

TABLE I
THE CONDITIONS FOR THE OPTIMAL SOLUTION

A	α_{opt}	Cond. for $\alpha > 0$	Cond. for $\alpha < 0$	Final Cond.
Zero	Positive	$F > 2\,\varepsilon$	$F > -2\,\varepsilon$	$F > 2\,\varepsilon$
	Negative	$F < 2\,\varepsilon$	$F < -2\,\varepsilon$	$F < -2\,\varepsilon$
	Zero	$F \leq 2\,\varepsilon$	$F \geq -2\,\varepsilon$	$-2\,\varepsilon \leq F \leq 2\,\varepsilon$
Positive	Positive	$F > 0$	$F > -2\,\varepsilon$	$F > 0$
	Negative	$F < 0$	$F < -2\,\varepsilon$	$F < -2\,\varepsilon$
	Zero	$F \leq 0$	$F \geq -2\,\varepsilon$	$-2\,\varepsilon \leq F \leq 0$
Negative	Positive	$F > 2\,\varepsilon$	$F > 0$	$F > 2\,\varepsilon$
	Negative	$F < 2\,\varepsilon$	$F < 0$	$F < 0$
	Zero	$F \leq 2\,\varepsilon$	$F \geq 0$	$0 \leq F \leq 2\,\varepsilon$

Table I shows the conditions for the optimal solution for the above three cases. For example, for $A = 0$, the optimal solution α_{opt} is either positive, negative or zero. Suppose that α_{opt} is positive. Then, from the condition for $\alpha > 0$ in (54), if $F > 2\,\varepsilon$ (Cond. for $\alpha > 0$) is satisfied, the optimum solution exists for $\alpha > 0$. And $Q(\alpha)$ needs to be monotonic for $\alpha < 0$. From the condition for $\alpha < 0$ in (54), this is satisfied by $F > -2\,\varepsilon$ (Cond. for $\alpha < 0$). By combining these conditions, $\alpha_{\mathrm{opt}} > 0$ for $F > 2\,\varepsilon$ (Final Cond.) is obtained.

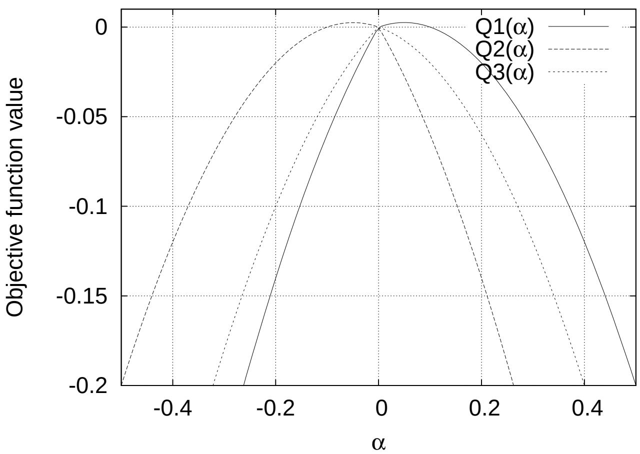

Fig. 1. Objective functions for different parameters

According to the sign of α_{opt}, we set the value to sign(0):

$$\mathrm{sign}(0) = \begin{cases} 1 & \text{for } \alpha_{\mathrm{opt}} > 0, \\ -1 & \text{for } \alpha_{\mathrm{opt}} < 0. \end{cases} \tag{57}$$

If $\alpha_{\mathrm{opt}} = 0$, the initial α is optimal and thus, we delete α from optimization.

Figure 1 shows the three cases for $A = 0$ with $\varepsilon = 0.1$: $F = 0.3$ ($\alpha_{\mathrm{opt}} > 0$), -0.3 ($\alpha_{\mathrm{opt}} < 0$), 0.1 ($\alpha_{\mathrm{opt}} = 0$) for $Q1(\alpha)$, $Q2(\alpha)$, and $Q3(\alpha)$, respectively. For example, if $F = 0.3$, $Q(\alpha)$ takes the maximum value for $\alpha > 0$. Therefore, sign(0) = 1.

For SMO, by (57), the objective function value is guaranteed to be non-decreasing. But for the working set size larger than two, the objective function value may decrease if some of the corrections given by (44) are opposite to the signs given by (57). We may solve this problem by deleting the associated variables and recalculate (44). But in our computer experiment in the subsequent section, we continued training even if this happened. The non-monotonic convergence did not cause any significant problem.

4) LS SVRs: Training of the LS SVR is the same as that of the LS SVM discussed in [12].

In the previous discussions, we replace the partial derivatives of $Q(\boldsymbol{\alpha}_{W'})$ by

$$\frac{\partial Q(\boldsymbol{\alpha}_{W'})}{\partial \alpha_i} = F_i - F_s - \alpha_i/C + \alpha_s/C \text{ for } i \in W', \tag{58}$$

$$\frac{\partial^2 Q(\boldsymbol{\alpha}_{W'})}{\partial \alpha_i \partial \alpha_j} = -K_{ij} + K_{is} + K_{sj} - K_{ss} - 2\,\delta_{ij}/C \quad \text{for } i, j \in W'. \tag{59}$$

Because $-\partial^2 Q(\boldsymbol{\alpha})/\partial \boldsymbol{\alpha}_{W'}^2$ is positive definite and there are no inequality constraints, $r = 1$ and the corrections are always possible. In the extreme case where $|W| = M-1$, the solution is obtained in one step without iterations.

B. Working Set Selection

We adapt the loop variable (LV) selection strategy developed for training SVMs [12] to function approximation. It is based on SMO with the second order information [7] and loop variable detection.

Let the variable associated with $\min \bar{F}_i$ ($\min F_i$ for the LS SVR) be $\alpha_{i_{\min}}$.

In the second order SMO, to reduce computational burden, fixing $\alpha_{i_{\min}}$, the variable that maximizes the objective function value is searched [7]:

$$i_{\mathrm{2nd}} = \arg \max_{i \in V_{\mathrm{KKT}}} \Delta Q(\alpha_i, \alpha_{i_{\min}}). \tag{60}$$

We call the pair of variables that are determined by the second order SMO, *SMO variables*.

To speed up convergence for a large C value, we add variables, which are selected in the previous steps as SMO variables into the working set in addition to the SMO variables.

When at least one of the current SMO variables has already appeared as an SMO variable at a previous step, we consider that a loop is detected and pick up the loop variables that are the SMO variables in the one step to l_{c} steps prior to the current step, where l_{c} is a user-defined parameter and we call the detected loop, *l_{c}-cycle loop*. To avoid obtaining an infeasible solution by adding loop variables to the working set, we restrict loop variables to be unbounded support vectors for the L1 SVR and support vectors for the L2 SVR. But for the LS SVR, any variables are selected.

Let $|W_{\mathrm{s}}|$ denote the maximum working set size. Then we set $|W_{\mathrm{s}}| = 2\,l_{\mathrm{c}} + 2$.

In the following we show the procedure of LV selection for the L1 SVR more in detail.

At the start of training, we initialize $status(i) = 0$ for $i = 1, \ldots, M$, where $status(i) = 0$ for α_i not being selected as an SMO variable, and $status(i) = 1$, already being selected, and ptr is the read pointer of the first-in last-out stack $filo$ with the stack size of $|W_{\mathrm{s}}|$. After $filo$ is full, ptr points to the last element of $filo$ and does not change afterwards. At each iteration step, after $i_{\min}$ and i_{2nd} are calculated, we do the following.

Loop detection and working set selection

1) (Loop detection) Set $W_1 = i_{\min}$ and $W_2 = i_{\mathrm{2nd}}$. If $status(i_{\mathrm{2nd}}) = 1$ or $status(i_{\min}) = 1$, then a loop is detected and go to 2. Else, $status(i_{\mathrm{2nd}}) = 1$, $status(i_{\min}) = 1$, $filo \leftarrow \{i_{\min}, i_{\mathrm{2nd}}\}$, and exit.
2) (Working set setting) Set $k = 1$.
   do $j = 1, ptr$
   if $filo(j) \notin W$ and $0 < |\alpha_{filo(j)}| < C$, then $k \leftarrow k + 1$, $W_k = filo(j)$
   end do
   Set $status(i_{\mathrm{2nd}}) = 1$ and $status(i_{\min}) = 1$, and $filo \leftarrow \{i_{\min}, i_{\mathrm{2nd}}\}$ and exit.

In Step 2, the condition of $filo(j) \notin W$ is to avoid duplicate indices in the working set and the condition $0 < |\alpha_{filo(j)}| < C$ is to avoid obtaining an infeasible solution. For the L2 SVR, the condition is changed to $C \neq 0$ and for the LS SVR, no condition is imposed on $\alpha_{filo(j)}$.

The advantage of the LV selection is that the working set size $|W|$ is determined automatically according to whether loop variables exist. Thus, the overhead caused by matrix inversion is reduced.

C. Calculating Corrections by Cholesky Factorization

We use the Cholesky factorization in calculating (44).

We set $\alpha_s = \alpha_{i_{\min}}$, which is the first element of W and $W' = W - \{i_{\min}\}$. Let $K = \{K_{ij}\} = -\partial^2 Q(\boldsymbol{\alpha})/\partial \boldsymbol{\alpha}^2_{W'}$ $(i, j = 1, \ldots, |W'|)$. Here, the set $\{1, \ldots, |W'|\}$ is a subset of V_{KKT} and the elements are renumbered from 1 to $|W'|$ and 1 corresponds to i_{2nd}. If K is positive definite, it is decomposed by the Cholesky factorization into

$$K = L\,L^{\top}, \tag{61}$$

where L is the regular lower triangular matrix.

Then during the Cholesky factorization, if the argument of the square root associated with the diagonal element is smaller than the prescribed value $\eta\,(>\,0)$, we stop factorizing the matrix and use the already-factorized matrices to obtain the corrections. This happens for the L1 SVR and for the L2, LS SVRs with extremely large C values. Otherwise, we use the full L to obtain the corrections.

For the L1 and L2 SVRs, we check whether the corrections satisfy the inequality constraints. If some of the variables do not satisfy the constraints, we recalculate the corrections, deleting the rows in the L after the rows associated with the variables that violate the inequality constraints. We repeat this procedure, until the feasible corrections are obtained (i.e., $r > 0$). The above procedure is done using the matrices factorized so far. For the LS SVR, $r = 1$.

Because we select the SMO variables as α_s and the first variable in W', the first diagonal element of $-\partial^2 Q(\boldsymbol{\alpha})/\partial \boldsymbol{\alpha}^2_{W'}$ is non-zero and the SMO variables give the feasible solution. Therefore, for the L1/L2 SVRs, SMO-NM reduces to SMO, at worst.

The Cholesky factorization requires $|W'|^3/3$ floating operations [16] compared to one division for SMO. Therefore, to speed up training using the Cholesky factorization over SMO, enough reduction of the number of iterations is necessary.

D. Training Procedure of SMO-NM

In the following we show the training procedure of SMO-NM for the L1 SVR using the LV selection strategy.

1) (Initialization) Set an appropriate value to l_c. Set $\alpha_i = 0$ for $i = 1, \ldots, M$ and select a pair i, j for corrections.
2) (Corrections) Calculate partial derivatives (45) and (46) and calculate corrections by (44). Then, modify the variables by (50).
3) (Convergence Check) Update F_i and calculate b_{up} and b_{low}. If (22) is satisfied, stop training. Otherwise calculate i_{2nd} by (60).
4) (Loop detection and working set selection) Do loop detection and working set selection and go to Step 2.

IV. Characteristics of Solutions

In this section, we discuss convergence of SMO-NM.

For the L1 and L2 SVRS, the following Theorem holds.

Theorem 1: Assume that the signs of variable corrections for the working set $W (|W| > 2)$ are the same as those given by (57). Then, the increase of the objective function value, $\Delta Q(\boldsymbol{\alpha}_{W'})$, is given by

$$\Delta Q(\boldsymbol{\alpha}_{W'}) = -\frac{1}{2} r_W (2 - r_W)\, \frac{\partial Q(\boldsymbol{\alpha}_{W'})}{\partial \boldsymbol{\alpha}_{W'}}^{\top} \left(\frac{\partial^2 Q(\boldsymbol{\alpha}_{W'})}{\partial \boldsymbol{\alpha}^2_{W'}}\right)^{-1} \frac{\partial Q(\boldsymbol{\alpha}_{W'})}{\partial \boldsymbol{\alpha}_{W'}} \geq 0, \tag{62}$$

where r_W is the scaling factor for W. Then if $r_W \geq r_{W_S}$,

$$\Delta Q(\boldsymbol{\alpha}_{W'}) \geq \Delta Q(\boldsymbol{\alpha}_{W'_{\mathrm{S}}}) \tag{63}$$

is satisfied, where $W \supset W_{\mathrm{S}}$. The strict inequality holds when some values of $\alpha_i \in W - W_{\mathrm{S}}$ are not equal to those of the optimal solution for the working set W.

The proof is similar to that given in [12]. If the assumption does not hold for the L1 and L2 SVRs, there may be cases where the objective function value decreases by the variable corrections.

For the LS SVR, the above theorem holds without the assumption and $r = 1$. Therefore, (63) holds for any working set size. This means that the number of iteration by SMO-NM is smaller than or equal to that by SMO.

Because by SMO the SMO variables, which improve the objective function value most, are selected, by variable corrections the objective function value increases monotonically. By SMO-NM, if the assumption holds for all the iterations steps, convergence to the optimal solution is guaranteed. But unlike for L1 and L2 SVMs, the monotonic convergence of SMO-NM is not theoretically guaranteed.

V. Performance Evaluation

Using the benchmark data sets downloaded from the LIBSVM homepage [17], we evaluated the convergence, including training time and the number of iterations, of the proposed method over that of SMO and LIBSVM, which is one of the fastest training tools based on SMO.

Because the tendency is similar for L1, L2, and LS SVRs, in the following we only show the results for the L1 SVR. Table II lists the seven data sets used in our study. It includes the number of input variables and the number of data samples for each data set. For all the data sets, we normalized the input range into $[-1, 1]$, set $\varepsilon = 0.1$, and used the RBF kernels:

$$K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma ||\mathbf{x} - \mathbf{x}'||^2/m), \tag{64}$$

where m is the number of inputs for normalization and γ is a spread of a radius.

We set $\eta = 10^{-9}$ and $\tau = 0.001$ [7]. We measured the training time using a personal computer (3GHz, 2GB memory, Windows XP operating system). If training time was shorter than 60 s, we measured training time five times and took the average. We prepared a cache memory with the size equal to

TABLE II
CROSS-VALIDATION RESULTS USING L1 SVR

Data	Inputs	Samples	C	γ	MAE
mpg	7	392	100	5	1.803
housing	13	506	100	5	2.110
mg	6	1385	10	5	0.093
space_ga	6	3107	10	15	0.075
abalone	8	4177	10000	0.5	1.457
cpusmall	12	8192	100	15	2.026
cadata	8	20640	10000	15	38247

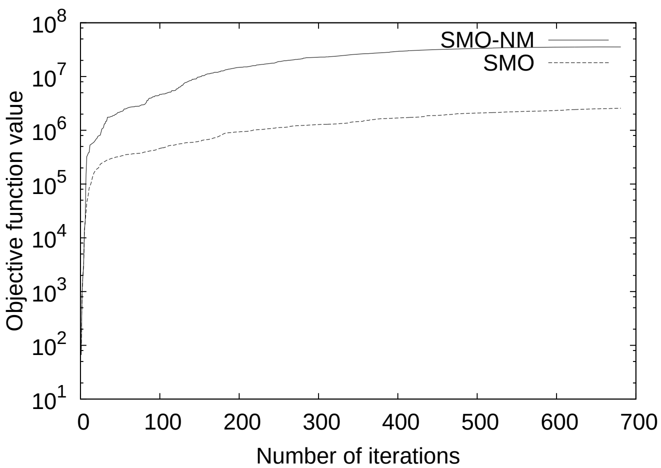

Fig. 2. Objective function values for the mpg data set with $C = 100000$.

the kernel matrix. This was possible for the data sets excluding the cadata set.

To show that large C values are necessary to realize best generalization ability, for each data set we carried out fivefold cross-validation selecting the C value from $\{10, 100, 1000, 10000\}$ and the γ value from $\{0.05, 0.1, 0.5, 1, 5, 10, 15\}$. The selected C and γ values and the associated mean absolute errors (MAEs) are listed in Table II.

As in [12], we set the maximum working set size $|W_{\rm s}|$ to be 600 (the number of cycles = 298).

Training time is also affected by the selection of the γ value. But here we set $\gamma = 1$ in (64), which is a default value in LIBSVM.

Figure 2 shows the change of the objective function values as the training proceeded for the mpg data set with $C = 100000$. For the SMO-NM, the monotonic convergence was violated only once but the decrease was so small, it did not appear in the graph. Although SMO converged monotonically but because the convergence was so slow there was a large gap of the objective function values at the iteration step near 700, where SMO-NM converged.

Figure 3 shows the change of the working set size for the mpg data set with $C = 10$ during convergence. The loop was detected at the 98th step. Afterwards, the working set size changed dramatically.

Table III shows the results for the number of iterations

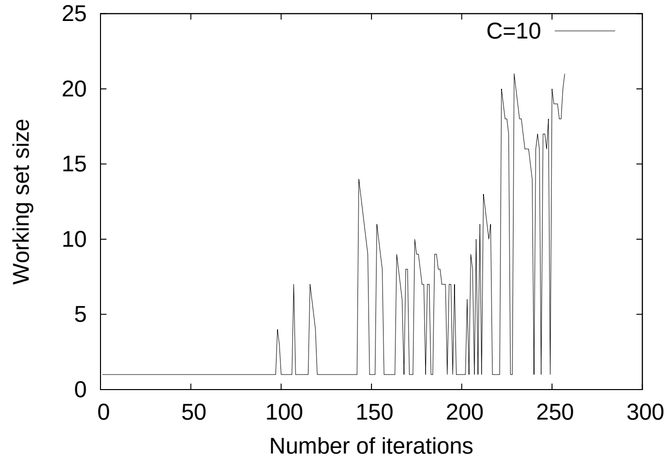

Fig. 3. Working set size for the mpg data set

(Iterations), the average working set size for SMO-NM, the training time (Time), the mean square error (MSE) of the training data set for SMO-NM, and the numbers of support vectors (SVs) for SMO-NM and LIBSVM. In the table, SMO and SMO-NM denote the second order SMO and the proposed method using the LV selection strategy with $|W_{\rm s}| = 600$. For "Iterations" and "Time" columns, the smallest and shortest values are shown in boldface, respectively.

The MSEs for SMO and LIBSVM were almost the same as that for SMO-NM and the SVs for SMO was almost the same for SMO-NM. The SVs for SMO-NM and LIBSVM were almost the same except for the space-ga data set with $C = 100000$. Therefore, almost the same solutions were obtained by the three methods.

Comparing SMO and LIBSVM, the number of iterations of SMO was usually smaller but training time was longer. In SMO, sophisticated optimization techniques such as shrinking were not implemented. This might make training time longer.

Comparing SMO and SMO-NM, the number of iterations by SMO-NM was always smaller and training time by SMO-NM was in most cases shorter and comparable even if longer.

SMO-NM was faster than LIBSVM for $C = 10^5$ except for the cadata set. Slower convergence for the cadata set was because the average working set size was only 2.73, and the speeding up by the Newton's method did not work.

According to the computer experiments, the SMO-NM worked to accelerate training over SMO for large C values. To speed up SMO-NM for small C values, it is better to combine Newton's method with LIBSVM, because SMO-NM can readily be implemented into LIBSVM.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number 25420438.

VI. CONCLUSIONS

We proposed training the support vector regressor (SVR) with the absolute variables by combining sequential minimum

TABLE III
PERFORMANCE COMPARISON FOR THE L1 SVR

Data	C	Iterations			WS size	Time (s)			MSE	SVs	
		SMO-NM	SMO	LIBSVM	SMO-NM	SMO-NM	SMO	LIBSVM	SMO-NM	SMO-NM	LIBSVM
mpg	10	**257**	740	731	5.50	**0.065**	0.087	**0.065**	6.882	378	379
	1000	**569**	23218	43159	35.25	**0.224**	1.090	0.265	4.689	374	374
	100000	**681**	2078552	5211733	93.82	**0.803**	112.1	27.92	3.086	374	374
housing	10	**295**	652	733	5.57	0.093	0.109	**0.087**	16.67	489	489
	1000	**710**	34164	41151	56.81	0.506	2.196	**0.375**	5.587	486	486
	100000	**773**	1744860	3727411	160.74	**3.297**	142.2	37.47	1.704	486	488
mg	10	**1304**	5118	5907	17.30	0.774	0.884	**0.228**	0.014	525	524
	1000	**1244**	346873	347689	58.17	**1.778**	50.01	2.925	0.012	492	490
	100000	**1442**	19272281	154899003	115.84	**5.400**	3192	988.9	0.010	501	546
space_ga	10	**2261**	5718	6822	12.90	2.325	2.293	**0.556**	0.012	967	969
	1000	**2902**	929034	610363	36.06	**5.475**	261.8	9.190	0.010	903	905
	100000	**2768**	41003167	357534630	82.79	**12.27**	13869	1905	0.009	861	1431
abalone	10	**2182**	3219	3886	3.32	2.650	2.778	**1.528**	4.648	3940	3941
	1000	**5449**	125129	189285	25.43	11.16	46.28	**3.468**	4.328	3953	3950
	100000	**9207**	7715526	21051891	71.37	**47.84**	3601	186.1	4.101	3953	3963
cpusmall	10	**4226**	4798	5028	2.81	10.30	10.13	**5.965**	32.96	7933	7931
	1000	**12367**	126940	233103	32.78	74.92	106.1	**13.29**	9.051	7820	7821
	100000	**17936**	12220687	44966369	117.48	**451.4**	11469	1000	7.639	7805	7835
cadata	10	**10321**	**10321**	13094	2.00	67.77	68.91	**35.62**	1.376E+10	20640	20640
	1000	**10339**	10411	12621	2.01	68.08	69.52	**33.61**	6.166E+09	20640	20640
	100000	**10767**	24681	24697	2.73	87.67	135.7	**33.31**	4.310E+09	20640	20640

optimization (SMO) and Newton's method. For SMO, we derived the partial derivative of an absolute variable at the zero point according to whether the optimal solution exists in the positive region, negative regions, or at the zero point. For the working set size more than two, we assumed the derivative values at the zero points by those of SMO. The proposed training method uses the working set strategy developed for SVMs and it reduced to SMO when the working set size is two.

By computer experiment using seven benchmark data sets, we showed that the proposed method was faster than SMO for the large margin parameter values.

REFERENCES

[1] D. Mattera, F. Palmieri, and S. Haykin. An explicit algorithm for training support vector machines. *IEEE Signal Processing Letters*, 6(9):243–245, 1999.

[2] J. C. Platt. Fast training of support vector machines using sequential minimal optimization. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods: Support Vector Learning*, pages 185–208. MIT Press, Cambridge, MA, 1999.

[3] A. Ghio, D. Anguita, L. Oneto, S. Ridella, and C. Schatten. Nested sequential minimal optimization for support vector machines. In A. E. Villa, W. Duch, P. Érdi, F. Masulli, and G. Palm, editors, *Artificial Neural Networks and Machine Learning – ICANN 2012*, volume 7553 of *Lecture Notes in Computer Science*, pages 156–163. Springer, 2012.

[4] G. W. Flake and S. Lawrence. Efficient SVM regression training with SMO. *Machine Learning*, 46(1–3):271–290, 2002.

[5] N. Takahashi, J. Guo, and T. Nishi. Global convergence of SMO algorithm for support vector regression. *IEEE Transactions on Neural Networks*, 19(6):971–982, 2008.

[6] S. S. Keerthi and E. G. Gilbert. Convergence of a generalized SMO algorithm for SVM classifier design. *Machine Learning*, 46(1–3):351–360, 2002.

[7] R.-E. Fan, P.-H. Chen, and C.-J. Lin. Working set selection using second order information for training support vector machines. *Journal of Machine Learning Research*, 6:1889–1918, 2005.

[8] Á. Barbero and J. R. Dorronsoro. Faster directions for second order SMO. In K. Diamantaras, W. Duch, and L. S. Iliadis, editors, *Artificial Neural Networks – ICANN 2010*, volume 6353 of *Lecture Notes in Computer Science*, pages 30–39. Springer, 2010.

[9] Á. Barbero and J. R. Dorronsoro. Momentum sequential minimal optimization: An accelerated method for support vector machine training. In *Proceedings of the 2011 International Joint Conference on Neural Networks (IJCNN 2011)*, pages 370–377, San Jose, CA, 2011.

[10] T. Joachims. Making large-scale support vector machine learning practical. In B. Schölkopf, C. J. C. Burges, and A. J. Smola, editors, *Advances in Kernel Methods: Support Vector Learning*, pages 169–184. MIT Press, Cambridge, MA, 1999.

[11] R. A. Hernandez, M. Strum, J. C. Wang, and J. A. Q. Gonzalez. The multiple pairs SMO: A modified SMO algorithm for the acceleration of the SVM training. In *Proceedings of the 2009 International Joint Conference on Neural Networks (IJCNN 2009)*, pages 1221–1228, Atlanta, GA, 2009.

[12] S. Abe. Fusing sequential minimal optimization and Newton's method for support vector training. *International Journal of Machine Learning and Cybernetics* (DOI: 10.1007/s13042-014-0265-x), 2014.

[13] S. Abe. *Support Vector Machines for Pattern Classification*. Springer-Verlag, London, UK, second edition, 2010.

[14] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy. Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation*, 13(3):637–649, 2001.

[15] J. López, Á. Barbero, and J. R. Dorronsoro. Momentum acceleration of least–squares support vector machines. In T. Honkela, W. Duch, M. Girolami, and S. Kaski, editors, *Artificial Neural Networks and Machine Learning – ICANN 2011*, volume 6792 of *Lecture Notes in Computer Science*, pages 135–142. Springer, 2011.

[16] G. H. Golub and C. F. Van Loan. *Matrix Computations, Third Edition*. The Johns Hopkins University Press, Baltimore, MD, 1996.

[17] C.-C. Chang and C.-J. Lin. LIBSVM–A library for support vector machines: http://www.csie.ntu.edu.tw/˜cjlin/libsvm/.