ESANN 2010 proceedings, European Symposium on Artificial Neural Networks - Computational Intelligence
and Machine Learning. Bruges (Belgium), 28-30 April 2010, d-side publi., ISBN 2-930307-10-2.

# Active Set Training of Support Vector Regressors

Shigeo Abe

Kobe University - Graduate School of Engineering
Kobe, Japan

**Abstract**. In our previous work we have discussed the training method of a support vector classifier by active set training allowing the solution to be infeasible during training. In this paper, we extend this method to training a support vector regressor (SVR). We use the dual form of the SVR where variables take real values and in the objective function the weighted linear sum of absolute values of the variables is included. We allow the variables to change signs from one step to the next. This means changes of the active inequality constraints. Namely, we solve the quadratic programming problem for the initial working set of training data by Newton's method, delete from the working set the data within the epsilon tube, add to the working set training data outside of the epsilon tube, and repeat training the SVM until the working set does not change. We demonstrate the effectiveness of the proposed method using some benchmark data sets.

## 1 Introduction

In a support vector machine (SVM), the input space is mapped into a high-dimensional feature space, and because the mapping function is not explicitly treated by the kernel trick, usually the SVM is trained in the dual form, where the number of variables is the number of training data in pattern classification and twice the number of training data for function approximation. Thus to reduce the number of variables in training, decomposition techniques are used. There are fixed-size chunking [1] and variable-size chunking [2]. In fixed-size chunking, we do not keep all the support vector candidates. The most well-known training method using fixed-size chunking is sequential minimal optimization (SMO) [3]. It optimizes two data at a time. In variable-size chunking, we keep support vector candidates in the working set and when the algorithm terminates, the working set includes support vectors. Training based on variable-size chunking is sometimes called active set training because the constraints associated with support vector candidates satisfy the equality constraints and are called active and the working set active set.

Because the coefficient vector of the hyperplane is expressed by the kernel expansion, substituting the kernel expansion into the coefficient vector, the SVM in the primal form can be solvable. Based on this idea Chapelle [4] proposed training the SVM in the primal form. By this method, at each step the quadratic programming program for the active set is solved by Newton's method and from the active set variables that are no longer support vectors are deleted and violating variables are added to the active set. In [5], this method is extended to dual L2 SVMs, where at each step variables are allowed to be infeasible. Because in the dual form kernel expansion is not used and the coefficient matrix is positive definite, training in the dual form was usually faster than in the primal.

In this paper we propose training L2 support vector regressors (SVRs) in the dual form in the similar way as in [5]. We use Mattera et al.'s formulation of the SVR [6], whose number of variables is the number of training data. Then the dual variables take real values. The training algorithm is as follows. Starting from the initial working set, we repeatedly solve the dual quadratic programming problem, delete from the working set the variables which are within the epsilon tube and add to the working set the data which are outside of the epsilon tube until the same working sets are obtained.

In Section 2, we explain L2 SVRs in the dual form, and in Section 3 we discuss training methods of SVRs. In Section 4, by computer experiment we demonstrate the effectiveness of the proposed method for some benchmark data sets.

## 2 L2 Support Vector Regressors in the Dual Form

Let the $M$ training input-output pairs be $(\mathbf{x}_i, y_i)$ $(i = 1, \dots, M)$, where $\mathbf{x}_i$ is the $i$th training input and $y_i$ is the associated output. The L2 SVR is trained by solving

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

$$\text{subject to} \quad y_i - \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) - b \le \varepsilon + \xi_i \quad \text{for} \quad i = 1, \dots, M, \tag{2}$$

$$\mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b - y_i \le \varepsilon + \xi_i^* \quad \text{for} \quad i = 1, \dots, M, \tag{3}$$

$$\xi_i \ge 0, \quad \xi_i^* \ge 0 \quad \text{for} \quad i = 1, \dots, M, \tag{4}$$

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, $\varepsilon$ is the parameter to define the epsilon tube, $\xi_i$ and $\xi_i^*$ are slack variables, and $C$ is the margin parameter that determines the trade-off between the magnitude of the margin and the estimation error of the training data.

The above optimization problem can be converted into the dual form introducing nonnegative slack variables $\alpha_i$ and $\alpha_i^*$ associated with the inequality constraints (2) and (3), 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 $\alpha_i$ and $\alpha_i^*$ appear only in the forms of $\alpha_i - \alpha_i^*$ and $\alpha_i + \alpha_i^*$ and both $\alpha_i$ and $\alpha_i^*$ are not positive at the same time, we can reduce the number of variables to half by replacing $\alpha_i - \alpha_i^*$ with real-valued $\alpha_i$ and $\alpha_i + \alpha_i^*$ with $|\alpha_i|$ [6]. Then, we obtain the following dual problem for the

L2 support vector regressor:

$$\begin{aligned} \text{maximize} \qquad & Q(\boldsymbol{\alpha}) = -\frac{1}{2}\sum_{i,j=1}^{M} \alpha_i\,\alpha_j \left( K(\mathbf{x}_i, \mathbf{x}_j) + \frac{\delta_{ij}}{C} \right) \\ & -\varepsilon \sum_{i=1}^{M} |\alpha_i| + \sum_{i=1}^{M} y_i\,\alpha_i \end{aligned} \tag{5}$$

$$\text{subject to} \qquad \sum_{i=1}^{M} \alpha_i = 0, \tag{6}$$

where $\alpha_i$ are dual variables associated with $\mathbf{x}_i$ and take real values, $K(\mathbf{x}, \mathbf{x}') = \boldsymbol{\phi}^\top(\mathbf{x})\,\boldsymbol{\phi}(\mathbf{x})$ is the kernel, and $\delta_{ij}$ is Kronecker's delta function. In the computer experiment we use the RBF kernel: $K(\mathbf{x}, \mathbf{x}') = \exp(-\gamma \|\mathbf{x} - \mathbf{x}'\|^2)$, where $\gamma$ is the parameter to control the radius of the spread.

The KKT complementarity conditions are

$$\alpha_i\,(\varepsilon + \xi_i - y_i + \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + b) = 0 \qquad \text{for} \quad \alpha_i \geq 0, \quad i = 1, \ldots, M, \tag{7}$$

$$\alpha_i\,(\varepsilon + \xi_i + y_i - \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) - b) = 0 \qquad \text{for} \quad \alpha_i < 0 \quad i = 1, \ldots, M, \tag{8}$$

$$C\,\xi_i = |\alpha_i| \qquad \text{for} \quad i = 1, \ldots, M. \tag{9}$$

Therefore, $b$ is obtained by

$$b = \begin{cases} y_i - \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) - \varepsilon - \dfrac{\alpha_i}{C} & \text{for} \quad \alpha_i > 0, \\[2ex] y_i - \mathbf{w}^\top \boldsymbol{\phi}(\mathbf{x}_i) + \varepsilon - \dfrac{\alpha_i}{C} & \text{for} \quad \alpha_i < 0, \quad i \in \{1, \ldots, M\}. \end{cases} \tag{10}$$

By the formulation, L2 SVRs are very similar to L2 SVMs. If the value of $\varepsilon$ is very small almost all training data become support vectors. In such a case the L2 SVR behaves very similar to the least squares (LS) SVRs. And for $\varepsilon = 0$ the L2 SVR is equivalent to the LS SVR.

## 3 Training Methods

We solve the equality constraint (6) for one variable and substitute it into (5). Then the optimization problem is reduced to the maximization problem without constraints. We divide the variables into the working set and the fixed set and solve the subproblem for the working set fixing the variables in the fixed set. In the next iteration process, we delete the variables that are within the $\varepsilon$-tube from the working set and add, from the fixed set, the variables that do not satisfy the KKT conditions and iterate optimizing the subproblem until the same solution is obtained. We discuss the method more in detail.

Consider solving (5) and (6) for the index set $S$. Solving the equality constraint in (6) for $\alpha_s$ $(s \in S)$, we obtain

$$\alpha_s = -\sum_{\substack{i \neq s, \\ i \in S}} \alpha_i. \tag{11}$$

Substituting (11) into (5), we obtain the following optimization problem

$$\text{maximize} \quad Q(\boldsymbol{\alpha}_S) = \mathbf{c}_S^\top \boldsymbol{\alpha}'_S - \frac{1}{2} {\boldsymbol{\alpha}'_S}^\top K_S \, \boldsymbol{\alpha}'_S, \tag{12}$$

where $\boldsymbol{\alpha}_S = \{\alpha_i | i \in S\}$, $\boldsymbol{\alpha}'_S = \{\alpha_i | i \neq s, i \in S\}$, $\mathbf{c}_S$ is the $(|S|-1)$-dimensional vector, $K_S$ is the $(|S|-1) \times (|S|-1)$ positive definite matrix, and

$$\begin{aligned} c_{S_i} &= \left\{ \begin{array}{lll} y_i - y_s & \text{for} & D(\mathbf{x}_i, y_i) \geq 0, D(\mathbf{x}_s, y_s) \geq 0 \\ y_i - y_s - 2\,\varepsilon & \text{for} & D(\mathbf{x}_i, y_i) \geq 0, D(\mathbf{x}_s, y_s) < 0 \\ y_i - y_s + 2\,\varepsilon & \text{for} & D(\mathbf{x}_i, y_i) < 0, D(\mathbf{x}_s, y_s) < 0 \quad i \neq s, \quad i \in S \end{array} \right. & (13) \\ K_{S_{ij}} &= K(\mathbf{x}_i, \mathbf{x}_j) - K(\mathbf{x}_i, \mathbf{x}_s) - K(\mathbf{x}_s, \mathbf{x}_j) \\ &\quad + K(\mathbf{x}_s, \mathbf{x}_s) + \frac{1 + \delta_{ij}}{C} \qquad \text{for} \quad i, j \neq s, \quad i, j \in S, & (14) \end{aligned}$$

where $c_{S_i}$ is the $i$th element of $\mathbf{c}_{S_i}$ and $D(\mathbf{x}, y) = y - \boldsymbol{\phi}(\mathbf{x}_i) + b$.

If $\varepsilon = 0$, in (13), $c_{S_i} = y_i - y_s$ irrespective of the signs of $D(\mathbf{x}_i, y_i)$ and $D(\mathbf{x}_s, y_s)$. Thus, similar to LS SVRs, we can solve (12) by a single matrix inversion.

Initially, for positive $\varepsilon$, we set some indices to $S$ and set $\alpha_i = 0$ $(i \in S)$ and $b = 0$. Therefore, $D(\mathbf{x}_i, y_i) = 0$. Thus, in (13), $c_{S_i}$ is set according to the signs of $y_i$ and $y_s$. We solve (12):

$$\boldsymbol{\alpha}'_S = K_S^{-1} \, \mathbf{c}_S. \tag{15}$$

We calculate $b$ using (10). Because of $\varepsilon$, the $b$ values calculated by different $\alpha_i$ in $S$ may be different. Thus we calculate the average of $b$s. If training data associated with the variables in $S$ are within the $\varepsilon$-tube, we delete these variables and add the indices of the variables to $S$ that violate KKT conditions. If the working sets and $\mathbf{c}_{S_i}$ are the same for the consecutive two iterations, the solution is obtained and we stop training. This stopping condition is different from that for the SVM discussed in [5] because of the absolute value of $\alpha_i$ in the objective function.

The procedure for training the L2 SVR is as follows.

1. Set the indices associated with $h$ training data to set $S$ and go to Step 2.

2. Calculate $\boldsymbol{\alpha}'_S$ using (15) and using (11) obtain $\alpha_s$. Calculate $b$s for $i \in S$ using (10) and calculate the average of $b$s.

3. Delete from $S$ the indices of $\mathbf{x}_i$ that satisfy $|D(\mathbf{x}_i, y_i)| \leq \varepsilon$. And add to $S$ the indices associated with at most $h$ most violating data, namely $\mathbf{x}_i$ that satisfy $|D(\mathbf{x}_i, y_i)| > \varepsilon$ from the largest $|D(\mathbf{x}_i, y_i)|$ in order. If the solution is obtained, stop training. Otherwise, go to Step 2.

Because the proposed method includes matrix inversion with the size of $|S| - 1$, training will become slow or prohibitive if the number of support vectors is very large. Therefore, the proposed method will be suited for small or medium size regression problems.

Table 1: Approximation errors of LS, L1, and L2 support vector regressors

| | LS | L1 | L2 |
|---|---|---|---|
| Abalone [7] | 1.51 | **1.48** | 1.51 |
| Boston 5 [8] | **0.0264** | 0.0270 | 0.0270 |
| Boston 14 [8] | **2.19** | **2.19** | **2.19** |
| Orange juice [9] | 4.78 | **4.68** | 4.79 |
| Water purification [10] | 0.976 | **0.958** | 0.976 |

For a large value of $\varepsilon$, there may be cases where a proper working set is not obtained and thus the solution does not converge. This is because the monotonic decrease of the objective function values is not guaranteed.

## 4 Performance Comparison

We compared the average estimation error and training time of the proposed method with those of other methods using the data sets listed in Table 1. For the data sets that are not divided into training and test data sets, i.e., abalone, Boston 5, and Boston 14, we randomly divided the set into two with almost equal sizes.

In our initial study we found that active set training for the L2 SVR failed to converge more frequently than for the L2 SVM [5] especially for a large value of $\varepsilon$. Therefore, to avoid non-convergence we set a value of $h$ larger than that for the L2 SVM, i.e., $h = 500$. Therefore, except for the abalone problem, initially all the training data were used for training.

We used the RBF kernel with the $\gamma$ value selected from $\{0.1, 0.5, 1, 5, 10, 15, 20\}$. The value of the margin parameter was selected from $\{1, 10, 50, 100, 500, 1,000, 2,000, 3,000, 5,000, 7,000, 10,000, 100,000\}$ and the $\varepsilon$ value for L1 and L2 SVRs from $\{0.001, 0.01, 0.05, 0.1, 0.5, 1\}$ and determined the parameter values by fivefold cross-validation. For the parameter values of the LS SVR and L1 SVR, we used the grid search. But for the L2 SVR, for a large value of $\varepsilon$, active set training sometimes failed to converge. Therefore, we first carried out cross-validation with $\varepsilon = 0.001$ and selected the $\gamma$ value. Then fixing the $\gamma$ value we carried out cross-validation changing $C$ and $\varepsilon$ values. The parameter values for the LS SVR and L2 SVR were very similar. Especially, the $\gamma$ values were the same because we started cross-validation of the L2 SVR with $\varepsilon = 0.001$, which was very close to the LS SVR.

Table 1 shows the average approximation errors. The smallest approximation errors are shown in boldface. The approximation errors for the three SVRs are almost identical.

Table 2 lists the training time in seconds, measured by a personal computer (3GHz, 2GB memory, Windows XP operating system). In the table "L1 PDIP" denotes that the L1 SVR was trained by the primal-dual interior-point method

Table 2: Training time comparison (s).

| Data | L1 PDIP | L2 NM | Active |
|---|---|---|---|
| Abalone | 446 | 198 | **58** |
| Boston 5 | 1.0 | 4.5 | **0.09** |
| Boston 14 | 1.6 | 1.4 | **0.09** |
| Orange Juice | 0.83 | 36 | **0.16** |
| Water purification | 0.80 | 0.23 | **0.13** |

and "L2 NM" denotes that the L2 SVR was trained by Newton's method with fixed-size chunking. For all the cases, active set training for the L2 SVR was the fastest.

## 5 Conclusions

In this paper we proposed a new training method for L2 SVRs. Namely, starting with an initial working set, we solve the subproblem expressed in the dual form by Newton's method, delete the data in the working set that are within the epsilon tube, add the data that are outside of the epsilon tube, and repeat solving the subproblem until the same working set is obtained. By computer experiments we show that the proposed method was faster than that of Newton's method with fixed-size chunking but for large epsilon value, there were cases where the training did not converge. We leave this problem in the future study.

## References

[1] E. Osuna, R. Freund, and F. Girosi, An improved training algorithm for support vector machines, In *Proc. NNSP '97*, pp. 276–285, 1997.

[2] C. Saunders, M. O. Stitson, J. Weston, L. Bottou, B. Schölkopf, and A. Smola, Support vector machine: Reference manual, Technical Report CSD-TR-98-03, Royal Holloway, University of London, 1998.

[3] J. C. Platt, Fast training of support vector machines using sequential minimal optimization, In B. Schölkopf et al., Eds., *Advances in Kernel Methods: Support Vector Learning*, pp. 185-208, MIT Press, 1999.

[4] O. Chapelle, Training a support vector machine in the primal, In L. Bottou et al., Eds., *Large-Scale Kernel Machines*, pp. 29–50. MIT Press, 2007.

[5] S. Abe, Is primal better than dual, In C. Alippi et al., Eds., *Proc. ICANN 2009 Part I*, pp. 854-863, 2009.

[6] D. Mattera, F. Palmieri, and S. Haykin, An explicit algorithm for training support vector machines, *IEEE Signal Processing Letters*, vol. 6, no. 9, pp. 243–245, 1999.

[7] A. Asuncion and D. J. Newman, UCI machine learning repository, http://www.ics.uci.edu/∼mlearn/MLRepository.html, 2007.

[8] Delve Datasets, http://www.cs.toronto.edu/˜delve/data/datasets.html.

[9] UCL Machine Learning Group, http://www.ucl.ac.be/mlg/index.php?page=home.

[10] S. Abe, *Support Vector Machines for Pattern Classification*, Springer-Verlag, 2005.