Fifth International Workshop on Computational Intelligence & Applications
IEEE SMC Hiroshima Chapter, Hiroshima University, Japan, November 10, 11 & 12, 2009

Support Vector Selection for Regression Machines

Wan-Jui Lee, Chih-Cheng Yang, and Shie-Jue Lee
Department of Electrical Engineering
National Sun Yat-Sen University
Kaohsiung 80424, Taiwan
leesj@mail.ee.nsysu.edu.tw

***Abstract*—In this paper, we propose a method to select support vectors to improve the performance of support vector regression machines. First, the orthogonal least-squares method is adopted to evaluate the support vectors based on their error reduction ratios. By selecting the representative support vectors, we can obtain a simpler model which helps avoid the over-fitting problem. Second, the simplified model is further refined by applying the gradient descent method to tune the parameters of the kernel functions. Learning rules for minimizing the regularized risk functional are derived. Experimental results have shown that our approach can improve effectively the generalization capability of support vector regressors.**

Keywords: Orthogonal least-squares, over-fitting, gradient descent, learning rules, error reduction ratio, mean square error.

I. Introduction

Support Vector Machines which form a new class of learning algorithms were motivated and derived from the research of statistical learning theory [8]. The decision boundary for classification problems is represented with a small subset of the training examples, called the support vectors. Although support vector machines were first developed for classification problems, due to the introduction of Vapnik's ε-insensitive loss function [1], they have been extended to solve a nonlinear regression estimation problem by representing the regression hyperplane with support vectors. Clearly, Support vector regression (SVR) is also expected to inherit the good support vector characteristic that it generalizes well to unseen data. Unlike the traditional methods that minimize the empirical training errors, support vector regression implements the structural risk minimization principle [9] which is based on the fact that the generalization error is bounded by the sum of the training error and a confidential interval term which depends on the Vapnik-Chervonenkis (VC) dimension. Consequently, SVR can find the optimal regression hyperplane that minimizes the training error for the training samples as well as maximizes the generalization capability for unseen testing samples.

Selecting a proper set of samples or rules to achieve better performance in system modeling has been studied for over a decade. In [2], an orthogonal least-squares (OLS) based learning algorithm for radial basis function (RBF) networks was proposed. Instead of randomly selecting RBF centers from samples, the OLS method selects a suitable set of RBF centers from them. Thus, the oversize and ill-conditioning problems occurring frequently in the random selection of centers can automatically be avoided. The OLS method involves the transformation of the sample set into a set of orthogonal basis vectors, and calculates the contribution of each individual basis vector to the desired output. Therefore, a significant subset of samples can be selected as RBF centers according to the error rate that a basis vector can reduce. Later, this method is adopted in [10] to simplify support vector based or fuzzy rule based models. However, it doesn't consider the reduction of similar (redundant) and correlated centers. By evaluating only the approximating capabilities of the rules, the OLS method often assign a high degree of importance to a set of redundant or correlated rules, and might result in poor generalization capabilities. To solve the problem, in [6], a method for detecting redundant and correlated rules is introduced to the OLS-based rule selection. If the inner product of a basis vector with itself is close to zero, it means that its corresponding rule is a linear combination of the previously selected rules, then it will not be selected.

In this paper, we adopt and extend the rule base reduction method in [6] to reduce the number of support vectors. The orthogonal least-squares method is applied to evaluate the support vectors based on their error reduction ratios. By selecting the representative support vectors, we can obtain a simpler model which helps avoid the over-fitting problem. Then the simplified model is refined by utilizing the structural risk minimization principle. We adopt the gradient descent method to derive learning rules for tuning the parameters of the associated kernel functions. Also, in order to capture the data structure of the training samples, we adjust the

distribution of all kernel functions in each dimension. Experimental results have shown that our approach can improve effectively the generalization capability of support vector regressors.

The rest of the paper is organized as follows. In Section II, the background of support vector regression is introduced. The construction of the regression hyperplane in the feature space and the effect of kernel functions is shown. Section III gives a description of our method. An Experiment is presented in Section IV. Finally, concluding remarks are given in Section V.

II. Support Vector Regression

Among the support vector regression (SVR) machines [7], [8], [1], the most commonly used is ε-SVR. The ε-SVR does not penalize the approximation errors less than the pre-specified tolerance ε and seeks to estimate the function

$$f(\mathbf{x}) =< \mathbf{w}, \mathbf{x} > +b, \mathbf{w}, \mathbf{x} \in \mathcal{R}^d, b \in \mathcal{R} \qquad (1)$$

based on the following set of training patterns:

$$(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n) \in \mathcal{R}^d \times \mathcal{R}, \qquad (2)$$

where n is the number of training patterns and $\mathcal{R}^d$ is the original space of the input patterns.

By ε-SVR, the function estimation problem can be converted to the following optimization problem:

$$\begin{aligned} \text{minimize } & \frac{\lambda}{2} < \mathbf{w}, \mathbf{w} > + \sum_{i=1}^{n} (\xi_i + \xi_i^*), \\ \text{subject to } & < \mathbf{w}, \mathbf{x}_i > +b - y_i \leq \varepsilon + \xi_i \\ & y_i - < \mathbf{w}, \mathbf{x}_i > +b \leq \varepsilon + \xi_i^* \\ & \xi_i, \xi_i^* \geq 0 \end{aligned} \qquad (3)$$

which, by introducing Lagrange multipliers, can be replaced by the following dual optimization problem:

$$\begin{aligned} \text{maximize } & -\varepsilon \sum_{i=1}^{n} (\alpha_i^* + \alpha_i) + \sum_{j=1}^{n} (\alpha_j^* - \alpha_j) y_i \\ & - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) < \mathbf{x}_i, \mathbf{x}_j >, \\ \text{subject to } & \sum_{i=1}^{n} (\alpha_i^* - \alpha_i) = 0, \alpha_i, \alpha_i^* \in [0, \frac{1}{\lambda}]. \end{aligned} \qquad (4)$$

Interestingly, the weight vector of the optimal regression hyperplane can be found to be

$$\mathbf{w} = \sum_{i=1}^{n} (\alpha_i^* - \alpha_i) \mathbf{x}_i \qquad (5)$$

and therefore the optimal regression hyperplane of Eq.(1) can be represented as

$$\begin{aligned} f(\mathbf{x}) &= (\mathbf{x}^{\mathbf{T}} \sum_{i=1}^{n} (\alpha_i^* - \alpha_i) \mathbf{x}_i) + b \\ &= (\sum_{i=1}^{n} (\alpha_i^* - \alpha_i) < \mathbf{x}, \mathbf{x}_i >) + b \end{aligned} \qquad (6)$$

where b can be derived from Eq.(6) as given below:

$$b = \frac{1}{n} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{n} (\alpha_j^* - \alpha_j) < \mathbf{x}_i, \mathbf{x}_j >). \qquad (7)$$

To solve a nonlinear problem, we can map the problem from the original space to a feature space through a nonlinear transformation with suitably chosen kernel functions and then finding a linear model in the feature space. The linear model obtained in the feature space corresponds to a nonlinear model in the original space. For this purpose, consider a mapping $\mathbf{\Phi}(\mathbf{x})$ from the input space into a feature space as

$$\Phi : \mathcal{R}^d \rightarrow \mathcal{H}. \qquad (8)$$

Then the training algorithm would only depend on the data through dot products in $\mathcal{H}$, i.e. on functions of the form $< \mathbf{\Phi}(\mathbf{x}_i), \mathbf{\Phi}(\mathbf{x}_j) >$. Suppose there is a kernel function K such that

$$K(\mathbf{x}_i, \mathbf{x}_j) =< \mathbf{\Phi}(\mathbf{x}_i), \mathbf{\Phi}(\mathbf{x}_j) >, \qquad (9)$$

we would only need to use K in the training algorithm, and would never need to explicitly even know what $\mathbf{\Phi}$ is. The dot product in the feature space can be expressed as a kernel function. Therefore, for a nonlinear problem, we have the following regression function

$$f(\mathbf{x}) = \sum_{i=1}^{n} (\alpha_i^* - \alpha_i) K(\mathbf{x}, \mathbf{x}_i) + b, \qquad (10)$$

and b is

$$b = \frac{1}{n} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{n} (\alpha_j^* - \alpha_j) K(\mathbf{x}_i, \mathbf{x}_j)). \qquad (11)$$

Note that the summation term of Eq.(10) does not include all the n training samples. Instead, only those input vectors $\mathbf{x}_i$ with $|\alpha_i^* - \alpha_i| > 0$ have contributions to the summation and these input vectors are called support vectors. In this paper, we will use the Gaussian function as the kernel function, therefore

$$K(\mathbf{x}_i, \mathbf{x}_j) = e^{-(\sum_{k=1}^{d} \frac{\|x_{ik} - x_{jk}\|^2}{\sigma_{jk}^2})}. \qquad (12)$$

Clearly, each support vector is the center of its corresponding Gaussian function.

III. Support Vector Selection

Usually, the number of support vectors derived from support vector learning is unnecessarily large due to the settings for the trade-off parameter λ and the kernel function. Similar or less important support vectors may be generated, resulting in a higher chance of over-fitting and decreasing in generalization capability. By selecting significant support vectors and ignoring insignificant support ones, we are able to construct a proper model without the need of optimal settings for the trade-off parameter and the kernel function, and avoid the over-fitting problem.

Suppose we are given a set of n training examples. By using ε-SVR, m support vectors are obtained. For simplicity, assume that these support vectors are $\mathbf{x}_1$, $\mathbf{x}_2$, $\ldots$, $\mathbf{x}_m$ with associated multipliers $\alpha_1^* - \alpha_1$, $\alpha_2^* - \alpha_2$, $\ldots$, $\alpha_m^* - \alpha_m$. We adopt the orthogonal least-squares method [3] to provide a systematic way for support vector selection. Only significant support vectors are kept and insignificant ones are ignored. As a result, the support vector regressor can be significantly simplified. By introducing all the n training patterns into Eq.(10), we can express the regressor as

$$\begin{aligned}
y_1 &= (\alpha_1^* - \alpha_1)K(\mathbf{x}_1, \mathbf{x}_1) + (\alpha_2^* - \alpha_2)K(\mathbf{x}_2, \mathbf{x}_1) \\
&\quad + \cdots + (\alpha_m^* - \alpha_m)K(\mathbf{x}_m, \mathbf{x}_1) + b + e_1 \\
y_2 &= (\alpha_1^* - \alpha_1)K(\mathbf{x}_1, \mathbf{x}_2) + (\alpha_2^* - \alpha_2)K(\mathbf{x}_2, \mathbf{x}_2) \\
&\quad + \cdots + (\alpha_m^* - \alpha_m)K(\mathbf{x}_m, \mathbf{x}_2) + b + e_2 \\
&\vdots \quad \vdots \quad \vdots \\
y_n &= (\alpha_1^* - \alpha_1)K(\mathbf{x}_1, \mathbf{x}_n) + (\alpha_2^* - \alpha_2)K(\mathbf{x}_2, \mathbf{x}_n) \\
&\quad + \cdots + (\alpha_m^* - \alpha_m)K(\mathbf{x}_m, \mathbf{x}_n) + b + e_n
\end{aligned}$$

where $e_1, e_2, \cdots, e_n$ are approximation errors between the desired and actual outputs. Note that $\mathbf{x}_i$ is a support vector if and only if $|\alpha_i^* - \alpha_i| > 0$. Let

$$\mathbf{P} = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1m} \\ p_{21} & p_{22} & \cdots & p_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nm} \end{bmatrix}, \quad (13)$$

$$\theta = [\theta_1 \theta_2 \cdots \theta_m]^T, \quad (14)$$

$$\mathbf{y} = [y_1, y_2, \cdots, y_n]^T, \quad (15)$$

$$\mathbf{E} = [b + e_1, b + e_2, \cdots, b + e_n]^T \quad (16)$$

where

$$\begin{aligned}
p_{ij} &= \frac{(\alpha_j^* - \alpha_j)K(\mathbf{x}_j, \mathbf{x}_i)}{\sum_{i=1}^m (\alpha_i^* - \alpha_i)K(\mathbf{x}_j, \mathbf{x}_i)}, \\
\theta_j &= \sum_{i=1}^m (\alpha_i^* - \alpha_i)K(\mathbf{x}_j, \mathbf{x}_i).
\end{aligned}$$

The regressor can be rewritten as

$$\mathbf{y} = \mathbf{P}\theta + \mathbf{E}.$$

Note that $\mathbf{E}$ includes the bias and the approximation error. From Eq.(13), we see that there is a one-to-one correspondence between the support vectors and the column vectors in $\mathbf{P}$. Each time a support vector is selected in such a manner that the variance increment of the desired output is maximized.

By using the Gram-Schmidt orthogonalization, $\mathbf{P}$ can be decomposed as

$$\mathbf{P} = \mathbf{WA} \quad (17)$$

where $\mathbf{A}$ is an $m \times m$, upper triangular matrix with 1s on the main diagonal, and $\mathbf{W}$ is an $n \times m$ matrix with orthogonal columns $\mathbf{w}_i, 1 \leq i \leq m$, such that

$$\mathbf{w}_i{}^T \mathbf{w}_j = 0, i \neq j. \quad (18)$$

Substituting Eq.(17) into Eq.(III), we have

$$\mathbf{y} = \mathbf{WA}\theta + \mathbf{E} = \mathbf{Wg} + \mathbf{E} \quad (19)$$

where $\mathbf{g} = \mathbf{A}\theta$. A least-squares solution for $\mathbf{g}$ is given by

$$\mathbf{g} = (\mathbf{W}^T\mathbf{W})^{-1}\mathbf{W}^T\mathbf{y} \quad (20)$$

and the ith coordinate of $\mathbf{g}$ is

$$g_i = \frac{\mathbf{w_i}^T\mathbf{y}}{\mathbf{w_i}^T\mathbf{w_i}}. \quad (21)$$

Since $\mathbf{w}_i$ and $\mathbf{w}_j$ are orthogonal for $i \neq j$, we have

$$\begin{aligned}
\mathbf{y^T y} &= \mathbf{g}^T\mathbf{W}^T\mathbf{W}\mathbf{g} + \mathbf{E}^T\mathbf{E} \\
&= \sum_{i=1}^m g_i^2 \mathbf{w_i}^T\mathbf{w_i} + \mathbf{E}^T\mathbf{E}. \quad (22)
\end{aligned}$$

Note that $\sum_{i=1}^m g_i^2 \mathbf{w_i}^T\mathbf{w_i}$ is related to the portion of the output energy explained by the regression, and the ith term in the summation represents the increment in the energy introduced by the inclusion of the i column vector in $\mathbf{P}$. Since there is a one-to-one correspondence between the column vectors in $\mathbf{P}$ and the support vectors. We can define the error reduction ratio due to the support vector x_i as

$$[err]_i = \frac{g_i^2 \mathbf{w_i}^T\mathbf{w_i}}{\mathbf{y^T y}}, 1 \leq i \leq m. \quad (23)$$

This ratio offers a simple criterion for selecting support vectors. Each time a support vector is selected such that the error reduction ratio is maximal.

Note that the error reduction ratio only tries to minimize the training error without considering the model structure. Thus, it is possible that a fairly redundant support vector has a high ratio because of its contribution to the output. Similar to the idea proposed in [6], we add a requirement of $h_k \geq \epsilon$ on selecting the kth most significant support vector, with $\epsilon > 0$ being a

user-defined value. The reason is that when h_k is too small, the corresponding column vector is nearly a linear combination of the column vectors corresponding to the previously selected $k-1$ support vectors and thus the support vector can be ignored.

When a subset of support vectors is selected, we proceed to refine the simplified regressor by applying the gradient descent (GD) method to tune the parameters of the kernels. The GD method is a well-known optimization method and can be represented by the following equation:

$$\omega^{(t+1)} = \omega^{(t)} + \eta \cdot \frac{\partial F(\omega)}{\partial \omega}|_{\omega=\omega^{(t)}}, \tag{24}$$

where ω is the parameter to be tuned, η is the learning rate and $F(\omega)$ is the error function on ω. In this section, we choose the kernel functions to be Gaussian functions which are most commonly used because of their good performance. However, the GD method can be applied to other differentiable kernel functions as well.

The objective of the learning process is to find a regression hyperplane $f(\mathbf{x})$ which minimizes the regularized risk functional

$$R_{ref} = \frac{\lambda}{2} < \mathbf{w}, \mathbf{w} > + R_{emp}[f(\mathbf{x})]. \tag{25}$$

Here, $< \mathbf{w}, \mathbf{w} >$ is the term which characterizes the model complexity, and

$$R_{emp}[f(\mathbf{x})] = \frac{1}{n}\sum_{i=1}^{n}(y_i - f(\mathbf{x}_i))^2 \tag{26}$$

measures the training error, with $\lambda > 0$ being a trade-off constant. Minimizing Eq.(25) captures the main insight of statistical learning theory. In order to obtain a small risk, one needs to control both the training error and the model complexity.

Suppose, after support vector selection, we have s support vectors $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m$ with associated Lagrange multipliers $\alpha_1^* - \alpha_1$, $\alpha_2^* - \alpha_2$, $\ldots$, $\alpha_m^* - \alpha_m$. Each support vector is the center of the corresponding kernel function. Keeping the centers fixed, we can improve the performance of the regressor by tuning the variances of the kernel functions. By applying the GD method, we are able to derive the learning rules for optimizing the variances, σ_{ij}, of the Gaussian functions, under the given support vectors and Lagrange multipliers. The learning rules are applied iteratively until convergence is achieved. Substituting Eq.(10) and Eq.(5) into Eq.(25), we have

$$\begin{aligned} R_{ref} &= \frac{\lambda}{2} < \mathbf{w}, \mathbf{w} > + R_{emp}[f(\mathbf{x})] \\ &= \frac{\lambda}{2}\sum_{i=1}^{s}\sum_{j=1}^{s}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)K(\mathbf{x}_i, \mathbf{x}_j) + \\ &\quad \frac{1}{n}\sum_{i=1}^{n}(\sum_{j=1}^{s}(\alpha_j^* - \alpha_j)K(\mathbf{x}_i, \mathbf{x}_j) + b - y_i)^2. \end{aligned}$$

Replacing the kernel function by the Gaussian function given in Eq.(12), we have

$$\begin{aligned} R_{ref} &= \frac{\lambda}{2}\sum_{i=1}^{s}\sum_{j=1}^{s}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) \\ &\quad e^{-\sum_{k=1}^{d}(\frac{x_{ik}-x_{jk}}{\sigma_{jk}})^2} \\ &\quad + \frac{1}{n}\sum_{i=1}^{n}(\sum_{j=1}^{s}(\alpha_j^* - \alpha_j)e^{-\sum_{k=1}^{d}(\frac{x_{ik}-x_{jk}}{\sigma_{jk}})^2} + \\ &\quad b - y_i)^2 \\ &= \frac{\lambda}{2}\sum_{i=1}^{s}\sum_{j=1}^{s}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j) \\ &\quad e^{-\sum_{k=1}^{d}(\frac{x_{ik}-x_{jk}}{\sigma_{jk}})^2} \\ &\quad + \frac{1}{n}\sum_{i=1}^{n}(\sum_{j=1}^{s}(\alpha_j^* - \alpha_j)e^{-\sum_{k=1}^{d}(\frac{x_{ik}-x_{jk}}{\sigma_{jk}})^2})^2 \\ &\quad + \frac{2}{n}\sum_{i=1}^{n}(b - y_i)\sum_{j=1}^{n}(\alpha_j^* - \alpha_j) \\ &\quad e^{-\sum_{k=1}^{d}(\frac{x_{ik}-x_{jk}}{\sigma_{jk}})^2} + \frac{1}{n}\sum_{i=1}^{n}(b - y_i)^2. \end{aligned} \tag{27}$$

Then, we can derive the gradient of R_{ref} with respect to σ_{fp}, $1 \le f \le s$, $1 \le p \le d$, as

$$\begin{aligned} \frac{\partial R_{ref}}{\partial \sigma_{fp}} &= \lambda(\alpha_f^* - \alpha_f)\sum_{i=1}^{n}(\alpha_i^* - \alpha_i)\frac{(x_{ip} - x_{fp})^2}{{\sigma_{fp}}^3} \\ &\quad e^{-\sum_{k=1}^{d}(\frac{x_{ik}-x_{fk}}{\sigma_{fk}})^2} \\ &\quad + \frac{4}{n}(\alpha_f^* - \alpha_f) \\ &\quad \sum_{i=1}^{n}\frac{(x_{ip} - x_{fp})^2}{{\sigma_{fp}}^3}e^{-\sum_{k=1}^{d}(\frac{x_{ik}-x_{fk}}{\sigma_{fk}})^2} \\ &\quad [\sum_{j=1}^{s}(\alpha_j^* - \alpha_j)e^{-\sum_{k=1}^{d}(\frac{x_{ik}-x_{jk}}{\sigma_{jk}})^2}] \\ &\quad + \frac{4}{n}(\alpha_f^* - \alpha_f)\sum_{i=1}^{n}(b - y_i)\frac{(x_{ip} - x_{fp})^2}{{\sigma_{fp}}^3} \\ &\quad e^{-\sum_{k=1}^{d}(\frac{x_{ik}-x_{fk}}{\sigma_{fk}})^2}. \end{aligned} \tag{28}$$

Furthermore, the gradient descent method in Eq.(24)

suggests the following learning rule for σ_{fp}:

$$\sigma_{fp}{}^{(t+1)} = \sigma_{fp}{}^{(t)} + \eta \cdot \frac{\partial R_{ref}}{\partial \sigma_{fp}}|_{\sigma_{fp}=\sigma_{fp}{}^{(t)}}, \qquad (29)$$

where $1 \leq f \leq s$ and $1 \leq p \leq d$. Therefore, we can adjust the variances of the kernel functions by Eq.(28) and Eq.(29).

IV. An Experiment

We use a high-dimensional synthetic dataset in this experiment to compare the performance of ε-SVR and our method. Consider the following nonlinear function:

$$y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + 0(x_6 + x_7 + x_8 + x_9 + x_{10}) + \varphi \qquad (30)$$

where φ is a Gaussian noise with zero mean and unit variance and the inputs $x_1, ..., x_{10}$ are sampled independently from a uniform [0,1] distribution. We randomly generate a training dataset with 1000 patterns, and a testing dataset with 200 patterns. There are no identical patterns in the training and testing datasets. We run ε-SVR and our method, respectively, on the training and testing datasets, and the results obtained under different values of σ are shown in Table I and Figure 1.

Table I shows the reduction percentage on the number of support vectors achieved by our support vector selection method. From the table, it is clear to see that support vector selection is effective, especially for the case with smaller values of σ (or equivalently, larger $\frac{1}{\sigma^2}$). For example, for the case with $\frac{1}{\sigma^2} = 5$, about 40% of the support vectors are removed. Fewer support vectors are reduced for the case with larger values of σ. The magnitude of σ indicates the range a support vector covers, and a larger value means a broader range covered. Therefore, it is more difficult to ignore those support vectors with large values of σ. The error comparison between ε-SVR and the simplified model after support vector selection is given in Figure 1(a). Clearly, the simplified model after support vector selection works almost equally well with ε-SVR even though the former has fewer support vectors. This indicates that the ignored support vectors are indeed insignificant ones, and therefore validates the effectiveness of our support vector selection method.

Then we proceed to parameter learning to refine the obtained simplified model. The error comparison between the refined model and ε-SVR is shown in Figure 1(b). Clearly, the refined model performs better than ε-SVR, especially for the case with smaller values of σ. For the case with a smaller σ, the corresponding kernel is more localized, and therefore ε-SVR tends to deteriorate due to over-fitting.

V. Conclusions

We have proposed an effective approach to improve the performance of support vector regressors. The orthogonal least-squares method is first applied to obtain a simpler support vector model. Significant support vectors are kept and insignificant ones are removed. The simplified model helps to avoid the over-fitting problem. Then the gradient descent method is applied to tune the parameters of the kernel functions. Learning rules for minimizing the regularized risk functional are derived. An experiment have been presented to show that the generalization capability of support vector regressors can be improved with a smaller number of support vectors and learned kernel functions.

Acknowledgment

This work was supported by the Ministry of Economic Affairs under the grant 98-EC-17-A-02-S2-0114.

References

[1] C. C. Chang and C. J. Lin , "Training ν-Support Vector Regression: Theory and Algorithms ," *Neural Computation* , vol. 14, pp. 1959-1977, 2002.

[2] S. Chen, C. F. N. Cowan, and P. M. Grant, "Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks," *IEEE Transactions on Neural Networks*, vol. 2, no. 2, pp. 302-309, 1991.

[3] F. M. Ham and I. Kostanic, *Principles of Neurocomputing for Science and Engineering*, Boston, USA: McGraw-Hill, 2001.

[4] K. Kobayashi, D. Kitakoshi, and R. Nakano, "Yet Faster Method to Optimize SVR Hyperparameters Based on Minimizing Cross-Validation Error," Proc. International Joint Conference on Neural Networks, 2005, pp. 871–876.

[5] S. J. Lee and C. L. Hou, "An ART-Based Construction of RBF Networks," it IEEE Transactions on Neural Networks, vol. 13, no. 6, pp. 1308-1321, 2002.

[6] M. Setnes and R. Babuška, "Rule Base Reduction: Some Comments on the Use of Orthogonal Transforms," *IEEE Transactions on System, Man, and Cybernetics-Part C: Applications and Reviews*, vol. 31, no. 2, pp. 199-206, 2001.

[7] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge, UK: Cambridge University Press, 2004.

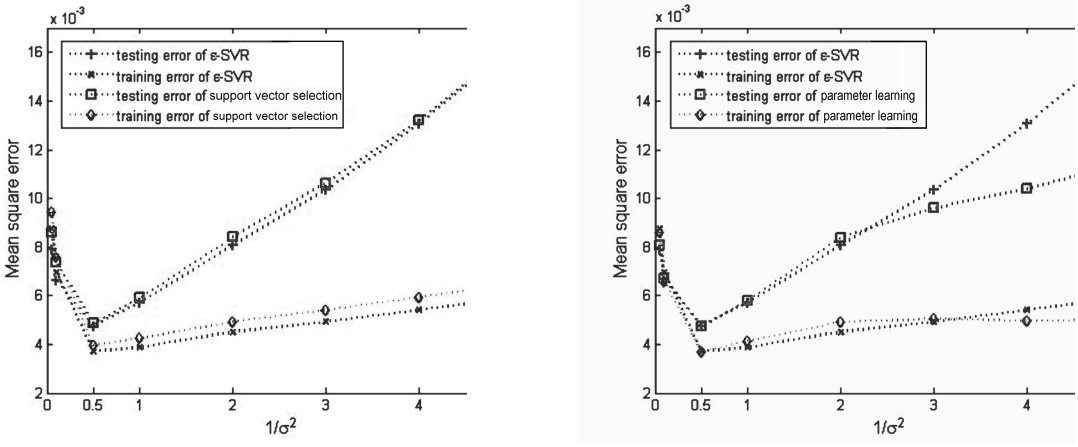
[8] A. J. Smola and B. Scholkopf, "A Tutorial On Support Vector Regression," *Statistics and Computing* , vol. 14, no. 3, pp. 199-222, 2004.

[9] Y. Tan and J. Wang, "A Support Vector Machine with a Hybrid Kernel and Minimal Vapnik-Chervonenkis Dimension," *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 4, pp. 385- 395, 2004.

[10] X. X. Wang, S. Chen, D. Lowe, and C. J. Harris, "Sparse Support Vector Regression Based on Orthogonal Forward Selection for the Generalizaed Kernel Models," *Neurocomputing*, vol. 70, no. 1-3, pp. 462-474, 2006.

TABLE I
REDUCTION PERCENTAGE ON THE NUMBER OF SUPPORT VECTORS (SVS)

$\frac{1}{\sigma^2}$	0.05	0.1	0.5	1	2	3	4	5
# of SVs by ε-SVR	218	189	116	140	192	237	285	341
# of SVs after selection	202	173	103	107	129	157	180	207
reduction percentage (%)	7.3	8.4	11.2	23.5	32.8	33.7	36.8	39.2



(a) (b)

Fig. 1. Performance comparison: (a) ε-SVR vs support vector selection; (b) ε-SVR vs (support vector selection + parameter learning).