Evolutionary Support Vector Regression based on Multi-Scale Radial Basis Function Kernel

Tanasanee Phienthrakul and Boonserm Kijsirikul

***Abstract*—Kernel functions are used in support vector regression (SVR) to compute the inner product in a higher dimensional feature space. The performance of approximation depends on the chosen kernels. The radial basis function (RBF) kernel is a Mercer's kernel that has been widely used in many problems. However, it still has the restriction in some complex problems. In order to obtain a more flexible kernel function, the multi-scale RBF kernels are combined by non-negative weighting linear combination. This proposed kernel is proved to be a Mercer's kernel. Then, the evolutionary strategy (ES) is applied for adjusting the parameters of SVR and kernel function. Moreover, subsets cross-validation is used for evaluating these parameters. The optimum values of these parameters are searched by (5+10)-ES. The experimental results show the ability of the proposed method that outperforms the statistical techniques.**

I. Introduction

SUPPORT vector machines (SVMs) are learning algorithms proposed by Vapnik et al. [1], based on the idea of the empirical risk minimization principle. They have been widely used in many applications such as pattern recognitions and function approximations. Basically, SVM operates a linear hyperplane in an augmented space by means of some defined kernels satisfying Mercer's condition [1], [2], [3].

These kernels map the input vectors into a very high dimensional space, possibly of infinite dimension, where a linear hyperplane is more likely [3]. There are many types of kernel functions such as linear kernels, polynomial kernels, sigmoid kernels, and RBF kernels. Each kernel function is suitable for some tasks, and it must be chosen for the tasks under consideration by hand or using prior knowledge [4].

The RBF kernel is a most successful kernel in many problems, but still has the restrictions in some complex problems. Therefore, we propose to improve the efficiency of approximation by using the combination of RBF kernels at different scales. These kernels are combined by including weights.

The weights, the widths of the RBF kernels, the deviation of approximation, and the regularization parameter of SVM are the adjustable parameters; these parameters are called *hyperparameters*. In general, these hyperparameters are usually determined by a grid search. The hyperparameters are varied with a fixed step-size in a range of values, but this kind of search consumes a lot of time.

Hence, we propose to use the evolutionary strategies (ESs) for choosing these hyperparameters. However, the objective function is an important part in evolutionary algorithms. There are many ways to measure the fitness of the hyperparameters. In this work, we propose to use subsets cross-validation for evaluating the hyperparameters in the evolutionary process.

We give a short description of support vector regression in Section II. In Section III and Section IV we propose the multi-scale RBF kernel and apply evolutionary strategies to determine the appropriate hyperparameters, respectively. The proposed kernels with the help of ES are tested in Section V. Finally, the conclusions are given in Section VI.

II. Support Vector Regression

The support vector machine is a learning algorithm that can be divided into support vector classification and support vector regression. Support Vector Regression (SVR) is a powerful method that is able to approximate a real valued function in terms of a small subset (called support vectors) of the training examples. Suppose we are given training data $\{(x_1, y_1), \ldots, (x_l, y_l)\} \subset X \times R$, where X denotes the space of input patterns.

In ε-SV regression, our goal is to find a function $f(x)$ that has at most ε deviation from the actually obtained target y_i for all the training data, and at the same time is as flat as possible. In other words, we do not care about errors as long as they are less than ε, but we do not accept any deviation larger than this [5].

We begin by describing the case of linear functions f, taking in form:

$$f(x) = \langle w, x \rangle + b \text{ with } w \in X,\ b \in R. \quad (1)$$

Flatness in this case means that one seeks a small w. One way to ensure this is to minimize the norm, i.e. $\|w\|^2 = \langle w, w \rangle$. We can write this problem as a convex optimization problem:

This work was supported by the Thailand Research Fund and the Royal Golden Jubilee Ph.D. Program.

T. Phienthrakul is with the Department of Computer Engineering, Chulalongkorn University, Bangkok 10330 Thailand (phone: 66-2-218-6956; fax: 66-2-218-6955; e-mail: tanasanee@yahoo.com).

B. Kijsirikul, Dr., is with the Department of Computer Engineering, Faculty of Engineering, Chulalongkorn University, Bangkok 10330 Thailand (e-mail: boonserm.k@chula.ac.th).

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|w\|^2 \\ \text{subject to} \quad & y_i - \langle w, x_i \rangle - b \leq \varepsilon \\ & \langle w, x_i \rangle + b - y_i \leq \varepsilon \end{aligned} \tag{2}$$

Sometimes, we may want to allow for some errors. Soft margin loss function was adapted to SV machines; one can introduce slack variables ξ_i, ξ_i^* to cope with otherwise infeasible constraints of the optimization problem [5]. Hence we will get

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \\ \text{subject to} \quad & y_i - \langle w, x_i \rangle - b \leq \varepsilon + \xi_i \\ & \langle w, x_i \rangle + b - y_i \leq \varepsilon + \xi_i^* \\ & \xi_i, \ \xi_i^* \geq 0 \end{aligned} \tag{3}$$

The constant $C > 0$ determines the trade-off between the flatness of f and the amount up to which deviations larger than ε are tolerated. In most cases this optimization problem can be solved more easily in its dual formulation [5]. An example of a linear SVR is shown in Fig. 1.

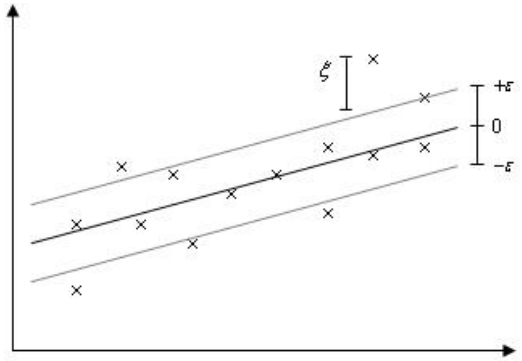

Fig. 1. An example of the soft margin for a linear SVR.

Moreover, seeking a proper linear hyperplane in an input space has the restrictions. There is an important technique that enables these machines to produce complex nonlinear approximation inside the original space. This performs by mapping the input space into a higher dimensional feature space through a mapping function Φ [6]. This can be achieved by substituting $\Phi(x_i)$ into each training example x_i. However, a good property of SVM is that it is not necessary to know the explicit form of Φ. Only the inner product in the feature space, called kernel function $K(x,y) = \Phi(x) \cdot \Phi(y)$, must be defined.

The function which can be a kernel must satisfy Mercer's condition [7]. Some of the common kernels are shown in Table I. Each kernel corresponds to some feature space and because no explicit mapping to this feature space occurs, optimal linear separators can be found efficiently in the feature space with millions of dimensions [8].

TABLE I
COMMON KERNEL FUNCTIONS

Kernel	Formula
Linear	$K(x,y) = x \cdot y$
Polynomial	$K(x,y) = (1 + x \cdot y)^d$
Sigmoid	$K(x,y) = \tanh(\alpha \ x \cdot y + \beta)$
Exponential RBF	$K(x,y) = \exp(-\gamma \|x - y\|)$
Gaussian RBF	$K(x,y) = \exp(-\gamma \|x - y\|^2)$
Multi-quadratic	$K(x,y) = -\sqrt{\|x - y\|^2 + c^2}$

III. MULTI-SCALE RBF KERNEL

The Gaussian RBF kernel is widely used in many problems. It uses the Euclidean distance between two points in the original space to find the correlation in the augmented space [3]. This correlation is rather smooth, and there is only one parameter for adjusting the width of RBF, which is not powerful enough for some complex problems. In order to get a better kernel, the combination of RBF kernels at different scales is proposed. The analytic expression of this kernel is the following:

$$\mathrm{K}(x,y) = \sum_{i=1}^{n} a_i K(x,y,\gamma_i), \tag{4}$$

where n is a positive integer, a_i for $i = 1,...,n$ are the arbitrary non-negative weighting constants, and $K(x,y,\gamma_i) = \exp(-\gamma_i \|x - y\|^2)$ is the RBF kernel at the width γ_i for $i = 1,...,n$.

In general, the function which maps the input space into the augmented feature space is unknown. However, the existence of such function is assured by Mercer's theorem [4]. The Mercer's theorem [2], [4] is shown in Fig. 2.

Mercer's Theorem. Any symmetric function $K(x,y)$ in the input space can represent an inner product in the feature space if

$$\iint K(x,y)\, g(x)\, g(y)\, dx\, dy \geq 0$$

is valid for all $g \neq 0$ for which $\int g^2(u)\, du < \infty$. Then the kernel function K can be expanded in terms of Φ_i

$$K(x,y) = \sum_{i=1}^{\infty} \lambda_i \Phi_i(x)\, \Phi_i(y)$$

with $\lambda_i \geq 0$. In this case, the mapping function from the input space to the feature space is expressed as

$$\Phi: x \rightarrow \left(\sqrt{\lambda_1}\, \Phi_1(x),\ \sqrt{\lambda_2}\, \Phi_2(x), \ldots \right)$$

such that K can be the inner product

$$\Phi(x) \cdot \Phi(y) = \sum_{i=1}^{\infty} \lambda_i \Phi_i(x)\, \Phi_i(y) = K(x,y).$$

Fig. 2. The Mercer's theorem.

The proposed kernel functions can be proved to be an admissible kernel by the Mercer's theorem. The proving process is shown in Fig. 3. The RBF is a well-known Mercer's kernel. Therefore, the non-negative linear combination of RBFs in (4) can be proved to be the Mercer's kernel.

Corollary. The non-negative linear combination of Mercer's kernels is a Mercer's kernel.

Proof. Let $K(x,y,\gamma_i)$ be the Mercer's kernels with the parameter γ_i, for $i=1,...,n$, and let $\mathrm{K}(x,y)=\sum_{i=1}^{n} a_i K(x,y,\gamma_i)$ when a_i for $i=1,...,n$ are non-negative real values.

According to Mercer's theorem, we know that

$$\iint K(x,y,\gamma_i)\, g(x)\, g(y)\, dx\, dy \;\geq\; 0\,,\ \forall g \quad \text{for } i=1,...,n\,.$$

By taking the linear combination with non-negative coefficients a_i, we will get

$$\sum_{i=1}^{n} a_i \iint K(x,y,\gamma_i)\, g(x)\, g(y)\, dx\, dy \;\geq\; 0\,.$$

Then

$$\iint \sum_{i=1}^{n} a_i K(x,y,\gamma_i)\, g(x)\, g(y)\, dx\, dy \;\geq\; 0\,.$$

Therefore,

$$\iint \mathrm{K}(x,y)\, g(x)\, g(y)\, dx\, dy \;\geq\; 0\,,\ \forall g\,.$$

Hence, the function $\mathrm{K}(x,y)=\sum_{i=1}^{n} a_i K(x,y,\gamma_i)$ is a Mercer's kernel. □

Fig. 3. The proof of the proposed kernel.

As shown in (4), there are $2n$ parameters when n terms of RBF kernels are used (n parameters for adjusting weights and n values of the widths of RBFs). However, we notice that the number of parameters can be reduced to $2n-1$ by fixing a value of the first parameter to 1. The multi-scale RBF kernel becomes as follows

$$\mathrm{K}(x,y)=K(x,y,\gamma_0)+\sum_{i=1}^{n-1} a_i K(x,y,\gamma_i)\,. \qquad (5)$$

This form of the multi-scale RBF kernel will be used in the rest of this paper.

IV. Evolutionary Strategies for SVR based on Multi-Scale RBF Kernel

Evolutionary strategies (ES, [9]) are based on the principles of adaptive selection found in the natural world. ES has been successfully used to solve various types of optimization problems [10]. Each generation (iteration) of the ES algorithm takes a population of individuals (potential solutions) and modifies the problem parameters to produce offspring (new solutions) [11]. Only the highest fit individuals (better solutions) survive to produce new generations [11].

In order to obtain appropriate values of the hyperparameters, ES is considered. There are several different versions of ES. Nevertheless, we prefer to use the $(\mu+\lambda)$-ES where μ parents produce λ offspring. Both parents and offspring compete equally for survival [11]. The (5+10)-ES is applied to adjust these hyperparameters, and this algorithm is shown in Fig. 4.

```
t = 0;
initialization( v_1,..., v_5 , σ );
evaluate f(v_1),..., f(v_5) ;
while (t < 500) do
    for i =1 to 10 do
        v'_i = recombination( v_1,..., v_5 );
        v'_i = mutate (v'_i) ;
        evaluate f(v'_i) ;
    end
    ( v_1,..., v_5 ) = select( v_1,..., v_5 , v'_1,..., v'_10 )
    σ = mutate_σ(σ) ;
    t = t+1;
end
```

Fig. 4. (5+10)-ES algorithm.

This algorithm uses 5 solutions to produce 10 new solutions by a recombination method. These new solutions are mutated and evaluated, but only the 5 fittest solutions are selected from 5+10 solutions to be the parents in the next generation. These processes will be repeated until a fixed number of generations have been produced or the acceptance criterion is reached.

A. Initialization

Let $\vec{v}$ be the non-negative real-value vector of the hyperparameters that has $2n+2$ dimensions. The vector $\vec{v}$ is represented in the form:

$$\vec{v}=(C,\varepsilon,n,\gamma_0,a_1,\gamma_1,a_2,\gamma_2,\ldots,a_{n-1},\gamma_{n-1}), \qquad (6)$$

where C is the regularization parameter, ε is the deviation of an approximation, n is the number of terms of RBFs, γ_i for $i=0,...,n-1$ are the widths of RBFs, and a_i for $i=1,...,n-1$ are the weights of RBFs.

The (5+10)-ES algorithm starts with the 0^{th} generation (t=0) that selects 5 solutions $\vec{v}_1,...,\vec{v}_5$ and standard deviation $\vec{\sigma}\in R_+^{2n+2}$ using randomization. These 5 initial solutions are evaluated to calculate their fitness. Our goal is to find $\vec{v}$ that optimizes the objective function $f(\vec{v})$.

B. Recombination

The recombination function will create 10 new solutions. We use the global intermediary recombination method for creating these 10 new solutions. Ten pairs of solutions are selected from conventional 5 solutions. The average of each pair of solutions, element by element, is a new solution.

$$\begin{aligned}\vec{v}_1' &= \frac{1}{2}(\vec{v}_1 + \vec{v}_2)\\ \vec{v}_2' &= \frac{1}{2}(\vec{v}_1 + \vec{v}_3)\\ &\vdots\\ \vec{v}_{10}' &= \frac{1}{2}(\vec{v}_4 + \vec{v}_5).\end{aligned} \tag{7}$$

C. Mutation

The $\vec{v}_i'$ for i =1,…,10 are mutated by adding each of them with $(z_1, z_2, \ldots, z_{2n+2})$, and z_i is a random value from a normal distribution with zero mean and σ_i^2 variation.

$$\begin{aligned}mutate(\vec{v}) = \; & (C + z_1, \varepsilon + z_2, n + z_3, \gamma_0 + z_4,\\ & \ldots, a_{n-1} + z_{2n+1}, \gamma_{n-1} + z_{2n+2})\\ & z_i \sim N_i(0, \sigma_i^2).\end{aligned} \tag{8}$$

In each generation, the standard deviation will be adjusted by

$$\begin{aligned}mutate_{\sigma}(\vec{\sigma}) = & (\sigma_1 \cdot e^{z_1}, \sigma_2 \cdot e^{z_2}, \ldots, \sigma_{2n+2} \cdot e^{z_{2n+2}})\\ & z_i \sim N_i(0, \tau^2),\end{aligned} \tag{9}$$

when τ is an arbitrary constant.

D. Evaluation

In general, percentage error is used to measure the efficiency of regression model. However, this function may overfit training data. Sometimes, data contain a lot of noise, and thus if the model fits these noisy data, the learned concept may be wrong. Hence, we would like to validate a set of hyperparameters on many training sets. A good hyperparameters should perform well on these training data. However, as we have only a fixed amount of training data. Therefore, 5-subsets cross-validation is proposed to avoid the overfit problems.

At the beginning, the training data are divided into five subsets, each of which has almost the same number of data. For each generation of ES, the classifiers with the same hyperparameters are trained and validated five times. In the j^{th} iteration (j =1, 2, 3, 4, 5), the classifier is trained on all subsets except for the j^{th} one. Then, the error of prediction is calculated for the j^{th} subset. The average of these five errors is used to be the objective function $f(\vec{v})$. These partitions are displayed in Fig. 5.

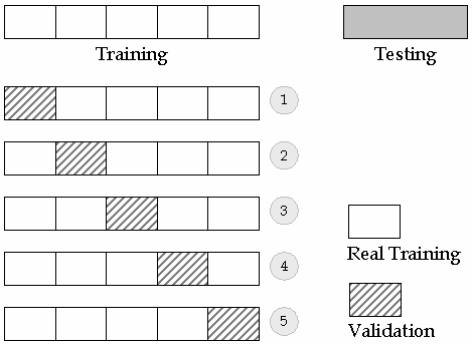

Fig. 5. Partition training data into 5 subsets.

V. Experimental Results

In order to verify the performance of the proposed method, SVRs with the multi-scale RBF kernels are trained and tested on 11 time series datasets from Neural Forecasting Competition [12]. These datasets are monthly time series drawn from homogeneous population of empirical business time series [12]. Since these datasets are used for competition, we can ensure that these data are not so easy. In each dataset, the data of 12 prior periods are used as the training data for predicting the next period.

The evolutionary strategies are used to find the optimal hyperparameters. The value of τ in evaluation process of these experiments is 1.0. The number of RBF terms is a positive integer that is less than or equal to 10. The widths of RBFs (γ_i), the weights of RBFs (a_i), the deviation of an approximation (ε), and the regularization parameter (C) are real numbers between 0.0 and 10.0. These hyperparameters are inspected within 500 generations of ES.

The performance of the proposed method is evaluated by the symmetric mean absolute percentage error (SMAPE) [13], which is defined as

$$\text{SMAPE} = \frac{1}{l}\sum_{i=1}^{l}\frac{|y_i - \hat{y}_i|}{(y_i + \hat{y}_i)/2} \times 100 \tag{10}$$

where y_i for $i = 1,...,l$ are the actually targets of the training data, $\hat{y}_i$ for $i = 1,...,l$ are the forecast values, and l is the number of training data. The experimental results are shown in Table II.

TABLE II
THE SMAPE VALUES ON TRAINING DATA OF EACH DATASET (%)

Datasets	Cumulative Mean	Moving Average N=5	Exponential Smoothing α =0.8	ES-SVR m-RBF
NN3_101	4.9062	3.3791	4.5050	**1.1838**
NN3_102	35.2892	38.3442	20.9627	**5.6823**
NN3_103	96.5719	93.7492	56.6141	**17.9259**
NN3_104	39.0983	40.6747	25.9709	**7.7985**
NN3_105	6.1532	3.3809	2.8427	**1.2307**
NN3_106	8.2873	8.4313	8.8172	**3.0020**
NN3_107	4.4867	3.7889	3.2979	**1.3381**
NN3_108	23.3428	21.7641	24.5493	**7.1894**
NN3_109	19.3230	7.8606	5.4935	**2.2521**
NN3_110	39.2046	42.5192	30.5229	**3.8448**
NN3_111	19.7143	20.4824	19.4045	**7.0860**
Average	26.9434	25.8522	18.4528	**5.3212**

The experimental results show the ability of the proposed method by SMAPE that outperforms the statistical techniques, i.e. the cumulative mean, the moving average, and the exponential smoothing. Moreover, the proposed method that combines two techniques of ES and SVR based on the multi-scale RBF kernel yields the SMAPE values less than SVR with the single RBF kernel in all datasets. The average SMAPE on 11 datasets of the proposed method is the best when compared with the other techniques.

The examples of the approximations are illustrated as graphs in Fig. 6. From these graphs, the proposed method is compared with the cumulative mean and the moving average. The proposed method and the moving average yield the results that are similar to the empirical data. However, with the help of ES, the proposed method can determine the optimal hyperparameters in a more convenient way. For the cumulative mean, the results of the approximation are not good enough; it does not perform well on these data, where there are both seasonal and trend characteristics.

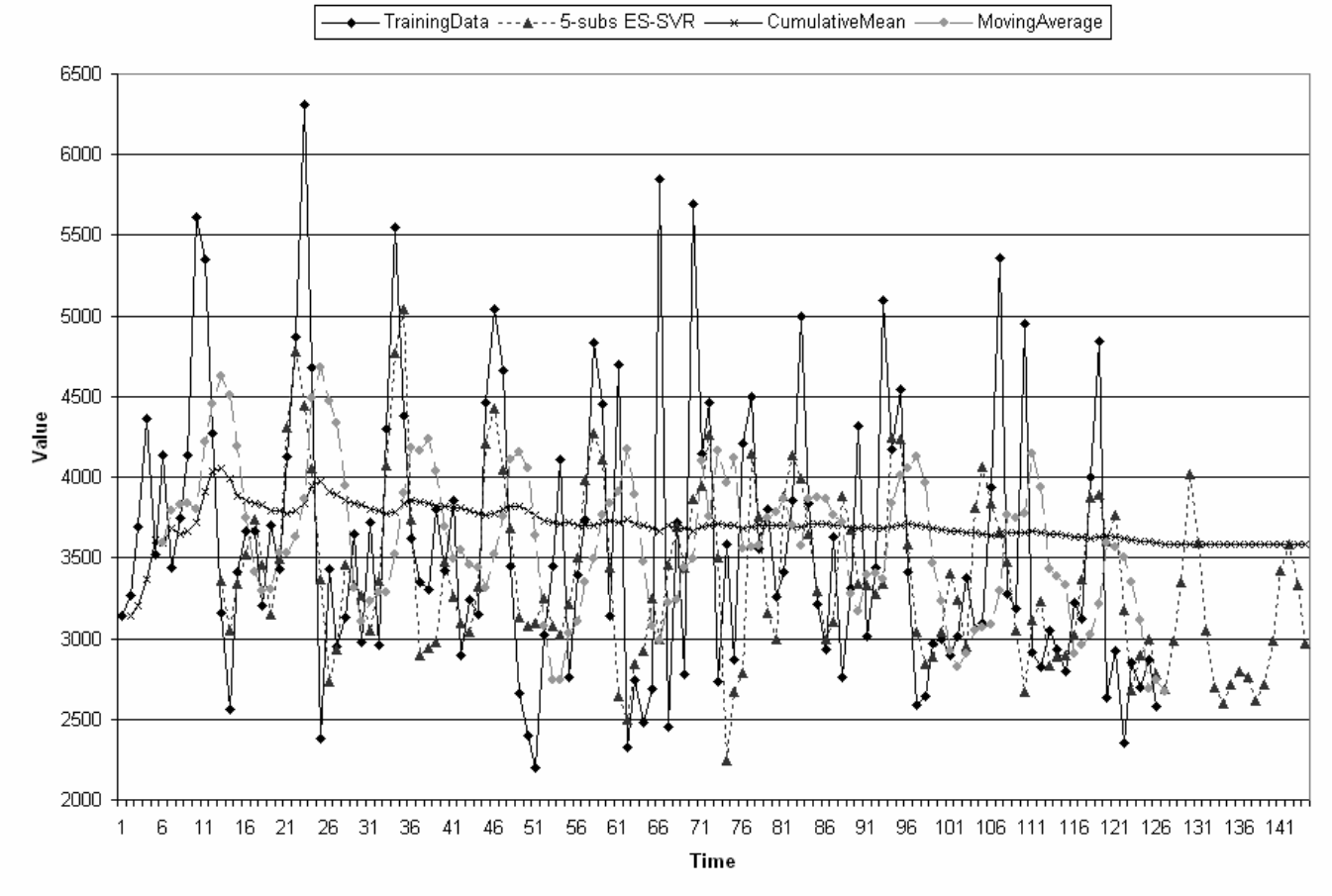

Fig. 6. Graph of the prediction by the proposed method, the cumulative mean, and the moving average on NN3_111 dataset.

VI. Conclusion

The non-negative linear combination of multiple RBF kernels with including weights is proposed for support vector regression. The proposed kernel is proved to be an admissible kernel by Mercer's theorem. Then, the evolutionary strategy is applied to the adjustment of the hyperparameters of SVR. The optimum values of these hyperparameters are searched. Moreover, subsets cross-validation on the error of prediction is considered to be the objective function in evolutionary process to avoid the overfit problems.

The experimental results show the ability of the proposed method through the symmetric mean absolute percentage error (SMAPE). When SVR uses the proposed kernel, it is able to learn from data very well. Furthermore, the evolutionary strategy is effective in optimizing the hyperparameters.

Hence, the proposed method is highly suitable for the complex problems where we have no prior knowledge about their hyperparameters. Besides, this non-negative linear combination can be applied to other Mercer's kernels such as sigmoid, polynomial, or Fourier series kernels, as the general form of linear combination of the Mercer's kernels has been proved to be a Mercer's kernel already.

References

[1] V.N. Vapnik, *Statistical Learning Theory*, John Wiley and Sons, Inc., New York, USA, 1998.

[2] B. Schölkopf, C. Burges, and A.J. Smola, *Advances in Kernel Methods: Support Vector Machines.* MIT Press, Cambridge, MA, 1998.

[3] N.E. Ayat, M. Cheriet, L. Remaki, and C.Y. Suen, "KMOD-A New Support Vector Machine Kernel with Moderate Decreasing for Pattern Recognition," *Proceedings on Document Analysis and Recognition*, pp. 1215-1219, Seattle, USA, 10-13 Sept. 2001.

[4] V. Kecman, *Learning and Soft Computing: Support Vector Machines, Neural Networks, and Fuzzy Logic Models*, MIT Press, London, 2001.

[5] A.J. Smola and B. Schölkopf, "A Tutorial on Support Vector Regression," *NEUROCOLT Technical Report NC-TR-98-030*, Royal Holloway College, London, 1998.

[6] B. Schölkopf, and A.J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, London, 2002.

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

[8] S. Russell and P. Norvig, *Artificial Intelligence: A Modern Approach*, Prentice-Hall, 2003.

[9] H.-G. Beyer and H.-P. Schwefel, "Evolution strategies: A comprehensive introduction," *Natural Computing*, Vol. 1, No. 1, pp. 3-52, 2002.

[10] D.B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence.* IEEE Press, Piscataway, NJ, 1995.

[11] E. deDoncker, A. Gupta, and G. Greenwood, "Adaptive Integration Using Evolutionary Strategies," *Proceedings of 3rd International Conference on High Performance Computing*, pp. 94-99, 19-22 December 1996.

[12] BI^{3}S-Lab, *Artificial Neural Network & Computational Intelligence Forecasting Competition*, Hamburg, Germany, http://www.neural-forecasting-competition.com/ datasets.htm [Accessed: March 2007].

[13] R.J. Hyndman and A.B. Koehler, "*Another Look at Measures of Forecast Accuracy*," International Journal of Forecasting, Vol. 22, Issue 4, Netherlands, 2006.