Evolutionary Strategies for Multi-Scale Radial Basis Function Kernels in Support Vector Machines

Tanasanee Phienthrakul
Department of Computer Engineering
Faculty of Engineering, Chulalongkorn University
Bangkok, Thailand 10330
tanasanee@yahoo.com

Boonserm Kijsirikul
Department of Computer Engineering
Faculty of Engineering, Chulalongkorn University
Bangkok, Thailand 10330
boonserm.k@chula.ac.th

ABSTRACT

In support vector machines (SVM), the kernel functions which compute dot product in feature space significantly affect the performance of classifiers. Each kernel function is suitable for some tasks. A universal kernel is not possible, and the kernel must be chosen for the tasks under consideration by hand. In order to obtain a flexible kernel function, a family of radial basis function (RBF) kernels is proposed. Multi-scale RBF kernels are combined by including weights. Then, the evolutionary strategies are used to adjust these weights and the widths of the RBF kernels. The proposed kernel is proved to be a Mercer's kernel. The experimental results show that the use of multi-scale RBF kernels result in better performance than that of a single Gaussian RBF on benchmarks.

Categories and Subject Descriptors

I.5.2 [**Pattern Recognition**]: Design Methodology – *classifier design and evaluation.*

General Terms

Algorithms, Performance, Design.

Keywords

Evolutionary Strategies, Support Vector Machines, Kernel Function, Radial Basis Function

1. INTRODUCTION

Support vector machines (SVM) are learning algorithms proposed by Vapnik et al. [1, 2], based on the idea of empirical risk minimization principle. It has been widely used in many applications such as pattern recognitions and function approximations. Basically, SVM operates a linear separation in an augmented space by means of some defined kernels satisfying Mercer's condition [2, 3, 4]. These kernels map the input vectors into a very high dimensional space, possibly of infinite dimension, where linear separation is more likely [4]. Then, a linear separating hyperplane is found by maximizing the margin between two classes in this space.

Hence, the complexity of the separating hyperplane depends on the nature and the properties of the used kernel [4]. There are many types of kernel functions such as linear kernel, polynomial kernel, sigmoid kernel, and RBF kernel. 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 RBF kernel by combining several terms of RBF kernels at different scales. These kernels are combined by including weights. These weights and the widths of the RBF kernels are the adjustable parameters in our kernel. In order to obtain good accuracy, a large number of kernel parameters are needed for testing. A question arises: how to search the best values of these parameters? We answer this question by proposing the use of evolutionary strategies for choosing these kernel parameters.

In this paper, we show that the proposed kernels with the help of ES provide better performance than the traditional RBF, and ES effectively searches good parameters for our kernel. In Section 2, we briefly review the support vector machines and the evolutionary strategies. In Section 3, an adaptive multi-scale RBF is proposed and proved to be the Mercer's kernel. Then, the evolutionary strategies are applied to adjust the weights and the widths of RBF kernels. After that, Section 4 illustrates the performances of the proposed kernel on benchmark datasets and gives a discussion. Finally, Section 5 draws a general conclusion.

2. NOTATION AND BACKGROUND

2.1 Support Vector Machines

A support vector machine is a classifier which finds an optimal separating hyperplane. In the simple pattern recognitions, SVM uses a linear hyperplane to create a classifier with a maximum margin [5]. Consider the problem of binary classification. The training dataset are given as

$$(x_1, y_1), (x_2, y_2), \ldots, (x_l, y_l),$$

GECCO'05, June 25-29, 2005, Washington, DC, USA.

where $x_i \in R^N$ and $y_i \in \{-1,1\}$ for $i = 1,\ldots,l$ when x_i is a sample data and y_i is its label [6]. A linear decision surface is defined by the equation:

$$w \cdot x + b = 0. \quad (1)$$

The goal of learning is to find $w \in R^N$ and the scalar b such that the margin between positive and negative examples is maximized. An example of the decision surface and the margin are shown in Figure 1.

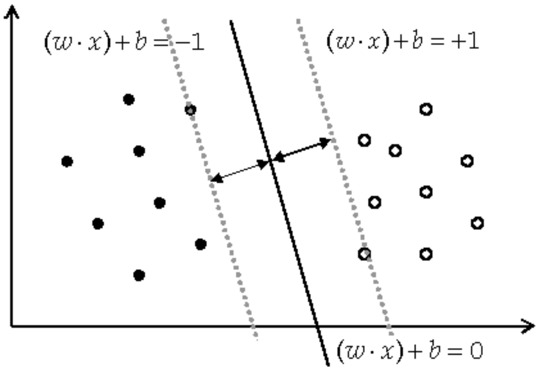

Figure 1. An Example of Decision Surface and Margin

In most cases, seeking a suitable linearly hyperplane in an input space has the restrictions. There is an important technique that enables these machines to produce complex nonlinear boundaries inside the original space. This performs by mapping the input space into a higher dimensional feature space through a mapping function Φ and separating there [7].

A good property of SVM is that it is not necessary to know the explicit form of Φ. Only the inner product in feature space, called kernel function $K(x,y) = \Phi(x) \cdot \Phi(y)$, must be defined. The decision function is the following equation:

$$f(x) = sign\left(\sum_{i=1}^{l} \alpha_i y_i K(x_i, x) + b\right) \quad (2)$$

where $\alpha_i \geq 0$ is the coefficient associated with a support vector x_i and b is an offset.

2.2 Evolutionary Strategies

Evolutionary strategies (ES, [8, 9]) are based on the principles of adaptive selection found in the natural world. They have been successfully applied to obtain good solutions in optimization problems. Each generation (iteration) of the ES algorithm takes a population of individuals (potential solutions) and modifies the problem parameters to produce offspring (new solutions). Both the parents and the offspring are evaluated but only the highest fit individuals (better solutions) survive to produce new generations. The ES has been successfully used to solve various types of optimization problems [10]. The basic ES algorithm is shown below.

1. Randomly generate a parent population of μ solutions.
2. Evaluate all parents to determine their fitness.
3. Apply reproduction operators to create λ offspring.
4. Evaluate and keep the μ fittest individuals.
5. Go to step 3 unless an acceptable solution has been found or a fixed number of generations has been produced and evaluated.

Every point in the search space is an individual. The ES uses a population of μ individuals to conduct the search for possibly better solutions [11]. During each generation, each individual is mutated to produce offspring. This means the ES is simultaneously investigating several regions of the search space, which greatly decreases the amount of time required to locate good solutions.

The initial population of individuals is randomly generated but, ideally, should be uniformly distributed throughout the search space so that all regions may be explored. Each individual in each generation is evaluated to determine its fitness. Individuals with high fitness represent approximations which produce low error estimates. The ES terminates after a fixed number of generations have been produced and evaluated or earlier if the acceptance criterion is reached [11].

There are several different versions of the ES. The ($\mu + \lambda$)-ES and (μ, λ)-ES are two of the more common versions. In the former, μ parents produce λ offspring. The parents and the offspring compete equally for survival. In the latter, μ parents produce $\lambda > \mu$ offspring, but only the μ best offspring survive. Thus the lifespan of any solution is only a single generation [11]. In this work, we use the ($\mu + \lambda$)-ES. Our results indicate that this method finds the parameters of SVM kernels which yield high accuracy. In the next section we discuss how the ($\mu + \lambda$)-ES are applied in our proposed kernel.

3. ADAPTIVE MULTI-SCALE RBF

All kernel functions in the literature are either dot product functions $K(x,y) = K(x \cdot y)$ or distance functions $K(x,y) = K(\|x - y\|)$ [4]. The examples of dot product kernels are linear, polynomial, and sigmoid kernels, while the examples of distance kernels are exponential RBF, Gaussian RBF, and multi-quadratic kernels. Each kernel is suitable for some datasets. One of the most widely used kernels is the Gaussian RBF kernel. In the several classification tasks, the Gaussian RBF kernel provided the better results among the other kernels. However, it has still restrictions in some complex problems. In this section, a family of RBF kernels is proposed. Multiple RBF kernels at different scales are combined and proved to be the Mercer's kernel. Moreover, the evolutionary strategies are applied to adjust the parameters of this kernel.

3.1 Multi-Scale RBF Kernels

The Gaussian RBF kernel uses the Euclidean distance between two points in original space to find the correlation in the feature space. The points very close to each other are strongly correlated whereas points far apart have uncorrelated image in the augmented space [4]. This correlation is rather smooth. 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, one possible way is to adjust the velocity of decrement in each range of distance between two points. Moreover, the obtained kernel should maintain the good characteristic of the RBF kernel. To implement this capability, the combination of RBF kernels at difference scale is proposed. The analytic expression of this kernel is following:

$$K(x,y) = \sum_{i=1}^{n} a_i K(x,y,\gamma_i) \tag{3}$$

where n is a positive integer, a_i for $i = 1,\ldots,n$ are the arbitrary nonnegative weighting constants, and

$$K(x,y,\gamma_i) = \exp(-\gamma_i \|x-y\|^2) \tag{4}$$

is the RBF kernel at the width γ_i for $i = 1,\ldots,n$.

The correlations in feature space (relations between the kernel functions and the distance between two points in the original space) of the multi-scale RBF kernels for n=1, 2, and 3 are displayed in Figure 2. The figure shows that the correlation of the RBF kernel is rather smooth, while 2-RBF and 3-RBF are more flexible. This can be interpreted that the increase of adjustable parameters provides a more adaptive kernel.

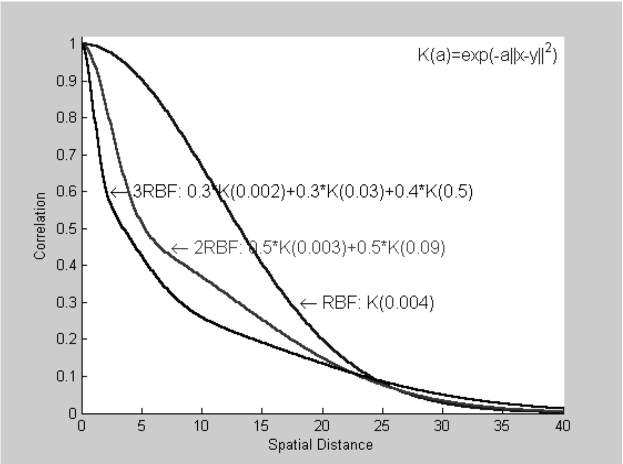

Figure 2. Graph of RBF, 2-RBF, and 3-RBF Kernels

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 [5]. The Mercer's theorem tells that any symmetric function $K(x,y)$ in the input space can represent an inner product in feature space if

$$\iint K(x,y)\, g(x)\, g(y)\, dx\, dy \quad \geq \quad 0 \tag{5}$$

be 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

$$K(x,y) = \sum_{i=1}^{\infty} \lambda_i\, \Phi_i(x)\, \Phi_i(y) \tag{6}$$

with $\lambda_i \geq 0$ [3, 5]. In this case, the mapping from input space to 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

$$\Phi(x) \cdot \Phi(y) = \sum_{i=1}^{\infty} \lambda_i\, \Phi_i(x)\, \Phi_i(y) = K(x,y). \tag{7}$$

In the next corollary, the proposed kernel functions will be proved to be an admissible kernel by the Mercer's theorem.

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

Proof. Let $K_i(x,y)$ be Mercer's kernel, for $i = 1,\ldots,n$, and let

$$K(x,y) = \sum_{i=1}^{n} a_i K_i(x,y) \tag{8}$$

According to the Mercer's theorem, we know that

$$\iint K_i(x,y)\, g(x)\, g(y)\, dx\, dy \quad \geq \quad 0,\ \forall g \tag{9}$$

for $i = 1,\ldots,n$.

By taking the linear combination with nonnegative coefficients a_i, we will get

$$\sum_{i=1}^{n} a_i \iint K_i(x,y)\, g(x)\, g(y)\, dx\, dy \quad \geq \quad 0. \tag{10}$$

And then

$$\iint \sum_{i=1}^{n} a_i K_i(x,y)\, g(x)\, g(y)\, dx\, dy \quad \geq \quad 0. \tag{11}$$

Therefore,

$$\iint K(x,y)\, g(x)\, g(y)\, dx\, dy \quad \geq \quad 0,\ \forall g. \tag{12}$$

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

□

The RBF is a well-known Mercer's kernel. Therefore, the linear combination of RBFs in equation 3 can be proved to be the Mercer's kernel. When the various RBF functions are combined, the results of classification are more flexible than using a single RBF function. Users can choose some of suitable RBF kernels for their problems. The examples of classification with a simple RBF kernel and a combination of two RBF kernels are shown in Figure 3 and 4, respectively.

In these examples, the training data are non-linearly separable. The SVM with a single RBF and 2-RBF (the proposed kernel with n =2) kernels can correctly classify the data. However, the 2-RBF kernel yields the result that is more flexible and easier to comprehend. Moreover, the margin of the 2-RBF kernel in this example is larger than the single RBF kernel. This means that the classification results of the 2-RBF kernel on unseen data are more plausible than those of the single RBF kernel.

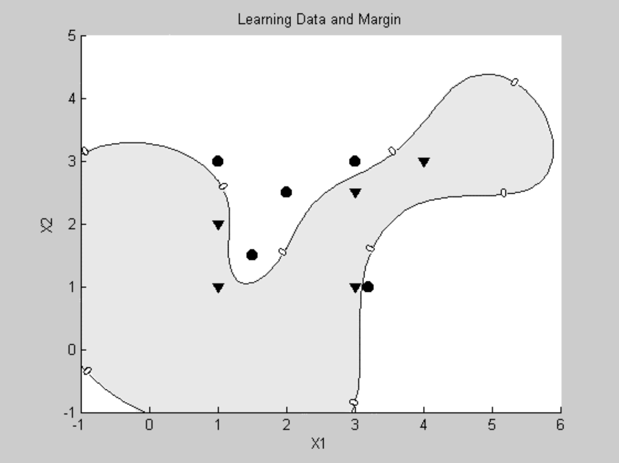

Figure 3. An Example of Classification with an RBF Kernel

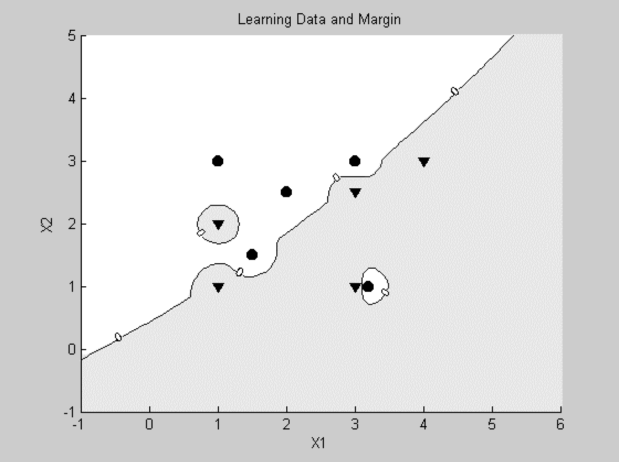

Figure 4. An Example of Classification with a 2-RBF Kernel

3.2 Evolving Multi-Scale RBF Kernels

As shown in equation 3, the proposed kernel has $2n$ parameters for adjusting correlation in the augmented space, when n terms of RBF kernels are used. There are n parameters for adjusting weights and n values of the widths of RBF. Though one may reduce the number of parameters to $2n-1$ (for example, by fixing a_1 to 1), we decide to use $2n$ parameters for ease of understanding. These values have influence to the efficiency of the proposed kernel. In order to obtain the optimal values, the evolutionary strategies (ES) are considered.

At the beginning, the training data are divided into five subsets, each of which has the same number of data. For each generation of ES, the classifier is trained and validated five times. In the i th iteration (i = 1, 2, 3, 4, 5), the classifier is trained on all subsets except the i th one. Then, the accuracy of classification is evaluated for the i th subset. These partitions are displayed in Figure 5. Only real training data sets are used to produce the classifiers by a set of parameters. Then, the validation set are used for calculating the accuracies of the classifiers. The average of these five accuracies is used to be the objective function $f(\vec{v})$ in this work. It is a rather good estimate of the generalization accuracy for adjusting the parameters. The testing data set is reserved for testing the final classifier with the best parameters found by the evolutionary strategy.

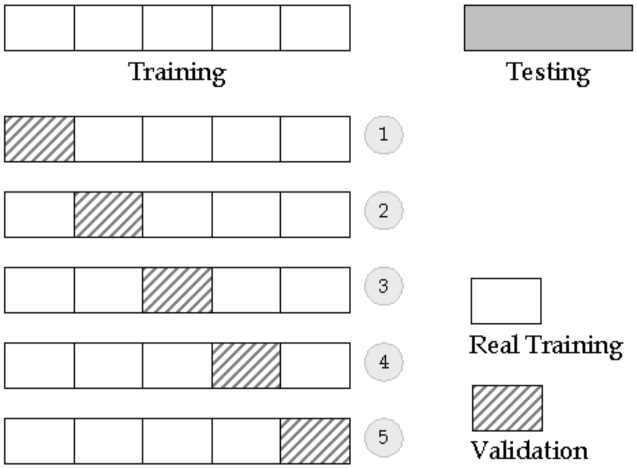

Figure 5. Partition Training Data into 5 Subsets

Let $\vec{v}$ be the vector of real number that has $2n$ dimensions. Our goal is to find $\vec{v}$ that makes the maximum $f(\vec{v})$. Each attribute of vector $\vec{v}$ is a parameter of the proposed kernel. The vector $\vec{v}$ is represented in form (a_1, γ_1, a_2, γ_2, … , a_n, γ_n), where a_i and γ_i for i =1,2,…, n are the parameters of the kernel in equation 3.

The (5+10)-ES is applied to adjust the parameters of our kernel. At the first generation, 5 solutions are selected randomly to be parents. All parents are evaluated to calculate their fitness. Then, these 5 solutions are used to create new 10 solutions, and all 5+10 solutions are evaluated. 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 good solutions are found. The algorithm of (5+10)-ES is shown in Figure 6.

This algorithm starts with 0th generation (t=0) which selects 5 solutions $\vec{v}_1, ..., \vec{v}_5$ and standard deviation $\vec{\sigma} \in R_{+}^{2n}$ using randomization or assigning initial values. Then, the recombination function will create a new solution. We use the global intermediary recombination method for creating 10 new solutions. Ten pairs of solutions are selected from conventional 5 solutions. The average of each pair of solutions is a new solution.

$$\vec{v}_1' = \frac{1}{2}(\vec{v}_1 + \vec{v}_2) \tag{13}$$

$$\vec{v}_2' = \frac{1}{2}(\vec{v}_1 + \vec{v}_3) \tag{14}$$

$$\vdots$$

$$\vec{v}_{10}' = \frac{1}{2}(\vec{v}_4 + \vec{v}_5) \tag{15}$$

```
t = 0;
initialization($\vec{v}_1,\ldots,\vec{v}_5, \vec{\sigma}$);
evaluation $f(\vec{v}_1),\ldots,f(\vec{v}_5)$;
while (t < 1000) do
    for $i$ =1 to 10 do
        $\vec{v}_i'$ = recombination($\vec{v}_1,\ldots,\vec{v}_5$);
        $\vec{v}_i'$ = mutate($\vec{v}_i'$);
        evaluate $f(\vec{v}_i')$;
    end
    ($\vec{v}_1,\ldots,\vec{v}_5$) = select($\vec{v}_1,\ldots,\vec{v}_5, \vec{v}_1',\ldots,\vec{v}_{10}'$)
    $\vec{\sigma}$ = mutate$_\sigma$($\vec{\sigma}$);
    t = t+1;
end
```

Figure 6. (5+10)-ES Algorithm

After that, these solutions are mutated by the following function:

$$mutate(\vec{v}) = (a_1 + z_1, \gamma_1 + z_2, \ldots, a_n + z_{2n-1}, \gamma_n + z_{2n}) \tag{16}$$

$$z_i \sim N_i(0, \sigma_i^2). \tag{17}$$

The $\vec{v}_i'$ for i =1,..,10 are mutated by adding $\vec{v}'$ with ($z_1, z_2, \ldots, z_{2n}$), and z_i is a random value from normal distribution with zero mean and σ_i^2 variation. In each generation, the standard deviation will be adjusted by equation 18.

$$mutate_\sigma(\vec{\sigma}) = (\sigma_1 \cdot e^{z_1}, \sigma_2 \cdot e^{z_2}, \ldots, \sigma_{2n} \cdot e^{z_{2n}}) \tag{18}$$

$$z_i \sim N_i(0, \tau^2), \tag{19}$$

when τ is an arbitrary constant. Then, this algorithm is repeated until *t* reaches a predefined value.

4. RESULTS AND DISCUSSION

In order to verify the performance, SVMs with the proposed kernel are tested on 10 datasets from UCI repository [12]. Each of datasets contains two classes. The number of attributes and the sample size of each dataset are shown in Table 1.

Table 1. Datasets from UCI Repository

Datasets	# Attributes	# Training Examples	# Test Examples
Checkers	2	128	64
Liver Disorder	6	230	115
Pima Diabetes	8	512	256
Glass	9	108	55
Parity of Bits	10	100	1024
Cleveland Heart	13	180	90
Australian	14	460	230
Random	20	100	3000
German-org	24	666	334
Ionosphere	34	234	117

In the experiment, the evolutionary strategies are used to find the optimal parameters of kernels in both the conventional RBF and the proposed kernels. Training examples (not including test data) are divided into five subsets with the same number of examples. For each generation, classifier with same parameters is trained and validated five times. For i =1,…,n, the widths of RBFs (γ_i) are between 0.0 and 10.0, and the weights of RBFs (a_i) are between 0.0 and 0.1, when n is the number of RBFs in equation 3. These parameters are inspected within 1000 generations of ES. Then, the best parameters will be used to test on unseen data (test data). The value of τ in these experiments is 1.0. The accuracies of the proposed kernel for n = 2, 3, 4, and 5 are compared with the single RBF in Table 2.

These results show the ability of the proposed kernel. All datasets, multi-scale RBF kernels yield the better accuracies than the single RBF on test data. 5-RBF provides the best accuracies in all datasets. There is a trend that the accuracy increases with the increase of the number of terms of RBF kernels. Moreover, the accuracies of multi-scale RBF kernels are significantly higher than those of the single RBF on some datasets. The experimental results also show the evolutionary strategy is effective in optimizing the kernel parameters, especially when the ranges of the parameters are large. Other methods for optimizing the parameters can also be used, such as gradient based methods. We decided to use (5+10)-ES because the ability to escape from local minima and the population size is not large so that it fast converges to an optimal solution. To avoid the over-fitting of the kernel parameters, the partition of training data into five subsets is employed so that the parameters which work well with all five subsets will have less chance to over-fit the data.

Table 2. Experimental Results

Datasets	Training Accuracies					Test Accuracies				
	RBF	**2-RBF**	**3-RBF**	**4-RBF**	**5-RBF**	**RBF**	**2-RBF**	**3-RBF**	**4-RBF**	**5-RBF**
Checkers	81.77	92.19	92.97	*94.53*	*94.53*	71.87	81.25	81.25	**82.81***	**82.81***
Liver Disorder	72.61	74.35	*78.26*	*78.26*	*78.26*	69.57	73.04	75.36	**78.26***	**78.26***
Pima Diabetes	*83.98*	83.66	83.01	83.79	81.25	72.66	75.13	75.39	75.39	**75.78**
Glass	78.70	88.89	*92.59*	*92.59*	88.98	81.82	85.45	**87.27**	**87.27**	**87.27**
Parity of Bits	67.67	89.67	*100.00*	*100.00*	*100.00*	50.39	52.96	**54.39****	**54.39****	**54.39****
Cleveland Heart	*82.78*	77.78	77.78	76.11	76.11	61.11	63.33	64.44	**65.56**	**65.56**
Australian	*93.48*	78.91	79.13	79.35	82.83	61.30	62.61	63.33	63.91	**65.22**
Random	*64.00*	*64.00*	*64.00*	*64.00*	*64.00*	50.33	**50.37**	**50.37**	**50.37**	**50.37**
German-org	68.62	*73.57*	72.37	71.02	71.02	72.75	72.85	72.85	**73.05**	**73.05**
Ionosphere	96.15	96.15	*97.01*	*97.01*	96.15	85.47	92.31*	95.73***	95.73***	**96.58*****

Statistical significant at level for the difference in test accuracies between the corresponding kernel and 'RBF':
* is 0.10, ** is 0.05, *** is 0.01

5. CONCLUSION

The linear combination of multiple RBF kernels with including weights is proposed for support vector classification. The RBF kernel is the most popular distance based kernel that is applied to various applications and yields good results. Here we show that the performance of the RBF kernel can be further enhanced by the combination of several RBF kernels.

The evolutionary strategy is applied to adjust weights and widths of RBFs in the proposed kernel. The proposed kernel is proved to be the admissible kernels by Mercer's condition. Moreover, the proposed kernel has more flexibility in complex problems.

The experiments were performed on 10 benchmarks. The results show the abilities of the proposed kernels through their accuracies on test examples. The combination of RBF kernels yields the better results. When the SVM uses this kernel, it is able to learn from data very well. Therefore, this method is very suitable for the problems where we have no prior knowledge about kernels.

Moreover, this combination can be applied to other Mercer's kernels such as sigmoid or polynomial kernels, as the general form of linear combination of the Mercer's kernels has been proved to be a Mercer's kernel already. Furthermore, there are the other combination techniques that can be used to improve the efficiency of SVM kernels, which will be investigated in the near future.

6. ACKNOWLEDGMENTS

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

7. REFERENCES

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

[2] Vapnik, V.N. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, USA, 1995.

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

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

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

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

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

[8] Beyer, H.G. and Schwefel, H.P. Evolution Strategies: A Comprehensive Introduction. *Natural Computing.* 1 (1), 2002, 3-52.

[9] Rechenberg, I. *Evolutionsstrateie*. Frommann-Holzboog Verlag, Stuttgart, Germany, 1994.

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

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

[12] Blake, C.L. and Merz, C.J. *UCI Repository of machine learning databases* [http://www.ics.uci.edu/~mlearn/ MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science, 1998.

[13] Boser, B., Guyon, I., and Vapnik, V. A Training Algorithm for optimal margin classifiers. In *Proceedings of 5th Annual ACM Workshop on COLT.* ACM Press, 1998, 144-152.

[14] Bottou, L., Cortes, C., Denker, J.S., Drucker, H., Guyon, I., Kackel, L.D., LeCun, Y., Sackinger, E., Simard, P., Vapnik, V., and Miller, U.A. Comparison of Classifier Methods: A case study in handwritten digit recognition. In *Proceedings of 12th International Conference on Pattern Recognition and Neural Network. 1994.*

[15] Burges, C. *A Tutorial on Support Vector Machines for Pattern Recognition.* Kluwer Academic Publishers, 1998.

[16] Cristianini, N. and Taylor, J.S. *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods.* Cambridge University Press, UK, 2000.

[17] Cortes, C. and Vapnik, V.N. Support Vector Networks. In *Machine Learning*. 20:273, AT&T Labs-Research, USA, 1995.

[18] Eads, D.R., Hill, D., David, S., Perkins, S.J., Ma, J., Porter, R.B., and Theiler, J.P. Genetic Algorithms and Support Vector Machines for Time Series Classification. In *Proceedings of SPIF*. No.4787, 2002, 74-85.

[19] Friedrichs, F. and Igel, C. Evolutionary Tuning of Multiple SVM Parameters. In *12th European Symposium on Artificial Neural Networks (ESANN 2004)*. 2004, 519-524.

[20] Fröhlich, H., Chapelle, O., and Schölkopf, B. Feature Selection for Support Vector Machines by Means of Genetic Algorithms. In *15th IEEE International Conference on Tools with AI (ICTAI 2003)*. 2003, 142-148.

[21] Goldberg, D.E. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, US, 1989.

[22] Gunn, R. *Support Vector Machines for Classification and Regression.* University of Southampton, 10 May 1998, http://www.isis.ecs.soton.ac.uk/isystems/kernel [Accessed: May 2004].

[23] Igel, C. Multi-objective Model Selection for Support Vector Machines. In *Proceedings of the Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005).* No.3410, 2005, 534-546.

[24] Jong, K., Marchiori, E., and Vaart, A. Analysis of Proteomic Pattern Data for Cancer Detection. In *Applications of Evolutionary Computing. EvoBIO: Evolutionary Computation and Bioinformatics.* Springer, 2004.

[25] Jong, K., Marchiori, E., Sebag, M., and Vaart, A. Feature Selection in Proteomic Pattern Data with Support Vector Machines. In *Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB),* IEEE, 2004.

[26] Miller, M.T., Jerebko, A.K., Malley, J.D., and Summers, R.M. Feature Selection for Computer-Aided Polyp Detection using Genetic Algorithms. In *Proceedings of SPIF*. No.5031, 2003, 102-110.

[27] Müller, K., Mika, S., Rätsch, G., Tsuda K., and Schölkopf, B. An Introduction to Kernel-Based Learning Algorithm. *IEEE Transactions on Neural Networks.* 12, 2 (March. 2001), 181-201.

[28] Osuna, E., Freund, R., and Girosi, F. *Support vector machines: Training and applications.* Technical Report, MIT, AI Memo No.1602, 1997.

[29] Runarsson, T.P. and Sigurdsson, S. Asynchronous Parallel Evolutionary Model Selection for Support Vector Machines. In *Neural Information Processing.* 3(3), 2004, 59-68.

[30] Smits, G.F. and Jordaan, E.M. Improved SVM Regression using Mixtures of Kernels. In *Proceedings of the 2002 International Joint Conference on Neural Networks.* 3, May 12-17, 2002, 2785–2790.

[31] Schwefel, H.P. *Evolution and Optimum Seeking.* John Wiley and Sons, New York, USA, 1995.