EVOLVING HYPERPARAMETERS OF SUPPORT VECTOR MACHINES BASED ON MULTI-SCALE RBF KERNELS

Tanasanee Phienthrakul and Boonserm Kijsirikul
Department of Computer Engineering, Chulalongkorn University, Thailand

Abstract: Kernel functions are used in support vector machines (SVMs) to compute dot product in a higher dimensional space. The performance of classification depends on the chosen kernel. Each kernel function is suitable for some tasks. In order to obtain a more flexible kernel function, a family of RBF kernels is proposed. Multi-scale RBF kernels are combined by including weights. These kernels allow better discrimination in the feature space, and are proved to be the Mercer's kernels. Then, the evolutionary strategies are applied for adjusting the hyperparameters of SVM. Subsets cross validation is used to be the objective function in evolutionary process. The experimental results show that the accuracy of the proposed method is better than the ordinary approach.

Key words: Support Vector Machines, Evolutionary Strategies, Kernel Methods, Radial Basis Function

1. INTRODUCTION

Support Vector Machines (SVMs) are learning algorithms that have been widely used in many applications such as pattern recognitions and function approximations [1]. Basically, SVM operates a linear separation 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 linear separation is more likely [3]. 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 [3]. 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 it still has the restrictions in some complex problems.

Therefore, we propose to improve the efficiency of classification by using the combination of RBF kernels at different scales. These kernels are combined by including weights. These weights, the widths of the RBF kernels, and regularization parameter of SVM are called *hyperparameters*. In general, the hyperparameters are usually determined by grid search. These hyperparameters are varied with a fixed step-size in a range of values, which consume a lot of time. Hence, we propose to use the evolutionary strategies (ESs) for choosing these hyperparameters. Moreover, we propose to use subset cross validation for evaluating our kernel in evolutionary process.

A short description of support vector machines is presented in Section 2. In Section 3, we propose the multi-scale RBF kernel and apply evolutionary strategies to determine the hyperparameters of the kernel. The proposed kernels with the help of ES are tested in Section 4. Finally, the conclusions are described in Section 5.

2. SUPPORT VECTOR MACHINES

Support vector machine is a classifier which finds an optimal separating hyperplane. In the simple pattern recognitions, SVM uses a linear separating hyperplane to create a classifier with a maximum margin [4]. Consider the problem of binary classification. The training dataset are given as $(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)$, where $x_i \in R^N$ and $y_i \in \{-1, 1\}$ for $i = 1, \dots, l$ when x_i is a sample data and y_i is its label [5]. A linear decision surface is defined by the equation:

$$w \cdot x + b = 0 . \tag{1}$$

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 is shown in Figure 1.

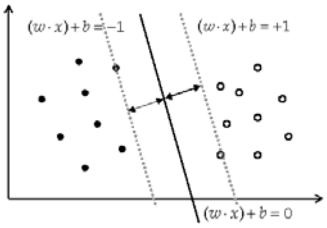


Figure 1. An example of decision surface and margin

This surface can be achieved by minimizing $\| w \|^2$, and the conditions for classification without training error are $y_i((w \cdot x_i) + b) \geq 1$ for $i = 1, \ldots, l$, that are a quadratic optimization problem [4]. This yields the decision function as

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

The data examples x_i which correspond to non-zero α_i values are called *support vectors*.

However, the quadratic programming solutions cannot be used in the case of overlapping because the constraints cannot be satisfied [4]. In such a situation, this algorithm must allow some data to be unclassified, or on the wrong side of a decision surface [4]. In practice, we allow a soft margin, and all data inside this margin are neglected. The width of soft margin can be controlled by a corresponding regularization parameter *C* that determines the trade-off the training error and the VC dimension of the model [4].

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 [6]. This can be achieved by substitution $\Phi(x_i)$ for 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 feature space, called kernel function $K(x, y) = \Phi(x) \cdot \Phi(y)$, must be defined. The decision function becomes the following equation:

$$f(x) = sign\left(\sum_{i=1}^{l} \alpha_i \, y_i \, K(x_i, x) + b \right). \tag{3}$$

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

3. EVOLVING MULTI-SCALE RBF KERNEL

The evolutionary strategies (ES) are the algorithms that imitate the natural processes (natural selection and survival of the fittest principle), which were developed by Rechenberg and Schwefel [7, 8, 9]. ES was developed for numerical optimization problems, and they are significantly faster than traditional genetic algorithms [10]. In this section, the multi-scale RBF kernel is proposed for SVM on classification problems. Then, the evolutionary strategies are applied to evolve hyperparameters of SVM.

3.1 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]. Although, the RBF kernel yields good results on various applications, it has 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 difference scale is proposed. The analytic expression of this kernel is following:

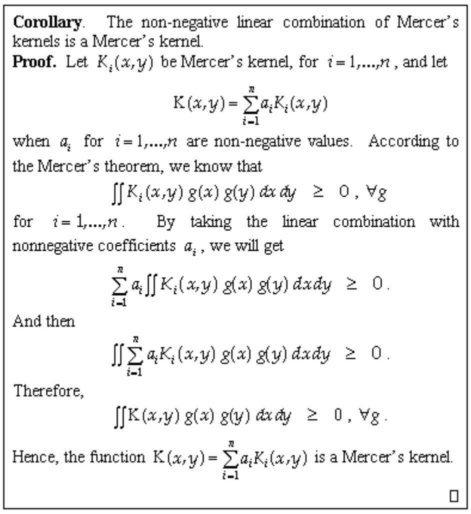
$$K(x,y) = \sum_{i=1}^{n} a_i K(x,y,\gamma_i)\ . \tag{4}$$

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

$$K(x,y,\gamma_i)\ =\ \exp(-\gamma_i \left\|x-y\right\|^2)\ . \tag{5}$$

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

The RBF is a well-known Mercer's kernel. Therefore, the non-negative linear combination of RBFs in equation 5 can be proved to be an admissible kernel by the Mercer's theorem [5] that is showed in Figure 2.



Corollary. The non-negative 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

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

when a_i for $i = 1,\ldots,n$ are non-negative values. According to the Mercer's theorem, we know that

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

for $i = 1,\ldots,n$. By taking the linear combination with nonnegative coefficients a_i , we will get

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

And then

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

Therefore,

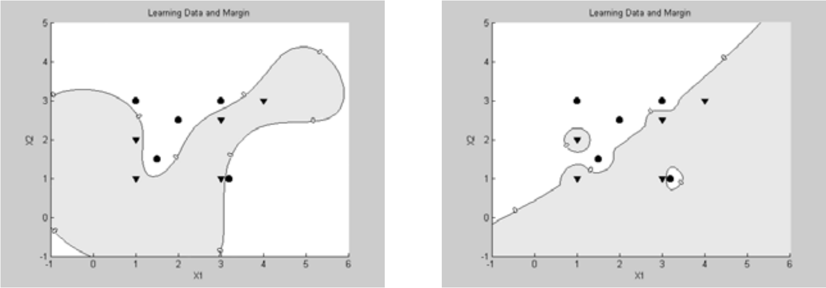
$$\iint \mathrm{K}(x,y)\, g(x)\, g(y)\, dx\, dy \ \ \geq\ \ 0\ ,\ \forall g\ .$$

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

□

Figure 2. Proving of the proposed kernel

When the various RBF functions are combined, the results of classification are more flexible than using a single RBF function. The examples of classification with a simple RBF kernel and a combination of two RBF kernels are showed in Figure 3.



(a) RBF kernel

(b) 2-RBF kernel

Figure 3. The examples of classification

In these examples, the training data are non-linearly separable. The SVM with a single RBF and 2-RBF (the multi-scale RBF 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.

3.2 Evolving hyperparameters of SVM

In this sub-section, the ES is applied to evolve the optimal hyperparameters of SVM. There are several different versions of the 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].

Form equation 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 RBF). 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 that will be used in the rest of this paper is in the form:

$$K(x,y) = K(x,y,\gamma_0) + \sum_{i=1}^{n-1} a_i K(x,y,\gamma_i) \ . \tag{6}$$

Let $\vec{v}$ be the non-negative real value of the hyperparameter vector that has $2n+1$ dimensions. The vector $\vec{v}$ is represented in the form:

$$\vec{v} = (C, n, \gamma_0, a_1, \gamma_1, a_2, \gamma_2, \dots, a_{n-1}, \gamma_{n-1}). \qquad (7)$$

where C is the regularization parameter, n is the number of RBFs, γ_i are the widths of RBFs, and a_i are the weights of RBFs. Our goal is to find $\vec{v}$ that maximizes the objective function $f(\vec{v})$. The (5+10)-ES is applied to adjust these hyperparameters. The algorithm of (5+10)-ES is showed in Figure 4.

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

Figure 4. (5+10)-ES algorithm

This algorithm starts with the 0th generation (t=0) in which 5 solutions $\vec{v}_1,\dots,\vec{v}_5$ and standard deviation $\vec{\sigma} \in R_+^{2n+1}$ are selected randomly. These initial solutions are evaluated. Then, the solutions are used to create 10 new solutions by the global intermediary recombination method. Ten pairs of solutions are selected from conventional 5 solutions. The average of each pair of vector solutions, element by element, is a new solution.

$$\vec{v}'_1 = \frac{1}{2}\left(\vec{v}_1 + \vec{v}_2\right) \qquad (8)$$

$$\vec{v}'_2 = \frac{1}{2}\left(\vec{v}_1 + \vec{v}_3\right) \qquad (9)$$

$$\vdots$$

$$\vec{v}'_{10} = \frac{1}{2}\left(\vec{v}_4 + \vec{v}_5\right) \qquad (10)$$

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

$$mutate(\bar{v}\,) = (C+z_1\,, n+z_1\,, \gamma_0+z_2 \;\; a_1+z_3\,, \gamma_1+z_4\,,\;\dots\;, a_{n-1}+z_{2n}\,, \gamma_{n-1}+z_{2n+1}\,) \quad (11)$$

$$z_i \sim N_i(0\,, {\sigma_i}^2)\,. \quad (12)$$

The $\bar{v}_i'$ for i =1,..,10 are mutated by adding $\bar{v}'$ with ($z_1\,, z_2\,,\dots, z_{2n+1}$), and z_i is a random value from normal distribution with zero mean and ${\sigma_i}^2$ variation. In each generation, the standard deviation will be adjusted by the equation 13.

$$mutate_{\,\sigma}\,(\,\bar{\sigma}\,) = (\,\sigma_1\cdot e^{z_1}\,,\; \sigma_2\cdot e^{z_2}\,,\;\dots\;,\; \sigma_{2n+1}\cdot e^{z_{2n+1}}\,) \quad (13)$$

$$z_i \sim N_i(0\,, \tau^2)\,, \quad (14)$$

when τ is an arbitrary constant.

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.

For evaluating the hyperparameters of SVM, there are many ways to define an objective function. Although, training rate will be the easiest objective function, it maybe over-fit with training data. In many time, our data has a lot of noise. If the decision functions over-fit to these noisy data, the target concept may be wrong. Therefore, we propose to train the decision function with subsets cross validation; a good set of parameters should perform well on all these subsets.

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.

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 $f(\bar{v})$. 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.

4. EXPERIMENTAL RESULTS

In order to verify the performance of the proposed method, SVMs with the multi-scale RBF kernel are trained and tested on datasets from the UCI repository [12]. The evolutionary strategies are used to find the optimal hyperparameters of

SVM. The proposed method is evaluated by 5 folds cross-validation. The regularization parameter, the widths of RBFs (γ_i), and the weights of RBFs (a_i) are real numbers between 0.0 and 10.0. The number of RBF terms is a positive integer that is less than or equal to 10. These hyperparameters are inspected within 1000 generations of ES. Then, the best hyperparameters will be used to test on validation data. The value of τ in evaluation process of these experiments is 1.0. The experiments are divided into 2 parts as two-class problems and multi-class problems.

4.1 Two-class problems

Fifteen datasets from UCI are used for testing. Each of datasets contains two classes. The proposed method is compared with GridSearch and the ES that uses training rate as the objective function. GridSearch is applied on single RBF kernel, while ES with training rates is applied on multi-scale RBF kernel. The number of attributes, the sample size, and the average accuracies on 5 folds of each dataset are shown in Table 1.

Table 1. Results of two-class problems

Datasets	No. of attributes	No. of examples	Average accuracy		
			RBF GridSearch	Multi-scale RBF kernel + ES (obj: training rates)	Proposed method
Checkers	2	192	**83.32**	81.73	83.31
Spiral	2	582	**100.00**	**100.00**	**100.00**
LiverDisorders	6	345	61.74	63.19	**66.38**
IndiansDiabetes	8	768	64.97	65.10	**76.16***
ThreeOfNine	9	512	53.51	53.51	**100.00***
TicTacToe	9	958	65.34	65.34	**99.48***
BreastCancer	10	699	86.41	88.41	**95.99***
ParityBits	10	1024	48.05	48.54	**57.71**
SolarFlare	10	1066	**80.87**	**80.87**	**80.87**
ClevelandHeart	13	270	55.56	55.55	**83.34***
Australian	14	690	55.51	55.51	**56.38**
German-org	24	1000	70.10	70.20	**74.80**
Ionosphere	34	351	66.10	66.38	**95.15***
Tokyo	44	959	81.02	82.17	**90.82***
Sonar	60	208	70.67	75.96	**89.41***

***** Statistical significance at level 0.01 for the difference between the proposed method and RBF GridSearch.

These results show the accuracies of the proposed method (using the multi-scale RBF kernel and ES with 5 subsets cross validation) that are significantly higher than GridSearch on almost all datasets. Although the training rates can be the objective function, their average accuracies is not higher than GridSearch for some datasets.

This is because it may over-fit training data when the kernel is more flexible. Hence, subsets cross validation is a good choice to avoid the over-fitting problem.

4.2 Multi-class problems

SVM is the binary classifier for two-class data. However, the multi-class classification problems can be solved by voting schema methods based on a combination of many binary classifiers [3]. One possible approach to solve *k*-class problem is to consider the problem as a collection of *k* binary classification problems. *k*-classifiers can be constructed, one for each class. The k^{th} classifier constructs a hyperplane between class *k* and the *k*-1 other classes [3]. A new example will be classified according to a classifier that yields the maximum value of decision function. This schema is commonly called *one against the rest* and showed in Figure 5.

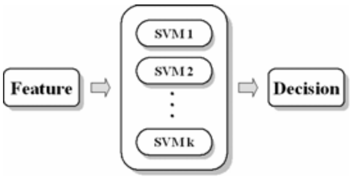


Figure 5. Multi-class problem

The proposed method has been tested on two multi-class problems from UCI. Both problems are composed of 3 classes. The experimental results are shown in Table 2. These results show that the accuracies of the proposed method are better than those of the RBF kernel using GridSearch on both problems.

Table 2. Results of multi-class problems

Datasets	No. of attributes	No. of examples	Average accuracies	
			RBF GridSearch	Proposed method
BalanceScale	4	625	85.92	**88.16**
Waveform	21	5000	33.92	**46.84**

5. CONCLUSIONS

The non-negative linear combination of multiple RBF kernels with including weights is proposed for support vector classification. The proposed kernel is proved to be the admissible kernels by Mercer's condition. Then, the evolutionary strategy is applied to adjust the hyperparameters of SVM. Subsets cross validation are considered to be the objective function in evolutionary process to escape from the over-fitting problem.

The experimental results show the abilities of the proposed method through their average accuracies on 5 folds cross validation. The multi-scale RBF kernel yields the better results. Furthermore, the experimental results also show the evolutionary strategy is effective in optimizing the hyperparameters, especially when the ranges of each parameter 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. Therefore, this method is very suitable for the problems where we have no prior knowledge about parameters.

ACKNOWLEDGEMENTS

The authors acknowledge the financial support provided by the Thailand Research Fund and the Royal Golden Jubilee Ph.D. Program.

REFERENCES

1. V.N. Vapnik, *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, USA, 1995.
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. J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, UK, 2004.
6. B. Schölkopf, and A.J. Smola, *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*, MIT Press, London, 2002.
7. I. Rechenberg, *Evolutionsstrategie: Optimierung technischer systeme nach prinzipien der biologischen evolution*, Frommann-Holzboog Verlag, Stuttgart, Germany, 1973.
8. H.-P. Schwefel, *Evolution and Optimum Seeking*, John Wiley and Sons, Inc., New York, 1995.
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.E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, US, 1989.
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. C.L. Blake and C.J. Merz, “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.