International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:5, No:3, 2011

Genetic Folding: Analyzing the Mercer's Kernels Effect in Support Vector Machine using Genetic Folding

Mohd A. Mezher, Maysam F. Abbod

***Abstract*—**Genetic Folding (GF) a new class of EA named as is introduced for the first time. It is based on chromosomes composed of floating genes structurally organized in a parent form and separated by dots. Although, the genotype/phenotype system of GF generates a kernel expression, which is the objective function of superior classifier. In this work the question of the satisfying mapping's rules in evolving populations is addressed by analyzing populations undergoing either Mercer's or none Mercer's rule. The results presented here show that populations undergoing Mercer's rules improve practically models selection of Support Vector Machine (SVM). The experiment is trained multi-classification problem and tested on nonlinear Ionosphere dataset. The target of this paper is to answer the question of evolving Mercer's rule in SVM addressed using either genetic folding satisfied kernel's rules or not applied to complicated domains and problems.

***Keywords*—**Genetic Folding, GF, Evolutionary Algorithms, Support Vector Machine, Genetic Algorithm, Genetic Programming, Multi-Classification, Mercer's Rules

I. Introduction

THIS document The feature mapping space [1] is a technique of extending the linear models to produce nonlinear models. The complexity of detecting linear relations to be learned depends on the way it is represented in the feature space. Ideally a representation of linear models that matches a specific problem should be chosen in the produced feature space. Therefore, one can map a nonlinear problem from input space to a new high dimensional space (called feature space) using suitable mapping functions to use a linear model in the feature space. This mapping function is chosen in advance and may be defined as

$$\Phi(x) = [\phi_1(x), ..., \phi_n(x)]' \quad (1)$$

This step is equivalent to mapping the input space $\Re^n$ into a new space $\Re^m$. Such mappings are corresponding the positive definite kernels and leads to solving a quadratic optimization problem with similar constraints as in H.

M. A. Mezher is with the School of Engineering and Design, Brunel University, UK (e-mail: mohd.mezher@ brunel.ac.uk).
M. F. Abbod, is with the School of Engineering and Design, Brunel University, UK (e-mail: maysam.abbod@ brunel.ac.uk)

Figure 1 shows an example of a feature mapping from a two dimensional input space to a two dimensional feature space, where the data cannot be separated by a linear SVM function in the input space, but can be in the feature space. The figure also shows the main components of the mapping function where the task of choosing the most suitable kernel and its parameters is known as model selection problems.

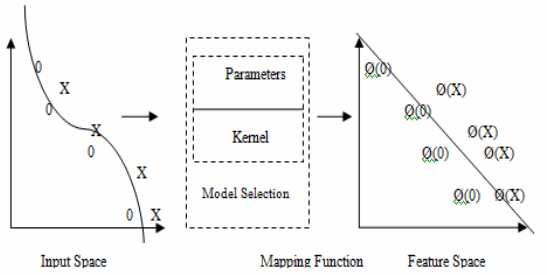


Fig. 1 A feature mapping space can simplify the input space

Consider an input data is referred to as the input space T, while $H = \{\phi(x) : x \in X\}$ is called the feature space. Usually, the dimension in H is much higher than in T. In fact, mapping data into higher space is simplify the task that has been known for a long time in ML, and then raises number of techniques for selecting the best representation of models. The decision function of mapping data can be expressed as:

$$f(x) = \sum_{i=1}^{N} w_i \phi_i(x) + b, \quad (2)$$
$$y = sign(f(x))$$

where w being normal vector to the hyperplane and b being the perpendicular distance of the hyperplane from the origin. However, in computational problems, the choice of $\Phi(x)$, curse of dimensionality and difficulties of generalization of the models can be sensitive when performing such a mapping [4, 49]. These problems are avoided in SVM in somehow by means of the 'implicit mapping' described in the next section.

In this paper, GF is applied for generating new kernel functions in SVM. Then, it is compared either undergoing to Mercer's rule techniques or not. The rest of the paper is organized as follows: support vector classification is

introduced in brief in Section 2. Making kernels structure is presented in Section 3. Details of model selection problems are shown in Section 4. An introduction of genetic folding algorithm is briefed introduced in Section 5. Experimental results on benchmark problems are given in Section 6. The conclusion of the paper comes later.

II. Support Vector Classification

SVC, class of SVM [1], classifies data by determining a set of Support Vectors (SVs) that outline a hyperplane in feature space. In general, produced classifier has the properties of maximizing the margin and minimizing the generalization error, based on a chosen kernel. Kernel functions improve the classifier's generalization capability, and it may affect the classification accuracy. However, finding a kernel function to fit a problem is not an easy mission. Therefore, diverse mechanisms have been investigated; some used predefined kernel functions such as Radial Base Function (RBF), others used Evolutionary Algorithm (EA) mechanisms. For a problem in binary classification let

$$T = ((x_1, l_1)...(x_n, l_n)) \text{ Since, } x \in \Re^m, l \in \{-1,+1\} \quad (3)$$

Suppose x_i is n-dimensional input data and l_i corresponds to the class associated with x_i. The task of classification function $f(x)$ is to learn mapping $x_i \to l_i$ using the training data (x, l) in T. SVM classifies data by constructing a hyperplane satisfies $< w.x > + b = 0$. SVM classifier finds SV points where are lying on $+l$ and $-l$ hyperplane with maximum margin and minimum classification errors between them. On the other hand, non-linearly separable dataset, SVM introduces a slack variable ξ to accept some errors and C is a cost parameter controls the compromise between training error and classifier complexity. Formally, the following primal optimization problem to be solved

$$\min_{w \in \Re} \frac{\|w\|}{2} + C\sum_{i=1}^{n} \xi_i \quad (4)$$
$$\text{Subject to } l(< xw > + b) \geq 1 - \xi$$

To transform this optimisation problem into its corresponding dual Lagrangian problem,

$$\min_{a \in R} W(\alpha) = \tfrac{1}{2}\sum_{i,j=1}^{t} \alpha_i \alpha_j l_i l_j < x_i \cdot x_j > - \sum_{i=1}^{t} \alpha_i \quad (5)$$
$$\text{Subject to} \sum_{i=1}^{t} \alpha_i l_i = 0 \; and \; C > \alpha_i \geq 0, \; i = 1,...,t$$

where α is Lagrange multiplier. Consider z is a new sample to be classified. Therefore, the discriminate function will be:

$$f = \sum_{i=1}^{t} \alpha_i l_i < x.z > + b, \; i = 1,...,t \quad (6)$$

SVM proved to be a promising classification algorithm in different fields. However, SVM has a risk in selecting optimum kernel function to be fitted to the problem in hand.

III. Making Kernels

In general, decision function cannot expect to obtain a very firm hyperplane i.e. tight to SV's points. Normally, real-world data are not linearly separable, even when the most outlying attributes are ignored [21]. However, SVM is stated completely in terms of inner products between vectors, and then the inner products can be replaced by a kernel function to be more flexible and stable. The kernel $K < x_i, x_j >$ again, maps the attributes implicitly into some feature space and then a suitable feature space is obtained. Since no explicit mapping is required, the problem is expressed completely in terms of finding fitted kernel function and its parameters in that feature space. Selecting proper kernel functions and free parameters to be fitted to the problem is very common problem in SVM.

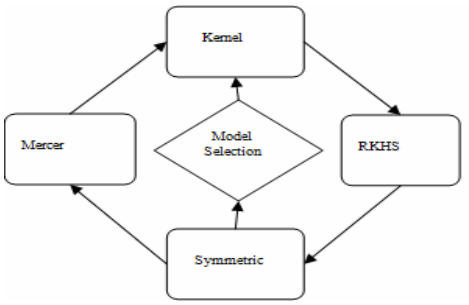


Fig. 2 Making kernels functions

Since the kernels have characterizations are required to be satisfied. The Figure 2 shows a cycle of the kernel's life are needed to testify the rules. The figure also shows that published works have been dedicated recently by using the cycle's component to develop new kernels function with their models. Some of the published work have used symmetric kernel matrix to produce a classifier in the feature space without satisfying other rules [6] especially Mercer's rule [3-5]. However, these approaches are developed EAs to combine standard SVM with either Genetic Algorithm [2, 8-9] or Genetic Programming [3-7] to produce new kernels. Some of these works have been recombined predefined kernels with either arithmetic or logical operators. Furthermore, in this paper, kernel functions that have been experimented satisfying all the rules shown in Figure 2, and it has been transferable easily to none satisfied rules.

In practical, the only information available to the in hand experiment is about the training data. The use of kernels is to map these data implicitly into a feature space and to train a

linear machine in such a space. The key to this approach is to find a kernel function that can be evaluated efficiently in that space. Once we computed the inner product $\langle x_i, x_j \rangle$ in feature space, it becomes possible to build a non-linear learning machine. A direct computation method is called kernel matrix.

Definition 1.Kernel Matrix (Gram Matrix) [1] given a function K to evaluate the inner products in a feature space with feature map ϕ such that for all $K \in R^{m \times m}$ and $x \in X^m$ a kernel matrix is defined for a set of vectors $\{x_1,...,x_m\}$ as:

$$[K]_{i,j} = \left\langle \phi(x_i), \phi(x_j) \right\rangle = k(x_i, x_j) \quad (7)$$

Moreover, the mapping function must be symmetric, and positive definite. Consider a finite input space $X = \{x_1,...,x_m\}$ and suppose $k(x_i, x_j)$ is a symmetric function on X, then the Mercer's theorem is defined as in definition 2.

Definition 2. Mercer's Rule [22] is a series of representation for kernels used to describe the corresponding Reproducing Kernel Hilbert Space (RKHS) for a symmetric function $K(x_i, x_j)$ on X, then there exists a kernel function such that $K(x_i, x_j) = <x_i, x_j>$ if and only if the matrix is positive semi-definite:

$$[K]_{i,j} = (K(x_i, x_j))_{i,j=1}^{n} \quad (8)$$

This means for all $\alpha \in R^m$ there are non-negative eigenvalues such as:

$$\langle \alpha, K\alpha \rangle \geq 0 \quad (9)$$

IV. Model Selection

The reformulation of kernels is critical to generalize a classifier that maximizing the margin and minimizing the test's error. For example; the polynomial kernel and its degree are effected the classifier generalization and the performance's results. This is due to the fact that different feature space results from different kernel functions and parameters in the original input space [2]. The downside of this, however, is that choosing kernel function and its parameters may be difficult or impossible. The large number of selection for a specific classifier causes problems both for requiring long memory and computational time due to the complexity of determining an obtainable model of a priori high dimensional dataset. In particular, in SVM, the learning of approximating functions is expressed as a specific model selection task which has to be found for a desired classifier.

In order to obtain an accurate classifier, SVM provides a number of control parameters that have to be tuned through given problems. The efficiency of a classifier is affected by the non-linear kernel function and their parameters [20]. Therefore, the control of fitting model selection in SVM is combined by a specific kernel and its parameters. However, the kernel-parameters are the few tunable settings in SVMs controlling the complexity of the resulting hypothesis [2]. In [12], stated that the selection models play a crucial role in building a prediction model with high accuracy and stability. In general, the aim of EA is to tune the hyper-parameters of SVM in order to achieve highest fitness value. The recent approaches for model selection can be distinguished into two common methodologies. First approach is the re-sampling techniques such as cross-validation search [12, 19], while second approach is involved by using one of the EAs algorithms [1, 3-9].

However, some works incorporate a priori fixed model selection into the learning method. Some has been used a polynomial kernel, RBF kernel and a Gaussian kernel widely. There is also another estimation method for model selection called grid search [19]. In [18], Vapnik mentioned that a different degree of values a different feature is computed and influenced. Although such model selection problem has got researchers' attention, new frameworks to analysis such these rules is still a challenging problem as mentioned.

V. Genetic Folding Algorithm

Genetic Folding (GF) [16-17] is a novel algorithm inspired by the folding mechanism in the RNA sequence. GF can represent an NP problem using a simple array of floating number instead of using a tree structure. GF starts with generating an initial population of randomly compound of arithmetic operations. Then, valid chromosomes (expression) will be evaluated. In this case, each chromosome has a fitness value depends on the fitness function we develop. The chromosome is then selected by the roulette wheel. After that, the fittest chromosome is subjected to the genetic operators to generate new populations in an independent way. In each population, the chromosomes are also subjected to a filter to check the validity of the chromosome. The genetic operators we used are the one-point crossover and the swap mutation operators. The whole process is repeated waiting for the optimum chromosome (kernel) to be achieved.

The chromosomes in GF are considered as the main distinguished structure in the algorithm. The chromosome can be represented as an individual (solution) in the search space. The main components of the GF chromosome are that each gene has three parts to be considered. The three parts are: an index number of the gene in the chromosome (father) and two points inside the gene (children). However, in the GF algorithm we included seven arithmetic operators to be conducted in our search [16-17]. The arithmetic operators are both one operand operator (sine, cosine, tanh, log) and two operands operator (plus, minus, multiply).

In general, GF encodes any equation by dividing it to two sides (left and right sides) and every time each side is divided into other divisions until a chromosome formed. This process depends on the property of the arithmetic operations used. The chromosome is formed by starting with the high property term

(e.g. the division operator) to end up with the less property values (terminals). In the meantime, the gene's structure has two components to be considered; the left side (ls) number and the right side (rs) number which represents the left and the right child respectively. However, there are three types of father's relationships; two children (two operands), one child (one operand) and no child (terminal).

In the meantime, to decode a chromosome, take the first gene which has two divisions. Suppose, the first father is an operator, therefore, it has two children. The children have operands; ls child and rs child. This process is defined as folding, for the way of folding the ls child (minus) and the rs child (multiply) over the father cell (plus). Repeatedly, for each father, one has a number of children to be called every time. However, GF encoding/decoding [16-17] is continuing until a kernel function is represented. Even though GF may use different genetic operators [14, 16-17], this work presents two types of operators to create new solutions in the search space; one point crossover and swap mutation operator. The new valid expression formulates the SVM kernel function using the arithmetic operators were conducted shown better results for valid expression satisfied the kernel's rules.

VI. Experimental Design

This section provides a detailed description of our experimental design. The experiments were conducted to analysis the efficiency/accuracy of satisfying the kernel's rules or without. The experimental results of SVM employs GF undergoes into either satisfied rules or not have been compared. The Ionosphere dataset considered in this experiment is appropriate to investigation the GF system since intermediate number of samples and high number of features (351×34). Leave N out of five cross-validations was used. In this case, we defined N as 10% for the testing data and 90% for the training data. Each time of the training/testing datasets, independently set of GF individuals conducted and evaluated.

The GF experiments utilize some functions from GPLAB toolbox [10]. The toolbox employs different types of arithmetic operators. Even though, GPLAB toolbox is not introduced for SVM and kernel function satisfied Mercer's rule for new solutions, the toolbox is developed to include GF concepts to draws GP's tree structure. Then, the proposed algorithm is compared by using the rules or without.

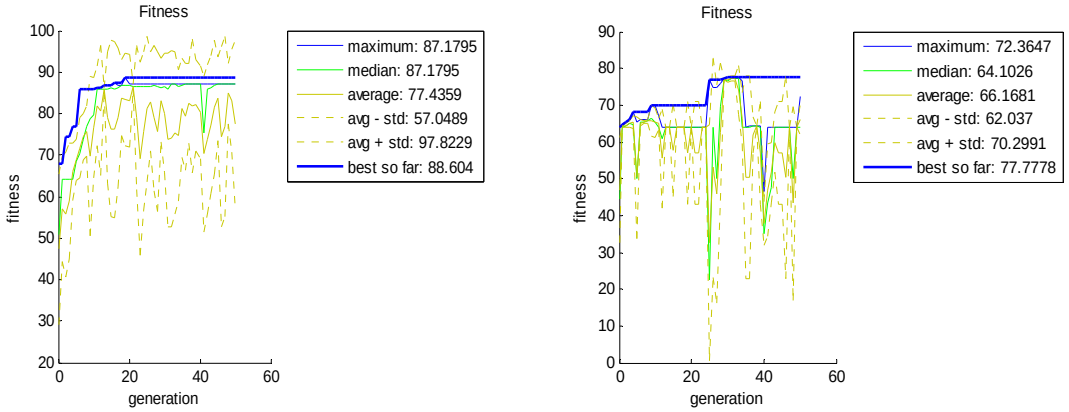
By the way, GF draws numbers of GF chromosomes have been found using SAMR operator [14]. Several parameters of values may be considered here, however, the main purpose here is to show a comparison between chromosomes have been defined either by using basic math's operator or with trigonometric operators undergo to either with kernel's rules or not. Table II shows complex chromosomes were folded with high accuracy values. The examination in the table shows also the importance of the operators which may affect the quality of solutions in GF chromosomes.

However, GF is compared by either using Mercer's rule or none Mercer's rule and it is experienced with Ionosphere dataset. GF methods used here is the palindrome method [15] which depends on numbering the terminals with their proper indices. Here, in this experiment is comparisons between GF algorithm satisfied Mercer's rules or not satisfied Mercer's rule is conducted. However, for comparison purposes, each GF chromosome was represented in GP tree structure by using GPLab package. The experiment shows different levels of comparisons have been included: fitness values, population diversity, Pareto front and the final produced kernel function. However, 50 generations with 20 individuals are built-in with basic and advanced math operators. In general, GF using Mercer's rule was able to converge more rapidly than without satisfied Mercer's rule. In addition, GF has obtained on optimum kernel function with less number of diversity (Figure 3(a)) and nodes (Figure 4(a)).

However, the accuracy values have been converging earlier in Figure 3(a) and better than without rules satisfied as in Figure 3(b). GF using Mercer's rule have drawn optimum kernel functions in Figure 6(a) with small number of diversity in the chromosomes over generations as shown in Figure 4(a). Figure 4 shows a good started accuracy value but with a better value in Figure 4(a).

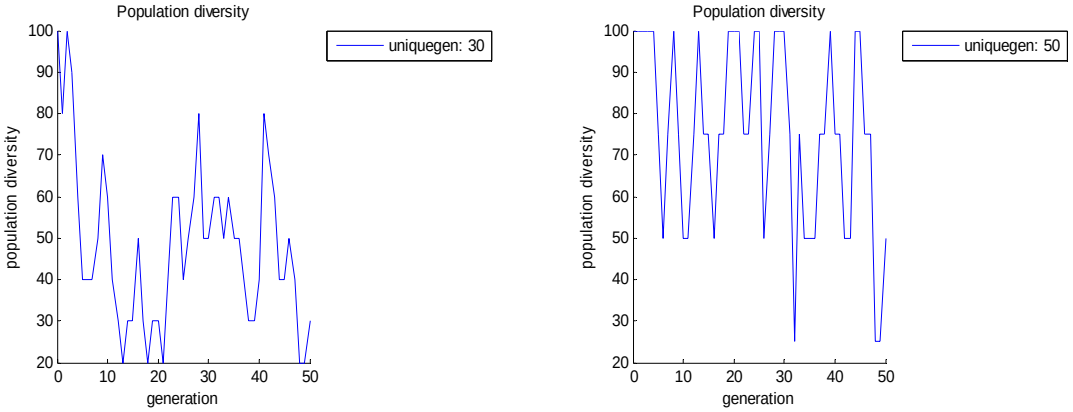
TABLE I
GF Chromosome of Ionosphere Dataset using all Math Operators

No	GP Sequence/ GF Sequence	Accuracy
1	sine(tanh(X,minus_s(Y,X)),minus_v(sine(Y,X),tanh(X,Y)))	
1	2.7 3.4 0.2 5.6 0.4 0.4 8.11 9.10 0.8 0.8 12.13 0.12 0.12 0.12	64.52
5	sine(tanh(X,minus_s(Y,X)),minus_v(sine(X,Y),tanh(X,Y)))	
5	2.7 3.4 0.2 5.6 0.4 0.4 8.11 9.10 0.8 0.8 12.13 0.12 0.12 0.12	64.52
10	sine(tanh(X,minus_s(Y,X)),Y)	
10	2.7 3.4 0.2 5.6 0.4 0.4 0.1	84.90
15	sine(tanh(X,minus_s(Y,X)),Y)	
15	2.7 3.4 0.2 5.6 0.4 0.4 0.1	84.90
20	sine(tanh(X,minus_s(Y,X)),Y)	
20	2.7 3.4 0.2 5.6 0.4 0.4 0.1	84.90
25	sine(tanh(X,minus_s(Y,X)),Y)	
25	2.7 3.4 0.2 5.6 0.4 0.4 0.1	84.90
30	sine(X,Y)	
30	2.3 0.1 0.1	87.17
35	sine(X,Y)	
35	2.3 0.1 0.1	87.17
40	sine(X,Y)	
40	2.3 0.1 0.1	87.17
50	sine(X,Y)	
50	2.3 0.1 0.1	88.60



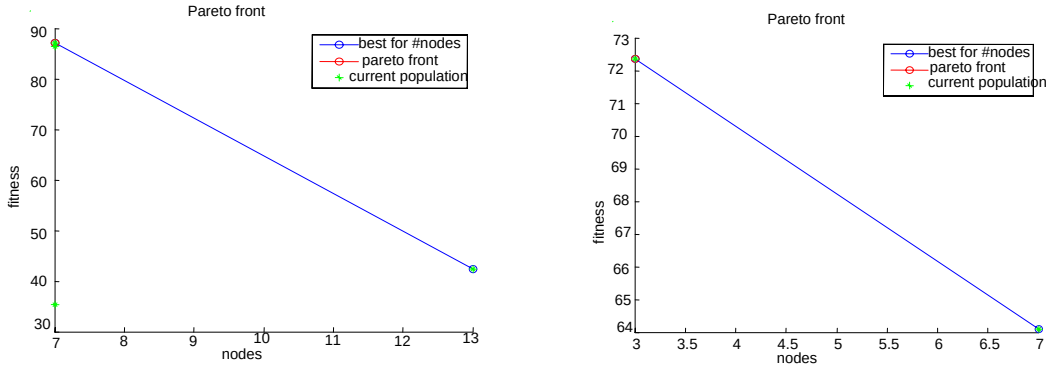
(a) GF with Mercer's Rule (b) GF without Mercer's Rule

Fig 3 GF using Mercer's Rule or without Comparison to the Fitness Values



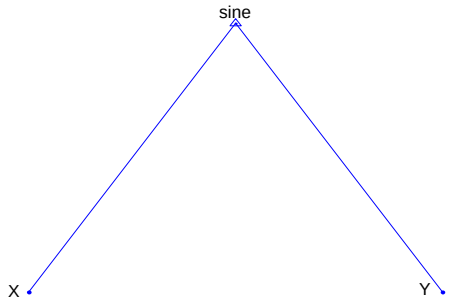
(a) GF with Mercer's Rule (b) GF without Mercer's Rule

Fig 4 GF using Mercer's Rule or without Comparison to the Diversity

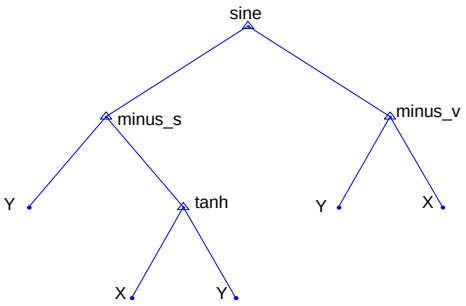


(a) GF with Mercer's Rule (b) GF without Mercer's Rule

Fig 5 GF using Mercer's rule or without Comparison to the Pareto Front



(a) GF with Mercer's Rule



(b) GF without Mercer's Rule

Fig 6 GF using Mercer's Rule or without Comparison the Predicted Optimum Tree Structure

VII. Conclusion

The question of evolving Mercer's rule in SVM was addressed using genetic folding. Due to the varied set of Math's operators and the high efficiency of the algorithm, it was possible to compare dissimilarly performing systems such as implementing kernels under Mercer's rule and seven Math's operators. The capability to introduce new kernel function allows GF to produce an accurate classifier for multi-classification problems. As most existing solutions either search evolutionary for models rely either on kernels or parameters, this analysis can help understand the different of applying kernel's rules satisfied Mercer's rule or not.

GF comes with many advantages state as; dynamic chromosomes, energetic GF operators also saving time and space of the memory. However, one main advantage shown here is the ability to predict new solutions early. The standard SVM has been conducted using GF algorithm for analyzing kernel's rules satisfied or not. The experimentation results show a promising outcome of GF in Ionosphere classification datasets in comparison to GP algorithms without satisfying the rules. Future improvements will involve the kernel's parameters instead of just using the involved produced kernels in the input space. GF for other multi-classification and regression datasets would be our next implementation research.

References

[1] Cristianini, N. and Shawe-Taylor, J., 'An Introduction to Support Vector Machines: and Other Kernel-Based Learning Methods', 1st edn. Cambridge University Press, (2000).
[2] Chen, P-W., Wang, J-Y. and Lee, H-M., 'Model Selection of SVMs Using GA Approach', IEEE International Joint Conference, vol. 3, 2035- 2040 (2004).
[3] Dioşan, L., Rogozan, A. and Pecuchet, J-P.,' Optimising Multiple Kernels for SVM by Genetic Programming', Evolutionary Computation in Combinatorial Optimization, vol. 4972, 230-241 (2008).
[4] Diosan, L., Rogozan, A. and Pecuchet, J-P. , 'Evolving Kernel Functions for SVMs by Genetic Programming', Machine Learning and Applications, ICMLA, 19-24 (2007).
[5] Gagné, C., Schoenauer, M., Sebag, M. and Tomassini, M., 'Genetic Programming for Kernel-based Learning with Co-evolving Subsets Selection', LNCS, no. 4193, 1008-1017 (2006).
[6] Howley, T. and Madden M., 'The Genetic Kernel Support Vector Machine: Description and Evaluation', Artificial Intelligence Review, vol. 24, no. 3-4, 379-395 (2005).
[7] Koza, J. R., 'Genetic Programming: on the Programming of Computers by Means of Natural Selection', 74-147, Cambridge, MA: The MIT Press, (1992).
[8] Lessmann, S., Stahlbock, R. and Crone, S. F., 'Genetic Algorithms for Support Vector Machine Model Selection', Proc. of the Intern. Joint Conf. on Neural Networks (IJCNN'06), Vancouver, Canada, (2006).
[9] Rojas, S.A. and Fernandez-Reyes, D., 'Adapting Multiple Kernel Parameters for Support Vector Machines using Genetic Algorithms', IEEE, vol. 1. 626-631 (2005).
[10] Silva S., 'GPLAB: A Genetic Programming Toolbox for MATLAB', (2007).
[11] Sivanandam, S. and Deepa, S., 'Introduction to Genetic Algorithm', Springer, 15-130 (2008).
[12] Staelin C., 'Parameter Selection for Support Vector Machines', HP Laboratories, (2003).
[13] Sullivan, K. and Luke, S., 'Evolving Kernels for Support Vector Machine Classification', Genetic And Evolutionary Computation Conference, 1702 – 1707 (2007).
[14] Mezher, M., Abbod, M. 'Evolving Self-Adaptive Genetic Algorithm using Nonlinear Support Vector for Classification Problems'. The International Journal Annals Computer Science Series. (2010).
[15] Mezher, M., Abbod, M. 'Palindrome Genetic Folding for Support Vector Regression Problems'. International Journal of Computer Systems Science and Engineering. Submitted on December (2010).
[16] Mohd Mezher, Maysam Abbod. 'Genetic Folding: A New Algorithm for Solving Multiclass SVM Problems'. Applied Soft Computing, Elsiver Journal. Submitted on September (2010).
[17] Mohd Mezher, Maysam Abbod. 'Genetic Folding: A New Class of Evolutionary Algorithm for SVM'. Society's Specialist Group on Artificial Intelligence (SGAI) International Conference on Artificial Intelligence. Cambridge, UK. August (2010).
[18] Vapnik V.N., 'Statistical Learning Theory'. 1998, John Wiley and Sons: USA.
[19] Chang C., Lin J., 'LIBSVM: A Library for Support Vector Machines'. in 8.1. (2001).
[20] Lessmann S., Stahlbock R. and Crone F. 'Genetic Algorithms for Support Vector Machine Model Selection'. in International Joint Conference on Neural Network. (2006). Vancouver, Canada,: Proc. of the Intern. Joint Conf. on Neural Networks (IJCNN'06).
[21] Kim H., Holand P., Park H. and Christianini N., 'Dimension Reduction in Text Classification with Support Vector Machines'. Journal of Machine Learning Research,. 6: 37-53. (2005)
[22] John Shawe-Taylor, Cristianini N., 'Kernel Methods for Pattern Analysis'. (2004), Cambridge University Press: UK.