Evolving Self-Adaptive Genetic Algorithm in Nonlinear Support Vector Machines for Classification Problems

Mohammad Mezher, Maysam Abbod
School of Engineering and Design,
Brunel University, West London, Uxbridge, UB8 3PH, UK

ABSTRACT: Support Vector Machines (SVM) has shown a range of promising applications on classification problems. In this paper, we propose the genetic algorithm that employs Self-Adaptive Mutation Rate (SAMR) to develop kernel functions for SVM classifiers. The proposed SAMR model implemented the hybrid model for three advanced non-linear classification algorithms and shows competitive results in comparing to Grid SVM. Five publicly available datasets, cross validation correctness for Area Under Curve (AUC) have been involved. Improvements achieved may lead to biomarkers results.
KEYWORDS: Non-linear systems, Classification, Support Vector Machine (SVM), Self-Adapt, Genetic Algorithm (GA), PSVM, SSVM, LSVM, RBF Kernel, Model/parameter selection, Grid Search.

Introduction

Support Vector Machine (SVM) is a supervised learning kernel-based method proposed by Vapnik and introduced to solve binary-class problems using structural risk minimization. Recently, SVM has received promising results in various application domains e.g. medical diagnosis, text and image recognition. SVM starts by classifying projected training data into positive and negative classes in the feature space to generalize a decision hyperplane for unseen data. The produced hyperplane has maximum parallel margin and number of support vectors. SVM implies mapping of non-linear problems into linearly by using dot products and kernel tricks of the projected data [BBL04], [CST00].

In order to obtain accurate class predictions SVM provides a number of control parameters that have to be tuned through the given task. The control of a parameter of fitting (model selection) in SVM combines specified kernel and kernel parameters.

SVM model selection can be distinguished into two common methodologies. First approach is the re-sampling techniques such as grid search or cross-validation search [Sta03], [CL09]. While second approaches consist of using one of the evolutionary algorithms [SL07], [Sta03], [RFR05]. In this work, we propose Self-adaptive mutation rate for utilizing a genetic algorithm (GA) for model selection of various algorithms to determine all properties of the classifier in a solely data-driven manner.

In this paper, we are interested in Self-Adaptive Mutation Rate (SAMR) [Bac92] in order to find optimum values of kernel function. In our previous work [MK09], SAMR has been proofed to be superior in bioinformatics problems. For this purpose, we propose a hybrid model that uses SAMR combined with three different types of SVM. The SVM algorithms that have been implemented are Lagrangian, Smooth and Proximal SVM (LSVM, SSVM and PSVM respectively). Their techniques are to boost our method to gain more accurate results.

Moreover, the idea of the propose system is to overcome various obstacles. First issue is the model/parameter selection problem. Second issue is the number of the new SVM algorithms. Third issue, SAMR avoided the time/space consuming by reducing the number of genetic operators to be only one operator and by using the self-adaptive mutation rate. The hybrid model proposed here has shown competitive results.

However, before we introduce some review of model selection techniques that have been proposed for SVM classification (Section Three); we outlined in Section Two the idea behind SVM and kernel tricks. We also describe the linear and nonlinear classification of SVM. The proposed SAGA is explained in section four. In section five, experimental results are presented to show the significant accuracy of the introduced algorithm.

1 SVM Classification

In this section, we give a brief review of SVM classification with some of the explanations for nonlinearity of SVM then we introduced the kernel function we used in our paper. For more details see [BBL04], [CST00], [Cam01].

Now, consider a set of training data (X,Y) and the $f(x)$ a binary classification function such as:

$$((x_1,y_1)...(x_i,y_j)) \text{ For } x \in \Re^n, y \in \{-1,+1\}$$

Suppose x_i is a new vector to be classified. So that the decision boundary $f(x)$ finds which data points belong to the separated classes. For this problem, consider w as a weight vector, the parameter ε can be considered as a regularization parameter and γ determine the distance of the vector from the origin. The SVM is given by the following primal quadratic program:

$$\min \frac{1}{2} w'w + \varepsilon \sum \xi \qquad (1)$$

Subject to $y(< x \cdot w > -\gamma) \geq 1$

Obviously, the generalization of optimal hyperplane will exist to the positive hyperplane:

$$f(x) = y = sign(\sum_{i=1}^{N} x.w - \gamma + \xi) \geq 1 = 1 \qquad (2)$$

While the negative hyperplane:

$$f(x) = y = sign(\sum_{i=1}^{N} x.w - \gamma + \xi) \leq -1 = -1 \qquad (3)$$

So, the decision boundary can be found by using the dot product and the weight of the sample:

$$f(x) = y = sign(\sum_{i=1}^{N} x.w - \gamma + \xi) - \gamma - \xi = 0 \qquad (4)$$

Therefore $\xi = 0$ for separable datasets and $\xi > 0$ for non-separable datasets or to accept some error. Notice, the margin is the distance of the two boundaries and the error variable can be minimized by maximizing the margin [C01].

In addition to the slack variable ξ to classify non-separable datasets, non-linear classification equations can be used. Unfortunately, non-linear classifications have some limitations, e.g., duality properties [CS00]. Stability, simplicity and duality of linear computations can be obtained by transforming the input vector space to a feature vector space. In the feature space linear kernel computations are implicitly produced.

Searching the optimal hyperplane in (1) is a Quadratic Programming (QP) problem, which can be simplified by substituting the weight in the primal presentation:

$$w = \sum_{i=1}^{N} \alpha y x_i \qquad (5)$$

Where α is the vector of nonnegative Lagrange multipliers associated with the constraints (2), (3) and (4).

1.1 Non-linear support vector machine

Projection technique ϕ maps the input vector $f : x_i$ to the feature vector $\phi(x_i)$ by using one of the non-linear functions *F*. Non-linear SVM can be reformulated from the linear equation on (2), (3) and (4) by: $f : x_i \rightarrow \phi(x_i) : F$ for positive, negative and decision hyperplane respectively.

Duality of nonlinear classification is a QP problem which can be constructed by replacing the weight value of the input vector x_i in (5) into equation (1) of the feature vector $\phi(x_i)$.

1.2 Kernel Support Vector Machine

The kernel functions of a non-linear classifier are to project the input vector space into a linearly high-dimensional space [CST00] [Cam01]. The projected data can be classified by substituting (2) in (5) to get:

$$F(K) = y = \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha y K(\phi_i(x), \phi_j(x)) - \gamma + \xi \geq 1 = 1 \qquad (6)$$

K can be any type of kernel functions in learning machines (e.g., feed-forward neural networks or polynomial classifiers). The RBF kernel proofed to produce good generalization and give a universal approximation through the use of RBF nodes in the hidden layer [CST00], [Cam01], [SL07], [Sta03], [RFR05].

Let $\phi_i(x), \phi_j(x)$ be two transformed feature space. So, to measure similarity between input vector space by using the RBF kernel:

$$K(\phi_i(x), \phi_j(x)) = \exp\left(\frac{-\left\| \phi_i(x) - \phi_j(x) \right\|^2}{\sigma^2}\right) \qquad (7)$$

Even though, using RBF kernel, in this paper, has simplicity (one parameter [RFR05]) generalization [Cam01] and consistency with no limitations (such as the polynomial's degree) [Sta03], it can be implemented

to any other types of kernel functions.

2 Prior Works and SVM Algorithms

Genetic algorithm (GA) is the most popular technique in evolutionary computation research [SD08]. Genetic algorithm in the model selection aims to tune the hyper parameter of SVM in order to achieve highest of classification accuracy. However, has been studied widely either by re-sampling techniques such as grid search or cross-validation search [Sta03], [CL09] or by using one of the evolutionary algorithms [SL07], [Sta03], [RFR05]. To our knowledge, no methods have been applied for SVM using Self-Adaptive Mutation Rate (SAMR) of GA. The aim of this paper is to construct and analyze SVM kernel with penalty of ε parameter and RBF parameter using SAMR. The proposed algorithm enables the mutation operators to be involved alone on a set of adjacent genes efficiently, without the needs of the other genetic operators [Bac92], [MK09].

Plenty of Non-linear SVM classifications have been implemented on an extensive range of datasets. Recent works from Wisconsin University claimed that the standard SVM can be improved by hybridization with other techniques [FM01], [MM01], [LM00]. The new algorithms have achieved better results over the standard SVM but with some limitation on determining the classifier's parameter. The efficiency of classifier may be affected by the non-linear classifier and the kernel function with their parameters [SL07], [Sta03], [LSC06]. Here, we adopted Proximal, Lagrangian and Smooth SVM (PSVM, LSVM and SSVM) to empower the influence of standard SVM and to overcome on the limitations exist on PSVM, LSVM and SSVM.

Proximal Support Vector Machine (PSVM) assigns the input data in the feature space to the closer of two parallel hyperplane that are pushed apart as far as possible from the centre [FM01]. Smooth Support Vector Machine (SSVM) is based on the smoothing function's method [LM00]. The Lagrangian Support Vector Machine (LSVM) is based on the Lagrangian formulation of the standard quadratic program of a linear support vector machine [MM01].

Here, we describe nonlinear LSVM, SSVM and PSVM briefly in point of mathematical view. So, consider following variables for the nonlinear classification problem for some $u > 0, \varepsilon > 0$ and $A \in R^{M \times N}$:

$$w = A' yu, \gamma = -e' yu, \quad (8)$$

In order to the Karush-Kuhn-Tucker (KKT) conditions to be satisfied, the gradients of Lagrange should be equal to zero [MM01]. Here we got:

$$v = (\frac{I}{\varepsilon} + GG')^{-1}, G = y[K - e] \quad (9)$$

The KTT optimality conditions can be obtained by setting the gradients equal to zero with respect to (γ, ω, y, u) and subject to $0 \le u \in R^N$ is,

$$\min_{u,\gamma} = \frac{\varepsilon}{2}\|y\|^2 + \frac{1}{2}\left\|\begin{bmatrix} u \\ \gamma \end{bmatrix}\right\|^2 - v(y(Kyu - e\gamma) + \gamma - e) \quad (10)$$

However, by recovering the variables (γ, ω) from (10) we can determine the separating surface of the PSVM.

Therefore, by using the same notations Lagrange in LSVM can be defined as:

$$\min_{0 \le y} \frac{u'}{2}(\frac{I}{\varepsilon} + yKy)u - e'u \quad (11)$$

And the Smoothing function in SSVM can be defined as:

$$\min_{u,\gamma} \frac{\varepsilon}{2}\|p(e - y(Kyu - e\gamma), \alpha\|_2^2 + \frac{1}{2}(u'u + \gamma^2) \quad (12)$$

However, we introduced all the previous functions with the common parameter ε for easy tracking function in our proposed system. The parameter ε>0 plays a main role in estimating the error value of w multiplied by the summation error of ξ. The bounding plane w and the error variable γ determine the maximal margin and the accepted noise (error) of the classified classes respectively.

Even though, the RBF's parameter (σ) and the regularization parameter (ε) determine the quality of the classifier. Unfortunately, the previous algorithms had no concerning of the parameters. Therefore, while we adopted these algorithms we take in our concern the SVM's parameters, which should be made carefully for good classifiers to be achieved.

3 Implementation

Self-Adaptive Mutation Rate (SAMR) [BAC92] uses the same GA life cycle to produce offspring [Bac92]. GA life cycle uses genetic operators to produce new solutions in the search space to reach an optimal solution. However, the idea behind SAMR is to implement the mutation operators

only without the needs for other genetic operators. Therefore, by using SAMR the time and the computation complexity will be consumed.

3.1 Genetic Algorithm Chromosome representation

In general, Self-Adaptive Genetic Algorithm (SAGA) implements self adaptive mutation operator to produce new offspring. SAGA has proof to be efficient to solve huge and complicated problems in the bioinformatics field such as RNA Folding [MK09]. SAMR is based on modifying mutation rate (μ) by evolution. In SAGA every individual has a different μ value encoded into its string. Genetic operators provide no direct feedback mechanism on how good or bad parameter values are. As a result the high-quality parameter values will present an evolutionary advantage to the individual it belongs to. At the end, the self-adaptation mechanism has been successfully applied to continue optimisation problems with evolving strategies and evolutionary programming to achieve optimum solutions [Thi02].

In details, Chromosomes combine a number of genes. Each gene might be represented in different type of representation. In order to facilitate a data driven determination of the kernel function by SAMR GA, we have to define a genotype encoding for the parameters. This is accomplished by using eight binary genes for the regularization parameter ε and eight binary genes for the RBF parameter σ. Genes encoded as four bits for one of three algorithms (i.e. PSVM, SSVM, and LSVM) and four genes for the mutation rate (μ). Therefore, the chromosome is set to 24 genes in length. Binary representation may save some time consuming of easy computations during the execution time.

All genes started with integer random values encoded into binary genes, then all genes subjected into the self-adaptive mutation operator achieved by μ proportion. In the fitness function as the genes will be decoded into integer values except μ real number, the mutation rate multiplies by 10 or 100 to find real values. In addition to the μ decoded into the chromosome, the ε value and σ parameter decoded as well in binary representation with 8 genes each. In contrast to the self-adaptation, fixed μ require pre-determined specific values at the beginning which will not be adapted with successive generations.

3.2. Evaluation function

The Receiver Operating Characteristic (ROC) [Faw05] analysis has been used in medicine, radiology, and other areas for many decades, and it has been introduced relatively recently in machine learning and data mining. ROC can measure the generalization performance of a classifier by computing the AUC value.

Let K a kernel function, c_i a chromosome, σ, ε, μ values decoded in c_i. Let X represents 5 cross-validations, and $M \times N$ is the dimension of data. The evaluation function can be computed by:

$$F = Avg\{AUC_i((K(\sigma), \varepsilon))_{i=1}^{N}\}_{j=1}^{X} \quad (13)$$

Acquiring the average value of AUC for a number of cross-validation improves the quality of the off-springs during the GA life cycle. By computing the average of AUC, we guarantee the efficiency of selecting the classifier's parameters.

The next section shows preliminary experimental results of RBF kernels adopted for these algorithms. The SVM using SAGA has achieved a significant classification performance in different UCI datasets (http:// archive.ics.uci.edu/ml/).

4. Simulation Results

Different SVM has been used to address the parameter/model selection such as [S03], [RFR05] and recently [LSC06]. LIBSVM [CL09] uses the grid-search on finding the parameters ε and σ using cross-validation and they found that trying exponentially growing sequences of ε and σ parameters is a practical method to identify good parameters.

In this work, the functionality of grid search in LIBSVM is adopted for finding the optimum parameters. The method is based on picking a parameter of (ε, σ) pairs with the best five cross-validation accuracy. Then the selected pairs are to be used on the test datasets. LIBSVM is widely used online, and it uses the grid search for model selection. Therefore, the standard SVM with the grid search is compared with the proposed SAMR GA.

Various improvements have been made to the nonlinear classification algorithms and simulation results were produced using five UCI datasets. The advantage of improving these algorithms is that can easily and quickly handle large datasets. All experimental results are based

on a simple MATLAB code. Such advantages can be considered for further improvements.

The proposed algorithm has been implemented in Matlab 7. The implementation of LSVM, SSVM and PSVM algorithms is based on their online codes [FM01], [MM01], [LM00] respectively. While the code for model selection of the RBF and SVM parameters, as well as the SAMR GA with the ROC fitness function was developed and added to the algorithms.

Various compressions have been taken to clarify the accuracy of the classifier that has been reached. Table 1 shows results of standard SSVM, PSVM, and LSVM in comparison to SAMR GA-SSVM, SAMR GA-PSVM, and SAMR GA-LSVM. The datasets used are Ionosphere (351×34), Pima Indians (768×8), Bladder Cancer (693×12), CLEVE (297×13) and BUPA Liver (345×6).

In all the experimental results, we demonstrate the three algorithms (SSVM, PSVM and LSVM) using SAMR GA or without in compression to the soft-margin nonlinear SVM using grid search. The algorithm ran for 100 generations using 50 random individuals in the search space. Then, the new offspring generated only by the self-adaptive mutation operator. The mutation rate (μ) could start randomly with probability $0.01 \le \mu \le 0.09$. For more details see [Bac92] and [MK09].

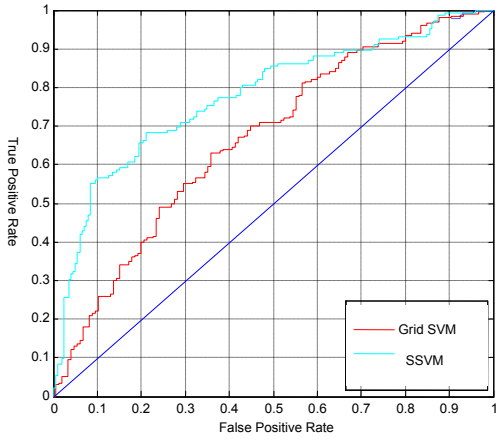
Using the roulette wheel [SD08] to reproduce offspring depends on selecting the fittest individual from the previous population. Combining SAMR GA with SSVM, LSVM and PSVM produced significant accuracy in comparing to the algorithms without SAMR.

The SAMR GA computes the optimum fitness value using one of the elected SVM algorithms. The best individual fitness represents the final parameters vector for σ, ε, μ and algorithm type. The mean of AUC of five cross-validation computed by the fittest parameters that have been self-adapted.

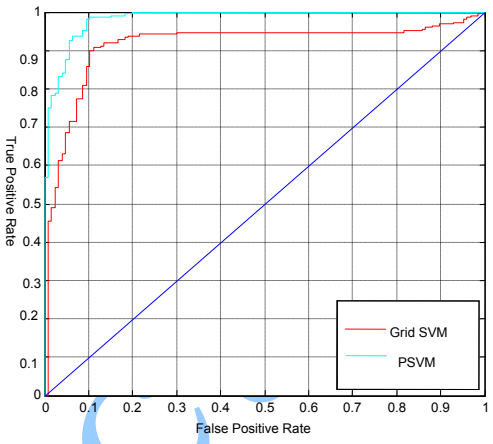
Many experiments with different datasets have been tested as shown in Table 1. Figure 1 shows the classifier with SAMR GA for different datasets in comparing to the Grid SVM and random classifier (the diagonal line). Generally, there is a good improvement for SVM with SAMR GA over the SVM without SAMR GA on both data sets. Figure 1(a, b, d) shows that the SAMR GA LSVM has succeeded to improve the accuracy of the classifier over the grid SVM.

SAMR GA draws good classifiers in various dataset by using the different type of algorithms. SAMR GA in BUPA dataset (Figure 1(a)) and CLEVE dataset (Figure 1(d)) draws optimum classifier by using SSVM.

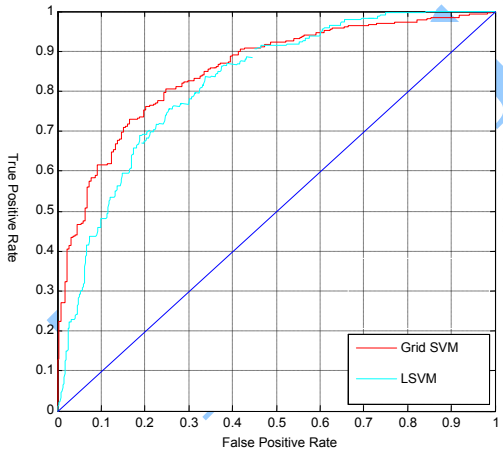
PSVM for IONO dataset (Figure 1(b)) and for PIMA dataset (Figure 1(e)) achieved 91.4% and 95.8% in comparison to 68.5% and 72.2% with grid-search technique respectively. The SAMR GA in Bladder Cancer (Figure 1(c)) shows no ability to beat the Grid search by using LSVM.



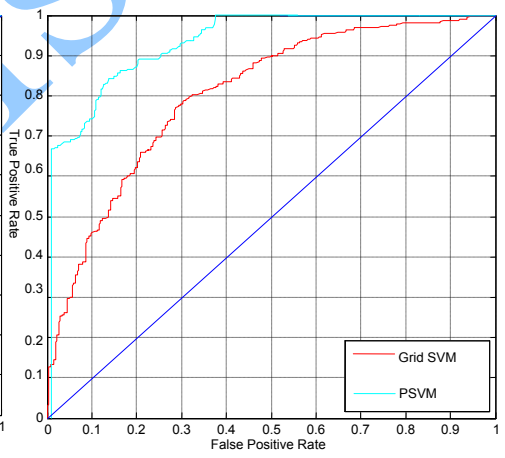
(a) BUPA dataset



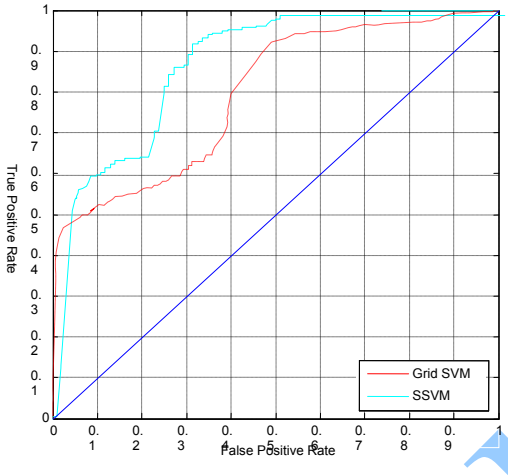
(b) IONO dataset



(c) Bladder cancer dataset



(d) CLEVE dataset



(e) PIMA dataset

Figure 1. ROC curve of Grid SVM vs. SAMR GA of SVM for all datasets

Figure 2 shows the grid search of standard SVM for Cleve dataset. The best accuracy that has been accomplished was 75.5% with log2(C) =-1 and log2 (sigma) =-1 and the best selected parameters were ε (C) =0.5 and σ (sigma) =0.5 where C and sigma represent ε, σ respectively.

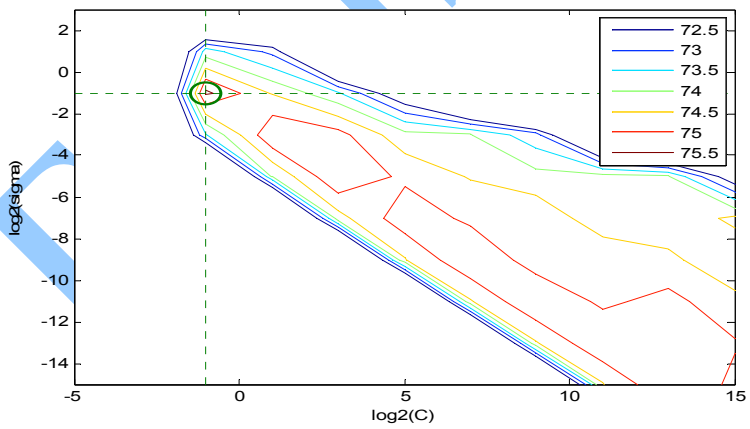


Figure 2. Grid Search of Cleve dataset

Table 1 shows better classification accuracy using SAMR GA in comparison to the non SAMR GA version. Significant results were achieved on the UCI dataset by SAMR GA using non standard SVM for model/parameter selection. All the introduced algorithms tuned model/parameter very well except on the Bladder Cancer dataset. LSVM

consumed time in huge dataset such as PIMA. AUC for all dataset (in %) has been computed for five experiment results. All the experimental results were conducted on a dual core processor machine with 2GB memory. Notice that # for the LSVM algorithm was out of memory.

Table 1. AUC for all dataset (in %) in form "average of five experiments"

Algorithm / Data	Grid SVM	Without SAMR GA			With SAMR GA
	SVM AUC	SSVM AUC	PSVM AUC	LSVM AUC	AUC (Algorithm)
BUPA	66.6	67.7	63.2	47.1	**71.8** (SSVM)
IONO	91.4	90.3	91.1	65.6	**92.8** (PSVM)
Bladder Cancer	**87.1**	80.4	79.5	81.2	81.2 (LSVM)
CLEVE	75.5	80.3	79.8	50.7	**88.1** (SSVM)
PIMA	68.5	67.7	66.4	#	**72.2** (PSVM)

Bold fonts are Best AUC fitness values. Optimum algorithm found is in parentheses.

Conclusion

In this work, three non-linear SVM algorithms have been combined with self adaptive mutation rate of the genetic algorithm. The soft-margin nonlinear SVM using grid search is utilized for tuning the RBF kernel parameter and the regularization parameter. The results of the Grid SVM compared with the three different nonlinear SVM with or without SAMR GA. The proposed algorithm has reliably found optimum parameter settings and an algorithm across a wide range of machine learning problems. Additionally, this proposed algorithm can be used to obtain a model/parameter selection suitable for adaptive kernels with the fittest algorithm, which can be sensitive to other conditions in the search space. Using SAMR GA tuning optimum kernel functions with their parameters can be extended as a future work.

References

[Bac92] **T. Back** - *Self-Adaptation in Genetic Algorithm,* in Proceedings of the 1992 First European Conference on Artificial Life (Cambridge, MA). MIT Press, Citeseer, 263-271, 1992.

[BBL04] **O. Bousquet, S. Boucheron, G. Lugosi** - *Introduction to Statistical Learning Theory,* Advanced Lectures on Machine Learning. 3176. 169-207, 2004.

[Cam01] **C. Campbell** - *An Introduction to Kernel Methods, Radial Basis Function Networks: Recent Developments in Theory and Applications*, Physica Verlag Rudolf Liebing KG. 66. 155-192, 2001.

[CST00] **N. Cristianini, J. Shawe-Taylor** - *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods and Other Kernel-based Learning Methods,* Cambridge, United Kingdom, Cambridge University Press, 2000.

[CL09] **C. Chang, C.J. Lin** - *LIBSVM: a library for support vector machines*, http://www.csie.ntu.edu.tw/~cjlin/libsvm/, 2009.

[Faw05] **T. Fawcett** - *An Introduction to ROC Analysis*. Pattern Recognition. 27. 861–874, 2005.

[FM01] **G. Fung, O. L. Mangasarian** - *Proximal Support Vector Machine Classifiers,* KDD 2001: Seventh ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, San Francisco, August 26-29. 77-86, 2001.

[LM00] **Y. Lee, O. L. Mangasarian** - *SSVM: A Smooth Support Vector Machine for Classification,* Computational Optimization and Applications. 20. 5-22. 2000.

[LSC06] **S. Lessmann, R. Stahlbock, S. F. Crone** - *Genetic Algorithms for Support Vector Machine Model Selection*, Proc. of the Intern. Joint Conf. on Neural Networks (IJCNN'06), Vancouver, Canada, 2006.

[MK09] **M. Mezher, T. Khader** - *Genetic Algorithm Self-Adaptive Mutation Rate for DNA Folding (GASAMR),* Online Journal of Bioinformatics. 10. 82-92, 2009.

[MM01] **O. L. Mangasarian, D. R. Musicant** - *Lagrangian Support Vector Machines*. Journal of Machine Learning Research. 1. 161–177, 2001.

[RFR05] **S. A. Rojas, D. Fernandez-Reyes** - *Adapting Multiple Kernel Parameters for Support Vector Machines using Genetic Algorithms*. IEEE. 1. 626-631, 2005.

[Sta03] **C. Staelin** - *Parameter Selection for Support Vector Machines,* HP Laboratories, 2003.

[SD08] **S. N. Sivanandam, S. N. Deepa** - *Introduction to Genetic Algorithms,* Springer, 2008.

[SL07] **K. M. Sullivan, S. Luke** - *Evolving Kernels for Support Vector Machine Classification,*.Genetic and Evolutionary Computation Conference, London, England. 1702-1707, 2007.

[Thi02] **D. Thierens** - *Adaptive Mutation Rate Control Schemes in Genetic Algorithms,* Proceedings of the 2002 Congress on Evolutionary Computation IEEE. No.1, 980–985, 2002.