A Multi-objective Genetic Algorithm for Model Selection for Support Vector Machines

Amal Bouraoui*, Yassine Ben Ayed, and Salma Jamoussi

Multimedia, InfoRmation systems and Advanced Computing Laboratory
MIRACL-Sfax University,
Sfax-Tunisia Technopole of Sfax: Av.Tunis Km 10 B.P. 242, Sfax-Tunisia 3021

Abstract. Selecting the proper Kernel function in SVMs and the specific parameters for that kernel is an important step in achieving a high performance learning machine. The objective of this research is to optimize SVMs parameters using different kernel functions. We cast this problem as a multi-objective optimization problem, where the classification accuracy, the number of support vectors and the margin define our objective functions. So, we introduce a method based on multi-objective evolutionary algorithm NSGA-II to solve this problem. We also introduce a multi-criteria selection operator for our NSGA-II. The proposed method is applied on some benchmark datasets. The experimental obtained results show the efficiency of the proposed method.

Keywords: Parameter selection, kernel function setting, multi-objective genetic algorithm NSGA-II, support vector machines (SVMs).

1 Introduction

Support vector machines (SVMs), proposed by Vapnik [1], are a powerful and popular techniques which have been widely used in various fields of classification problems such as pattern recognition, bioinformatics and finance [2, 4].

With SVM, a classification model is generated in the training process using the training data. The model is then used to classify data. The crucial and largest problems encountered in establishing the SVM model are how to select the kernel function, its corresponding parameters values and the misclassification penalty parameter C. The setting quality of SVM kernel functions and hyper-parameters influence the performance of learning and generation. In fact, inappropriate kernel function and inappropriate hyper-parameters tuning lead to poor classification results [5].

Genetic algorithms (GAs) have been successfully applied to solve the problem of parameters selection for SVM classification due to its ability to discover good solutions for complex searching and optimization problems. The drawback of GAs that considering one objective to be either maximized or minimized. But, it has been observed that a single objective is no sufficient and there are many

* Corresponding author.

D.-N. Pham and S.-B. Park (Eds.): PRICAI 2014, LNAI 8862, pp. 809–819, 2014.
© Springer International Publishing Switzerland 2014

objectives that may to be taken into account for obtaining an effective SVM classifier. These objectives are most often conflicting in nature.

For that, motivated by the multi-objective optimization problems, in this paper we introduce a new method to simultaneously set the appropriate kernel function, its parameters and SVM parameters for SVM classification based on multi-objective evolutionary algorithm NSGA-II for which we have implemented several improvements to perform the desired task.

The remainder of this paper is organized as follows: We begin in Section 2 with an overview of related work of model selection. Section 3 and 4 describe respectively SVMs and kernel selection and multi-objective genetic algorithm NSGA-II. In section 5, we present our proposed approach. The experimental results on chosen benchmark datasets are discussed in Section 6. We conclude this work in Section 7.

2 Related Work

The most simple way to tune hyperparameters (model selection) is the grid search algorithm [6]. It trains SVMs with all desired combinations of hyperparameters, evaluates their performance and outputs the settings that achieved the highest accuracy. The results obtained by this technique demonstrate that it is robust and works effectively and efficiently on a variety of problems [7]. Nevertheless, it is time consuming. Moreover, a better region on the grid must be specified before doing a grid-search. More recently, the model selection was seen as an optimization task and many optimization algorithms were suggested to perform model selection like gradient descent (such as [8,9]). However, these methods have the drawback that the kernel function and the score function (or at least an accurate approximation of this function) for evaluating the performance of the hyperparameters should be differentiable with respect to all hyperparameters [10]. Moreover, their results depend on the initialization. The genetic algorithms (GAs) are also applied to tune SVM parameters (such as [11, 15]). One very complete study is that of Huang and Wang [16], where they aim to perform feature and parameters selection simultaneously. They show that their method improves significantly SVMs accuracy. In this approach three criteria: classification accuracy, number of selected features, and the feature cost were combined to create a single objective function. In [14], Zhao et al. have made an area search table, based on the asymptotic behaviors of support vector machines and after analyzing it, they proposed at first a parametric distribution mode fused with a genetic algorithm in order to improve classification performance. The results obtained indicate that classification accuracy of genetic algorithm based on parametric distribution model is better than that of grid search [14]. Then, they used this approach to simultaneously optimize the SVM parameters and the feature subset. In [15], a method based on crossbreed genetic algorithm has been proposed to choose the kernel function and its parameters. This method uses two fitness functions which are produced according to the two criterion of SVM's performance: empirical estimators and theoretical bounds for the

generalization error for improving the performance of SVMs. It has been able to avoid premature convergence and, consequently, improves predictive accuracy [15]. Other researchers have used the PSO (Particle Swarm Optimization) methods. For example, Lin and al. [17] have proposed a method based on particle swarm optimization for SVM parameters tuning with and without feature selection. They use as objective function the classification accuracy. The comparison of their obtained results with the obtained results of other methods based on PSO algorithm show the efficiency of their approach.

3 SVMs and Kernel Function

Originally, SVMs are designed as a statistical learning technique which can solve linear and nonlinear binary classification. Then, it extended for multiclass problems by designing a number of two-class SVMs. The basic idea of SVMs is to map the input space x into a high dimensional feature space ($z = \Phi(x)$), and to classify the transformed feature by a hyperplane ($w \cdot z - b = 0$). SVMs aim to find the optimal separating hyperplane, which maximizes the margin between the two classes. The mapping is performed by a kernel function $k(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$. There are many kernel functions and the most popular kernel functions are given in Table 1.

Table 1. The most popular kernel functions

Linear Kernel	$k(x, y) = x \cdot y$
RBF Kernel	$k(x, y) = exp\left(-\gamma \ \|x - y\|^2\right)$
Polynomial Kernel	$k(x, y) = (\gamma(x\, y) + r)^d, \gamma \succ 0$
Sigmoid Kernel	$k(x, y) = tanh\left(\gamma x^T\, y + r\right)$

When using RBF kernel in SVM, two major parameters C and γ must be set appropriately. The choice of value for C influences on the classification outcome. If C is too large, then the classification accuracy rate is very high in the training phase, but very low in the testing phase. If C is too small, then the classification accuracy rate unsatisfactory, making the model useless. Parameter γ has a much greater influence on classification outcomes than C, because its value affects the partitioning outcome in the feature space. An excessively large value for parameter γ results an over-fitting, while a disproportionately small value leads to under-fitting [18].

4 Multi-objective Genetic Algorithm NSGA-II

NSGA-II is one of the most efficient multi-objective evolutionary algorithms [19]. It appears as one of the reference algorithms in multi-objective optimization.

This algorithm is characterized by a sorting procedure of the population (parent and offspring) in successive fronts according to the non-dominance relation and a selection method (called Fast Non-dominated Sort) based on a performance criterion (called crowded comparison) and a so-called niching technology. For more details readers can refer to [19]. The practical implementation of NSGA-II on our specific problem differs to the standard NSGA-II especially at the selection operator (which is binary tournament for standard NSGA-II).

5 The P-SVM Proposed Method:From NSGA-II to SVM Model Selection Using Several Kernels

This section is devoted to analyzing and describing our multi-objective P-SVM (Parameter selection for SVM) method for simultaneous kernel choice and hyper-parameters determination for the SVM. We consider the task of model selection as an optimization problem that requires the choice of several efficient criteria to be optimized. We decided to choose the classification accuracy, the margin and the total number of support vectors as criteria to be optimized simultaneously. Thus, an individual (chromosome) with high classification accuracy, a high margin value and a low total number of support vectors present an optimal compromise between these criteria. The details of our NSGA-II to simultaneous SVM kernel and hyper-parameters tuning are given by the following issues.

(1) Chromosome design and initialization
For genetic algorithms, one of the key issues is to encode a solution of the problem into a chromosome.

To implement our proposed approach, we have used the most popular kernel functions to be optimized simultaneously. These kernels are recapitulated in Table 1. So, five parameters need to be optimized: the misclassification penalty parameter C, the SVM parameter ϵ and parameter(s) (d, γ or r) according to the treated kernel function. Each individual in the population must encode these five real values in order to represent a point in the search space (see Figure 1). In our application, we used a real-coded scheme because it is more accurate and better suited to continuous optimization problems. The search ranges of values of C,ϵ, γ, d and r are respectively $\{0.0001, \ldots, 10000.0\}$, $\{0.00002, \ldots, 0.2\}$, $\{0.0005, \ldots, 5\}$, $\{1, \ldots, 9\}$ and $\{0.0, \ldots, 20.0\}$.

First of all, we randomly selected T potential solutions. T denotes the size of the population and it was fixed at 15 in our case after several experiments to decide which is the best value.

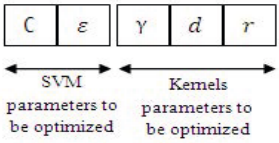

Fig. 1. Chromosome design

(2) Genetic Operator

Genetic operator consists of three basic operators: selection, crossover and mutation. In this paper, a new selection approach is presented to pick fittest individuals.

– *Selection operator*

During the selection step we select best individuals in current population as parents to generate offspring. Since we aim to optimize three different criteria: the margin, the number of the support vectors and the classification accuracy, we had the idea of crossing optimal individuals within the sense of different criteria. This idea consists on randomly select six individuals and keep only three, each of them is optimal for one criterion. The three solutions chosen will be crossed. The process is repeated a number of times to have at least T children in the child population. Indeed, crossing the individual judged best for a criterion with other individual judged best for an other criteria, we hope to exploit the best genes of these individuals to create new better solutions. This proposal led to satisfactory results.

– *Crossover operator*

The crossover allows the creation of new individuals according to a given process. Its aim is to enrich the diversity of the population by manipulating the structure of chromosomes (individuals). As we employed the real-encoding scheme for C, ϵ and kernel parameters, we used the SBX crossover operator [20] which is defined as follows.

$$\begin{cases} x_i\left(e_1\right) = 0.5\left[\left(1+\bar{\beta}\right)x_i\left(p_1\right) + \left(1-\bar{\beta}\right)x_i\left(p_2\right)\right] \\ x_i\left(e_2\right) = 0.5\left[\left(1-\bar{\beta}\right)x_i\left(p_1\right) + \left(1+\bar{\beta}\right)x_i\left(p_2\right)\right] \end{cases}.$$

where $\bar{\beta} = \begin{cases} (\alpha u)^{\frac{1}{\eta_c+1}} & if\ u \le 1/\alpha \\ \left(\frac{1}{(2-\alpha u)}\right)^{\frac{1}{\eta_c+1}} & otherwise \end{cases}$, $\alpha = 2-\beta^{-(\eta_c+1)}$ and $\beta = 1 + \frac{2}{x_i(p_2)-x_i(p_1)}\min\left\{\min\left(x_i\left(p_1\right), x_i\left(p_2\right)\right) - x_i^{min}, x_i^{max} - \max\left(x_i\left(p_1\right), x_i\left(p_2\right)\right)\right\}$. η_c represents the distribution index and u a random number between 0 and 1.

The three retained individuals at the selection step are crossed each other. For example, if we take the parent having the best accuracy rate so he is crossed with the best parent for the margin and with the best parent for the number of support vectors.

– *Mutation operator*

The mutation consists of randomly changing the value of one (or more) component(s) of the individual. In our algorithm, we used the polynomial mutation operator [21], which is defined as: $x_i{}' = x_i + \Delta_{max}\bar{\delta}$.

Where $\bar{\delta} = \begin{cases} \left(2u + (1-2u)(1-\delta)^{\eta_m+1}\right)^{\frac{1}{\eta_m+1}} - 1 \;\; if \; u \leq 0.5 \\ 1 - \left(2(1-u) + 2(u-0.5)(1-\delta)^{\eta_m+1}\right)^{\frac{1}{\eta_m+1}} \;\; otherwise \end{cases}$,

$\delta = \frac{\min\left(x_i - x_i^{min}, x_i^{max} - x_i\right)}{x_i^{max} - x_i^{min}}$, η_m represents the distribution index and $\Delta_{max} = x_i^{max} - x_i^{min}$.

In this work and after many experiments, we used the value of 0.015 as mutation probability.

– *Replacement operator*

The NSGA-II is based on an elitist replacement. Indeed, the population of parents and the population of their descendants are assembled and sorted according to the criterion of not-dominance to identify different fronts. The best individuals will be found in the first fronts. To form the new population the fronts were added until a T number of individuals is reached. Here we used the same principle of the NSGA-II replacement operator.

(3) Stopping criterion

Each genetic algorithm must have a break point. We used as stopping criterion the maximum number of generations. It was set at 40 after many comparisons based on several experiments on tested databases.

The flow chart of our multi-objective genetic algorithm using several kernel is shown in Figure 2:

To classify individuals in each generation, we need to calculate our objective functions (the classification rate, the margin and the number of the support vectors). To do this, we must train and evaluate our SVM classifier using the four considered kernel functions. When the RBF kernel is selected, only the parameters (C, ϵ and γ) were used for building SVM model. Indeed for each individual in the population we treat the four used kernel functions. Then, we compare their results in terms of accuracy rate and adopt as kernel function for this individual one that gives the best model (which has the highest value of classification accuracy). The function of dominance will depend on the best kernel selected for each individual. Therefore, the Pareto fronts represent not only a set of best parameters but also the best kernel for the treated problem. Choosing the best kernel for each individual also affects the selection and replacement operators.

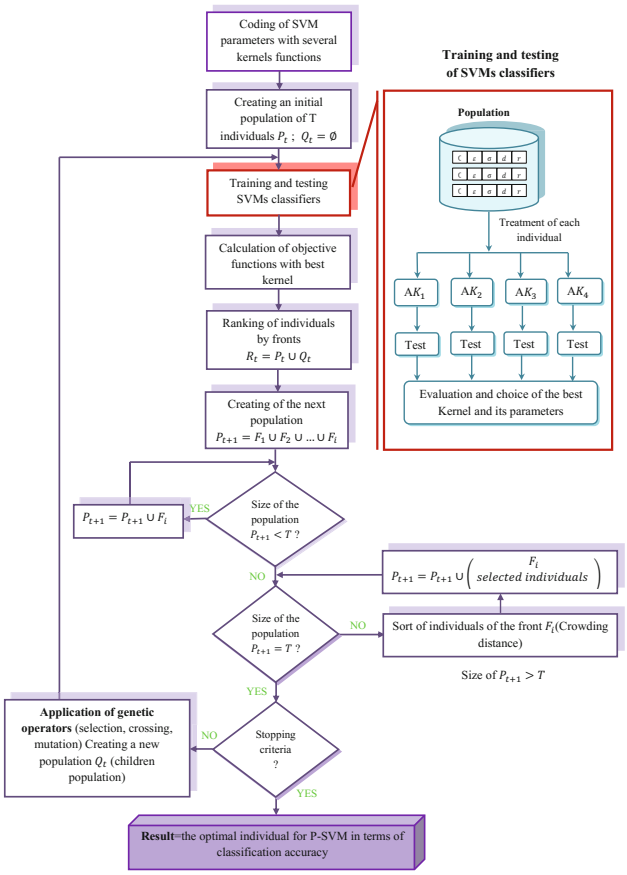

Fig. 2. Our SVM model selection with several kernels

6 Experimental Results

To implement our proposed approach, we have developed our method using the java langage. We also used the LibSvm which is a library for support vectors classification. A general use of LibSvm involves two steps: first, a training data set is used to obtain a model and subsequently the model is validated on a test data set for predictive power. Our proposed method is evaluated and validated on five UCI datasets which can be taken from LibSvm webpage [1] or ML-Repository webpage [2].

[1] `www.csie.ntu.edu.tw/~cjlin/`
[2] `http://archive.ics.uci.edu/ml/`

Table 2 describes these datasets in terms of number of attributes, instances and classes.

Table 2. Information about the used datasets

Benchmark	Classes	Attributes	Instances
Australian	2	14	690
Ionosphere	2	34	351
Heart	2	13	270
Pima	2	8	768
Glass	7	9	214

We tested our method, at first, in optimizing the SVM parameters with the RBF kernel only (P-SVM (RBF only)) in order to compare our results with those of literature.

Scaling was applied to prevent feature values in greater numeric ranges from dominating those in smaller numeric ranges, and to prevent numerical difficulties in the calculation. Scaling the feature value may contribute to improve the classification accuracy of SVM [14]. The range of each feature value was scaled to the range $[-1, +1]$.

Table 3 reports the results obtained by our method P-SVM (RBF only) compared with the results obtained by the NSGA-II method.

Table 3. Results obtained by P-SVM (RBF only)

Benchmark	NSGA-II		P-SVM (RBF only)	
	AA	MA	AA	MA
Australian	88.12	89.78	**88.26**	**89.85**
Ionosphere	100.0	100	**100**	**100**
Heart	87.59	89.99	**88.9**	**90.74**
Pima	83.25	85.45	**86.04**	**87.66**
Glass	**82.56**	**90.67**	81.4	**90.67**

Where AA and MA design respectively average accuracy and maximum accuracy.

A comparison between the results obtained by applying NSGA-II and those obtained by applying P-SVM (RBF only) shows that the latter is more effective. Indeed, crossing optimal individuals within the sense of different criteria allows the detection of more satisfactory solutions. Consider the case of the Heart and Pima databases, we can detect classification accuracies equal to 90.74% and 87.66%; more accurate than those obtained by the classic selection strategy of NSGA-II (89.99% and 85.45%).

For this, we adopt the strategy of crossing optimal individuals within the sense of different criteria to implement our proposal of optimizing SVMs parameters using several kernels.

In Table 4, the results obtained by our proposal (P-SVM (RBF only)) were compared with other picked up for state of the art algorithms [16], [17] and [14] which optimized a one kernel function: the RBF.

Table 4. A comparison between results obtained by P-SVM-RBF and [16], [17] and [14]

Bench.	[16]	[14]	[17]	P-SVM (RBF only)	
				AA	MA
Aust.	88.09	86.81	88.09	88.26	89.85
Iono.	96.61	98.57	97.5	100	100
Heart	94.58	91.11	88.17	88.9	90.74
Pima	82.98	81.97	80.19	86.04	87.66
Glass	-	-	78.04	81.4	90.67

It is noted that our approach allows accuracies rates better than those achieved by the three mentioned works. Such as, for the Ionosphere, Pima and Glass databases we could detect accuracy rates equal to 100%, 87.66% and 90.67%.

Table 5 presents the results obtained when applying our method of optimizing SVM parameters using several kernel functions, compared with those obtained by optimizing only the RBF kernel.

Table 5. Results of simultaneous optimization

Benchmark	P-SVM		P-SVM (RBF only)	
	AA	MA	AA	MA
Australian	**91.87**	**92.86**	88.26	89.85
Ionosphere	100	100	100	100
Heart	**91.86**	**94.23**	88.9	90.74
Pima	**86.43**	**88.5**	86.04	87.66
Glass	**89.46**	**93.38**	81.4	90.67

The obtained results prove the effectiveness of our proposed approach (P-SVM). We were able to achieve accuracies rate more accurate than those achieved by the P-SVM-RBF. Such as, for the Australian, Heart and Glass databases we could detect accuracy rates equal to (92.86%, 94.23% and 93.38%) instead of (89.85%, 90.74% and 90.67%) when using the RBF once.

7 Conclusion

In literature, the most proposed methods for the SVM model selection are tackling the misclassification penalty C and kernel parameters. The choice of the kernel function and its parameters is an important step which influence the performance of SVM. Whereas, this choice is very difficult as it is dependent on databases. In the context, we have proposed a method enable to select the best kernel for a used dataset while selecting different combinations of parameters to fine tune. As SVM model selection can be considered as multi-objective optimization problem, we have applied NSGA-II to implement our approach according to three criteria: the classification accuracy, the margin and the total number of support vectors. We have implemented a new strategy concerning the selection operator. The results obtained show the effectiness of our proposal. In future, the simultaneous model selection and feature selection may be studied. The work can be extended for regression problems or unsupervised context. Moreover, the application of our algorithm on known biological data as the challenges that they present particularly those of dimensionality and the high precision sought can be envisaged.

References

1. Vapnik, V.: The nature of statistical learning theory. Springer, New York (1995)
2. Kotsia, I., Pitas, I.: Facial expression recognition in image sequences using geometric deformation features and support vector machines. IEEE Transactions on Image Processing 16(1), 172–187 (2007)
3. Ma, J., Nguyen, M.N., Rajapakse, J.C.: Gene classification using codon usage and support vector machines. IEEE/ACM Transactions on Computational Biology and Bioinformatics 6(1), 134–143 (2009)
4. Yu, L., Chen, H., Wang, S., Lai, K.K.: Evolving least squares support vector machines for stock market trend mining. IEEE Transactions on Evolutionary Computation 13(1), 87–102 (2009)
5. Keerthi, S.S., Lin, C.-J.: Asymptotic behaviors of support vector machines with Gaussian kernel. Neural Computation 15, 1667–1689 (2003)
6. Hsu, C.-W., Lin, C.-J.: A comparison of methods for multiclass support vector machines. IEEE Trans. Neural Networks 13(2), 415–425 (2002)
7. Staelin, C.: Parameter selection for support vector machines. Hewlett-Packard Co. Tech. Rep. Hpl-2002-354r1 (2003)
8. Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing multiple parameters for support vector machines. Machine Learning 46(1), 131–159 (2002)
9. Chung, K., Kao, W., Sun, C., Lin, C.: Radius margin bounds for support vector machines with rbf kernel. Neural Comput. 15(11), 2643–2681 (2003)
10. Frauke, F., Igel, C.: Evolutionary Tuning of Multiple SVM Parameters. In: Proceedings of the 12th European Symposium on Artificial Neural Networks (ESANN 2004). d-side publications, Evere (2004)
11. Liang, X., Liu, F.: Choosing multiple parameters for SVM based on genetic algorithm. In: 6th International Conference on Signal Processing, August 26-30, vol. 1, pp. 117–119 (2002)

12. Liu, H.-J., Wang, Y.-N., Lu, X.-F.: A method to choose kernel function and its parameters for support vector machines. In: Proceedings of 2005 International Conference on Machine Learning and Cybernetics, August 18-21, vol. 7, pp. 4277–4280 (2005)
13. Liu, S., Jia, C.-Y., Ma, H.: A new weighted support vector machine with GA-based parameter selection. In: Proceedings of 2005 International Conference on Machine Learning and Cybernetics, August 18-21, vol. 7, pp. 4351–4355 (2005)
14. Zhao, M., Fu, C., Ji, L., Tang, K., Zhou, M.: Feature selection and parameter optimization for support vector machines: A new approach based on genetic algorithm with feature chromosomes. Expert Systems with Applications 38(5), 5197–5204 (2011)
15. Liu, H.-J., Wang, Y.-N., Lu, X.-F.: A method to choose kernel function and its parameters for support vector machines. In: Proceedings of IEEE International Conference on Machine Learning and Cybernetics, vol. 7, pp. 4277–4280 (2005)
16. Huang, C.L., Wang, C.J.: A GA-based feature selection and parameters optimization for support vector machine. Expert Systems with Application 31(2), 231–240 (2006)
17. Lin, S.W., Ying, K.C., Chen, S.C., Lee, Z.J.: Particle swarm optimization for parameter determination and feature selection of support vector machines. Expert Systems with Applications 35(4), 1817–1824 (2008)
18. Liu, H., Yu, L.: Toward integrating feature selection algorithms for classification and clustering. IEEE Transactions on Knowledge and Data Engineering 17(4), 491–502 (2005)
19. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
20. Deb, K., Beyer, H.: Self-adaptive genetic algorithms with simulated binary crossover. Complex Systems 9, 431–454 (1999)
21. Deb, K., Tiwari, S.: Omni-optimizer: A generic evolutionary algorithm for single and multiobjective optimization. European Journal of Operational Research 185(3), 1062–1087 (2008)