CHAPTER 7

A GENETIC ALGORITHM WITH AN IMPROVED CROSSOVER TO OPTIMIZE THE SUPPORT VECTOR MACHINE PARAMETERS FOR SHOEPRINT RECOGNITION

Two parameters, C and $\sigma$, must be cautiously predetermined in establishing an efficient Support Vector Machine (SVM) model. Therefore, the purpose of this chapter is to develop a genetic-based SVM (GA-SVM) model that can involuntarily determine the optimal parameters, C and $\sigma$, of SVM with the highest predictive accuracy and simplification ability concurrently. This chapter pioneered on employing a real-valued Genetic Algorithm (GA) with an improved crossover operator to optimize the parameters of SVM for shoeprint classification. Experimental results show that the proposed GA-SVM model performs the best predictive accuracy, implying that integrating the GA with traditional SVM model is very successful.

7.1 INTRODUCTION

Before the training of SVMs, the value of their parameters must be stipulated by the user. According to Muller et al., (2001), the proper choice of these values affects the performance of the obtained classifier. In the decomposition of multiclass shoeprint classification problem, the ideal parameter values for each of the binary SVMs may differ. The empirical
search for these values through a trial-and-error approach is clearly impractical. This chapter investigates the use of Genetic Algorithms (GAs) (Michalewicz, 1996) to automatically tune the parameters of the binary SVMs contained in common decomposition strategies. Thus, the parameter adjustment problem was formulated as a search for combinations of parameter values able to minimize the error rates obtained in the multiclass problem solution. Two approaches are compared to the use of default values for the parameters. In the first approach, the GA searches for a set of parameter values which will be common to all classifiers in the decomposition. In the second approach, a set of values is selected for each binary classifier in the decomposition.

Min and Lee (2005) stated that the optimal parameter search on SVM plays a crucial role to build a prediction model with high prediction accuracy and stability. To make an efficient SVM model, two extra parameters: C and $\sigma^2$ have to be carefully predetermined. The first parameter, C, determines the trade-offs between the minimization of the fitting error and the minimization of the model complexity. The second parameter, $\sigma^2$, is the bandwidth of the Radial Basis Function (RBF) kernel. Consequently, the purpose of this study is to propose a model that can determine the optimal parameters (C and $\sigma^2$) of SVMs to yield the highest predictive accuracy and generalization ability for predicting model. The model was tested on the shoeprint features to compare its accuracy with that of other models that based on multivariate statistics and AI approaches.

The rest of the chapter is organized as follows: The following section introduces the basic GA and its operators. Section 7.3 presents the parameters to be tuned for SVM classifier. Section 7.4 illustrates the proposed GA with improved crossover operation and brief about the hybridization of GA with SVM for shoeprint image classification. Section 7.5 presents the
experimental results and analyzes the performance of the proposed GA-SVM model and the chapter is summarized in section 7.6.

7.2 GENETIC ALGORITHMS

Based on concepts from genetics and natural selection, GAs evolves a population of solutions in order to solve an optimization problem. In this process, the solutions, named individuals, are transformed during a sequence of interactions in a stochastic search into individuals with better quality or fitness. GAs have been successfully applied to several problems from the areas of control, planning, combinatorial optimization and Machine Learning (Beasley, 1997).

7.2.1 Basic GA

Given an initial population of possible solutions to the problem to be solved, a GA searches for the global solution through an iterative process. At each iteration, also named generation, a new population is produced, which contains evolutions of individuals selected from the previous generation. The initial population is generally composed of random solutions. The individuals are codified by a data structure named chromosome. In the basic or standard GA, the chromosomes are represented by bit strings. Each bit, also named gene, represents the presence (value 1) or absence (value 0) of a specific characteristic in the individual (Back, 2000). This is the representation adopted in Figure 7.1.

At each generation, the individuals are evaluated, quantifying their fitness to solve the problem. This evaluation is performed by a fitness function, which decodes the information contained in each individual chromosome into a measure of its quality.
Figure 7.1 Genetic operators: (a) one-point cross-over and (b) bit-flip mutation

According to their evaluation, some individuals in the population are selected for reproduction, producing descendants which will form a new population. This selection must privilege the fittest individuals, according to the natural selection principles. One of the existing selection methods is the binary tournament (Mitchell, 1998).

In the reproduction of the selected individuals, their characteristics or genes are combined to obtain two descendants. This combination is performed with the application of the cross-over genetic operator, which is a binary operator applied to two individuals. These individuals are named parents and their chromosomes are combined to produce two new individuals, named offspring. For the bit-string representation, a common cross-over operator is the one-point cross-over. Given two parents, a cut point is randomly chosen. The offspring are obtained by permuting the parents’ parts that are posterior to the cut point. Figure 7.1a represents this process. Simulating the stochastic nature of evolution, the cross-over is applied according to a rate $p_c$.

A second genetic operator usually applied is the mutation, which enforces a genetic variability in the new solutions. The mutation alters genes from the individuals generated in the cross-over step. For the bit-string representation, a bit-flip mutation, in which the value of a random gene from
the individual is altered according to a rate $p_m$, is commonly used. Figure 7.1b illustrates the bit-flip operator.

Other type of selection frequently used is the elitism, in which a proportion $p_e$ of the best individuals from the current population is directly sent to the new population. This selection prevents the loss of the current best solutions.

The procedures of population generation, evaluation of its individuals, selection and application of the genetic operators are iterated, forming the basis of the GAs. Depending on the initial population, the GA may produce distinct solutions to the same problem. Therefore, the algorithm is usually run several times with different initial populations. To stop the GA, different criteria may be used. The GA may be stopped, for example, when a maximum number of generations are reached or when the best individual is not modified for a given number of generations.

7.2.2 GA for Continuous Problems

The bit-string representation is generally used when the solutions are constituted by Boolean or discrete-valued variables. Nevertheless, when the variables are continuous, the use of a binary representation may present disadvantages. First, it is necessary to use several bits to represent the continuous value. The higher the precision of the float number, the larger the number of bits that must be added to the chromosome. As a consequence, the search space increases, making the problem solution more difficult. Another problem with the binary encoding of continuous values is that two close points in the continuous space usually are not close in the binary representation. To minimize this discrepancy, Grey encoding may be adopted, in which two consecutive values differ by one bit (Michaelewicz, 1996). Nevertheless, to make the GA solution conceptually closer to the continuous
variable space, it is common to use a real-valued string representation. Several studies point advantages in approximating the individuals’ representation to the characteristics of the problem (Mitchell, 1998). According to Michaelewicz (1996), the real encoding representation is more stable, faster and makes the development of special tools to lead with non-trivial restrictions easier.

**7.3 TUNING OF SVMs’ PARAMETERS**

Like most ML techniques, SVMs have parameters to be tuned, which influence their performance. They are the value of the regularization constant C and the Kernel type, with its respective parameters. The use of a decomposition approach in multiclass problems increases the number of parameter values to be estimated, since each binary classifier confronts a different classification problem and may have distinct ideal parameter values. Three approaches can be followed to define the value of the parameters:

1. Use default values. Different SVM simulation tools may define, for each kernel, different default values for the parameters;
2. Define the values manually by trial and error and
3. Adjust the values through an optimization technique, such as GAs.

This chapter will investigate the influence of the parameter adjustment performed by GAs on the classification performance obtained by SVMs when they are used in multiclass decompositions. GAs have been frequently used together with other Machine Learning techniques. They have been successfully applied to tune Neural Networks (NNs) parameters, to train NNs, to extract rules from NNs, to select a subset of features, to decompose multiclass problems into binary sub-problems, among others.
7.3.1 Related work

A few works in the literature employed GAs to tune the parameters of SVMs in multiclass decompositions. Xu and Chan (2003a) used GAs to optimize the value of the standard-deviation parameter of the Gaussian Kernel for binary SVMs in the 1AA decomposition. The value of C was kept constant. In their experiments with two datasets, first a rough selection of the parameter values was performed. Next, the GA refined the values of $\sigma$ in the Gaussian Kernel for some fixed C values. The SVMs tuned by the GA obtained results slightly better than those reported in Hsu and Lin (2002), in which a trial-and-error approach was followed. Nevertheless, the results were close and no statistical comparison was performed.

Xu and Chan (2003b) proposed a method where the regularization values were optimized, while the parameters of a Gaussian Kernel were kept constant. In a wheel bearing fault detection dataset with three classes, the GA was able to produce SVMs with accuracy results better than those of a non-GA method, although again no statistical comparisons were carried out. In both Xu and Chan (2003a, 2003b) a binary encoding was adopted in the GA representation of the parameters’ values.

Liepert (2003) also investigated the use of binary-encoding GAs to tune the parameter values for SVMs in the application of topological fields chunking for the German language, a multiclass classification problem. The author observed that varying the $\sigma$ values of the Gaussian Kernel for each binary SVM in AAA decomposition did not seem advantageous and that the use of common values for all binary SVMs was sufficient to obtain good accuracy results. In Souza and Carvalho (2004), GAs were employed to select features from multiclass gene expression datasets. The values of the regularization constants for the SVMs in the 1AA decomposition strategy
were also determined by GAs. A binary encoding was adopted. A similar work was presented in Frohlich et al., (2003) but using the AAA decomposition strategy. In both works, the focus was not on the parameter selection process, but on the selection of relevant feature sets from gene expression datasets.

In previous experiments, the authors of the present work verified the potential of using GAs to search for SVMs’ parameter values in multiclass decompositions (Lorena and de Carvalho, 2006). Nevertheless, the focus of that work was not only on the parameter tuning process, but on how GAs could be used to determine multiclass decompositions. Although the parameter values are continuous, a binary encoding was adopted. Besides GAs, other optimization techniques such as Simulated Annealing (SA), Particle Swarm Optimization (PSO), and gradient search were also employed in the SVM parameter search process.

This chapter investigates the use of GAs with a real valued encoding to jointly search the regularization constant and Kernel parameters in multiclass decompositions. Only the Gaussian Kernel was considered, since it is general and shows less numerical difficulties than other common Kernel types, as the polynomial and the sigmoid Kernels. The proposed GA is general and can be employed to find the parameter values for SVMs in the one-against-all decompositions.

7.4 PROPOSED ALGORITHM

This section describes how the GA proposed in this work was implemented. After showing how the individuals are represented, it discusses how they are evaluated and presents the main steps of the proposed algorithm.
7.4.1 Real-valued Genetic Algorithm (RGA)

Recently, Genetic Algorithms (GAs) have been widely and successfully applied to various optimization problems. GAs are well suited to the concurrent manipulating of models with varying resolutions and structures since they can search non-linear solution spaces without requiring gradient information or a priori knowledge about model characteristics. The problem existing in the binary coding lies in the fact that a long string always occupies the computer memory even though only a few bits are actually involved in the crossover and mutation operations. This is particularly the case when a lot of parameters are needed to be adjusted in the same problem and a higher precision is required for the final result. To overcome the inefficient occupation of the computer memory, the underlying real-valued crossover and mutation algorithms are employed. In contrast to the Binary Genetic Algorithm (BGA), the Real-valued Genetic Algorithm (RGA) uses a real value as a parameter of the chromosome in populations without performing coding and encoding process before calculates the fitness values of individuals. Namely, RGA is more straightforward, faster and more efficient than BGA. Since this study is concerned with finding optimal values of SVM parameters whose precise values are unknown, the aforementioned properties of RGA are highly advantageous.

7.4.2 Parameter Optimization

To design an effective SVM model, values of parameters in SVM have to be chosen carefully in advance (Lin, 2001; Duan et al., 2003; Min & Lee, 2005). These parameters include the following: (1) regularization parameter C, which determines the tradeoff cost between minimizing the training error and minimizing the complexity of the model; (2) parameter sigma (σ or d) of the kernel function which defines the non-linear mapping
from the input space to some high-dimensional feature space. This investigation only considers only the Gaussian kernel, the variance of whose function is sigma squared $\sigma^2$; (3) a kernel function used in SVM, which constructs a non-linear decision hypersurface in an input space. Campbell et al., (1999) proposed the Kernel-Adatron Algorithm which can automatically select models without testing on a validation data. Unfortunately, this algorithm is ineffective if the data have a flat ellipsoid distribution (Campbell, 2002). Therefore, one possible way to solve the problem is to consider the distribution of the data. Interestingly, various specific functions in SVM, after the learning stage, can create the decision hypersurfaces of the same type (Kecman, 2001). To solve the problem, Lin (2001) provided a systematic method for selecting SVM parameters. His systematic design for selecting parameters of support vector regression was adopted the concept of the sampling theory into Gaussian Filter. Min and Lee (2005) proposed a grid search technique using 5-fold cross validation to find out the optimal parameters values of kernel function of SVM. In contrast to abovementioned methods of parameter optimization on SVM, this chapter develops a new method, named GA-SVM, for optimizing the two SVM parameters ($C$ and $\sigma^2$) simultaneously with an improved crossover operator. In this section, we describe the design of our proposed model, a genetic-based SVM model, for shoeprint classification.

7.4.3 The Proposed GA Approach

In the proposed GA-SVM model, the SVM parameters are dynamically optimized by implementing the RGA evolutionary process and the SVM model then performs the prediction task using these optimal values. Namely, the RGA tries to search the optimal values to enable SVM to fit various datasets. The optimal values of SVM’s parameters are searching by GAs with a randomly generated initial populations consisting of
chromosomes. The values of the two parameters, $C$ and $\sigma^2$ are directly coded in the chromosomes with real-valued data. The proposed model can implement either the roulette-wheel method or the tournament method for selecting chromosomes. An improved crossover method and boundary mutation method were used to modify the chromosome. The single best chromosome in each generation is survives to the succeeding generation. The proposed model is able to handle huge data sets and easily be combined with the RGA.

7.4.3.1 Chromosome Representations

Unlike the traditional BGA, the RGA used to solve optimization problems, directly codes all of the corresponding parameters or variables in a chromosome. Hence, the representation of the chromosome is straightforward in the RGA. The two parameters, $C$ and $\sigma$, of SVM were directly coded to form the chromosome in the proposed method. The chromosome $X$ is represented as $X = \{p_1, p_2\}$, where $p_1$ and $p_2$ denote the regularization parameter $C$ and sigma $\sigma$ (the parameter of the kernel function), respectively.

7.4.3.2 The Fitness Function

A fitness function, assessing the performance of each chromosome, must be designed before starts to search optimal values of SVM parameters. Several measurement indicators have been developed and applied to evaluate the predictive accuracy of models.

7.4.3.3 Genetic operators

The real-valued genetic algorithm uses selection, crossover, and mutation operators to generate the offspring of the existing population.
**Selection** – The proposed GA-SVM model incorporates two well-known selection methods – the roulette wheel method and the tournament method. The tournament selection method is adopted here to decide whether a chromosome can survive to the next generation. The chromosomes that survive to the next generation are placed in a matting pool for crossover and mutation operations.

**Crossover** – Once a pair of chromosomes has been selected for crossover, one or more randomly selected positions are assigned to the to-be-crossed chromosomes. The newly crossed chromosomes are then combined with the rest of the chromosomes to generate a new population. However, overloading problem frequently occurs when the RGA is used to optimize values. This chapter uses the method proposed by Adewuya (1996) to prevent overload of post-crossover when genetic algorithm with real-valued chromosomes are applied.

\[
X_1^{\text{old}} = \{x_{11}, x_{12}, \ldots, x_{1n}\}, \quad X_2^{\text{old}} = \{x_{21}, x_{22}, \ldots, x_{2n}\}
\]

Move closer:

\[
X_1^{\text{new}} = X_1^{\text{old}} + \sigma (X_1^{\text{old}} - X_2^{\text{old}})
\]

\[
X_2^{\text{new}} = X_2^{\text{old}} + \sigma (X_1^{\text{old}} - X_2^{\text{old}})
\]

Move away:

\[
X_1^{\text{new}} = X_1^{\text{old}} + \sigma (X_2^{\text{old}} - X_1^{\text{old}})
\]

\[
X_2^{\text{new}} = X_2^{\text{old}} + \sigma (X_2^{\text{old}} - X_1^{\text{old}})
\]

\(X_1^{\text{old}}\) and \(X_2^{\text{old}}\) represent the pair of populations before crossover operation; \(X_1^{\text{new}}\) and \(X_2^{\text{new}}\) represent the pair of new populations after crossover operation. In addition, \(\sigma\) is a random micro number controls the variance of each crossover operations. With this crossover operator, the parent chromosomes are chosen for exchanging the information to produce new
children. The parents’ selection is done in random so far, but if the matting is performed with appropriate parents could lead the GA to earlier convergence. For example, if the matting is performed with two chromosomes with highest fitness, it may not produce bigger change in the optimal solution construction. If the crossover matting is performed between stronger and weaker chromosomes, means that the chromosome with higher fitness with the chromosome has lower fitness could produce an effective chromosome to explore the search space wider. This concept motivates the idea for generating a new crossover operator as discussed below.

Initially the chromosomes are sorted in decreasing order based on their fitness values and this sorted list is divided into two halves, ascertain that one half has the chromosome with higher fitness the other with the lower fitness values. Now the matting is performed between one parent from each half. The experimental results show that this kind of matting enhances the performance of GA.

**Mutation** – The mutation operation follows the crossover operation and determines whether a chromosome should be mutated in the next generation. In this study, uniform mutation method is applied and designed in the presented model. Consequently, researchers can select the method of mutation in GA-SVM best suited to their problems of interest.

\[
X^{old} = \{x_1, x_2, ..., x_n\}
\]

\[
x^new_k = LB_k + r (UB_k - LB_k)
\]

\[
X^{new} = \{x_1, x_2, ..., x^new_k, ..., x_n\}
\]

where \(n\) denotes the number of parameters; \(r\) represents a random number in the range \((0, 1)\), and \(k\) is the position of the mutation. \(LB\) and \(UB\) are the low and upper bounds on the parameters, respectively. \(LB_k\) and \(UB_k\) denote the low and upper bound at location \(k\). \(X^{old}\) represents the population before
mutation operation; \( X_{\text{new}} \) represents the new population following mutation operation.

### 7.5 EXPERIMENTAL RESULTS AND DISCUSSION

The same set of shoeprint image database as discussed in the previous chapters is used here for the experiment here. Initially the features are extracted from the shoeprint images using the proposed feature extraction approaches CPCA, SR-CWT (Subband Relevance based CWT), SR-DWT, and SR-RDWT. Then the features are classified with the standard SVM classifier as described in section 5.3, and the proposed GA-SVM classifier. The parameters were varied in the following intervals: \( C \in [1, 1000] \) and \( \sigma \in [0.001, 1] \). The following Table 7.1 presents the classification accuracy from the SVM and GA-SVM classifiers for both full and partial shoeprint images with the optimal \( C \) and \( \sigma \) values. The same results are depicted in the following Figures 7.2 and 7.3. From the results it is clearly shown that the proposed GA-SVM model outperforms the standard SVM.

**Table 7.1 Classification Performance for SVM vs. GA-SVM classifiers**

<table>
<thead>
<tr>
<th>Feature Extraction Methods</th>
<th>Classification Accuracy</th>
<th>Full Shoeprint</th>
<th>Partial Shoeprint</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SVM</td>
<td>GA-SVM</td>
<td>C</td>
</tr>
<tr>
<td>SR-RDWT</td>
<td>94.82</td>
<td>95.71</td>
<td>262</td>
</tr>
<tr>
<td>SR-DWT</td>
<td>93.16</td>
<td>95.05</td>
<td>14</td>
</tr>
<tr>
<td>SR-CWT</td>
<td>91.57</td>
<td>92.67</td>
<td>340</td>
</tr>
<tr>
<td>CPCA</td>
<td>89.88</td>
<td>91.19</td>
<td>374</td>
</tr>
</tbody>
</table>
Figure 7.2  Full Shoeprint Classification Performances with SVM vs. GA-SVM

Figure 7.3  Partial Shoeprint Classification Performances with SVM vs. GA-SVM
7.6 SUMMARY

This chapter investigated the development of an automatic system for the selection of the parameter values of the SVMs contained in shoeprint image classification. The empirical search of these values is in general costly. As the number of classes increases, the number of possible combinations increases considerably. Taking advantage of GAs as a robust search and optimization technique, a hybrid system for this tuning process was developed. The LibSVM was verified that the GA with improved crossover operator showed benefits in the choice of parameters mainly for a dataset in which the default values showed a low performance. In the only statistically significant results, the use of GAs to select the parameter values for the SVMs classification better than the use of the default value. The results obtained here were not compared with results from related work because different experimental setup and partitions of the dataset were used in the experiments. One of the main contributions of this work was in the study of how the genetic optimization of the parameters of the binary SVMs in a multiclass decomposition affects the classification performance. The GAs may be easily extended to binary problems. They can also be adapted to optimize other machine learning techniques, which present parameters to be tuned.