GENETIC ALGORITHMS WITH DIVERSITY MEASURES FOR MULTIPLE CLASSIFIER SYSTEMS

Leidys Cabrera Hernández [*], Alejandro Morales Hernández[*], Gladys M. Casas Cardoso[*], Yailen Martínez Jiménez [*]

[*]Departamento de Computación, Facultad Matemática, Física y Computación, Universidad Central "Marta Abreu" de Las Villas, Cuba.

e-mail: leidysc@uclv.edu.cu, alejandrom@uclv.edu.cu, gcasas@uclv.edu.cu, yailenm@uclv.edu.cu

Abstract

The combination of classifiers is an active research area of the machine learning and pattern recognition communities. Many theoretical and empirical studies have been published demonstrating the advantages of the paradigm of combination of classifiers over the individual classifiers. When combining classifiers it is important to guarantee the diversity among them. Some statistical measures can be used to estimate how diverse the ensembles of classifiers are.

Genetic algorithms play a significant role as search technique for handling complex spaces in many fields. They are based on the underlying genetic process in biological organisms and on the natural evolution principles of populations. These algorithms process a population of chromosomes, which represent search space solutions, with three operations: selection, crossover and mutation. Under its initial formulation, the search space of solutions is coded using the binary alphabet.

In this work some diversity measures are presented and a variant of a Genetic Algorithm is implemented in order to obtain, from all the possible combinations of a large number of base classifiers, a combination that ensures greater diversity among the chosen classifiers and multiple classifier system accuracy.

Keywords: Genetic Algorithms, Measures of Diversity, Classifier, Multiple Classifiers

Introduction

Genetic algorithms (GAs) are emerging as tools for solving complex search and optimization problems, as a result of the analysis of adaptive systems in nature. The search and optimization methods have been studied since the early years of computing, ranging from methods based on calculation to enumerative methods, up to random search algorithms. All these methods are analyzed and criticized in terms of robustness, but this does not mean they are not useful; they can be used as a complement to more robust schemes in order to create hybrid approaches. The term Genetic Algorithm is used because they simulate the processes of the Darwinian evolution through the use of genetic operators which operate on a population of individuals that "evolves "from one generation to another. The development of the whole theory concerning the subject has not only served to achieve an efficient search method but allowed to abstract and rigorously explain the adaptive process in natural systems. Moreover, this has made possible to design artificial systems that include these natural mechanisms [9].

On the other hand, the issue of classification has been widely discussed and continues to develop. Choosing the best classifier depends largely on the problem to be solved, for each case the selected classifier establishes the best decision boundary to separate the classes. In the search for better methods of classification there is a trend to combine several classifiers for the solution of same problem. This is the idea on which the so-calls multiple classifier systems algorithms are based on. They use several classifiers and combine their outputs [1] with the aim of achieving a better result.

Dietteric [2] suggests three reasons why a multiple classifier system can be better than a single classifier. The first is statistical, because if each classifier has a hypothesis, the idea of combining these hypotheses, results in a hypothesis that cannot be the best, but at least avoids selecting the worst. The second justification is computational, since some algorithms execute queries that can lead to different local optima: each classifier starts the search from a different point and ends it closest to the optimum. There is an expectation that some combination pathway can lead to a classifier with a better approximation. The last justification is figurative because it is possible that the hypothesis space contains the hypotheses considered not optimal, but the approximation of several decision boundaries can result in a new space outside the initial hypothesis that is closest to the optimum.

There are several ways in which multiple classifier systems may be built. There are a number of algorithms developed, some for general problems as bagging and boosting and others for specific problems, but they all have the choice of the fundamental parts of the base classifiers and the choice of how to combine the outputs [3].

The selection of basic classifiers is the first step in building multiple classifier systems. Among the variants to combine the base classifiers are the ones used for bagging and boosting, which use the same classification model trained with different case subsets. The first selects random case subsets and the second selects the subsets iteratively based on the result of the previous iteration. Another variant is used by Stacking, which is used in different classification models trained with the same initial base.

It could be said that these two paradigms are the most general and used in building multiple classifier systems. Although which of the two variants is the best has not been demonstrated. Individual multiple classifier systems, like simple classifiers, are not intrinsically better than others, but have to be selected on the basis of which gives better results for each specific problem [4].

The diversity between the base classifiers is very important since this will largely depend on the final result of multiple classifier systems. Each classifier achieved a percentage of cases classified correct. The more diverse the results of the classifiers are the higher probability to cover a higher percentage of cases, by combining the outputs thereof. Some multiple classifier systems ensure diversity using different sets of training bases, but this only works for classifiers that are sensitive to changes, such as decision trees. Others use different sets of features and thus also vary the training base. Others use different base classifiers. In the latter case it is difficult to know when a great diversity is ensured, making it necessary to use some statistical measures that help to determine how diverse they are. Some measures of diversity are described by Kuncheva in [5]. They can be classified as: measures in the form of pairs (pairwise) and measures group (nonpair-wise). In this paper we will work with the pairwise measures.

Section 1: Measures forms of pairs (pairwise)

These measures are calculated for pairs of classifiers. Its outputs are binary (0, 1) indicating whether the instance was classified correctly or not. Table 1 shows the results of two classifiers (C_i, C_j) for a given instance, depending on whether or not they were correctly classified. If we consider all instances between the pair of classifiers (Ci, Cj) the following results are obtained, see Table 2:

	C_j correct (1)	C_j incorrect (0)
C_i correct (1)	a	b
C_i incorrect (0)	c	d
a + b + c + d = 1		

Table 1: Binary matrix for one instance

	C_j correct (1)	C_j incorrect (0)
C_i correct (1)	A	B
C_i incorrect (0)	C	D
A +B + C + D = N		

Table 2: Binary matrix for N instances

N is the total number of cases. A set of L classifiers produces L(L-1)/2 pairs of values. To obtain a single result these values should be averaged.

Correlation coefficient ρ

The coefficient of correlation [4], is one of the measures for pairs of classifiers, it is calculated as:

$$\rho_{ci,cj} = \frac{\mathrm{A}\times\mathrm{D}-\mathrm{B}\times\mathrm{C}}{\sqrt{(\mathrm{A}+\mathrm{B})\times(\mathrm{C}+\mathrm{D})\times(\mathrm{A}+\mathrm{C})\times(\mathrm{B}+\mathrm{D})}} \quad (1)$$

A better diversity is obtained for smaller values of ρ. The values of ρ will be in the interval [-1, 1].

Q Statistics

The Q statistic is one of the measures for pairs of classifiers

$$\mathrm{Qci,cj} = \frac{\mathrm{A}\times\mathrm{D}-\mathrm{B}\times\mathrm{C}}{\mathrm{A}\times\mathrm{D}+\mathrm{B}\times\mathrm{C}} \quad (2)$$

It has been proved that ρ and Q have the same sign. Also, it can be demonstrated that $|\rho| \leq |Q|$ [5].

The Measure of Differences

The measure of differences was introduced by Skalak [6], it is the most intuitive measure between a pair of classifiers, and it is equal to the probability that the two classifiers disagree in their predictions. The diversity increases when the value of D increases.

$$D_{ci,cj} = \frac{\mathrm{B}+\mathrm{C}}{\mathrm{N}} \quad (3)$$

The Double-Fault Measure

Another measure to be analyzed is known as double fault measure, which was introduced by Giacinto and Roli [7] and considers the failure of two classifiers simultaneously. This measure is based on the concept that it is more important to know when simultaneous errors are committed, that when both have a correct classification. The diversity increases when the value of DF decreases.

$$D_{ci,cj} = \frac{\mathrm{D}}{\mathrm{N}} \quad (4)$$

Section 2: Genetic Algorithms

Genetic algorithms (GAs) are search methods based on the general purpose principles of natural genetics, they are search algorithms based on the mechanisms of natural selection and genetics. Genetic algorithms are an example of a method that exploits the random search "guided" that has gained popularity in recent years due to their applicability in a wide range of fields and few requirements imposed by the problem [8], [9].

The basic idea is to maintain a population of chromosomes, which represent candidate solutions to the concrete problem; this population evolves over time through a process of competition and controlled variation. Each chromosome in the population has an associated fitness to determine which chromosomes are used to form new ones in the competition process, which is called selection. The new ones are created using genetic operators such as crossover and mutation. GAs have had a great measure of success in search and optimization problems. The reason for a great part of their success is their ability to exploit the information accumulated about an initially unknown search space in order to bias subsequent searches into useful subspaces, i.e., their adaptation. This is their key feature, particularly in large, complex, and poorly understood search spaces, where classical search tools (enumerative, heuristic) are inappropriate, offering a valid approach to problems requiring efficient and effective search techniques.

To use the GAs is necessary to find a structure for representing the possible solutions. Thinking about this issue as the problem of searching in a state space, an instance of this structure represents a point or a state in the search space of all possible solutions. Thus, a data structure in the GA consist of one or more chromosomes (often one), which is commonly represented as a string of bits. Each chromosome is a concatenation of a number of subcomponents called genes. The position of a gene in the chromosome is known as locus alleles. In the string of bits, a gene is a bit, a locus is the position in the string and an allele is its value (0 or 1 if it is a bit).

Fixed length and binary coded strings for the representation of the solutions have dominated the GA research since there are theoretical results that show them to be the most appropriate ones [10], and as they are easy to implement.

In order to optimize the GA structure, a measurement of the quality of each frame in the search space is needed. The fitness function is responsible for this task. In a maximization function, the objective function often acts as the fitness function. The GA usually works with maximization functions, for minimization problems the objective function values can be negated and transferred in order to take positive values, thus producing adaptability [11, 12].

The simple mechanism of GA is as follows:

- ✓ The simple GA randomly generates a population of n structures (strings, chromosomes or individuals)
- ✓ The population operators act transforming the population. Once the actions of the three operators are completed, we can say that a generational cycle has expired.
- ✓ Then the previous step is repeated while the stopping criterion of the AG is not guaranteed.

The selection operator makes the selection of strings according to their adaptability for the next steps. The crossover operator performs the recombination of genetic material from two parent’s strings. The mutation operator, like the natural mutation operator, performs the mutation of a gene within a chromosome.

A probability is associated to each of these operators. The mode of operation of a GA can be summarized as shown in Figure 1. The AG runs for a fixed number of generations or until some stopping criterion is satisfied.

Most experts on this subject agree that the GA can solve the difficulties represented in real life problems that sometimes are insoluble by other methods. The focus of research in GA is robustness: the balance between effectiveness and efficiency needed to survive in many different environments.

Section 3: Genetic Algorithm to detect good ensembles of classifiers

A genetic algorithm is implemented using diversity measures for combining diverse classifiers and provides the best possible accuracy.

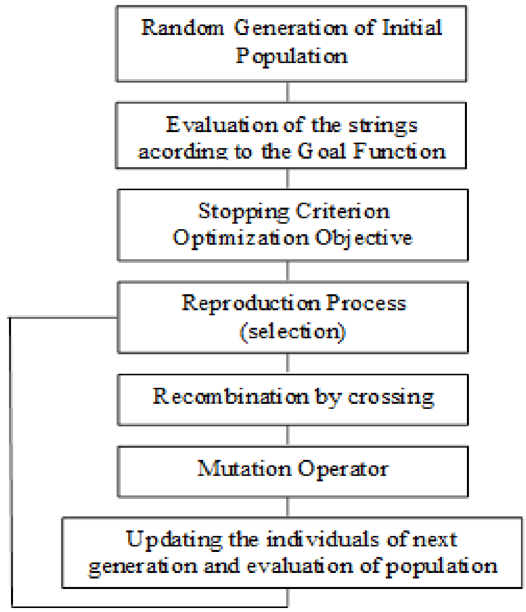

Figure 1: Functional diagram of a genetic algorithm

The settings of all the parameters of the genetic algorithm and the definition of the objective function are:

Description of the Goal Function

The target is Max f(X), where f is the goal function. The value of f is the result of the evaluation of the multiple classifier system plus the result of the evaluation of the selected diversity measures. X is the chromosome.

Chromosome configuration

Gen: binary variable. It takes the value 1 if the classifier belongs to the combination and it takes the value 0 otherwise. Chromosome: Arrangement of genes representing the set of all base classifiers that will be used as a basis in multiple classifier systems.

Configuration of the population

The population size can be calculated as: $2^{\frac{Sc}{2}}$, where Sc denotes the size of the chromosome. The initial population was randomly generated.

Mutation operator

The traditional mutation operator was implemented: randomly choose a chromosome, randomly choose a gene to mutate and change its status: 0 for 1 or 1 to 0. If the resulting chromosome was already analyzed, choose another mutation point and repeat. The probability of mutation was 0.2.

Crossover operator

The traditional crossover operator was implemented: randomly choose two different chromosomes: A and B. Randomly choose a point of crossing and attach the first part of the chromosome A to the second part of B. Do the same with the other two parts. If one of the resulting chromosomes (or both) is (are) already analyzed, apply the mutation operator. The probability of crossover was 0.75.

Selection operator

The roulette method was implemented. A probability value was assigned to each chromosome depending on their goal function. Greater chromosomes are more likely to be selected.

Input parameters

Generations: 100

The diversity measures used were the D and DF because they are simpler than the others. We also tested the sum of those two.

The calculations of classifiers were made using the platform of intelligent learning WEKA (Waikato Environment for Knowledge Analysis), with the "Percentage Split" option and default parameters of the software. In this option, a portion of the data is used for training and the other one for validation. Table 3 shows the used classifiers together with their accuracy.

Section 4. Application

The Hepatitis data base, from the UCIML Repository (UCI Repository of Machine Learning Databases) was used as a study case to exemplify our contribution. This base is binary, and it has 13 nominal features, 6 discrete features and 155 cases in total. All the calculations were done using Weka.

Used Classifiers

Classifier	Accuracy
Naivesbayes	0.83
Multilayer Perceptron	0.81
Lazy.IBK	0.79
Trees.J48	0.79
Functions.Logistic	0.84
Random Tree	0.83
Trees.LMT	0.84
Functions.SMO	0.88
Lazy.KStar	0.90
Functions.SGD	0.88

Table 3: Base classifiers accuracies

Multiple classifier system

By combining classifiers we are aiming at a more accurate classification decision at the expense of increased complexity. Numerous algorithms have been proposed to combine base classifiers in order to construct a new multiclassifier. The average combination rule was used in this research.

The results obtained are shown in Table 4: The resulting chromosome provides a combination of classifiers which improves to 94% the multiple classifier system accuracy.

Hepatitis data base	Measure value	Goal Function	Chromosome	Multiple Classifier System
D	0.17	1.11	0100010101	0.94
DF	0.94	1.88	0100010101	0.94
(D + DF)/2	0.56	1.50	0100010101	0.94

Table 4: Results of the genetic algorithm

The resulting chromosome corresponds to the combination of the following classifiers: MultilayerPerceptron(); weka.classifiers.trees.RandomTree(); weka.classifiers.functions.SMO() y weka.classifiers.functions.SGD();

Conclusions

This paper shows a novel technique using genetic algorithms to find a good subset of diverse classifiers. The objective function involves the classification accuracy of the multiclassifier and the results of the diversity of the base classifiers.

A case of study of Hepatitis data was used to exemplify the contribution. Ten base classifiers were applied and their individual accuracies were not higher than 90%. By using the proposed genetic algorithm with diversity measures we obtain a multiclassifier with a better accuracy: 94%.

References

[1] Polikar, R. Ensemble based systems in decision making. IEEE Circuits and Systems Magazine.6 (3) ed. 2006.

[2] Dietterich, T. G. Ensemble methods in machine learning. Multiple classifier systems. Berlin: Springer-Verlag Berlin.2000.

[3] Bonet, I. Modelo para la clasificación de secuencias, en problemas de la bioinformática, usando técnicas de inteligencia artificial. Tesis Doctoral, UCLV. 2008.

[4] Kuncheva, L. I. Combining Pattern Classifiers, Methods and Algorithms, New York, NY, Wiley Interscience.2004.

[5] Kuncheva, L. I. & Whitaker, C. J. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. Machine Learning, 51, 181-207.2003.

[6] Skalak, D. B. The Sources of Increased Accuracy for Two Proposed Boosting Algorithms, 1996.

[7] Giacinto, G. & Roli, F., Design of effective neural network ensembles for image classification purposes. Image and Vision Computing, 2001. 19(9-10): p. 699-707.

[8] Holland, J.H. (1975).Adaptation inNatural andArtificial Systems. TheUniversity of Michigan Press.

[9] Goldberg, D.E. (1989a). Genetic Algorithms in Search, Optimization, and Machine Learning. AddisonWesley, New York.

[10] Goldberg D.E. (1991a). RealCoded Genetic Algorithms, Virtual Alphabets, and Blocking. Complex Systems 5, 139–167.

[11] Antonisse, J. (1989). A new interpretation of schema notation that overturns the binary encoding constraint. Proc. of the Third Int. Conf. on Genetic Algorithms, J. David Schaffer (Ed.) (Morgan Kaufmann Publishers, San Mateo), 86–91.

[12] Radcliffe N.J. (1992). NonLinear Genetic Representations. Parallel Problem Solving from Nature 2, R.M¨anner and B. Manderick (Ed.) (Elsevier Science Publichers, Amsterdam), 259–268.