Data Mining II, C.A. Brebbia & N.F.F. Ebecken (Editors)
© 2000 WIT Press, www.witpress.com, ISBN 1-85312-821-X

Designing optimized pattern recognition systems by learning Voronoi vectors using genetic algorithms

Cláudio M. N. A. Pereira [1,3] and Roberto Schirru [2]
[1] *Instituto de Engenharia Nuclear, CREA,*
Comissão Nacional de Energia Nuclear, Brazil.
[2] *Programa de Engenharia Nuclear,*
Universidade Federal do Rio de Janeiro, Brazil.
[3] *Computer Science Department, Universidade Iguaçu, Brazil.*

Abstract

In this work is described a methodology for developing optimized pattern recognition systems by means of genetic machine learning. The idea is to redefine the set of classes that must be learned by the classification system, mapping them into another set which the Voronoi vectors can successfully classify a sample into one of the original classes. The main objective of this approach is to find the minimum number of classification rules (by minimizing the number of classes in the new set) that maximizes the number of correct classifications. To accomplish that, a genetic algorithm was design. In the genetic model, a classical fixed length genotype encodes the vectors positions in such a way that a variable number of vectors is allowed. The objective function is designed so that it rewards the correctness in the classification and penalize high number of vectors. In order to verify the accuracy and robustness of the genetic algorithm in finding the optimum Voronoi vectors set, simple benchmarks, which the expected optima are well known, were used. The success obtained in the tests with benchmarks has motivated the application of the methodology in the design of a nuclear power plant transients diagnosis system. The obtained results reveal the ability of the genetic algorithm in learning optimized classification rules that lead to efficient diagnosis, dealing with the compromise between number of rules and certainty in the classification.

Data Mining II, C.A. Brebbia & N.F.F. Ebecken (Editors)
© 2000 WIT Press, www.witpress.com, ISBN 1-85312-821-X

1 Introduction

Pattern classification is a very important feature of the human cognition. Observing similarities among patterns may involve a lot of human abilities, and often, the procedures realized automatically by the human brain can not be exactly described. Hence, making computers to be able to pattern classification is a hard task.

Some areas of the Artificial Intelligence (AI), specially artificial Neural Networks (NN) ([1], [2]) have been widely applied in order to face such kind of problem, and the specific topic called Vector Quantization (VQ) is of special interest for this work. Such technique consists in dividing an input space into a number of distinct regions for which a reproduction vector - Voronoi Vector (VV) - is defined. Finding the optimum VV set is a difficult optimization problem and several techniques could be used to solve it. Learning Vector Quantization (LVQ) is one of these techniques, proposed by Kohonen [3]. It is a supervised learning technique that moves the Voronoi vectors in order to optimize the classification. In order to provide a global scope in the optimization, Ng [4] proposed an approach to VQ using Genetic Algorithms (GA) and Generalized Lloyd Algorithm [5].

In this work, it is proposed the use of GA in order to optimize the classification system by transforming the original set of classes into a new one in which the classification by the VV's can be mapped back to a correct classification in the original set of classes. The objective is to find the minimum number of classes and theirs Voronoi vectors.

For example, consider samples of Class 0 and Class 1 in Figure 1, represented by points in plane.

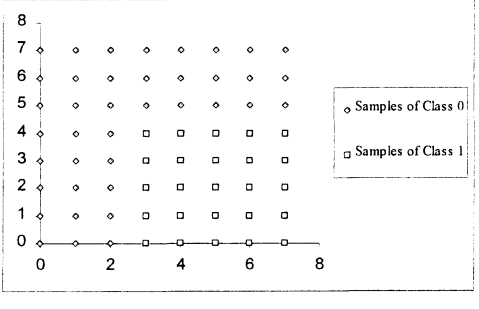


Figure 1: Example of two classes represented by points in the plane

Note that the use of one vector per class may lead to misclassifications. However, if more than one vector could be used to represent a single class, the problem of misclassification could be solve, as can be seen in Figure 2.

Data Mining II, C.A. Brebbia & N.F.F. Ebecken (Editors)
© 2000 WIT Press, www.witpress.com, ISBN 1-85312-821-X

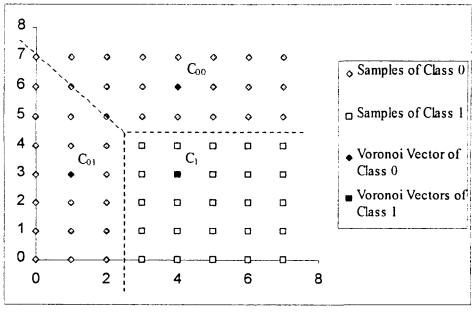


Figure 2: Representation of classes by more than one vector per class.

Figure 2 contains an optimum representation of the classes 0 and 1 by three vectors (in bold). It is clear that, if Class 0 was divided into two other classes, Class 00 and Class 01, then the vectors C_{00}, C_{01} and C_1 would be the VV's of the new set of classes. In this case, the final classification may have the rules that map the transformed set of classes into the original one, such as:

IF ((Class' = C_{00}) OR (Class' = C_{01})) THEN Class = C_0;
ELSE IF ((Class' = C_1) THEN Class = C_1;

Where Class' is the classification in the new set, and Class is the classification in the original one

Such transformation is, therefore, a hard optimization problem, once the number of classes in the new set, as well as the position of the vectors are unknown. In order to deal with this complexity, and the poor previous knowledge about the search space, genetic algorithms [6] were used as the optimization technique.

Genetic algorithms, created by Holland [7], are powerful search techniques based on the metaphor of species evolution via natural selection inspired in Darwin's theory [8]. In the GA, populations of candidates to optimum solution are adapted through a simulated artificial evolution. The candidates (many times called individuals), represented by fixed length bit-strings, receive a fitness that dictate its adaptability to the environment (the objective of the optimization problem). Selected according to their fitness, the individuals may undergo crossover and mutation.

GA`s have been used to optimize classification systems by optimizing time partitions of the classes represented by time series [9]. This approach intend to be more flexible, once the VV can be placed anywhere (respecting the discretization imposed by the GA) in the search space.

Data Mining II, C.A. Brebbia & N.F.F. Ebecken (Editors)
© 2000 WIT Press, www.witpress.com, ISBN 1-85312-821-X

2 Genetic modeling description

2.1 Genotype structure

In the present application, individuals are fixed length bit-string representing the number of classes in the new set as well as the corresponding VV, as can be seen in Figure 3.

	Vector 1			Vector 2			Vector 3			...	Vector N		
N	v_1	...	v_n	v_1	...	v_n	v_1	...	v_n	...	v_1	...	v_n

Figure 3: Genotype representation

The number of genes may change according to the first gene, *N*, value. In order to accomplish this, respecting the classical paradigm of the GA where the genotype must have its length fixed, the length of each gene may change, affecting, of course, the precision of the representation.

Consider a hypothetical 2-dimensional problem where the number of classes (vectors) may vary between 1 and 4, and each variable, between 0 and 15. The minimum precision in the representation of the vectors occur when the number of classes is maximum (4 in this case). Suppose that 4 bits is enough precise. Then, the length, L, of the genotype is given by:

$$L = \left\lceil \begin{matrix} \text{Minimum number of bits} \\ \text{for representation of the} \\ \text{maximum number of classes} \end{matrix} \right\rceil + \left\lceil \begin{matrix} \text{Number} \\ \text{of} \\ \text{Variables} \end{matrix} \right\rceil \cdot \left\lceil \begin{matrix} \text{Minimum number} \\ \text{of bits per} \\ \text{variable} \end{matrix} \right\rceil \cdot \left\lceil \begin{matrix} \text{Maximum} \\ \text{number of} \\ \text{classes} \end{matrix} \right\rceil \quad (1)$$

So, for this problem, $L = 2 + 2 \,.\, 4 \,.\, 4 = 34$.

Figure 4 exemplifies two possible genotypes for the above example.

	Vector 1		Vector 2		Vector 3		Vector 4	
N	v_1	v_2	v_1	v_2	v_1	v_2	v_1	v_2
11	0010	0111	1001	0000	0001	0011	1111	0100
4	2.0	7.0	9.0	0.0	1.0	3.0	15.0	4.0

	Vector 1		Vector 2	
N	v_1	v_2	v_1	v_2
01	00000001	11111111	00000100	00000000
2	0.5	15.0	2.0	0.0

Figure 4: Two genotypes with different number of classes and theirs respective phenotype.

Data Mining II, C.A. Brebbia & N.F.F. Ebecken (Editors)
© 2000 WIT Press, www.witpress.com, ISBN 1-85312-821-X

2.2 Fitness function

The fitness of each individual may increase with the number of correct classification and may be penalized if the number of vectors increase. The compromise between correct classification and number of vectors are imposed by two importance coefficients, as follow:

$$F = k_1.C - k_2.V \tag{2}$$

Where F is the fitness function, C is the number of correct classifications (percent), V the number of vectors, k_1 and k_2 the importance coefficients for C and V respectively.

The choice of k_1 and k_2 must be made according to the restrictions and objectives of the problem. In the next section, it is better explained.

3 Method application and results

3.1 Tests with benchmarks

In order to test the method, the genetic algorithm, which the implementation was based on the Genetic Search Implementation System - Genesis [10], was applied to 4 hypothetical cases (see Figure 5) which the minimum number of vectors that lead to 100% of correct classification are known. These tests was created to verify the robustness of the GA in finding the global optimum.

The two first problems presents samples of different classes spread in the plane. Note that the use of one vector per class is a bad solution and should lead to misclassifications. In Problem 1, the class 1 must be subdivided into two classes in order to get 100% correct classification, while in Problem 2, the new set of classes is exactly the original one, so 4 vectors must be found. Problems 3 and 4 were created in order to observe the method application to time series classification.

The number of classes was encoded into a bit-string with 4 bits, and decoded into values between the number of classes in the original set of classes, and this number plus 15. For example, in Problem 1, the number of classes may vary between 2 and 17. In Problem 3, between 3 and 18, and so one. The minimum length for each gene was 4.

The importance coefficients k_1 (applied to the number of correct classifications) and k_2 (applied to the number of vectors) were chosen so that if 100% correct classification is possible with the maximum number of classes, then it must win the 99% correct with minimum number of classes. So:

$$k_1.100 - k_2.MaxClasses \geq k_1.\ 99 - k_2.MinClasses$$

Where MaxClasses and MinClasses are the maximum and the minimum number of classes, respectively.

If $k_1 = 1$, then:

$$100\text{-}k_2.MaxClasses \geq 99 - k_2.MaxClasses \Rightarrow 1 \geq k_2.(MaxClasses\text{-}MinClasses)$$
$$\Rightarrow 1 \geq 15\ k_2 \Rightarrow k_2 \leq 0.066666$$

As the difference between two fitness values may be very small, the genetic selection was made using the ranking option, in order to distribute the surveillance probabilities according to the fitness classification, and not to the fitness value itself.

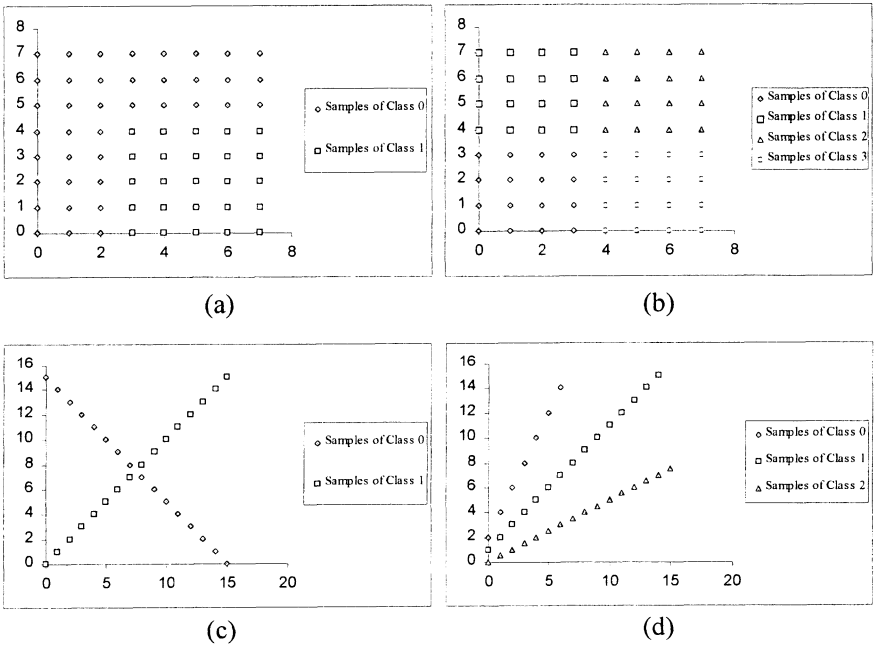


Figure 5: Examples cases. (a) Problem 1; (b) Problem 2; (c) Problem 3 (d) Problem 4.

The optimum number of Voronoi vectors found by the GA, as well as the corresponding percentage of correct classification are compared with the expected values in Table 1.

Note that optimum configurations were found in the 4 examples cases. For the Voronoi vectors set, however, many configurations that led to the global

Data Mining II, C.A. Brebbia & N.F.F. Ebecken (Editors)
© 2000 WIT Press, www.witpress.com, ISBN 1-85312-821-X

optimum were found. Figure 6 shows one Voronoi vectors configuration for each example.

Table 1: Results of the application of the method to the 4 examples.

	Number of Vectors		Correct classifications (%)	
	Expected	Found by GA	Expected	Found by GA
Problem 1	3	3	100	100
Problem 2	4	4	100	100
Problem 3	4	4	100	100
Problem 4	3	3	100	100

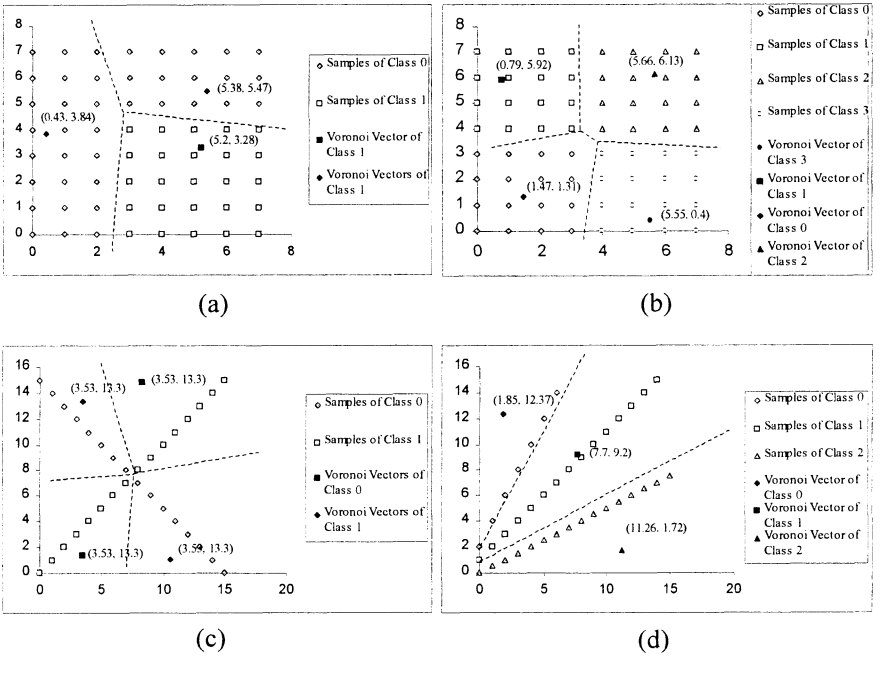


Figure 6: Voronoi vectors obtained by the application of the method to: (a) Problem 1; (b) Problem 2; (c) Problem 3; (d) Problem 4.

The results presented here were obtained using the following genetic parameters: population size of 50 individuals, crossover rate of 60%, mutation rate of 5%, ranking selection end elitism. Nevertheless, they reflect those results obtained by many experiments with different adjust of the genetic parameters and random seeds for the optimization.

Data Mining II, C.A. Brebbia & N.F.F. Ebecken (Editors)
© 2000 WIT Press, www.witpress.com, ISBN 1-85312-821-X

3.2 Application to nuclear transients classification

After testing the GA with the simple hypothetical problems, specially created to validate the method basis, it was applied to a more practical case. Now, three nuclear power plant transients, represented by 18 variable time excursions each one, are considered. The training patterns were obtained by precise simulation and each variable excursion is constituted by 61 samples with time step of 1 second.

Considering from 3 (the number of classes) to 34 vectors, and a minimum of 4 bits to represent the variables values, the best relation between correct classification and number of vectors, proposed by the GA, was *97% of correct classifications with the 19 vectors* shown in Table 2.

Table 2: Best Voronoi Vectors found by the GA and theirs respective classes

Vect..	*Vector: {t, v1,v2,v3,v4,v5,v6,v7,v8,v9,v10,v11,v12,v13,v14,v15, v16,v17,v18}*	*Class*
0	{56.00, 7.33, 288.04, 117.63, 55667.35, 1.37, 49.61, 2.92, 229.32, 484.87, 117.28, 49.29, 13.88, 5.20,25.11, 55.70, 31.90, 10.29, 203.15}	2
1	{13.18, 33.40, 55.05, 166.08, 59457.49, 25.14, 5.71, 4.57, 29.45, 52.81, 99.37, 27.29, 47.06, 4.46, 13.34, 79.63,14.52, 41.69, 93.67}	2
2	{32.47, 98.59, 42.25, 146.47, 47850.23, 33.38,25.80, 0.40, 292.43, 72.25, 91.31, 37.07, 40.94, 11.77, 96.89, 61.59, 17.74, 28.88, 211.67}	0
3	{45.35, 53.99, 25.70, 11.58, 52794.92, 5.52, 48.31, 4.37, 321.04, 449.98, 52.13, 90.79, 9.92, 7.59, 17.33, 85.85, 27.35, 42.77, 58.62}	2
4	{49.13, 50.71, 25.70, 263.99, 28537.79, 6.31, 15.94, 1.75, 156.30, 302.50, 75.50, 101.43, 37.80, 7.09, 62.22, 70.89, 10.79, 9.79, 63.50}	0
5	{8.03, 1.64, 226.20, 159.78, 19500.83, 48.89, 45.82, 2.68, 456.22, 11.88, 116.84, 56.44, 25.04, 0.87, 66.16, 65.37, 41.02, 7.25, 68.39}	1
6	{47.72, 97.34, 208.21, 57.89, 57075.59, 16.17, 49.31, 0.62, 80.26, 332.86, 118.64, 55.62, 57.64, 1.74, 72.46, 29.14, 29.87, 43.49, 300.43}	2
7	{55.28, 103.06, 185.08, 215.36, 41379.80, 6.31, 23.41, 7.18, 511.14, 133.33, 210.31, 46.62, 48.66, 3.48, 81.91, 13.39, 28.06, 0.00, 227.15}	0
8	{46.77, 89.97, 203.07, 90.31, 29489.05, 45.74, 45.32, 2.62, 33.79, 519.38, 19.77, 86.70, 30.71, 5.47, 71.67, 46.47, 26.63, 35.15, 139.22}	0
9	{41.10, 76.89, 205.64, 155.15, 21403.35, 17.74, 49.81, 2.81, 46.47, 337.20, 82.69, 70.34, 23.15, 4.98, 13.39, 77.19, 10.43, 27.91, 134.34}	0
10	{58.58, 6.54, 146.52, 0.00, 52794.92, 0.79, 29.39, 1.31, 490.01, 194.06, 217.50, 1.64, 36.38, 11.45, 44.89, 25.99, 43.18, 24.64, 92.82}	1
11	{25.04, 103.88, 259.62, 226.94, 58978.11, 46.53, 35.36, 1.62, 308.37, 345.88, 224.69, 11.45, 51.50, 0.25, 60.65, 0.00, 28.79, 30.81, 65.95}	2
12	{36.38, 0.00, 43.70, 266.30, 9988.23, 14.19, 5.48, 4.06, 384.41, 410.94, 154.59, 67.07, 15.12, 14.80, 22.05, 74.04, 34.90, 43.49, 36.64}	2
13	{21.26, 75.25, 280.18, 219.99, 9512.60, 2.37, 61.26, 7.62, 88.71, 250.45, 82.69, 94.07, 15.12, 1.00, 64.59, 59.07, 9.72, 41.32, 119.68}	0
14	{57.17, 6.54, 197.93, 50.95, 42806.69, 39.03, 56.78, 4.75, 536.48, 363.23, 167.17, 76.89, 29.76, 15.80, 48.83, 96.88, 28.43, 7.97, 31.75}	2
15	{41.10, 31.90, 205.64, 113.47, 15695.79, 31.54, 36.36, 5.18, 29.57, 228.76, 213.91, 87.52, 19.84, 0.75, 79.55, 62.22, 6.12, 35.52, 156.32}	1
16	{33.54, 27.81, 2.57, 275.57, 1902.52, 42.98, 43.83, 4.06, 274.58, 124.66, 59.32, 49.90, 11.81, 4.98, 11.03, 45.68, 20.87, 2.54, 227.15}	2
17	{0.00, 30.26, 267.33, 37.05, 11415.12, 46.53, 61.76, 0.50, 354.84, 124.66, 165.37, 89.16, 58.11, 2.99, 23.63, 37.81, 15.83, 45.67, 12.21}	2
18	{11.34, 68.71, 125.95, 69.47, 52794.92, 31.94, 14.94, 4.62, 80.26, 42.24, 107.85, 47.44, 6.61, 15.05, 27.57, 81.91, 10.07, 36.24, 266.23}	0

Data Mining II, C.A. Brebbia & N.F.F. Ebecken (Editors)
© 2000 WIT Press, www.witpress.com, ISBN 1-85312-821-X

Note that it was found 7 vectors representing Class 0, 3 vectors representing Class 1 and 9 vectors representing Class 2. These numbers really show an expert knowledge discovered by the GA, that is: *Classes 0 and 2 are more confusing than class 1.* That's why they need more representing vectors.

This result was obtained using population size of 100 individuals, crossover rare of 60%, mutation rate of 2%, ranking and elitism. The approach used to calculate k_1 and k_2 was the same described in the hypothetical problems. In this case, the values used for k_1 and k_2 were 1.0 and 0.032 respectively. According to k_1 and k_2 values, the search may emphasize the *minimization of the number of vectors* or *minimization of misclassifications*. Here, the *minimization of misclassifications* was chosen in order to be more conservative in terms of precision in the diagnosis.

The best fitness convergence can be seen in Figure 7.

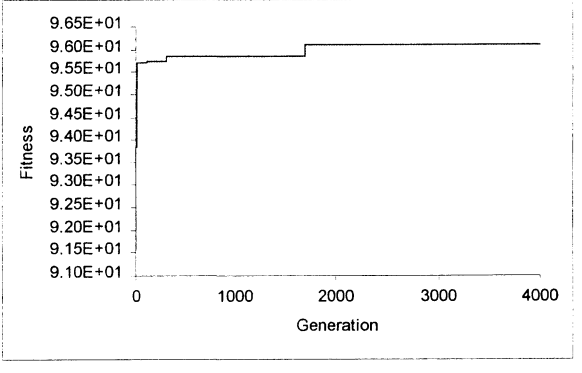


Figure 7: Best fitness convergence

The best fitness in the initial population was 91.58 (7 vectors / 92% of correct classification). In 9th generation, the number of correct classifications reached 96%, but with 34 vectors. Three generations later (12nd generation) the 96% could be found with 27 vectors. In 13rd generation, the 97% could be found with 32 vectors. After this generation, the number of vectors was gradually decreasing, when finally, the value of 19 was found in the 1962nd generation.
Because of the values of k_1 and k_2 used, the GA quickly find a high number of correct classification, and then, start trying to decrease the number of vectors. However, as the number of vectors (1st gene) change, the rest of the genotype may be completely different, in order to increase the fitness. In other words, the building blocks may change drastically according to any modifications in the first gene. This fact really slows down the search process, making the optimization computationally hard.

Data Mining II, C.A. Brebbia & N.F.F. Ebecken (Editors)
© 2000 WIT Press, www.witpress.com, ISBN 1-85312-821-X

4 Conclusions

Genetic search demonstrated to be a powerful technique to compute the best Voronoi vectors. The results obtained in the simple examples were the global optimum for most of the experiments made, demonstrating accuracy and robustness.

The search for the Voronoi vectors, instead of time partitions, as proposed by Pereira [9] is more flexible, allowing different number of vectors per class, however, there is an overhead in terms of computational time. This is, in part, due to the genotype structure which the first gene (number of vectors), encoded by few bits, govern the building blocks, making the convergence harder.

Parallel GA is under development in order to allow the obtainment of better results in real problems which are often associated to topologically complex search spaces.

References

[1] Haykin, S, *Neural Networks-A Comprehensive Foundation*, Macmillan College Publishing Company, New York, 1994.

[2] Steven K. Rogers and Matthew Kabrisky, *An Introduction to Biological and Artificial Neural Networks for Pattern Recognition*, Spie Optical Engineering Press, 1991.

[3] Kohonen, T., *Learning Vector Quantization for Pattern Racognition*, Technical Report, TKK-F-A601, Helsinki University of Technology, Finland, 1986.

[4] Ng, W., Choi, S. and Ravishankar, C. V., An Evolutionary Approach to Vector Quantizer Design, *IEEE International Conference on Evolutionary Computation*, 1995.

[5] Lloyd, S. P., Least Square Quantization in PCM. *IEEE Transactions on Information Theory*, 28 (2), pp. 127-135, 1982.

[6] Goldberg, D. E., *Genetic Algorithms in Search Optimization and Machine Learning*, Addison-Wesley, 1989.

[7] Holland, J. H., *Adaptation in Natural and Artificial Systems*, An Arbor, University of Michigan, 1975.

[8] Darwin, C., The *Origin of Species by Means of Natural Selection*, John Murray, London, 1859.

[9] Pereira, C. M. N. A., Schirru, R. And Martinez, A. S., Learning an Optimized Classification System from a data Base of Time Series Using Genetic Algorithms, *First International Conference on Data Mining*, Brazil, pp. 21-34, 1998.

[10] Grefenstette, J. J., *A User's Guide to Genesis Version 5.0*, 1990.