Genetic Algorithms Applied to Clustering

Fernández V.[1,2]; García Martínez R. [1,2]; González R. [1]; Rodriguez L. [1]

1.- Intelligent Systems Laboratory. Computer Science Department. School of Engineering. University of Buenos Aires
2.- Artificial Intelligence Laboratory. Buenos Aires Institute of Technology
E-mail: rgm@itba.edu.ar

Abstract

In the following paper it is developed an image compression technique using clustering with genetic algorithms. This technique tries to avoid the problems associated with limitations from the operating system (MS-DOS), as Mariano O'kon [1] proposses in his tesis on genetic algorithms.

The main problem, appears to be the large size of the data structures used by this kind of algorithms, which can become unmanageable for the operating system. Our solution consists in classifying vectors by parts, obtaining in this way shorter chromosomes.

Key Words:

Genetic Algorithms - Machine Learning.

I. A Brief introduction to Genetic Algorithms

Genetic algorithms are search algorithms based on natural genetic and selection combining the concept of survival of the fittest with an structured interchange, but aleatory of the information. These concepts involve the preservation of the characteristics of the best exponents of a generation in the next generation; moreover introducing aleatory changes in the new generation composition by means of crossing over and mutation operations. This aleatory component prevents getting stuck into a local maximun from which you can not escape to reach a global maximun. This would represent one of the main advantages of genetic algorithm in opposition to the traditional search methods as the gradient method. Another advantage is its utility for real time applications, in spite of not providing the optimal solution to the problem it provides almost the better solution in a shorter time, including complex problems to solve by traditional methods.

II. Clustering with Genetic Algorithms

Clustering purpose is to divide a given group of objects in a number of groups (clusters), in order that the objects in a particular cluster would be similar among the objects of the other ones. This technique tries to solve how to distribute N object in M clusters (in this paper *c_elem* objects in *n_clase* clusters) according to the minimization of some optimization criterion additive over the clusters. Once the optimization criterion is selectied, the clustering problem is to provide an efficient algorithm in order to search the space of the all possible classifications and to find one on which the optimization function is minimized.

The problem is to classify a group of samples. These samples form clusters of points in a n-dimensional space. These clusters form groups of similar samples. The more formal procedures use an optimization criterion such as minimize the distance additions of each sample to the clusters centre, which can be considered as the gravitatory of a cluster; This means a unique point X which better represents all points from this cluster. This optimization criterion was used in our work and the minimization process is performed by genetic algorithm.

III. Clustering technique applications to the image compression

It is used 128 x 128 pixels images, each one can take a value bewteen 0 and 255, what gives a resolution of 256 grey tones. In this way, a byte is enough to codificate each pixel, therefore the size of the image used comes from a program called TARGA BITMAPS in a file *.tga and it occupies 17170 bytes due to a header that was added. The passage to byte format, without header (*.img) carried out by TGA2RAW program.

Reading the file *.img is performed constructing arrays of 4x4 elements as it is shown in the following picture.

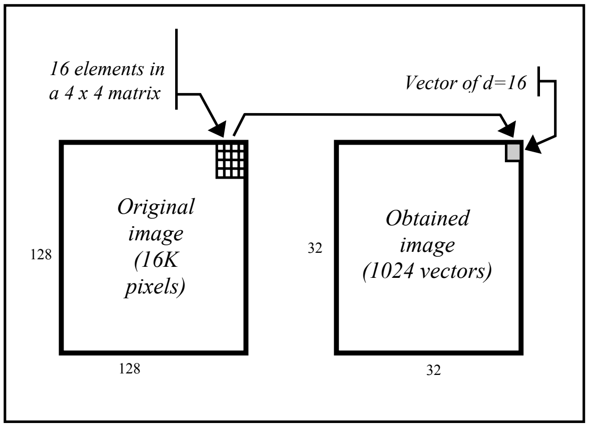


Fig. 1

As you can see, the final image has 32 x 32 =1024 vectors of 16 bytes each one (16 Kbytes) stored in a file *.vec.

The clustering idea exposed, tends to these vectors that contain information of the former image, in a determined amount of forms. As the amount of possible grey tones obtained with the codification used is of 256, this could be the quantity of types of the 1024 vectors. So now, the size of compressed image would be:

256 x16 +1024= 5120 bytes = 5Kbytes

that means that it will be composed of 256 vectors of 16 bytes each. These vectors will be the most representative ones of each type. The other 1024 bytes will indicate to which type belongs each one of the vectors of the non-compressed image. It is worth comparing the result obtained with the original image size (16 Kbytes). The compression ratio resulting from the classification is:

16/5 = 3,2

IV. Implementation of image compression in stages

This section is devoted to the description of the main aspects of the implementation of clustering techniques for image compression and to the introduction of the concept of "the implementation in stages" that tries to eliminate the problems emerged from the memory limitations of the operating system used.

Starting from the array of 32 x 32 vectors of 16 elements each stored in the file *.vec, 8 groups of 8 x 16 vectors each are generated. The election of the 128 vectors integrating each of the 8 groups is not selective. They are formed by dividing the original array in 8 rectangles. This division process is implemented by the program VEC2GRP and the reorganized array is saved in a *.grp file.

Once the 8 groups are formed, the clustering algorithm is executed to carry out the classification by parts. Each of these groups are independently classified into 32 clusters. Once the clustering for the 8 groups is finished, 256 clusters will be obtained.

In the implementation of the clustering algorithm the following codificacion has been used. Each chromosome is a vector of 128 elements (bytes) which indicate the vector indentified by the index that belongs to. For instance, if the element number 56 of the chromosome contains the number 10, it is understood that the vector 56 of the group in cuestion belongs to class 10.

The optimization criterion that has been used was the **Square of the Sum of Errors**. As the aim is to minimize the distance of each vector to the center of the cluster to which it belongs to, the fitness function to maximize is:

$$f = \begin{cases} M - e & \text{si } M - e \geq 0 \\ 0 & \text{si } M - e < 0 \end{cases}$$

where M is a constant that in the first place is equal to the maximum error possible and e is given by:

$$e = \sum_{i=1}^{NC} \sum_{j=1}^{n^i} \left\| \overline{x}_j^i - \overline{z}^i \right\|^2$$

Where:

NC: number of classes

n^i: number of vectors in class i

$\overline{x}_j^i$: vector j of class i

$\overline{z}^i$: medium value of class i

The medium value of de class i is calculated as:

$$\overline{z}^i = \frac{\sum_{i=1}^{n^j} \overline{x}_i^j}{n^j}$$

The principal data structures used in the implementation of this algorithm were:

c: Is the vector that contains the information about to which cluster belongs the vector indicated by the index.

n: Is the vector that contains the quantity of elements in the class indicated by the index.

z: Is the vector that contains the coordinates of the center of the cluster indicated by the index.

Thus the memory required for the algorithm is:

c+z+n

Therefore for each chromosome of a generation the require:

128+32*16+32=672 bytes

If the generation has 51 chromosomes (N=51):

51*672=34272 bytes

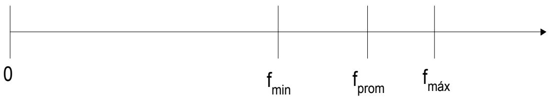
In case of not using implementation by parts the memory required for each chromosome is:

1024+254*16+256=5376

For a generation of 51 chromosomes (N=51):

51*5376=274176

During the algorithm implementation we detected that after a certain number of iterations (generations), the population was very diverse but the fitness average for all the chromosomes was so high that the minumun fitness and the maximun fitness of each generation were very close to it.



To solve this problem, we scaled the fitness function with a lineal function in the following way:

f'(x)=af(x)+b

a and b were calculated considering that the fitness average was the same for the original function (*f(x)*) and for the function transformed (*f'(x)*) and requiring that f'_{min}=0.

$$\begin{cases} f'_{prom} = f_{prom} \\ f'_{min} = 0 \end{cases}$$

$$f'_{prom} = af_{prom} + b = f_{prom} \quad \Rightarrow \quad b = f_{prom}(1-a)$$

$$f'_{min} = 0 = af_{min} + b = af_{min} + f_{prom}(1-a) = a(f_{min} - f_{prom}) + f_{prom}$$

$$\Rightarrow \quad a = \frac{f_{prom}}{f_{prom} - f_{min}}$$

Once the transformation was implemented the results were much better than the previous ones.

This process is accomplished by the program GRP2CMP, which save in a file (*.cmp) the mapping and the cluster centers for each group classified. When the classification process finishes, the file size which contains the compress image is:

nrogrp*(c+z)

where:

nrogrp: is the number of groups in which the original array was partitioned, that in this case is 8.

Therefore the file size is:

8*(128+32*16)= 5120 bytes = 5 Kbytes

In order to retrive the image, the file *.cmp is decompressed by the program CMP2GRP into a file *.grp of 16 Kbytes. Then it is reorganized to the format *.vec by means of the program GRP2VEC that starts the process inverse to the one before. Once this file is obtained the programs VEC2IMG and IMG2TGA are called to obtain the original file with the original format *.tga.

V. Conclusions

In this paper, the technique of clustering with genetic algorithm was applied to image compression. Basic operators of selection, crossing over and mutation were utilized. A lineal scale change of the fitness function was accomplished. This change provideded a very significant improvement to the results obtained. Finally, the concept introduced was the implementation by stages, which appeared as a possible solution to the memory limitations of the operating system adopted (MS-DOS). The main idea is to divide the problem into parts and to apply the clustering tecnique to each part independently.

The original image was divided in eight rectangles. This was done without following any criterion regarding selectivity. With this group conformation the results obtained were not satisfactory. The following step is to develop this implementation with different selective criterion of the vectors that form each group. This would have to consider the directions of minor variations of the pixels tones, so in this way, avoiding sudden changes that occur after the division has already been implemented.

VI.References

[1] O'kon Mariano. Tesis de Ingeniería Electrónica. Algoritmos Genéticos. Facultad de Ingeniería Electrónica. Universidad de Buenos Aires. 1994.

[2] Jay N Bhuyan, Vijay V. Raghavan, Venkatesh K. Elayavalli. Genetic Algorithm for Clustering with an Ordered Representation.

[3] Bhuyan J. N., "Genetic Algorithm for Clustering with an Ordered Representation", Proceedings of the fourth international conference in Genetic Algorithms, University of California, San Die go, 1991, páginas 408-415.