Genetic Algorithms and Clustering: an Application to Fisher's Iris Data[1]

Roberto Baragona, Claudio Calzini
Dipartimento di Sociologia, Università La Sapienza, Via Salaria 113, I-00198 Roma, Italy, e-mail: baragona@uniroma1.it

Francesco Battaglia
Dipartimento di Statistica, Probabil. e Stat. Appl., Università La Sapienza, Piazzale Aldo Moro 5, I-00185 Roma, Italy, e-mail: battag@pow2.sta.uniroma1.it

Abstract: Fisher's iris data constitute a hard benchmark for clustering procedures, and attracted much work based on statistical methods and new approaches related to evolutionary algorithms and neural networks. We suggest two genetic algorithms effective for simultaneously determining both the optimal number of groups and the assignment of items to groups. The grouping genetic algorithm proposed by Falkenauer (1998) forms the basis of our method, where the variance ratio criterion and the Marriott's method provide two fitness functions that both allow for fast computation and include the number of groups explicitly as a parameter. Specialized crossover operators, specific for each of the two fitness functions, are designed to accelerate convergence and minimize the number of iterations. Some simple implementations of our genetic algorithms are presented, that allow to classify correctly as many iris plants as the best alternative procedures proposed for this data set. Therefore genetic algorithms seem to constitute a good alternative choice for handling clustering problems.

Keywords: Classification, Crossover, Encoding, Evolutionary Computation, Fitness Function, Mutation

1. Introduction

In a classical paper Fisher (1936) considered a data set consisting of 150 items of iris specimen belonging to different varieties: 50 *setosa*, 50 *versicolor*, and 50 *virginica*. Four measurements were taken on each of them. Fisher noted that whilst the *setosa* group is neatly discriminated, the other two groups contain items which are similar and difficult to distinguish. Further interest arose about this data set. Mezzich and Solomon (1980) list several contributions and rank them according to several optimality criteria. Friedman and Rubin (1967) reached a correct classification for all but three items (two *versicolor* and one *virginica*), the

[1]The present paper is financially supported by Ministero della Ricerca Scientifica e Tecnologica, Italy

S. Borra et al. (eds.), *Advances in Classification and Data Analysis*

best results to date. Also, iris data have been analyzed by means of evolutionary methods (the Beagle algorithm of Forsyth, 1986) and neural networks (see Adorf and Murtagh, 1988).

Genetic algorithms have been introduced by Holland (1975), though a pioneering contribution of Box (1957) is worth to mention, and have been successfully adopted to handle problems with high computational complexity. Some fields of applications are listed in Mitchell (1996, pp. 15-16). Pattern recognition, machine vision, image reconstruction, identification of objects in images, seem to be the fields where genetic algorithms have been used most frequently, as far as clustering is concerned. Clustering, partitioning and grouping problems are characterized by high degree of complexity, especially if the number of groups is unknown, and genetic algorithms have been proposed as useful tools for searching for solutions (see Jones and Beltramo, 1991). However, few contributions are found in the statistical literature (see Chatterjee, Laudato and Lynch, 1996), where no examples are reported of applications of genetic algorithms to cluster analysis. A stochastic searching algorithm was examined by Fortier and Solomon (1966) which seems to envisage the use of mutation in genetic algorithms. As a matter of fact, unless the set of items and the number of partitions are smallest, an exhaustive approach is simply not feasible. Genetic algorithms are likely to perform more efficiently than other discrete search methods, in that they do not process a single path to solution, but handle simultaneously a population of candidate solutions. These individuals are evolved in a parallel fashion, though controlled interaction may take place, which is aimed at improving the average fitness. So, the search for solutions is performed in a region larger than other methods may explore. In addition, the stochastic nature of the genetic approach is likely to act in such a way that the risk for the algorithm to be trapped in local optima is greatly reduced.

In most applications, however, the encoding does not look to be well tailored for the specific clustering problem, as primary interest is focused on items instead of on groups. Furthermore, the number of groups is assumed to be known, or a preliminary analysis is required to be performed to find the best number of clusters. The grouping genetic algorithm (*GGA*) proposed by Falkenauer (1998) seems to constitute the best procedure to capture the very structure of the clustering problem. Falkenauer showed how to implement his procedure for some particular problems, but suggested resorting to local problem-dependent heuristics to handle different problems. As a matter of fact, to be effective for classifying the iris data, the fitness function and the genetic operators have to be given a peculiar formulation that we provide in the present paper. We discuss how genetic algorithms may be employed to partition the iris data into optimal number of non overlapping groups and report how our applications produced similar results in comparison with the best published ones. We found that genetic algorithms perform efficiently and constitute a simple and effective alternative for handling clustering problems. The plan of the paper is as follows. In Section 2 a brief account of applications of genetic algorithms to clustering problems is given. The *GGA* is outlined in Section 3, and our proposal of two fitness functions and

heuristic crossovers is introduced in Section 4. Application to the iris data is presented in Section 5. In Section 6 conclusions are drawn.

2. Clustering by genetic algorithms

A genetic algorithm evolves a finite discrete population of individuals, each of which has to encode a single solution to the clustering problem. We assume that every item has to belong to one and only one group of the partition and the assignment of items to groups is made by attempting to maximize the fitness (objective) function. Encoding is a crucial point which may drive the choice of what genetic operators to be used in conjunction with, and several methods have been proposed. Some of them, that we may mention here, include the linear code (often in its group-numbers form) and the Boolean matching code, permutation encoding, binary string, and ordinal and ordered representations. Redundancy and context insensitivity are envisaged by Falkenauer (1998, Chapter 4) as serious drawbacks of such encoding techniques. In addition, we may notice that, when these encoding methods are adopted, either the number of groups has to be held fixed, or, even if specific devices are employed to allow the number of groups to vary, this event is unlike to occur in practice, because the number of groups is never taken explicitly into account. So the algorithm is almost always confined to search the optimum for some given groups' number. As a matter of fact, there are as many optima as the numbers of groups ranging from one to the maximum allowed, but only one is the global optimum we are seeking for.

3. The grouping genetic algorithm

We shall outline the basic steps of the *GGA*. The encoding, crossover and inversion are reported as introduced in Falkenauer (1998, Chapter 5). The procedure of reproduction and the mutation operator are chosen by us, among the alternatives therein proposed, as the best ones for our specific problem. Let n denote the number of items to classify, p the number of measurements taken on each item (variables), and g the maximum allowed number of clusters.
Let s be the size of the population that is being processed by the genetic algorithm. Each individual h, $h=1,\ldots,s$, represents a candidate solution, which is characterized by a number of groups k, $1 \le k \le g$, and a partition (mutually exclusive) of the n items into the k groups. The chromosome for encoding such a candidate solution is a string of length $n+k$ of integer values. Genetic operators, but mutation, are applied only to the genes at the loci $n+1$, $n+2$, ..., $n+k$. So, for the ease of exposition, we will consider two distinct chromosomes, the first one of fixed length n, and the second one of variable length k. The first one follows the group-numbers straightforward encoding: each gene relates to a specific item, and may take a value between 1 and g, indicating the cluster to which the item itself belongs. The second one encodes the clusters' labels as genes. Therefore its length

k is variable, and ranges from 1 through g. The allelic values of its genes still are integers within the range $[1,g]$.

The so-called roulette wheel is the most common way of modeling the reproduction of the individuals in a given population. Copies of an existing individual are generated with probability proportional to its fitness function. In a population of s individuals, each of which has fitness function f_h, h=1,...,s, the expected number of copies of the h-th individual is $sf_h/\Sigma f_h$, or f_h/f^*, where f^* is the average fitness $\Sigma f_h/s$. So individuals characterized by above average fitness have a probability of reproduction higher than the remaining ones. In our implementation, the new population entirely replaces the old one, except the individual, which, in the past population, had the highest fitness function. If this individual exited the population, it is recovered to replace the individual that, within the current population, has the worst fitness function. This is our choice for the implementation of the elitist strategy. Rudolph (1994) advocated using the elitist strategy as necessary for the genetic algorithm to converge.

A kind of two-point crossover is performed on the second chromosomes of the parent's individuals. Two cutting points are randomly selected for each parent. Then two children are formed, the one by inserting the genes in between the crossing sites of parent two just before the first crossing point of the parent one, the other by reversing roles of the parents individuals. If necessary, the resulting groups have to be adapted according to the hard constraints and the fitness function. At this stage, local problem-dependent heuristics may be applied. The two children replace both parents in the new population. Typically, crossover applies only to a portion of the population. The percentage of pairs to be selected for crossover is the crossover's rate p_c. The candidate pairs of individuals are $s/2$. The number of crossovers, which are performed at a given step, is a binomial random variable with parameters $s/2$ and p_c. So, it turns out that the expected number of crossovers is $p_c s/2$.

The mutation operator acts so that a small number of genes are allowed to change at random their allelic values. The purpose of mutation is to maintain diversity among individuals in the population. Furthermore, it is the only way to recover some solutions that were lost all along the evolutionary path of the population, or to explore new regions of the solutions' space. Also, mutation is a useful operator as far as convergence of the genetic algorithm is concerned (Rizzi, 1997). In general, mutation is often defined as the smallest possible modification of a chromosome. For the grouping problem, it seems appropriate to consider that a mutation occurs when an item is moving from a group to another. In practice, with a small probability p_m, any gene of the first chromosome of each individual may change its allelic value to another one chosen randomly among the allelic values within the second chromosome.

Inversion is sharing features from both mutation and crossover. In fact, like mutation, it applies to a single individual. Then, it requires two crossing sites be chosen at random, so as to look alike two-point crossover. The genes between the crossing sites are reversed in order. Each individual is assigned a usually small probability for inversion p_i to occur. There seems a general agreement to exist that

it should only be applied when the interpretation of each gene is not locus dependent, just like happens for the second chromosome, which lists the groups' labels. Each gene, therein, refers to a group whose meaning depends only on what items are belonging to it. Inversion does not change the composition of the groups. The reason for using it resides in that, the promising genes (well performing groups), if close together, are more likely to be transmitted to individuals in the next generation.

4. Two fitness functions

Let B, W and T denote, as usual, the between-groups, within-groups and total sum of squares $p \times p$ matrices. We shall introduce two fitness functions that we tried as criteria for separating the iris data. The first one is aimed at minimizing trace(W), the second one det(W). Note that, under Gaussian multivariate model, the classification maximum likelihood approach reduces essentially to minimize det(W), if the observations within all groups are assumed to have common covariance matrix, and to minimize trace(W), if the common covariance matrix is diagonal (see Banfield and Raftery, 1993). When the covariance matrices may differ from group to group, then the evaluation of det(W_j), j=1,...,k, where W_j is the sum of squares matrix of the items that belong to the j-th group, is too time-consuming with respect to the (at present) available computing resources. In addition, Scott and Symons (1971) noticed that, in practice, the maximum likelihood methods would always partition the data into the maximum number g of partitions allowed. Methods for determining the optimal number of groups that are including as well a criterion for optimal assignment of items to groups seem to be only the variance ratio criterion (*VRC*) (Calinski and Harabasz, 1974), and the Marriott's method (*MM*) (Marriott, 1971). Required computations may be performed quickly enough to make their use in conjunction with genetic algorithm viable in practice.

We designed two different versions of *heuristic crossover*, according to what fitness function is concerned.

The variance ratio criterion $VRC=\{\text{trace}(B)/(k-1)\}/\{\text{trace}(W)/(n-k)\}$ offers directly a suitable fitness function, because it takes only positive values and has to be maximized for optimal partition of items into groups. The crossover may be adapted to the fitness function, in order to try to accelerate the convergence towards the solution. Following Falkenauer (1998, pp. 100-101), the items occurring twice are put aside, and each is re-assigned to the group with nearest centroid in terms of squared Euclidean distance. This is a well-known device in non-hierarchical clustering algorithms, which leads to increase trace(B). As $B+W=T$, and k is fixed, this implies that the *VRC* increases too.

Marriott's method consists of finding that partition that minimizes $k^2\det(W)$. The latter is a positive function, but a fitness function must be a non-decreasing one. So we may define $MM=\det(T)/\{k^2\det(W)\}$ as a suitable fitness function, where det(T) is constant over partitions and independent of k (Marriott, 1971, p. 503).

The crossover is performed as follows. The items, put aside because are being assigned to two different groups, are each re-assigned to the valid group whose mean minimize the Mahalanobis' generalized distance from the item itself. We used the formula reported in Marriott (1971, p. 508), where the within-groups dispersion matrix is computed by taking into account only the items which are not involved in the re-assignment step. Moreover, the co-ordinates of the groups' centroids are not updated at each re-assignment, for computations would turn out to be too much cumbersome, in the presence of little improvements.

5. Application to iris data

Since the work of Fisher (1936) many authors noticed that the species *setosa* does not overlap with the other two species whilst these latter are overlapping somewhat. Evidence of this circumstance is given by plots of the data and by exploratory methods (see Cerioli and Zani, 1999). A variety of clustering methods has been tentatively entertained in order to produce a partition of the iris data, which may exactly match the three species. We shall label the plants by taking consecutively down the rows of the originally published table by Fisher. So, labels 1-50 denote iris *setosa*, 51-100 *versicolor* and 101-150 *virginica*. Friedman and Rubin (1967, Section 6.2) applied the min trace(W) and the max det(T)/det(W) criteria to this data. The first one gave, for fixed k=3, 10 misclassified items, partly *versicolor* and partly *virginica*. For k=4, still *setosa* resulted as a separate group, then a group was obtained including only *virginica*, a group only of *versicolor* except for plant 107 of *virginica*, and a group containing a mixture of *versicolor* and *virginica*. The second criterion yielded a partition into 3 groups, which recovered the three species except for 3 plants: 71 and 84 went with *virginica*, and 134 went with *versicolor*. Another criterion they applied, consisting of maximizing trace($W^{-1}B$), produced as well, for fixed k=3, a partition with 3 misclassified items (but not the same as before). Duran and Odell (1974, p. 103), by using Mahalanobis' distance and k=9, obtained 22 misclassifications, and, by using Euclidean distance and k=17, 8 misclassified plants. A clustering method based on neural networks was proposed by Adorf and Murtagh (1988). Example on the iris data with k=3 was reported, where their method led to correctly classify 48 iris *setosa* in the first group, 35 *versicolor* in the second one, and 33 *virginica* in the third group. However, 34 plants were misplaced. Everitt (1993, p. 116) reported application of normal mixtures for the analysis of the iris data, with only 5 misclassified of *versicolor* added to *virginica*. 5 misclassifications were reported as well by Fraley and Raftery (1998) as a result of *EM* algorithm.

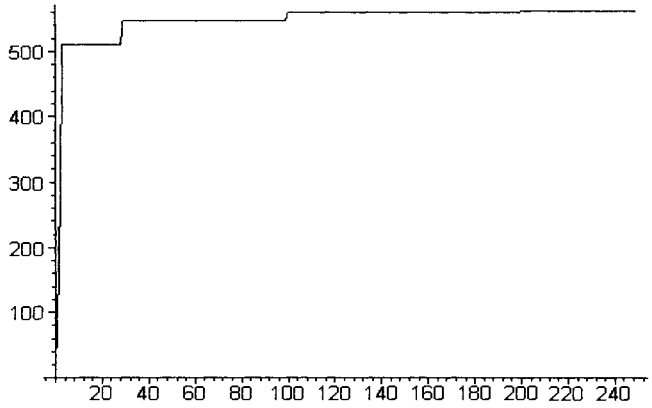
Of course, genetic algorithms cannot perform better than the fitness function they are being used with. Their usefulness resides in that they are able to find solutions that have not been possibly reached by other methods. As far as problems of this kind are concerned, where the space of solutions cannot be explored exhaustively, we shall never be confident that the global optimum is actually achieved, but some confirmation or some improvement may as well be obtained. For implementation

of a genetic algorithm, quite a few parameters are needed to specify. We made our choice mainly on the guidelines provided by Mitchell (1996, Section 5.6). We assumed s=200 as population size, the rates of crossover, mutation and inversion p_c=0.8, p_m=0.005, and p_i=0.1 respectively. The number of items is n=150 and that of measurements is p=4. For the maximum number of groups, we considered both g=3 and g=5. The initial population of s individuals was generated as follows: for each individual, the number of groups k was chosen as a random integer uniformly ranging from 1 through g; then, each item was uniformly randomly assigned to one of the k available groups. Renumbering was applied for the groups' labels to coincide with the integers 1,2,...,k. Number of iterations has been 50,000 for both fitness functions, though a much smaller number proved to be really needed.
Let us consider the *VRC*-based genetic algorithm and let g=5 are the maximum allowed number of clusters. In Table 1 the assignment of species to groups is displayed which maximized the fitness function.

Table1: *Assignment of species to groups on VRC-based fitness function*

group	*setosa*	*versicolor*	*virginica*	
1	50			50
2		48	14	62
3		2	36	38
	50	50	50	150

Figure 1: *Increasing of VRC over iterations*



The first group is composed only of *setosa*, the second one is a mixture of 48 *versicolor* and 14 *virginica*, the third group is composed of 36 *virginica* and the *versicolor* 53 and 78. Summing up, we have 16 misclassified items. Nevertheless, the maximum of the fitness function turns out to equal 561.63, whilst the figure reported by Friedman and Rubin for their min trace(W) criterion, given k=3, translated to *VRC*, corresponds to 560.43, in spite of the fact that, in the latter

case, the misclassified items are only 10. This circumstance is not surprising, however, as we may compute 487.33 the *VRC* for the partition which assigns exactly the three species to each of three groups. In Figure 1 the fitness is plotted as a function of the iterations. Its behavior seems to support that genetic algorithms may quickly find promising regions of the search space, with sudden increasing of the fitness function, followed by slower heuristic problem-dependent search for the true optimum (see Mitchell, 1996, p. 124). The maximum is achieved in correspondence of the 200-th step. Note that the genetic algorithm include simultaneously in its search all candidate solutions with k ranging from 1 through 5. This means that the *VRC* points at k=3 as the optimal number of group.
Let us now take the *MM*-based genetic algorithm with g=3 into account. After 200 iterations, exactly the same result, obtained by Friedman and Rubin by using their max det(T)/det(W) criterion, is achieved. The best fitness figure turns out to be 5.04. Note that the fitness computed when assigning each species to a separate group equals 4.74.
For g=5, results yielded by the *MM* criterion are displayed in Table 2. The optimal number of groups turns out to equal k=4, and the corresponding fitness function is equal to 6.8, a figure that encompasses the one which we can compute from the value 74.13 reported by Friedman and Rubin for max det(T)/det(W), that they obtained by assuming 4 groups. This seems to indicate that genetic algorithm may perform a better search towards the true optimum. The fitness function increases as displayed in Figure 2. The maximum value is reached after 109 iterations. By looking at Table 2, we may observe that *setosa* are well separated again, and form a group on their own, whilst *virginica* are split in two groups, the first one adding *versicolor* 84 and the second one *versicolor* 71. The remaining group is solely composed of *versicolor*. So, we have only 2 misclassified items. These findings are in agreement with Scott and Symons (1971, p. 394), who supported the use of max det(T)/det(W) as a criterion appropriate as far as iris data are concerned.

6. Concluding remarks

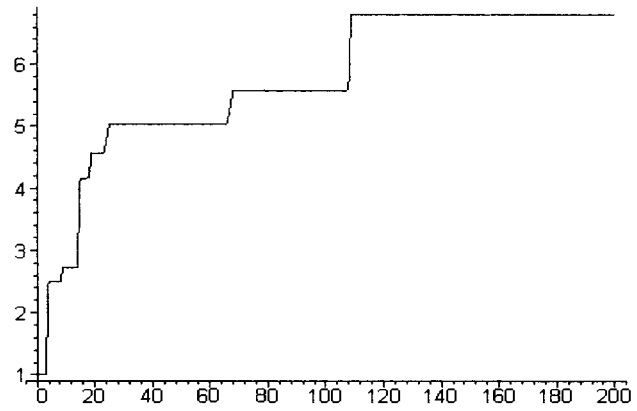
Fisher's iris data cannot be partitioned neatly into species as far as three groups are assumed. If we allow the number of clusters, and the assignment of items to them, both vary, the misclassified items may reduce to only 2, at the expense of splitting iris *virginica* into two groups. We found that genetic algorithms can perform efficiently the search moving around the space of solutions in a stochastic way driven by the genetic evolutionary mechanism. In order to avoid being trapped in local optima, variable length chromosomes, all along with grouping encoding and heuristic crossover, seem most efficient, since the number of clusters is explicitly modeled and the algorithm may take advantage from knowledge of the distance, properly assessed according to the fitness function, of items from groups' means.
So, genetic algorithms seem to constitute a good alternative choice for handling clustering problems. In fact, most statistical methods are able to explore only a

small region of the solutions' space, so that they are likely to find out local optima, and not the global one. On the other hand, some new proposals, such as neural networks and Gibbs sampler (Bensmail, Celeux, Raftery and Robert, 1997) need much heavier both design and computation. Genetic algorithms may explore the whole space of solutions and their implementation with respect to a given problem seems to be more straightforward. Computations are less cumbersome than for other methods, since the basic steps turn out to be very simple. In fact, most time is spent in the evaluation of the fitness function, a task that any algorithm must accomplish anyway.

Table 2: *Assignment of species to groups on MM-based fitness function*

group	*setosa*	*versicolor*	*virginica*	
1	50			50
2		48		48
3		1	17	18
4		1	33	34
	50	50	50	150

Figure 2: *Increasing of MM criterion over iterations*



References

Adorf, H.-M., Murtagh, F. (1988) Clustering based on neural network processing, in: *Compstat 1988*, Physica-Verlag Heidelberg for IASC (International Association for Statistical Computing), 239-243.

Banfield, J. D., Raftery, A. E. (1993) Model-based Gaussian and non-Gaussian clustering, *Biometrics* 49, 803-821.

Bensmail, H., Celeux, G., Raftery, A. E., Robert, C. P. (1997) Inference in model-based cluster analysis, *Statistics and Computing* 7, 1-10.

Box, G. E. P. (1957) Evolutionary operation: a method for increasing industrial productivity, *Applied Statistics* 6, 81-101.

Calinski, T., Harabasz, J. (1974) A dendrite method for cluster analysis, *Communications in Statistics* 3(1), 1-27.

Cerioli, A., Zani, S. (1999) Exploratory methods for detecting high density regions in cluster analysis, in: *Book of Short Papers CLADAG99*, Meeting held in Rome, Italy, 5-6 July 1999.

Chatterjee, S., Laudato, M., Lynch, L. A. (1996) Genetic algorithms and their statistical applications: an introduction, *Computational Statistics & Data Analysis* 22, 633-651.

Duran, B. S., Odell, P. L. (1974) *Cluster Analysis: A Survey*, Springer, New York.

Everitt, B. S., (1993) *Cluster Analysis* (Third edition), Edward Arnold, London.

Falkenauer, E. (1998) *Genetic Algorithms and Grouping Problems*, Wiley, New York.

Fisher, R. A. (1936) The use of multiple measurements in taxonomic problems, *Annals of Eugenics* 7, 179-188.

Forsyth, R. (1986) Evolutionary strategies, in: *Machine Learning: Applications in Experts Systems and Information Retrieval*, Forsyth & Rada, (Eds.), Ellis Horwood, London.

Fortier, J. J., Solomon, H. (1966) Clustering procedures, in: *Multivariate Analysis*, Paruchuri & Krishnaiah (Eds.), 493-506.

Fraley, C., Raftery, A. E. (1998) *MCLUST: Software for Model-Based Cluster Analysis*, Technical Report no. 342, University of Washington, The Statistics Department, Seattle, WA.

Friedman, H. P., Rubin, J. (1967) On some invariant criterion for grouping data, *Journal of the American Statistical Association* 63, 1159-1178.

Jones, D. R., Beltramo, M. A. (1991) Solving partitioning problems with genetic algorithms, in: *Proceedings of the Fourth International Conference on Genetic Algorithms*, Belew & Booker (Eds.), Morgan Kaufmann Publishers, San Mateo, California, 442-449.

Marriott, F. H. C. (1971) Practical problems in a method of cluster analysis, *Biometrics* 27, 501-514.

Mezzich, J. E., Solomon, H. (1980) *Taxonomy and Behavioral Science: Comparative Performance of Grouping Methods*, Academic Press, London.

Mitchell, M. (1996) *An Introduction to Genetic Algorithms*, The MIT Press, Cambridge, Massachusetts.

Rizzi, A. (1997) A convergence theorem for genetic algorithms, *Metron* 55, 69-83.

Rudolph, G. (1994) Convergence analysis of canonical genetic algorithms, *IEEE Transactions on Neural Networks, Special Issue on Evolutionary Programming*, 1-11.

Scott, A. J., Symons, M. J. (1971) Clustering methods based on likelihood ratio criteria, *Biometrics* 27, 387-397.