Genetic Algorithm Guided Clustering

James C. Bezdek\textsuperscript{1}, Srinivas Boggavarapu\textsuperscript{2}, Lawrence O. Hall\textsuperscript{2}, and Amine Bensaid\textsuperscript{2}

\textsuperscript{1}Department of Computer Science
University of West Florida
Pensacola, Fl. 32514

\textsuperscript{2}Department of Computer Science and Engineering
University of South Florida
Tampa, Fl. 33620
hall@ece.usf.edu

Abstract—
Genetic algorithms provide an approach to optimization. Unsupervised clustering algorithms attempt to optimize the placement of like objects into homogeneous classes or clusters. In this paper, we describe an approach to using genetic algorithms to optimize the clusters created during unsupervised clustering. Hard partitions of the feature space are the members of the population. They evolve into better partitions based upon the fitness function which is a version of the hard c-means optimization function. The method of crossover and mutation are described. An example of the clustering performance of this approach is shown with the Iris data. The genetic guided clustering is shown to outperform hard c-means on the Iris data in terms of the number of patterns which are correctly placed into a partition whose majority class is the same as the assigned pattern.

1 Introduction

Consider a set of \( n \) objects \( X = \{x_1, x_2, \ldots, x_n\} \) to be clustered into groups of like objects. Each \( x_i \in \mathbb{R}^p \) is a feature vector consisting of \( s \) real-valued measurements describing the \( p \) features of the object represented by \( x_i \). The features could be length, width, color etc. Particular clusterings of the objects can be represented by a crisp partition matrix, which is called a hard partition. The set of all \( c \times n \) non-degenerate constrained hard partition matrices is denoted by \( M_{\text{hen}} \) and is defined as

\[
M_{\text{hen}} = \{U \in \mathbb{R}^{c \times n} | \sum_{i=1}^{c} U_{ik} = 1, \sum_{k=1}^{n} U_{ik} \geq 1, \text{ and } U_{ik} \in \{0, 1\} \text{ for } 1 \leq i \leq c \text{ and } 1 \leq k \leq n\}. \tag{1}
\]

Given a criteria for performing a hard clustering, the problem is to find the corresponding best partition matrix in \( M_{\text{hen}} \).

The clustering criterion considered here is the function:

\[
J(U, V) = \sum_{i=1}^{c} \sum_{k=1}^{n} U_{ik} D_{ik}(v_i, x_k) \tag{2}
\]

where,
\( U \in M_{\text{hen}} \) is a fuzzy partition matrix; 
\( V = [v_1, \ldots, v_c] \) is the matrix of prototype parameters; and 
\( D_{ik}(v_i, x_k) \) is a measure of the distance from \( x_k \) to the \( i^{th} \) cluster prototype. A search is undertaken for a good representation of the cluster structure of \( U \) based upon a \((U, V)\) minimizer of (2). Typically, the optimal \((U, V)\) pair is sought using an alternating optimization scheme of the type generally described in [3].

1.1 Optimization by genetic algorithms

Genetic algorithms [6] are an important method for finding good, possibly optimal, solutions to large-scale optimization problems. The optimization is done by evolving, via crossover and mutation operations, a solution from some initial state guided by a fitness function. The initial state is a population whose values are to be optimized based upon the fitness function.
The fitness function for hard clustering will be based on $J(U,V)$ from (2). Our initial population will consist of $n$ U matrices, as opposed to encoded $v$'s (cluster centers) which was used in [1]. Our approach is distinct from that in [4], where partitions of data are represented as binary strings. Since we have only U matrices in our initial population, we will rewrite $J$ as:

$$J_1 = 0.5 \times \sum_{i=1}^{c} n_i s_i$$

(3)

where $n_i$ = number of points in hard cluster $i$, and $s_i = \frac{1}{n_i^2} \times \sum_{i=1}^{n_i} \sum_{j=1}^{n_i} \|x_i - x_j\|^2$. $J_1$ is based solely upon the U matrix [5].

In order to do genetic optimization of clustering, we need to define the crossover and mutation operators. Figure 1 shows how crossover is done. The crossover point and number of columns in the two U matrices chosen for reproduction are randomly chosen. The columns of the matrices are combined to create the children matrices.

$$U_1 = \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \quad U_2 = \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

Crossover point with distance of 2

$$\text{Child 1} = \begin{bmatrix} 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix} \quad \text{Child 2} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Figure 1: The crossover operation.

Figure 2 shows an example of mutation. Mutation consists of randomly choosing an element of a column to have the value 1, such that it is a different element than the one currently having a value of 1. Hence, the pattern represented by a column is randomly assigned to a new class.

$$\text{Column before mutation} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} \quad \text{Column after mutation} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

Figure 2: The mutation operation.

In our implementation an initial population of size $P$, consisting of U matrices is pseudo randomly generated such that each has at least one 1 in every row ($\sum_{j=1}^{n} U_{ij} \geq 1 \forall i$) and each column sums to 1, i.e. $\sum_{i=1}^{c} U_{ij} = 1, \forall j$. The U matrices are sorted by $J_1$ value and the R with the lowest $J_1$ values are chosen to
reproduce. The reproduction pairs are randomly chosen from the available pool of R. After reproduction and mutation, the R child U matrices are added to the population with the P - R U matrices with the greatest J1 values dropped from the population. The reproduction and survival of fittest process continues for some set number of generations.

2 Performance of Genetic guided clustering

In order to evaluate the performance of the genetic guided clustering, the simple Iris data set was used [3]. It consists of 150 patterns belonging to 3 classes with 50 patterns in each class. There are 4 real-valued features, petal-length, petal-width, sepal-length and sepal-width. We evaluated the results of the genetic guided clustering against hard c-means. Three different distance metrics [2] were used; the Euclidean, Mahalanobis and Diagonal which are defined below. The different metrics allow different shapes for the optimal clusters in \( \mathbb{R}^n \).

\[
D_{ik}(v_i, z_k) = d_{ik}^2 = (v_i - z_k)^T A (v_i - z_k),
\]

where \( A \) is a positive definite \( (n \times n) \) weight matrix. Let \( C_Y \) be the sample covariance matrix of data set \( Y \) [5]. Let \( \{a_i\} \) denote the diagonal values of \( C_Y \). Let \( D_Y \) be the Diagonal matrix with diagonal elements \( (d_Y)_{ii} = a_i \). Finally, let \( I \) be the identity matrix. Then our three norms are defined from equation (4) by using

\[
A = I \sim \text{Euclidean Norm}
\]

\[
A = D_Y^{-1} \sim \text{Diagonal Norm}
\]

\[
A = C_Y^{-1} \sim \text{Mahalanobis Norm}.
\]

In each test from which results are shown, clustering is done into \( c=3 \) classes. Hard c-means (HCM) is sensitive to the particular initialization used. Hence, we used 2 different kinds of initialization. The first is a partly random initialization of the U matrix and the second was to initialize the U matrix with the first 50 columns belonging to class 1, the second 50 columns belonging to class 2 and the third 50 columns belonging to class 3. The second initialization starts with a correct partition.

The partly random initialization is obtained as follows. For each cluster center \( v_i \), we choose the \( k^{th} \) element of the cluster center to be the \( k^{th} \) feature of a randomly chosen pattern to be clustered. This is done for each of the \( s \) elements of a cluster center. The process is repeated for each cluster center. An initial U matrix is then generated from the cluster centers. For a GA, population \( P \) (the population size) U matrices are generated in this manner.

In Table 1, the results of the 3 distance metrics within hard c-means are shown. In each line of the table, we show the iteration at which HCM converges (the U matrix is unchanged from the last iteration), how many errors exist, and the iteration at which the minimum number of errors was found. Errors are determined by finding the class of the majority of examples assigned to a cluster and then any examples labeled a different class are considered erroneously assigned. It is worth noting that the minimum number of errors is always found before convergence, except with a random initialization of U and the Mahalanobis distance metric. This behaviour of the clustering algorithm is typical of other initializations and occurs because we are not optimizing the number of errors, but \( J_1 \). One can conclude that \( J_1 \) does not optimize the number of classification errors for the Iris data set.

Table 2 shows the results in terms of classification errors of HCM for each distance metric averaged over 30 random initializations. It can be seen that the Euclidean norm has the least variance, followed by the Diagonal norm, and Mahalanobis norm.

Table 3 shows results for the GA guided clustering. These results are averaged over 30 runs with different random initializations of our population. In this case we report the number of generations rather than iterations, where the generation number is fixed. Again the standard deviation of the error results is reported. The average generation at which the minimum number of errors is obtained and the average number of minimum number of errors and average \( J_1 \) value are also included in the table. The Mahalanobis norm has a greater standard deviation and requires significantly more generations to get error totals lower
than the average obtained with straight HCM. On average, the GA approach provides a better clustering in terms of the number of errors. With the Diagonal norm, the GA approach provides a clear improvement in terms of the number of classification errors.

Next, if the random initial population is reduced by half to a size of 50. Table 4 reports the results from this experiment. The standard deviation for the errors is a little higher in this case, but the errors are about the same. \( J_1 \) does not get reduced as much in the smaller population, which appears to indicate that reducing \( J_1 \) does not necessarily reduce the error. This finding is consistent with the results shown in all the tables. The minimum error is observed at higher values of \( J_1 \) than its final value.

Table 5 shows results with a population of size 200. The results in terms of error and minimal \( J_1 \) value are improved with less or equal generations for each of the norms.

In Table 2, we show a graph of the classification errors of the top U matrix (as defined by minimum \( J_1 \) values) in a population and the second best member over the generations and in 3b a graph of the \( J_1 \) values of the top member and second best member of the same population for the Mahalanobis distance metric. The population size is 50 with the top 24 being chosen for reproduction. The behaviour is shown for one initialization that ends in 28 errors. It is similar for other initializations. It is clear that there is little difference between the top 2 members of the population. This metric shows a steady decrease in the measured errors in the classification as the \( J_1 \) value decreases.

In Table 3a, the classification errors over the generations are shown for the top and second best members of a 50 element population for the Euclidean metric. As before the best (in terms of minimal \( J_1 \) values) 24 elements are allowed to reproduce. The behaviour of the error curve is typically parabolic with the minimum errors occurring before the minimum \( J_1 \) value (Table 3b) is achieved. The \( J_1 \) value decreases very slowly (in fact fractionally) with several noticeable downward jumps.

Table 1: Hard c-means results with different norms and initializations.

<table>
<thead>
<tr>
<th></th>
<th>Final results</th>
<th>Optimum results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>errors</td>
<td>iteration</td>
</tr>
<tr>
<td>Euclidean norm (random U init)</td>
<td>17</td>
<td>12</td>
</tr>
<tr>
<td>Euclidean norm (Optimal U init)</td>
<td>17</td>
<td>5</td>
</tr>
<tr>
<td>Mah. norm (random U init)</td>
<td>39</td>
<td>12</td>
</tr>
<tr>
<td>Mah. norm (Optimal U init)</td>
<td>29</td>
<td>5</td>
</tr>
<tr>
<td>Diagonal norm (random U init)</td>
<td>25</td>
<td>6</td>
</tr>
<tr>
<td>Diagonal norm (Optimal U init)</td>
<td>25</td>
<td>3</td>
</tr>
</tbody>
</table>

Table 2: Hard c-means error results averaged over 30 random initializations for each distance metric.

<table>
<thead>
<tr>
<th></th>
<th>Final results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>errors</td>
</tr>
<tr>
<td>Euclidean norm</td>
<td>16.7</td>
</tr>
<tr>
<td>Mah. norm</td>
<td>38.9</td>
</tr>
<tr>
<td>Diagonal norm</td>
<td>31.2</td>
</tr>
</tbody>
</table>

Table 3: GA clustering results with different norms over 30 trials with Population=100 and R=50.
<table>
<thead>
<tr>
<th>Final results</th>
<th>Optimum results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>errors</td>
<td>Std. Dev.</td>
</tr>
<tr>
<td>Euclidean norm</td>
<td>15.9</td>
</tr>
<tr>
<td>Maha. norm</td>
<td>35.8</td>
</tr>
<tr>
<td>Diagonal norm</td>
<td>24.7</td>
</tr>
</tbody>
</table>

Table 4: GA clustering results with different norms over 30 trials with Population=50 and R=24.

<table>
<thead>
<tr>
<th>Final results</th>
<th>Optimum results</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>errors</td>
<td>Std. Dev.</td>
</tr>
<tr>
<td>Euclidean norm</td>
<td>16.2</td>
</tr>
<tr>
<td>Maha. norm</td>
<td>32.0</td>
</tr>
<tr>
<td>Diagonal norm</td>
<td>25.0</td>
</tr>
</tbody>
</table>

Table 5: GA clustering results with different norms over 30 trials with Population=200 and R=100.

2.1 Effects of Crossover and Mutation

In order to determine how crossover and mutation affect the clustering results, we kept the initial population the same and used different seeds for crossover and mutation to force different choices of crossover point and distance as well as mutations. Also, the mutation frequency was varied. There was no positive effect on the error and only a small effect on the $J_1$ value. This lead us to conclude that the GA clustering method is sensitive primarily to the initial population it is provided.

3 Summary

The genetic algorithm guided clustering method is sensitive to the initial population in a manner analogous to hard c-means sensitivity to its initialization. The hypothesis that the GA method of clustering on the Iris data provides better performance on average for each of the three norms holds at the 0.05 significance level under some conditions. The Mahalanobias norm requires populations of size 100 or 200 and more generations than the other norms. The hypothesis holds for the Euclidean norm for populations of 50 or 200. The performance of the GA clustering method with the Diagonal norm is clearly superior to that of HCM. In all cases, $J_1$ the minimization criterion is steadily reduced by the GA clustering method.

In general, it was found that more generations reduce the standard deviation and error regardless of the norm. Of course, more generations cause the clustering to take longer. It takes about 0.025 seconds per population element on a SUN Sparc 10 to complete a single generation or about 24 seconds for 200 generations of a population of size 50.

There are issues that remain to be solved, such as the number of generations to allow or a good stopping criterion for the clustering. $J_1$ can remain relatively flat for some time before a significant reduction is obtained, so small changes in $J_1$ do not seem to provide a reliable stopping criterion. Population size for effective clustering is another open question. Larger initial populations appear to allow better average results, though we do not believe in the case of the Iris data that a GA approach with HCM can provide much improved results. There are uninvestigated possibilities for the crossover operation.

The GA clustering approach presented here can provide a slight but significant increase in accuracy for different choices of norms. Finally, a clear advantage of the GA approach is that it can be performed on a parallel processor in a straightforward fashion. This holds the promise of faster GA clustering with no loss in accuracy.

Acknowledgements: This research partially supported by National Cancer Institute grant CA59425-01 and a grant from the Whitaker Foundation.

---

$^1$SUN and Sparc 10 are registered trademarks of SUN microsystems.
Figure 3: a) Mahalanobis error vs. generation b) Mahalanobis $J_1$ value vs. generation. Population = 50.

Figure 4: a) Euclidean error vs. generation b) Euclidean $J_1$ value vs. generation. Population = 50.

References


