Genetic Algorithms for Subset Selection in Model-Based Clustering

Luca Scrucca

Abstract Model-based clustering assumes that the data observed can be represented by a finite mixture model, where each cluster is represented by a parametric distribution. The Gaussian distribution is often employed in the multivariate continuous case. The identification of the subset of relevant clustering variables enables a parsimonious number of unknown parameters to be achieved, thus yielding a more efficient estimate, a clearer interpretation and often improved clustering partitions. This paper discusses variable or feature selection for model-based clustering. Following the approach of Raftery and Dean (J Am Stat Assoc 101(473):168–178, 2006), the problem of subset selection is recast as a model comparison problem, and BIC is used to approximate Bayes factors. The criterion proposed is based on the BIC difference between a candidate clustering model for the given subset and a model which assumes no clustering for the same subset. Thus, the problem amounts to finding the feature subset which maximises such a criterion. A search over the potentially vast solution space is performed using genetic algorithms, which are stochastic search algorithms that use techniques and concepts inspired by evolutionary biology and natural selection. Numerical experiments using real data applications are presented and discussed.

1 Introduction

In the model-based approach to clustering, each cluster is represented by a parametric distribution, and then a finite mixture model is used to model the observed data. Parameters are estimated by optimising the fit, expressed by the likelihood (eventually penalised), between the data and the model. This approach, based on probabilistic models, includes a number of advantages, such as the choice of the number of clusters.

Recently, several authors have argued that the clustering structure of interest may be contained in a subset of available variables. Law et al. [22] proposed a method for obtaining feature saliency, based on the assumption that features are

L. Scrucca (✉)
Department of Economics, Università degli Studi di Perugia, Perugia, Italy
e-mail: luca@stat.unipg.it

M.E. Celebi, K. Aydin (eds.), *Unsupervised Learning Algorithms*,
DOI 10.1007/978-3-319-24211-8_3

conditionally independent given the clustering labels. Raftery and Dean [30] recast the problem of comparing two nested variable subsets as a model selection problem. They used the Bayesian information criterion (BIC) to compare two models, by assuming that the irrelevant variables are regressed on the entire set of relevant variables. Maugis et al. [23] improved on Raftery and Dean's approach by allowing the irrelevant variables to be explained by only a subset of relevant variables. Later, [24] introduced an algorithm leading to a general variable role modelling.

Regardless of the different approaches taken to solve the problem, all the authors agree that selecting the relevant variables enables parsimony of unknown parameters to be achieved, which results in more efficient estimates, a clearer interpretation of the parameters and often improved clustering partitions.

However, the problem of selecting the best subset of clustering variables is a non-trivial exercise, especially when numerous features are available. Therefore, search strategies for model selection are required to explore the vast solution space. The classical method to search for a feasible solution is to adopt a forward/backward stepwise selection strategy. The main criticism of this approach is that stepwise searching rarely finds the best overall model or even the best subset of any size.

This paper discusses a computationally feasible approach based on genetic algorithms (`clustGAsel`), which address the potentially daunting statistical and combinatorial problems presented by subset selection in model-based clustering via finite mixture modelling.

The reminder of this article is set out as follows. Section 2 reviews the model-based approach to clustering. Section 3 discusses subset selection as a model comparison problem, and introduces a BIC-type criterion to select the relevant clustering variables. A comparison with other existing approaches is also discussed. Section 4 introduces genetic algorithms to search the potentially vast solution space. Genetic operators for a global search over any subset size and for a pre-specified subset size are presented. Section 5 illustrates empirical results based on two real data examples. The final Section provides some concluding remarks and ideas for further improvements.

2 Model-Based Clustering

2.1 Finite Mixture Modelling

Model-based clustering assumes that the observed data are generated from a mixture of G components, each representing the probability distribution for a different group or cluster [14, 26]. The general form of a finite mixture model is $f(\boldsymbol{x}) = \sum_{g=1}^{G} \pi_g f_g(\boldsymbol{x}|\boldsymbol{\theta}_g)$, where π_g represents the mixing probabilities, so that $\pi_g > 0$ and $\sum \pi_g = 1$, $f_g(\cdot)$ and $\boldsymbol{\theta}_g$ are the density and the parameters of the g-th component ($g = 1, \ldots, G$), respectively. With continuous data, we often take the density for each mixture component to be the multivariate Gaussian $\phi(\boldsymbol{x}|\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$ with

parameters $\boldsymbol{\theta}_g = (\boldsymbol{\mu}_g, \boldsymbol{\Sigma}_g)$. Thus, clusters are ellipsoidal, centred at the mean vector $\boldsymbol{\mu}_g$, with other geometric features, such as volume, shape and orientation, determined by $\boldsymbol{\Sigma}_g$. Parsimonious parameterisation of covariance matrices can be adopted through eigenvalue decomposition in the form $\boldsymbol{\Sigma}_g = \lambda_g \boldsymbol{D}_g \boldsymbol{A}_g \boldsymbol{D}_g^\top$, where λ_g is a scalar controlling the volume of the ellipsoid, $\boldsymbol{A}_g$ is a diagonal matrix specifying the shape of the density contours, and $\boldsymbol{D}_g$ is an orthogonal matrix which determines the orientation of the corresponding ellipsoid [2, 5]. Fraley et al. [15, Table 1] report some parameterisations of within-group covariance matrices available in the `MCLUST` software, and the corresponding geometric characteristics. Maximum likelihood estimates for this type of mixture models can be computed via the EM algorithm [11, 25]. A recent survey of finite mixture modelling for clustering is contained in [27].

2.2 BIC as a Criterion for Model Selection

One important aspect of finite mixture modelling is model choice, i.e. the selection of the number of mixture components and the parameterisation of covariance matrices. This is often pursued by using the Bayesian information criterion [BIC; 32]. For a recent review of BIC in model selection see [28].

From a Bayesian point of view, comparisons between models can be based on Bayes factors. Let us consider two candidate models, M_1 and M_2. The Bayes factor is defined as the ratio of the posterior odds to the prior odds:

$$B_{12} = \frac{p(M_1|\boldsymbol{X})/p(M_2|\boldsymbol{X})}{p(M_1)/p(M_2)} = \frac{p(\boldsymbol{X}|M_1)}{p(\boldsymbol{X}|M_2)}.$$

Model M_1 is favoured by the data if $B_{12} > 1$, but not if $B_{12} < 1$. When there are unknown parameters, B_{12} is equal to the ratio of the integrated likelihoods, with the integrated likelihood for model M_k (integrated over the model parameters) defined as

$$p(\boldsymbol{X}|M_k) = \int p(\boldsymbol{X}|\boldsymbol{\theta}_k, M_k) p(\boldsymbol{\theta}_k|M_k) d\boldsymbol{\theta}_k, \tag{1}$$

where $p(\boldsymbol{\theta}_k|M_k)$ is the prior distribution of the parameter vector $\boldsymbol{\theta}_k$ of model M_k. The integral in (1) is difficult to evaluate. However, assuming prior unit information, it can be approximated by

$$2 \log p(\boldsymbol{X}|M_k) \approx \mathrm{BIC}_k = 2 \log p(\boldsymbol{X}|\hat{\boldsymbol{\theta}}_k, M_k) - \nu_k \log(n),$$

where $p(\boldsymbol{X}|\hat{\boldsymbol{\theta}}_k, M_k)$ is the maximised likelihood under model M_k, ν_k is the number of independent parameters to be estimated, and n is the number of observations available in the data. Kass and Raftery [20] showed that BIC provides an approximation to the Bayes factor for comparing two competing models, i.e.

$$2 \log B_{12} = 2 \log p(\boldsymbol{X}|M_1) - 2 \log p(\boldsymbol{X}|M_2) \approx \mathrm{BIC}_1 - \mathrm{BIC}_2.$$

BIC has been widely used for mixture models, both for density estimation [31] and for clustering [13]. Keribin [21] showed that BIC is consistent for choosing the number of components in a mixture model, under the assumption that the likelihood is bounded. This may not be true in general of Gaussian mixture models (GMM), but it does hold if, for instance, the variance is bounded below, a constraint which is imposed in practice in the `MCLUST` software [15]. The use of BIC for model selection can also be seen as a way to penalise the likelihood based on model complexity. Finally, we note that other criteria have been proposed, such as the integrated complete-data likelihood (ICL) criterion [4].

3 Subset Selection in Model-Based Clustering

Raftery and Dean [30] discussed the problem of variable selection for model-based clustering by recasting the problem as a model selection problem. Their proposal is based on the use of BIC to approximate Bayes factors to compare mixture models fitted on nested subsets of variables. A generalisation of this approach was recently discussed by Maugis et al. [23, 24].

Let us suppose the set of available variables is partitioned into three disjoint parts: the set of previously selected variables, $\boldsymbol{X}_1$; the set of variables under consideration for inclusion or exclusion from the active set, $\boldsymbol{X}_2$; and the set of the remaining variables, $\boldsymbol{X}_3$. Raftery and Dean [30] showed that the inclusion (or exclusion) of variables can be assessed by using the Bayes factor

$$B_{12} = \frac{p(\boldsymbol{X}_2|\boldsymbol{X}_1, M_1)p(\boldsymbol{X}_1|M_1)}{p(\boldsymbol{X}_1, \boldsymbol{X}_2|M_2)}, \tag{2}$$

where $p(\cdot|M_k)$ is the integrated likelihood of model M_k ($k = 1, 2$). Model M_1 specifies that given $\boldsymbol{X}_1$, $\boldsymbol{X}_2$ is conditionally independent of the cluster membership, whereas model M_2 specifies that $\boldsymbol{X}_2$ is relevant for clustering, once $\boldsymbol{X}_1$ has been included in the model. An important aspect of this formulation is that set $\boldsymbol{X}_3$ of remaining variables plays no role in (2). Minus twice the logarithm of the Bayes factor in Eq. (2) can be approximated by the following BIC difference:

$$\text{BIC}_{\text{diff}} = \text{BIC}_{\text{clust}}(\boldsymbol{X}_1, \boldsymbol{X}_2) - \text{BIC}_{\text{not clust}}(\boldsymbol{X}_2|\boldsymbol{X}_1), \tag{3}$$

where $\text{BIC}_{\text{clust}}(\boldsymbol{X}_1, \boldsymbol{X}_2)$ is the BIC value for the "best" clustering mixture model fitted using both $\boldsymbol{X}_1$ and $\boldsymbol{X}_2$ features, whereas $\text{BIC}_{\text{not clust}}(\boldsymbol{X}_2|\boldsymbol{X}_1)$ is the BIC value for no clustering for the same set of variables. Large, positive values of BIC_{diff} indicate that $\boldsymbol{X}_2$ variables are relevant for clustering.

Raftery and Dean [30] adopted a stepwise greedy search algorithm to evaluate the inclusion or exclusion of a single feature from the already included set of variables. They show that the second term in the right-hand side of Eq. (3) can be written as

$$\text{BIC}_{\text{not clust}}(X_2|\boldsymbol{X}_1) = \text{BIC}_{\text{clust}}(\boldsymbol{X}_1) + \text{BIC}_{\text{reg}}(X_2|\boldsymbol{X}_1), \tag{4}$$

i.e. the BIC value for the "best" clustering model fitted using $\boldsymbol{X}_1$ plus the BIC value for the regression of the candidate X_2 variable on the $\boldsymbol{X}_1$ variables. In all cases, the "best" clustering model is identified with respect to the number of mixture components (assuming $G \geq 2$) and to model parameterisations. In the original proposal, the variable X_2 in the linear regression model term was assumed to depend on all the variables in $\boldsymbol{X}_1$. However, it may depend only on subset of them, or none (complete independence). To accommodate these situations, a subset selection step in the regression model has been proposed [23]. This approach is implemented in the R package `clustvarsel` [36].

The adopted stepwise algorithm is known to be suboptimal, as there is a risk of finding only a local optimum in the model space. To overcome this drawback and increase the chance of global optimisation, we propose the use of genetic algorithms to search the entire model space. However, moving on from the stepwise perspective, we need to reformulate the problem, as discussed in the next subsection.

3.1 The Proposed Approach

Let us suppose the set of variables previously included for clustering is empty and we want to evaluate the clustering model obtained from the candidate subset $\boldsymbol{X}_k$ of dimension $k \leq p$. The Bayes factor for comparing a candidate clustering model (M_k), fitted using the subset of k variables $\boldsymbol{X}_k$, against the no clustering model (M_0) on the same set of variables, is given by

$$B_{k0} = \frac{p(\boldsymbol{X}_k|M_k)}{p(\boldsymbol{X}_k|M_0)}.$$

Using the same arguments discussed previously, we can approximate the above ratio by the following BIC difference:

$$\text{BIC}_k = \text{BIC}_{\text{clust}}(\boldsymbol{X}_k) - \text{BIC}_{\text{not clust}}(\boldsymbol{X}_k). \tag{5}$$

Thus, the goal is to maximise the difference between the maximum BIC value from the finite mixture model for clustering, i.e. assuming $G \geq 2$, and the maximum BIC value for no clustering, i.e. assuming $G = 1$, with both models estimated on the candidate subset of variables $\boldsymbol{X}_k$.

By evaluating the criterion in Eq. (5) for a large number of candidate subsets of different sizes, we may choose the "best" subset as being the one which provides the largest BIC_k. However, the number of possible subsets of size k from a total of p variables is equal to $\binom{p}{k}$. Thus, the space of all possible subsets of size k, ranging from 1 to p, has a number of elements equal to $\sum_{k=1}^{p}\binom{p}{k} = 2^p - 1$. An exhaustive search would not be feasible, even for moderate values of p. Genetic algorithms are a

natural candidate for searching this potentially very large model space. In Sect. 4, we discuss the application of genetic algorithms to address this problem. Two strategies are introduced to deal with (a) the general case of subset size not specified a priori, and (b) the case of a fixed subset size.

3.2 Models for No Clustering

When a no clustering model is fitted, an eigen decomposition of the marginal covariance matrix, $\boldsymbol{\Sigma} = \lambda \boldsymbol{D}\boldsymbol{A}\boldsymbol{D}^\top$, can be adopted. This yields three possible models: (1) spherical, $\lambda \boldsymbol{I}$; (2) diagonal, $\lambda \boldsymbol{A}$; and (3) full covariance. The numbers of estimated parameters are equal to 1, k, and $k(k-1)/2$, respectively. Although these models include many interesting situations, there are some which are not included, but which may be useful in practical applications. For example, based on the above parameterisation, variables can be either all uncorrelated (case 1 and 2), or all correlated (case 3). However, a variable is often correlated with a subset of other variables, yet at the same time it can be nearly uncorrelated with the remainder. A simple strategy, although computationally demanding in high-dimensional cases, consists of calculating the BIC for each possible configuration of null covariance between pairs of variables. This would allow us to identify only a subset of correlated features for each variable, whereas the remaining features are left uncorrelated.

4 Genetic Algorithms

Genetic algorithms (GAs) are stochastic search algorithms, based on concepts of biological evolution and natural selection [18], which have been applied to find exact or approximate solutions to optimisation and search problems [16, 17].

A GA begins with a set of randomly generated individuals, or solutions, called the population. A fitness or performance is assigned to each individual in the population. The choice of the fitness function depends on the problem at hand. In optimisation problems, it is usually the objective function which needs to be minimised or maximised. A new population is formed by applying specific genetic operators. With the *selection* operator, individuals are selected to form new offspring according to their fitness value. Other genetic operators, such as *crossover* (by exchanging substrings of two individuals to obtain a new offspring) and *mutation* (randomly mutates individual bits), are applied probabilistically to the selected offspring to produce a new population of individuals. The new population is then used in the next iteration of the algorithm.

A GA is thus based on a sequence of cycles of evaluation and genetic operations, which are iterated for many generations. Generally, the overall fitness of the population improves, and better solutions are likely to be selected as the

search continues. The algorithm usually terminates when any one of the following conditions is fulfilled: (1) a maximum number of generations has been produced, (2) a satisfactory fitness level has been reached, and (3) the algorithm has achieved a steady state.

Note that a GA is a stochastic search method. There is no guarantee, therefore, that the algorithm will find a solution in a given specific case. However, the algorithm can, on average, be reasonably expected to converge. Another popular stochastic algorithm is simulated annealing (SA), which resembles a GA in that it has one individual in the population, no crossover, and a decreasing mutation rate. SA may find local improvements more efficiently, but it is not efficient in exploring large solution spaces. On the contrary, GAs represent a very effective way of searching through vast, complex solution spaces and are, therefore, more suitable for subset selection [7, 38].

4.1 GAs for Subset Selection in Model-Based Clustering

In this section we describe specific points of our use of GAs for subset selection in model-based clustering.

4.1.1 Genetic Coding Scheme

Each subset of variables is encoded as a string, where each locus in the string is a binary code indicating the presence (1) or absence (0) of a given variable. For example, in a model-based clustering problem with $p = 5$ variables, the string 11001 represents a model where variables 1, 2, and 5 are included, whereas variables 3 and 4 are excluded from the model.

4.1.2 Generation of a Population of Models

The population size N (i.e. the number of models fitted at each generation) is an important parameter of the GA search. A sufficiently large set of models ensures that a large portion of the search space is explored at each step. On the other hand, if there are too many individuals (i.e. models) in the population, GA tends to slow down. After a certain limit, which depends on the encoding and the optimisation problem, increasing population size is not beneficial. Since there are $2^p - 1$ possible subsets for p variables, we set $N = \min(2^p - 1, 50)$ by default.

4.1.3 Fitness Function to Evaluate the Model Clustering

The BIC criterion in (5) is used to assign a fitness value to each model, which corresponds to an individual of the GA population.

4.1.4 Genetic Operators

They are at the core of any implementation of GAs, and typically include the following:

- *Selection*: this operator does not create any new solution. Instead, it selects models for mating, based on their fitness. The basic idea is that solutions with improved fitness have a higher probability of surviving. Thus, relatively good solutions from a population are likely to be selected, whereas it is highly probable that not-so-good solutions will be deleted. We adopted a *linear rank selection* scheme from among the many possible schemes [see 1, Chaps. 22–29] to assign a probability to each model. For a set of models forming the current GA population, let o_i ($i = 1, \ldots, N$) be the rank obtained from their arrangement in non-increasing order, based on their fitness. Then, a new population is randomly extracted with probability of selection equal to $p_i = 2/N - 2/(N(N-1)) \times (o_i - 1)$. This ensures better models have a higher chance of being included in the next generation, but at the same time it does not impose too much pressure to avoid premature convergence.
- *Crossover*: this operator creates new offspring by means of the recombination of genetic material from their parents. Let p_c be the probability that two random strings (parents) are selected from the mating pool to generate a child. In *single point crossover*, one crossover point is selected, binary values from beginning of the string to the crossover point are copied from one parent, and the remainder are copied from the second parent. By default, we set $p_c = 0.8$.
- *Mutation*: this operator introduces random mutations in the population to ensure the search process can move to other areas of the search space. With probability p_m a randomly selected locus of a string can change from 0 to 1 or from 1 to 0. Thus, a randomly selected variable is either added to or removed from the model. The mutation probability is usually small, and we set $p_m = 0.1$ by default.
- *Elitism*: to improve the performance of GAs, a number of particularly fit individuals may survive through generations. By default the model with the best fitness is retained at each iteration.

4.2 Computational Issues

The computational effort needed by GAs mainly depends on the dimension of the search space, its complexity and the time required to calculate a fitness value for every individual of the population. In our case, this final step involves selecting

the "best" clustering model for a given candidate subset of variables, which is a time-consuming process. However, since a portion of the search space has probably already been explored at each iteration, we can avoid re-evaluating the models already estimated, by saving new string solutions and the corresponding fitness values during the iterations. This may save a great deal of computing time. Another possibility would be to use parallel computing on a computer cluster (more on this point is included in the final Section).

4.3 Random-Key GAs to Select a Fixed Size Subset

Let us suppose we are interested in finding the "best" clustering model using a fixed size subset of the most relevant variables, e.g. $k = 3$. The genetic coding scheme discussed in Sect. 4.1 is likely to produce illegitimate solution strings since no constraints are imposed on the size of active variables. Therefore, we must employ a different genetic encoding which ensures GAs only search the space of feasible solutions.

Bean [3] introduced random-key GAs, which guarantee each string corresponds to a feasible encoding of a solution. In practice, random-key GAs indirectly explore the solution space, by searching the space of random keys and using the decoder to evaluate the fitness of the random key. Let $(u_1, \ldots, u_N)$ be a random sequence of N values from $\mathscr{U}(0, 1)$, and $(r_1, \ldots, r_N)$ be the corresponding ranks. Then, the solution string $(s_1, \ldots, s_N)$, with generic element $s_i = I(r_i > N - k)$, where $I(\cdot)$ is the indicator function, guarantees the resulting binary string has exactly k active variables (those for which $s_i = 1$) and $N - k$ non-active variables (those for which $s_i = 0$).

The adoption of the random-key encoding function always produces feasible solutions, but also mating will produces feasible offspring. As a result, the genetic operators for selection and crossover described in the previous section can be directly applied over the space of random-keys. Nevertheless, other genetic operators could also be adopted. In our implementation, for instance, we use *uniform constrained crossover*, which generates a child by selecting at random a value for each locus from the parents, i.e. each bit of the child string is given by $u\text{Parent}_1 + (1 - u)\text{Parent}_2$, where $u \in \{0, 1\}$ is drawn at random. Finally, a small modification of the mutation operator is required. A randomly selected locus of a string can be substituted here with a random value extracted from $\mathscr{U}(0, 1)$.

5 Data Examples

In this section, we present some examples based on real data applications. The clustering partition obtained by the subset of selected variables is evaluated by calculating the classification error and the adjusted Rand index [ARI; 19]. The latter

is a measure of agreement between two data partitions, whose expected value is zero when two random partitions are compared, and it achieves the maximum value of one when two partitions perfectly coincide.

5.1 Birds, Planes and Cars

Cook and Forzani [8] analysed a dataset from a pilot study to evaluate the possibility of distinguishing birds, planes, and cars on the basis of the sounds they emit. The sample consisted of 58 recordings identified as birds, 43 as cars, and 64 as planes. Each recording was processed and represented by 13 SDMFCCs (Scale-Dependent Mel-Frequency Cepstrum Coefficients). Further details can be found in the previously cited paper.

We focus here on the clustering of records based on the extracted standardised sound characteristics. A GMM fitted on all the available features selected a (VVV,3) model, i.e. a model with three components and different full covariance matrices within clusters. The accuracy was good with an ARI equal to 0.74.

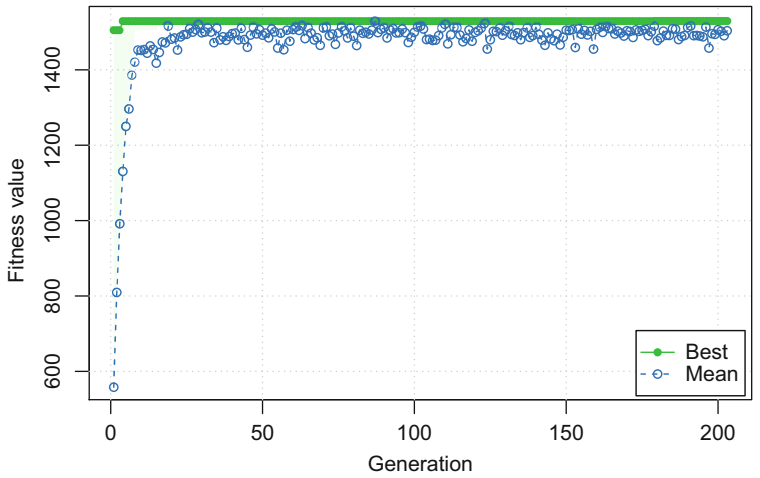
Then, we investigated the possibility of improving the accuracy of the final partition by using a subset of sonic features. Figure 1 shows the GA search path using the BIC_k criterion in (5) as the fitness function. The green filled points represent the best fitness values at each generation, whereas the blue open circle points represent the corresponding fitness averages. Every generation consists of 50 binary strings, each of which represents a candidate feature subset. Other tuning parameters for genetic operators can be read from the output on the right-hand side of the graph. The algorithm soon achieves the optimal solution, and it terminates once there are no improvements in 200 consecutive iterations.

The subset selected by the GA approach has 11 sonic features. The corresponding GMM fitted on this subset has again the (VVV,3) structure. However, by removing two features from the complete set, we improve clustering accuracy, achieving an ARI of 0.84.

To show the clustering obtained with the selected subset, we may project the data using the methodology proposed by Scrucca [33], and recently extended to show the maximal separation of clusters in [35]. The resulting graph is reported in Fig. 2, where we can see that cars appear to be well separated from the other two groups (in fact, no cars are misclassified), but planes and birds overlap and they are clearly more difficult to separate.

5.2 Italian Wines

Forina et al. [12] reported data on 178 wines grown in the same region in Italy, yet obtained from three different cultivars (Barolo, Grignolino, and Barbera). A total of 13 measurements of chemical and physical properties were made for each wine,




```
+--------------------------------------------+
|          Genetic Algorithm for             |
| Model-based Clustering Variable Selection  |
+--------------------------------------------+

GA settings:
Type                  =  binary
Population size       =  50
Number of generations =  1000
Elitism               =  1
Crossover probability =  0.8
Mutation probability  =  0.1

GA results:
Iterations             = 203
Fitness function value = 1529.475
Selected subset        =
(X1, X3, X4, X5, X6, X7, X8, X9, X10, X11, X13)
```

Fig. 1 Summary of GA search applied to the birds–planes–cars data example

such as the level of alcohol, the level of magnesium, the colour intensity, etc. The dataset is available at the UCI Machine learning data repository http://archive.ics.uci.edu/ml/datasets/Wine.

Often, a preliminary step in cluster analysis is to perform a principal components (PCs) analysis, followed by a clustering algorithm on the features obtained. The first few PCs are usually retained, but, as discussed by Chang [6], these may not contain the most important information for clustering. Here we present an analysis based on random-key GAs, aimed at finding the best subset of PCs.

Table 1 reports the results from the application of random-key GAs to select subsets of PCs of various sizes, where the PCs are obtained from standardised variables. The subset with the largest BIC_k is the one with five PCs: the first two, the 5th, the 6th, and the last PCs. The Gaussian mixture model fitted using the best subset found enables us to correctly estimate the number of components which, in turn, yields a large adjusted Rand index and a small error rate. Another potentially

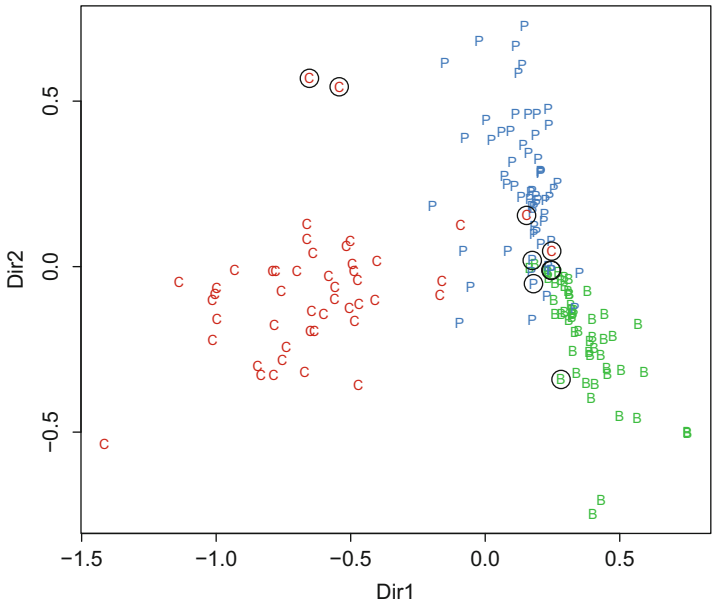
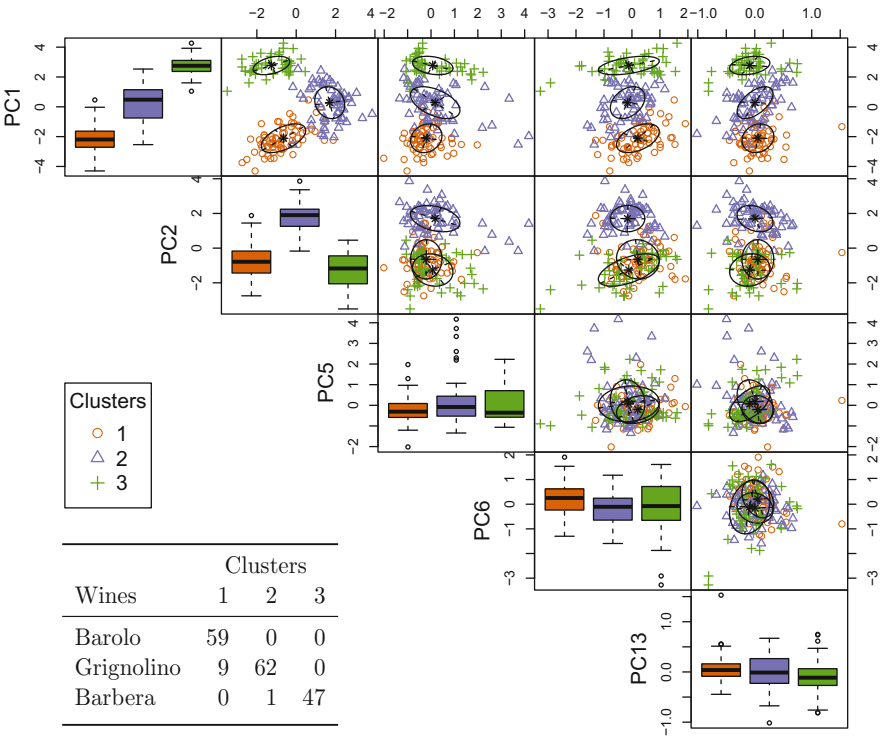


Fig. 2 Plot of data projected along the first two GMMDR directions for the birds–planes–cars data example. Points are marked according to the prevalent cluster membership: P = planes, C = cars, and B = birds. The misclassified points are identified by an *open circle*

Table 1 Results for random-key GAs for the selection of PCs of increasing subset size applied to Italian wine data (large values of BIC are preferable)

Size (k)	PCs subset	BIC_k	Model	G	Error (%)	ARI
1	5	45.04	V	2	60.11	0.0135
2	1 2	173.22	EEV	4	17.42	0.7099
3	1 2 6	200.77	EEV	3	15.73	0.5893
4	1 2 5 7	202.52	EII	7	31.46	0.5536
5	1 2 5 6 13	218.06	EEV	3	5.62	0.8300
6	1 2 3 5 6 13	213.38	VEV	3	1.12	0.9637
7	1 2 3 5 6 7 13	207.03	VEI	6	26.40	0.6722
8	1 2 3 5 6 7 11 13	193.78	VEI	6	26.40	0.6661
9	1 2 3 4 5 7 10 11 13	181.05	VEI	6	32.58	0.6040
10	1 2 3 4 5 6 7 8 10 13	175.27	VEI	5	17.42	0.7602
11	1 2 3 4 5 7 8 9 10 11 13	156.70	VEI	5	17.98	0.7394
12	1 2 3 4 5 6 7 8 10 11 12 13	128.60	VEI	4	16.85	0.7470
13	1 2 3 4 5 6 7 8 9 10 11 12 13	110.92	VEI	4	17.98	0.7372

interesting subset, with a BIC_k value very close to the maximum, is the subset which adds the third PC to the previous features. This enables us to achieve the largest adjusted Rand index and the smallest error rate. Note that, in general, the best



Wines	Clusters 1	2	3
Barolo	59	0	0
Grignolino	9	62	0
Barbera	0	1	47

Fig. 3 Scatterplot matrix of PCs selected by random-key GAs with points marked according to the clustering obtained by the estimated GMM

subset of a given size k does not contain the first k PCs, and the selected subsets for increasing sizes do not necessarily form a hierarchy. Figure 3 shows the scatterplot matrix of the selected PCs with points marked in accordance with the predicted clusters partition. The estimated GMM allows us to correctly identify the Barolo wines, and all but one of the Barbera wines. The Grignolino appears to be the most difficult wine to classify, with nine cases assigned to the Barolo cluster.

A comparison with the `clustGAsel` approach discussed in this paper is reported in Table 2. We considered the original greedy search proposed by Raftery and Dean [30] as implemented in the R package `clustvarsel` version 1 [9], and the updated version 2 of the package [10] which includes a subset selection procedure also for the regressors [36]. Furthermore, we considered the methods of Maugis et al. [23, 24] as implemented, respectively, in the software `SelvarClust` and `SelvarClustIndep`. By looking at the results in Table 2, we can notice that `clustvarsel` ver. 1 selects all the PCs. This is due to the orthogonality of PCs that leads to overestimate the relevance of each component at the inclusion step. The improved ver. 2 of `clustvarsel` identifies a subset of five PCs, which are the same components selected by `clustGAsel` with the exception of the 7th PC that

Table 2 A comparison of subset selection procedures using different algorithms for the identification of relevant PCs for clustering using the Italian wine data

Algorithm	PCs subset	BIC_k	Model	G	Error (%)	ARI
`clustvarsel` ver. 1	1–13	110.92	VEI	4	17.98	0.7372
`clustvarsel` ver. 2	1 2 5 7 13	214.08	EEI	6	19.66	0.7279
`SelvarClust`	1 2 4 5 6 13	199.96	VEV	3	8.99	0.7410
`SelvarClustIndep`	1 2 5 6 13	207.11	VEV	3	7.30	0.7843
`clustGAsel`	1 2 5 6 13	218.06	EEV	3	5.62	0.8300

is included instead of the 6th PC. The BIC_k criterion is equal to 214.08, only slightly smaller than the highest value we found (218.06), but the final GMM has six mixture components. This yields an ARI = 0.7279, thus a less accurate clustering partition. The `SelvarClust` algorithm selected a six-components subset, with the 4th PC added to the best subset found by `clustGAsel`. The number of clusters is correctly identified, but the final partition is only slightly better than that provided by the model with all the PCs included (ARI = 0.7410). Furthermore, a better subset may be found for the size $k = 6$, as shown in Table 1. Finally, the `SelvarClustIndep` algorithm selected the same PCs identified by `clustGAsel`, the same number of clusters, but a different GMM model. The accuracy of the final partition is better than that obtained by previous methods (ARI = 0.7843), but worse than the accuracy provided by `clustGAsel` (ARI = 0.83).

6 Conclusions

In multivariate datasets, variables are often highly correlated with each other or do not carry much additional information about clustering. The performance of clustering algorithms can be severely affected by the presence of those variables which only serve to increase dimensionality, while adding redundant information. The elimination of such variables can potentially improve both the estimation and the clustering performance. This aspect of model selection has recently been discussed in literature.

In this paper, we have addressed this problem by means of genetic algorithms, using a BIC-type criterion as the fitness function for subset selection in model-based clustering. The criterion adopted generalises the method introduced by Raftery and Dean [30] by allowing a subset of clustering variables to be uncorrelated with the irrelevant features. The same problem was also addressed by Maugis et al. [23, 24] who proposed the use of a different criterion and a stepwise search.

The improvement in computational efficiency of GAs is clearly an area for further research. Considerable savings in elapsed time could be gained by evaluating each proposed model in parallel. A simple form of parallelism could be based on a master process which runs all the genetic operators, with the exception of the

evaluation of the fitness function. The calculation of the BIC-type criterion in (5) for each individual of the current GA population would run in parallel, by worker processes operating on separate processors. Results from the worker processes could then be sent to the master, which would collect all the results before continuing with the next generation. There are some important issues in this standard master–slave parallelisation model, such as fault tolerance and reproducibility, which must be taken into account. See [37] for a discussion on these aspects.

The algorithms presented in this paper have been implemented using the `GA` package [34] for the open source statistical computing environment R [29].

References

1. Back, T., Fogel, D.B., Michalewicz, Z.: Evolutionary Computation 1: Basic Algorithms and Operators. IOP Publishing, Bristol and Philadelphia (2000)
2. Banfield, J., Raftery, A.E.: Model-based Gaussian and non-Gaussian clustering. Biometrics **49**, 803–821 (1993)
3. Bean, J.C.: Genetic algorithms and random keys for sequencing and optimization. ORSA J. Comput. **6**(2), 154–160 (1994)
4. Biernacki, C., Celeux, G., Govaert, G.: Assessing a mixture model for clustering with the integrated completed likelihood. IEEE Trans. Pattern Anal. Mach. Intell. **22**(7), 719–725 (2000)
5. Celeux, G., Govaert, G.: Gaussian parsimonious clustering models. Pattern Recogn. **28**, 781–793 (1995)
6. Chang, W.C.: On using principal components before separating a mixture of two multivariate normal distributions. Appl. Stat. **32**(3), 267–275 (1983)
7. Chatterjee, S., Laudato, M., Lynch, L.A.: Genetic algorithms and their statistical applications: an introduction. Comput. Stat. Data Anal. **22**, 633–651 (1996)
8. Cook, D.R., Forzani, L.: Likelihood-based sufficient dimension reduction. J. Am. Stat. Assoc. **104**(485), 197–208 (2009)
9. Dean, N., Raftery, A.E.: clustvarsel1: variable selection for model-based clustering. (2009). http://CRAN.R-project.org/package=clustvarsel, R package version 1.3
10. Dean, N., Raftery, A.E., Scrucca. L.: clustvarsel: variable selection for model-based clustering. (2014). http://CRAN.R-project.org/package=clustvarsel, R package version 2.1
11. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum likelihood from incomplete data via the em algorithm (with discussion). J. R. Stat. Soc. Ser. B Stat. Methodol. **39**, 1–38 (1977)
12. Forina, M., Armanino, C., Castino, M., Ubigli, M.: Multivariate data analysis as a discriminating method of the origin of wines. Vitis **25**, 189–201 (1986). ftp://ftp.ics.uci.edu/pub/machine-learning-databases/wine, wine Recognition Database
13. Fraley, C., Raftery, A.E.: How many clusters? Which clustering method? Answers via model-based cluster analysis. Comput. J. **41**, 578–588 (1998)
14. Fraley, C., Raftery, A.E.: Model-based clustering, discriminant analysis, and density estimation. J. Am. Stat. Assoc. **97**(458), 611–631 (2002)
15. Fraley, C., Raftery, A.E., Murphy, T.B., Scrucca, L.: MCLUST version 4 for R: Normal mixture modeling for model-based clustering, classification, and density estimation. Technical Report 597, Department of Statistics, University of Washington (2012)
16. Goldberg, D.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Professional, Boston, MA (1989)
17. Haupt, R.L., Haupt, S.E.: Practical Genetic Algorithms, 2nd edn. Wiley, New York (2004)
18. Holland, J.H.: Genetic algorithms. Sci. Am. **267**(1), 66–72 (1992)

19. Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**, 193–218 (1985)
20. Kass, R.E., Raftery, A.E.: Bayes factors. J. Am. Stat. Assoc. **90**, 773–795 (1995)
21. Keribin, C.: Consistent estimation of the order of mixture models. Sankhya Ser. A **62**(1), 49–66 (2000)
22. Law, M.H.C., Figueiredo, M.A.T., Jain, A.K.: Simultaneous feature selection and clustering using mixture models. IEEE Trans. Pattern Anal. Mach. Intell. **26**(9), 1154–1166 (2004)
23. Maugis, C., Celeux, G., Martin-Magniette, M.L.: Variable selection for clustering with gaussian mixture models. Biometrics **65**(3), 701–709 (2009)
24. Maugis, C., Celeux, G., Martin-Magniette, M.L.: Variable selection in model-based clustering: a general variable role modeling. Comput. Stat. Data Anal. **53**(11), 3872–3882 (2009)
25. McLachlan, G.J, Krishnan, T.: The EM Algorithm and Extensions, 2nd edn. Wiley, Hoboken, NJ (2008)
26. McLachlan, G.J., Peel, D.: Finite Mixture Models. Wiley, New York (2000)
27. Melnykov, V., Maitra, R.: Finite mixture models and model-based clustering. Stat. Surv. **4**, 80–116 (2010)
28. Neath, A.A., Cavanaugh, J.E.: The Bayesian information criterion: background, derivation, and applications. Wiley Interdiscip. Rev. Comput. Stat. **4**(2),199–203 (2012). doi:10.1002/wics.199
29. R Core Team (2014) R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria. http://www.R-project.org/
30. Raftery, A.E., Dean, N.: Variable selection for model-based clustering. J. Am. Stat. Assoc. **101**(473), 168–178 (2006)
31. Roeder, K., Wasserman, L.: Practical bayesian density estimation using mixtures of normals. J. Am. Stat. Assoc. **92**(439), 894–902 (1997)
32. Schwartz, G.: Estimating the dimension of a model. Ann. Stat. **6**, 31–38 (1978)
33. Scrucca, L.: Dimension reduction for model-based clustering. Stat. Comput. **20**(4), 471–484 (2010). doi:10.1007/s11222-009-9138-7
34. Scrucca, L.: GA: A package for genetic algorithms in R. J. Stat. Softw. **53**(4), 1–37 (2013). http://www.jstatsoft.org/v53/i04/
35. Scrucca, L.: Graphical tools for model-based mixture discriminant analysis. Adv. Data Anal. Classif. **8**(2), 147–165 (2014)
36. Scrucca, L., Raftery, A.E.: Clustvarsel: A package implementing variable selection for model-based clustering in R http://arxiv.org/abs/1411.0606. J. Stat. Soft. Available at http://arxiv.org/abs/1411.0606 (2014, submitted)
37. Ševčíková, H.: Statistical simulations on parallel computers. J. Comput. Graph. Stat. **13**(4), 886–906 (2004)
38. Winker, P., Gilli, M.: Applications of optimization heuristics to estimation and modelling problems. Comput. Stat. Data Anal. **47**(2), 211–223 (2004)