UNIVERSIDAD CARLOS III DE MADRID working papers

Working Paper 14-08
Statistics and Econometrics Series (04)
March 2014

Departamento de Estadística
Universidad Carlos III de Madrid
Calle Madrid, 126
28903 Getafe (Spain)
Fax (34) 91 624-98-48

RECOMBINING PARTITIONS FROM MULTIVARIATE DATA: A CLUSTERING METHOD BASED ON BAYES FACTORS

Adolfo Álvarez[(1)], Daniel Peña[(2)]

Abstract

We introduce SAGRA (Split And Group Recombining Algorithm), a cluster analysis methodology which split the data set into small homogeneous groups and later recombine those groups using Bayes factors. We compare the performance of SAGRA with other three cluster analysis algorithms: SAR, M-clust and K-means, using five quality measures: Purity, number of groups, Rand index, adjusted Rand index, and F1, over four different data configurations. Results indicate that the SAGRA algorithm obtain consistently similar or better indexes than the other algorithms over all measures and data configurations.

Keywords: Cluster analysis, Bayes factors, SAR, SAGRA.

(1) Álvarez, Adolfo, Departamento de Estadística, Universidad Carlos III de Madrid, C/ Madrid 126, 28903 Getafe (Madrid), Spain, e-mail: aaapinto@est-econ.uc3m.es.
(2) Peña, Daniel, Departamento de Estadística, Universidad Carlos III de Madrid, C/ Madrid 126, 28903 Getafe (Madrid), Spain, e-mail: dpena@est-econ.uc3m.es.

Work partially supported by Spanish Ministry of Science and Innovation, research projects SEJ2007-64500 and ECO2012-38442.

1. Introduction

This article deals with the problem of recombining partitions in multivariate data. In this context, we present a new clustering methodology which, based in a strategy of splitting, cleaning, and recombining, is able to detect groups inside a data sample. As splitting rule we use the discriminator function, where points sharing the same discriminator are classified into the same group, defining in this way a partition of the sample. For cleaning we detect and purge the outliers of each group, and finally for recombining, we propose the use of a Bayes factor to weight the likelihood of the sample given two models: one where all data is generated from a single distribution, and other when the distribution is a mixture estimated from the obtained partition.

We follow the same split and recombine approach as the SAR algorithm for exploratory data analysis proposed by Peña et al. (2004), which split the sample using an heterogeneity measure based in the Mahalanobis distance, to later enlarge the resulting small groups one data point at a time to form several possible data configurations. Nevertheless, we modify the splitting and recombining processes from the SAR algorithm, incorporating to the splitting an outlier detection process which tries to avoid mixing observations from different clusters in the same basic group. Second, in the recombining process we merge the groups obtained in the splitting, while the SAR test each observation to be incorporated to a group. In this way, we use the information given by those original partitions, increasing the efficiency of the procedure, and obtaining only one data configuration as an output.

The split and recombine methodology has been followed by several authors in cluster analysis. In fact, classical methods as k-means proposed by MacQueen (1967) can be considered as a "split and recombine" method, although the split process is based only in a few observations that are considered the starting points for the aggregation. Some algorithms take as input the partition process from outside procedures as those which are focused on recombining normal samples, particularly useful when a mixture of normal distributions, as those obtained by the M-clust algorithm (Fraley and Raftery, 2002), overestimates the real number of clusters in a sample. For example, Tantrum et al. (2003) propose the use of the dip test of unimodality (Hartigan and Hartigan, 1985) to recombine such mixtures, and Baudry et al. (2010) propose the use of the Integrated Completed Likelihood (ICL) criteria, established by Biernacki et al. (2000) where is assumed than a non-observable component containing the assigning labels of the data to

the groups can be incorporated to the likelihood. This criteria penalize the BIC by the Mean Entropy leading to a smaller number of components than BIC.

Beyond M-clust, a set of methods to merge Gaussian distributions based on misclassification probabilities are proposed by Hennig (2010a). The first is based on the Bhattacharaya distance defined by Fukunaga (1990), which measures the Bayes misclassification probability between two distributions. The second is called "DEMP method" and uses directly estimated misclassification probabilities, being these probabilities given by the EM algorithm. The last one is the "prediction strength method" where the misclassification is calculated splitting the data in two halves and use one half to predict the cluster membership of the second half.

More recent approaches to the problem of merging Gaussian components can be found in literature, including a topology based methodology using manifolds (Hennig, 2010b), averaging the clustering results of several models (Wei and McNicholas, 2012), applying k-means over the components means (Li, 2005), or measuring the Kullback-Leibler divergence between two distributions (Popović et al., 2012). A deeper review of this topic can be found in Hennig (2010a).

Atkinson and Riani (2007) follow a similar approach than Peña et al. (2004) in their clustering proposal. The authors use a forward search starting from random subsets of the data sample and calculate robust Mahalanobis distances between each element and the initial subset. If the subset is of size m, in each step the starting group is enlarged in one element by selecting the m+1 smaller distances, recalculating again the distances until all data sample is included. By plotting the Mahalanobis distances is possible to identify the original groups of the sample observe the peaks produced in the distances when observations belonging to different clusters are added to the subset.

A more recent clustering methodology using a split and recombine approach can be found in Fraiman et al. (2011) who propose an algorithm inspired in the CART technique for supervised classification problems (Classification and Regression Trees, Breiman et al., 1984). The process is done defining a nodes structure starting with one node with all data set and successively splitting the space where the data set lays, perpendicularly to the axes of each dimension of the data set, conforming a binary tree. In a second stage, a merging process is done via combining the different nodes based on distances between each pair of nodes, and setting an expected number of

clusters, or a cut-off as a stop rule.

Casella and Fuentes (2009) propose a different approach to detect clusters inside a data sample. They establish a test for the hypothesis $H_0 : \kappa = 1$ vs. $H_1 : \kappa = k$, where κ represents the number of clusters. Our procedure is in the same direction, so we will review it in deep.

Let $X = X_1, X_2, ..., X_n$ be the data sample, where each $X_i, i = 1, ..., n$ is an element of p-dimensions. Then a partition ω_k is a n-dimension vector which assigns each element of the sample X to one of the k groups, representing a way to cluster n elements into k groups.

For example, when $n = 3$, we have $X = X_1, X_2, X_3$ and the set of possible partitions are for $k = 1 : \{(X_1, X_2, X_3)\}$, leading to $\omega_1 = \{(1, 1, 1)\}$ for $k = 2 : \{(X_1, X_2), (X_3)\}, \{(X_1, X_3), (X_2)\}, \{(X_1), (X_2, X_3)\}$, so $\omega_2 \in \{(1, 1, 2), (1, 2, 1), (1, 2, 2)\}$ and for $k = 3 : \{(X_1), (X_2), (X_3)\}$, there is only one possible partition $\omega_3 = \{(1, 2, 3)\}$.

The number of ways to divide a set of n objects into k non-empty subsets, is called the "Stirling number of the second kind", and can be calculated as:

$$S_{n,k} = \frac{1}{k!} \sum_{j=0}^{k-1} (-1)^j \binom{k}{j} (k - j)^n. \tag{1}$$

From Equation (1) is clear that the Stirling number of the second kind grows exponentially, even with relatively small sizes of n and k. For example $S_{20,3} = 580,606,446$. For more details, a complete review of this measure can be found in Moll (2012).

Casella and Fuentes test is based in a Bayes factor associated with the null and alternative hypotheses as given by Equation (2), where $m(X|\kappa = k)$ represents the marginal of the likelihood of the data, X, given that there are k clusters.

$$BF = \frac{m(X|\kappa = k)}{m(X|\kappa = 1)} \tag{2}$$

Considering the total number of all partitions, given by $S_{n,k}$, the Bayes factor can be written as:

$$BF = \sum_{\omega \in S_{n,k}} \frac{m(X|\omega)}{m(X|\omega_1)} \frac{\pi(\omega)}{\pi(\omega_1)} \tag{3}$$

where $\pi(\omega)$ is the prior probability of the partition ω, and the posterior probability of H_0 is calculated then in terms of the Bayes factor, $P(H_0|X) = 1/(1+BF)$.

For each $\omega \in S_{n,k}$ the authors assume that the observations in the cluster j are distributed $N(\mu_j, \Sigma_j)$, so the likelihood of the sample and the marginal given a partition ω can be described by Equations (4) and (5) respectively.

$$L(\mu, \Sigma, \omega | X_1, X_2, \ldots, X_n) = \prod_{j=1}^{k} \prod_{l=1}^{n_j} N(X_l{}^{(j)} | \mu_j, \Sigma_j) \tag{4}$$

$$m(X|\omega) = \iint \prod_{j=1}^{k} \prod_{l=1}^{n_j} N(X_l{}^{(j)} | \mu_j, \Sigma_j) \cdot p(\mu_j, \Sigma_j) d\mu_j d\Sigma_j \tag{5}$$

Finally, as priori distribution for the mean and variance, the authors propose the use of: $p(\mu_j, \Sigma_j) = p(\mu_j|\Sigma_j) \cdot p(\Sigma_j)$ with:

$$p(\mu_j|\Sigma_j) \sim N(\mu_0^{(j)}, \tau^2 \Sigma_j) \tag{6}$$

where $\Sigma_j = diag(\sigma_{1j}^2, \sigma_{2j}^2 \ldots, \sigma_{rj}^2)$ and $\sigma_{rj}^2 \sim InverseGamma(a, b)$ with fixed values for $a = 2.01$ and $b = (a-1)^{-1}$

One of the issues of this methodology is that the sum of all possible partitions is in general too big, so is necessary to apply a Metropolis Hasting algorithm to sum over the subset of partitions that contribute more to the total sum in Equation (3). Nevertheless, the main idea of comparing models by a Bayes factor is an useful tool to decide whether to recombine a set of groups to detect clusters, and we will follow this approach in the next sections.

The article is structured as follows: in Section 2 we will develop the fundamentals of the recombining method, where we propose the use of a Bayes factor to compare two possible models explaining the data. In Section 3 we describe the algorithm which integrate the splitting and recombining processes, and finally we show the results of the application of the proposed method to four different data configurations and conclusions in Sections 4 and 5 respectively.

2. The splitting, cleaning, and recombining proposals

As in the original SAR process, the core of our methodology is based in the use of splitting and recombining procedures, plus a cleaning step between

them. The splitting process will be based on the discriminator function, the cleaning in an outlier detection process based on robust Mahalanobis distances, and finally for recombining we propose the use of Bayes factors. In this section we will review these procedures.

2.1. Splitting

With the aim of split the original data set into smaller groups, after detecting and potentially eliminating the outliers, the authors of the SAR define x_l as the discriminator of x_i if the latter observation appears as most discrepant (using the heterogeneity measures) with respect to the rest of the data set when the discriminator is deleted from the sample. The underlying idea is the following: If two observations are identical, they must have the same discriminator, thus, if they are close enough to each other, they should still have the same discriminator.

Formally, x_l in the multivariate case, assuming normality, is equivalent to:

$$x_l = arg \underset{j}{max} \ (x_i - \bar{x}_{(ij)})' \hat{V}_{(ij)}^{-1} (x_i - \bar{x}_{(ij)}) \tag{7}$$

which is the Mahalanobis distance between the element x_j and the rest of the sample, when the i_{th} and j_{th} elements are removed.

In the univariate case, Peña et al. (2004) shows that the discriminator are always the extreme points, while in the multivariate case, Rodriguez (2002) generalize this result demostrating that the discriminators belong to the convex hull of the sample. Therefore, Rodriguez (2002) proofs that discriminators are invariant to scale and positions transformation, because they are a monotonic function of the Mahalanobis Distance. Using these two properties, the observation x_l will be the discriminator of x_i if and only if:

$$x_l(x_i) = \underset{x_j \ \in \ Convex \ Hull}{arg \ max} \ \frac{\left(\frac{1}{n} + x_i' x_j\right)^2}{\left(\frac{n}{n-1} - x_j' x_j\right)} \tag{8}$$

Which is an efficient definition in terms of computational time, so it will be used in the algorithms included in this research.

To illustrate the discriminator function in the multivariate case, we present the widely known Old Faithful data set from Azzalini and Bowman (1990),

considering the waiting time between eruptions and the duration of them from the geyser "Old Faithful" in Yellowstone Park, Wyoming, USA. This data set form two groups as shown in the Figure 1.

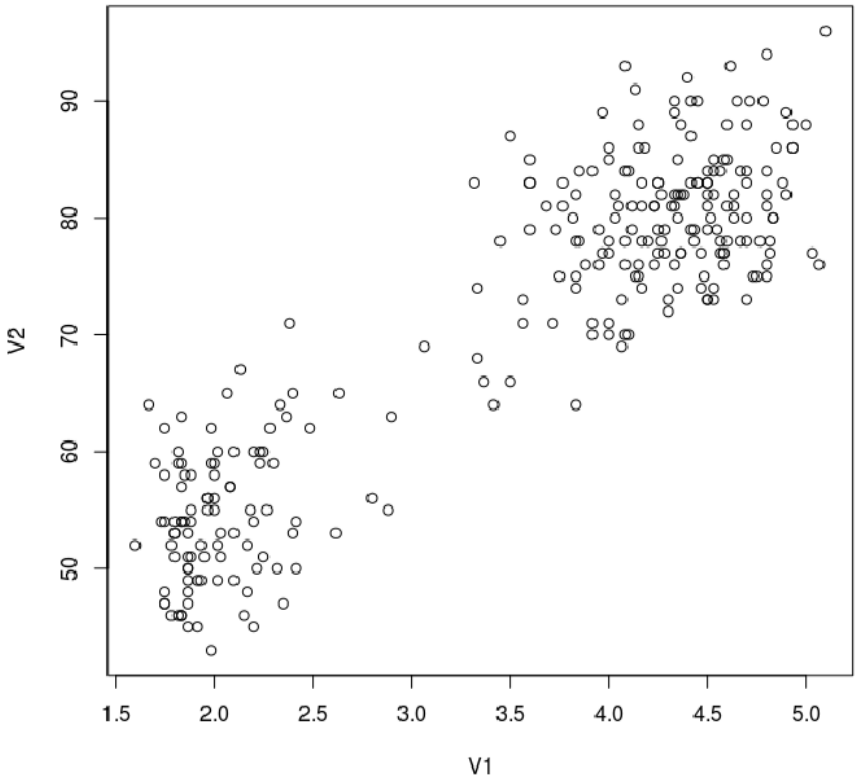

Figure 1: The Old Faithful data set

Applying the discriminator function, each data point is assigned to one discriminator following Equation (8) as showed in Figure 2, where is possible to see that the use of the discriminator function split the data into groups, assigning each point to one of the discriminators (observations 19, 58, 76, 149, 158, 161, 197, and 265) and this measure will be used in the SAR to perform the cluster analysis as we will see in the next section.

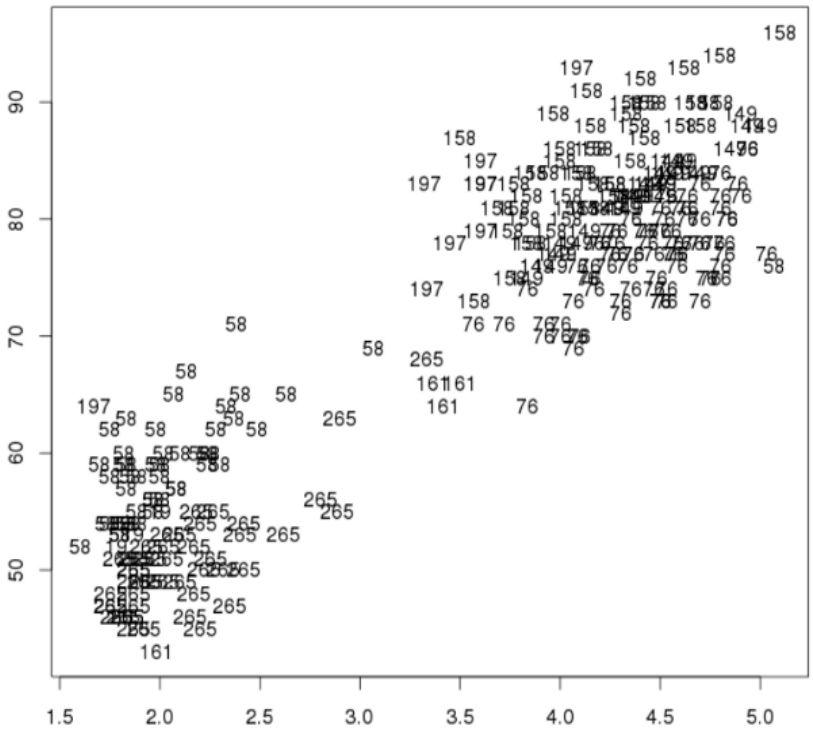

Figure 2: Discriminator function relationships, the number of the discriminator point is plotted

2.2. Cleaning

The outlier detection problem has been widely studied by the literature and is always a hot topic in statistical applications, because, the presence of outliers can bring inference complications as biased estimation, loss of efficiency, or bad predictions. Recent applications of outlier detection can be found in different areas such as cancer diagnostic (Kothari et al., 2013), climate change (Cho et al., 2013) or wireless networks (Branch et al., 2012).

To develop a new efficient method of outlier detection is far beyond the goals of this research, and even a comprehensive literature review of this topic can take hundreds of journal articles and books. Good recent examples of such reviews are Pahuja and Yadav (2013), Hodge and Austin (2004) and the book of Aggarwal (2013).

Among this big number of possibilities of outlier detection methodologies that we could incorporate to our proposal, the first and natural option to consider was the original SAR outlier detection presented in Peña and Tiao (2006). A more classic approach to the problem of outlier detection is the work of Rousseeuw (1985) who also propose the use of Mahalanobis distances to detect outliers. Under (multivariate) normality, the Mahalanobis distances are approximately distributed as a chi-square with p degrees of freedom (χ^2_p), but given that outliers can influence in those distances, is necessary to estimate them using a robust procedure. The method, known as MCD (Minimum covariance determinant) estimate the covariance matrix by the subset of h observations which minimises its determinant. Rousseeuw and Driessen (1999) shows that the MCD method is a computationally fast algorithm that can be used to calculate robust Mahalanobis distances based on those estimators and detect outliers. In practice, we use the implementation of the method proposed by Filzmoser et al. (2005) who also incorporate flexible critical values for those robust distances and that is available under the "mvoutlier" library for the R statistical language (Filzmoser and Gschwandtner, 2013).

Let $G_n(u)$ be the empirical distribution of the squared robust distances RD_i^2 calculated with the MCD method, is possible to compare the tails of that distribution and the theoretical distribution χ^2_p to detect outliers. The tails will be defined by $\delta = \chi^2_{p;1-\alpha}$ for a certain small α, so the departure of the empirical from the theoretical distribution in the tails is given by:

$$p_n(\delta) = \sup_{u \geq \delta} (\chi^2_p - G_n(u))^+ \tag{9}$$

where + indicates only the positive differences. Using this measure is still important to distinguish between extremes of the distribution and real outliers. To do so, a critical value (p_{crit}) is proposed by Filzmoser et al. (2005):

$$p_{crit} = \frac{0.24 - 0.003p}{\sqrt{n}} \text{ for } p \leq 10. \tag{10}$$

$$p_{crit} = \frac{0.252 - 0.0018p}{\sqrt{n}} \text{ for } p > 10. \tag{11}$$

Finally, the threshold value for the outliers is determined by

$$c_n(\delta) = G_n^{-1}(1 - \alpha_n(\delta)) \tag{12}$$

where

$$\alpha_n(\delta) = \begin{cases} 0, & \text{if } p_n(\delta) \leq p_{crit}(\delta, n, p). \\ p_n(\delta), & \text{if } p_n(\delta) > p_{crit}(\delta, n, p). \end{cases} \tag{13}$$

To depart from multivariate normality assumptions, a third option could be a more general algorithm of outlier detection in multivariate analysis proposed by Peña and Prieto (2001). The proposal is based on the idea of using projections to identify outliers, where each outlier must be an extreme point along the direction from the mean of the uncontaminated data to the outlier. In order to determine the direction of the projections, the authors claim that the presence of outliers in the projected data will imply particularly large (or small) values for the kurtosis coefficients, so they propose to use those directions that maximize or minimize the kurtosis.

2.3. Recombining

Given a partition $\omega_k = (l_1, l_2, ..., l_n)$ where $l_i \in \{1, 2, ...k\}, i = 1, 2, ...n$ are the labels assigning n data points $X_1, X_2, ..., X_n$ into $k > 1$ clusters, generated by the splitting process, we use a Bayes factor to compare the probability of the observed data given that partition against the data given the partition $\omega_1 = (1, 1, 1, ..., 1)$, implying all data points come from the same cluster in a similar way as used by Casella and Fuentes (2009).

Under the framework of recombine cluster subpartitions (or basic groups) as those obtained by a splitting procedure, we improve the Casella and Fuentes' Bayes factor in two ways:

a) We do not need to sum over all possible partitions or perform an importance sampling to estimate the Bayes factor, since we can use the information obtained by the splitting process. For example, consider two basic groups of sizes n_1 and n_2, so the Bayes factor to test if these two basic groups should be recombined will be:

$$BF = \frac{m(X|\omega_2) \cdot \pi(\omega_2)}{m(X|\omega_1) \cdot \pi(\omega_1)} \tag{14}$$

, where $\omega_2 = (\underbrace{1, 1, 1, ..., 1}_{n_1}, \underbrace{2, 2, 2, ..., 2}_{n_2})$ and $\omega_1 = (\underbrace{1, 1, 1, ..., 1}_{n_1+n_2})$

This approach has two main advantages: is more efficient in terms of computation time, and it also uses the information obtained by the partition process, which can be relevant to find the underlying structure of the data.

b) As a prior distribution for the mean and variance to derive the marginal given a certain partition, Casella and Fuentes use a restrictive approach where the covariance matrix is assumed to be diagonal. We propose to use a more flexible but also simple alternative, the use of the standard non informative given by Jeffreys:

$$p(\mu_j, \Sigma_j) \propto |\Sigma|^{\frac{-p-1}{2}}$$

Under this priori, Geisser (1964) shows that the posterior probability of an observation given a cluster defined by a $N(\bar{x}_i, S_i)$ is:

$$\begin{aligned} p(z|\bar{x}_i, S_i) &= \iint p(z|\mu_j, \Sigma_j) \cdot p(\mu_j, \Sigma_j|\bar{x}_i, S_i) d\mu_j d\Sigma_j \\ &= \left\{ \frac{N_i}{N_i + 1} \right\}^{p/2} \frac{\Gamma\{\frac{1}{2}(N_i)\}}{\Gamma\{\frac{1}{2}(N_i - p)\}|(N_i - 1)S_i|^{1/2}} \\ &\times \left[1 + \frac{N_i(\bar{x}_i - z)' S_i^{-1} (\bar{x}_i - z)}{(N_i + 1)(N_i - 1)} \right]^{-1/2(N_i)} \end{aligned} \tag{15}$$

Given a partition ω_k defined by k subsamples from the splitting process, in order to test if we will combine them ($H_0 : \kappa = 1$ *vs* $H_1 : \kappa = k$,), we will use the Bayes factor given by Equation (14), where $m(X|\omega)$ is the likelihood of the data given by partition ω, under the assumption that each of the groups in the partition follows a multivariate normal distribution, and using non informative priors for μ and Σ as given by Equation (15).

$$m(X|\omega) = \prod_{i=1}^{k}\prod_{j=1}^{n_i}\left\{\frac{N_i}{N_i+1}\right\}^{\frac{p}{2}} \frac{\Gamma\{\frac{1}{2}(N_i)\}}{\Gamma\left\{\frac{(N_i-p)}{2}\right\}|(N_i-1)S_i|^{\frac{1}{2}}} \times \left[1+\frac{N_i(\bar{x}_i-y_{ij})'S_i^{-1}(\bar{x}_i-y_{ij})}{(N_i+1)(N_i-1)}\right]^{\frac{-N_i}{2}} \tag{16}$$

In a similar way to Casella and Fuentes (2009), we use the marginal distribution of the number of clusters in a Dirichlet process proposed by Pitman (1996) as priors for partitions $\pi(\omega)$. In this configuration, the priors only depend on the number of elements in each group of the partition.

$$\pi(\omega_k) = \frac{\prod_{i=1}^{k}\Gamma(n_i)}{\Gamma(n+1)} \tag{17}$$

As $\pi(\omega_1) = \frac{\Gamma(n)}{\Gamma(n+1)}$ and $\pi(\omega_k) = \frac{\prod_{i=1}^{k}\Gamma(n_i)}{\Gamma(n+1)}$, then:

$$\frac{\pi(\omega_k)}{\pi(\omega_1)} = \frac{\prod_{i=1}^{k}\Gamma(n_i)}{\Gamma(n)} \tag{18}$$

When H_0 is true, the Bayes factor will be bigger than 1, but in order to have a standard measure to decide whether combine groups, we propose the use of a transformation that remains in the domain $[0, 1]$ and can be equivalent to $P(H_0)$. Following Casella and Fuentes (2009), we have:

$$P(H_0) = \frac{1}{1+BF}$$

And finally, we will reject the null hypothesis when $P(H_0)$ is smaller than a critical value, typically 0.05 or 0.01. In this case we will separate the partitions, being merged otherwise.

2.4. Examples of Bayes factor application

As an example of the performance of the Bayes factor, we will apply it to arbitrary partitions under the existence of one and two groups respectively.

Example 1. Two independent samples:

Two independent samples of sizes $n_1 = n_2 = 100$, are generated from a bivariate normal distributions with means $\mu_1 = (-1, -1)$; $\mu_2 = (1, 1)$ and with covariance matrices $\Sigma_1 = \Sigma_2 = \begin{bmatrix} 1/4 & 0 \\ 0 & 1/4 \end{bmatrix}$. We arbitrarily split the sample according to the line $X = -0.5 - 0.75X$, as shown in Figure 3, to separate the two samples, so we can check if the Bayes factor is able to keep them separate.

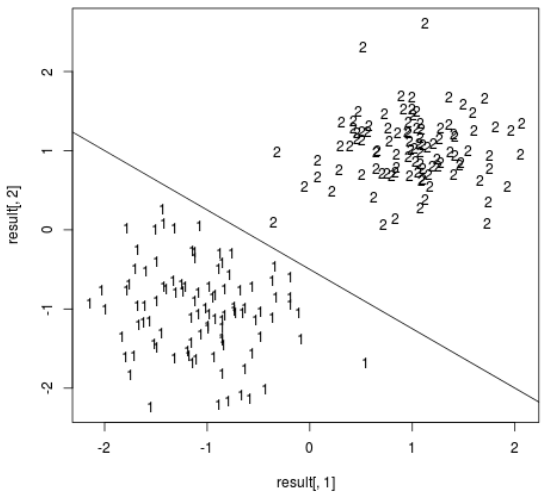

Figure 3: Bayes factor Example 1, two normal samples

When comparing $H_0 : \kappa = 1$ vs. $H_1 : \kappa = 2$, the following results are obtained:

$$m(X|\omega_2) = 2.662409e - 120,\ m(X|\omega_1) = 2.944844e - 224$$

$$\pi(\omega_1) = 0.005,\ \pi(\omega_2) = 1.115536e - 63$$

then:

$$\frac{m(X|\omega_2)}{m(X|\omega_1)} = 9.040915e + 103, \frac{\pi(\omega_2)}{\pi(\omega_1)} = 2.231071e - 61$$

and finally,

$$BF = 2.017093e + 43 \text{ and } p(H_0) = \frac{1}{1 + BF} = 4.95763e - 44$$

As expected, there is a strong evidence against H_0 and the two groups should be separated.

Example 2: One sample

In this example only one sample of size $n = 200$ is generated from a Normal distribution with mean $\mu_1 = (1, 1)$ and with covariance matrix $\Sigma_1 = \begin{bmatrix} 1/4 & 0 \\ 0 & 1/4 \end{bmatrix}$, while the arbitrary partition on this occasion will take place in the line $X = X$. This configuration is shown in Figure 4.

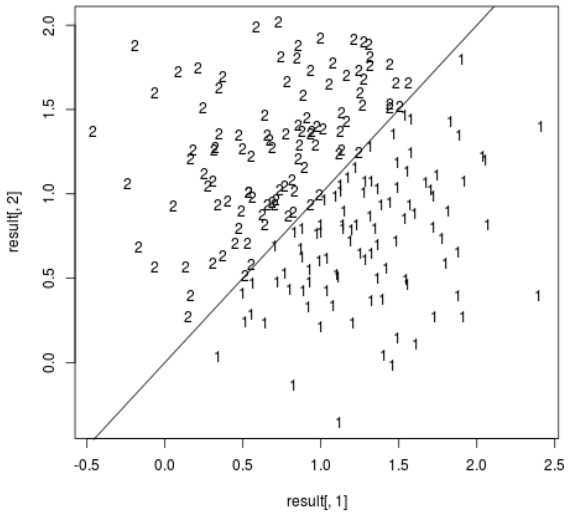

Figure 4: Bayes factor Example 2, one normal sample

Testing in this occasion $H_0 : \kappa = 1$ vs. $H_1 : \kappa = 2$, we obtain the following results:

$m(X|\omega_2) = 9.515897e - 83$, $m(X|\omega_1) = 1.498073e - 127$

$\pi(\omega_1) = 0.005$, $\pi(\omega_2) = 1.149685e - 63$

where: $\frac{m(X|\omega_2)}{m(X|\omega_1)} = 6.352093e + 44$, $\frac{\pi(\omega_2)}{\pi(\omega_1)} = 2.2993e - 61$

Finally,

$$BF = 1.460581e - 16 \text{ and } P(H_0) = \frac{1}{1 + BF} \approx 1$$

Now we have evidence in favour of H_0 so the two partitions should be merged, again as expected. More examples of this test under the framework of the proposed cluster algorithm will be given later in Section 4.

3. The splitting and group recombining algorithm (SAGRA)

Our algorithm proposal is based on the splitting, cleaning, and recombining processes described in the previous section, and it has as a main goal to return a vector with cluster classes given a multivariate data set. Such classes should help the researcher to unveil the structure of the data, being in this way a tool for exploratory research.

The algorithm holds the usual assumptions for cluster analysis, i.e. every observation is assigned to a group, all observations are classified, and the internal variability of the classes is smaller than between groups. One of the advantages of this procedure is that the data set does not need to be standardized since we use Mahalanobis distances.

The procedure to form the final data configuration is organized in six steps: two splitting steps, one outlier cleaning process and three recombining steps. Each step is detailed in the following subsections.

3.1. Split step 1

The input of the algorithm is a sample $x = x_1, x_2, ...x_n$ coming from a p-variate unknown distribution with sample mean $\bar{x}$, and sample covariance matrix S. To start the procedure, the discriminator of each observation is obtained. Recalling Subsection 2.1, the discriminator of one data point is the most discrepant point respect to the rest of the sample (Equation (7)).

We also saw in Subsection 2.1 that in an univariate sample, the discriminators are the extreme values of the sample, while in multivariate dimensions they lay into the convex hull. When the discriminator function is applied to the data sample, typically all observations will be assigned to a subset of the data points belonging to the convex hull, defining with this step the first split.

To illustrate step by step the behaviour of the proposed clustering procedure, we will apply it to the "Old Faithful" data from Figure 1. With the discriminator function, the dataset is split into 8 groups with the distribution given by Table 1.

Table 1: Discriminator function distribution for the geyser data set

group	1	2	3	4	5	6	7	8	Total
discriminator observation	19	58	76	149	158	161	197	265	
group size	3	49	79	34	48	4	9	46	272

A minimum size m_0 of the groups is needed to avoid over splitting into small highly homogeneous groups, which can be difficult to merge in the recombining stage of the algorithm. For this reason we set a minimum size equivalent to the 5% of the size of the sample, although it can depend on the complexity of the data set, other alternative could be to use the minimum size proposed by Peña et al. (2004), where $m_0 = p + log(n - p)$. In our example, $n = 272$, so $m_0 = 13.6$.

When this first partition leads to groups such that all of them are of sizes smaller than minimum size, the splitting stop. Otherwise we eliminate the small groups, classifying their observations as isolated observations.

Also the discriminators are extracted from the obtained groups, and temporarily assigned as isolated observations. This is because in deeper levels we want to discover new structures and not define the same partitions in the following steps, as it will happen if we keep the discriminator in. Therefore, the groups which are of sizes smaller than the minimum size are also considered isolated observations, as in the case of groups number 1, 6 and 7, whose sizes are 3, 4, and 9 observations respectively.

As a result of this procedure we obtain a first cluster structure and a set of isolated data points. In the example we get 5 groups with the distribution given by Table 2.

Table 2: First splitting step of SAGRA cluster distribution of the geyser example

group	1	2	3	4	5	isolated	Total
size	79	34	46	47	45	21	272

Formally, the step 1 is expressed as:

Require: data set $D = y_1, y_2, ..., y_n$
Split step 1.
for $i = 1 \to n$ **do**
$y_l(y_i) \leftarrow arg \max_j (y_i - \bar{y}_{(ij)})' \hat{V}^{-1}_{(ij)} (y_i - \bar{y}_{(ij)})$
end for
$L \leftarrow \{y_j \in D \mid \exists y_i \in D, y_j = y_l(y_i)\}$
Compute $C_1, C_2, ...C_K$, where $K = |L|, i = 1, 2, ..., n$
s.t. $y_i, y_j \in C_k \Leftrightarrow y_l(y_i) = y_l(y_j) \forall i, j = 1, 2, ..., n; k = 1, 2, ..., K$
s.t. $|C_k| \geq m_0 \forall k = 1, 2, ..., K$
for $k = 1 \to K$ **do**
$C_k \leftarrow C_k \setminus L$
end for
Output: $C = \{C_1, C_2, ..., C_K\}$

3.2. Split step 2

The second step is to apply the same previous discrimination procedure to each of the previous groups, finding its internal cluster structure.

For each of this "second level" cluster structures, we test if the groups should be split into the basic groups obtained by the splitting, or maintained as in the previous level. We use the recombining test introduced in the previous section setting $H_0 : \kappa = 1$ vs. $H_1 : \kappa = K_i$, being k_i the number of partitions found (second level) in the group i (first level). When $p(H_0) < \alpha$ we reject H_0 and we split into the groups defined by the discrimination of the second level.

All groups which are not split (i.e. $p(H_0) > \alpha$) are separated from the procedure and saved as "candidate groups". These candidate groups will be not split again, and only can be recombined, so they are separated from the rest of the groups. For each of the remaining groups we repeat the procedure until no further partition can be done so we added all sub partitions to the candidate groups.

In the geyser example, the second level partition is shown in Table 3:

The second step splits the group 1 into four subgroups. Then the Bayes factor for this four groups obtains a p-value of 1, indicating that the likelihood of these partitions is too low, so we do not split this group. For the other groups from the previous split (groups 2 - 5), there is no splitting, so $BF = 1$

Table 3: Second splitting step of SAGRA cluster distribution of the geyser example

Level 1	1	1	1	1	2	3	4	5
Level 2	1	2	3	4	1	1	1	1
size	21	20	18	15	33	45	19	21
P-value			1		0.5	0.5	0.5	0.5

and $P(H_0) = 1/2$. Since all p-values in second stage are bigger than $\alpha = 0.01$, we set all those partitions as candidate groups and the splitting procedure is finished and the groups in this step remains as they were in the previous splitting step.

Formally, step 2 is expressed as:

Require: data set $D = y_1, y_2, ..., y_n$, $C = \{C_1, C_2, ..., C_K\}$
  Split step 2.
  $GC \leftarrow \emptyset$
  if $K = 1$ **then**
    $GC \leftarrow D$
  else
    repeat
      $C \leftarrow \emptyset$
      for $i = 1 \rightarrow K$ **do**
        Apply step 1 to C'_i to obtain $C'_{1i}, C'_{2i}, ..., C'_{Ki}$
        Compute $p(H_0), H_0 : \kappa = 1 \; vs \; H_1 : \kappa = Ki$
        if $p(H_0) > \alpha$ **then**
          $GC \leftarrow GC \cup \{C'_i\}$
        else
          $C \leftarrow C \cup \{C'_{1i}\} \cup \{C'_{2i}\}, ..., \cup \{C'_{Ki}\}$
        end if
      end for
    until $C = \emptyset$
  end if
  Output: $GC = \{GC_1, GC_2, ..., GC_{K'}\}$

3.3. Cleaning process

As a result of the two split steps we get a set of "candidate groups" and some isolated data points from the last step. Nevertheless, since a minimum size was established, is possible that some of the candidate groups are still formed by a mix of observations from different clusters. To avoid undesirable recombination due to this misclassified observations, is necessary to apply an "outlier" detection and cleaning process before the recombination steps. The idea is to have pure basic groups with no elements from different clusters.

From the three outliers detection methods we considered in Section 2.2, currently our algorithm incorporate the MCD method for efficiency reasons, but future versions of the code will allow the user to choose among those outliers methods.

Coming back to the example, the cleaning is performed inside each group using the MCD method, leading to 16 observations removed from candidate groups. The new group distribution is given by Table 4:

Table 4: Cleaning step of SAGRA cluster distribution of the geyser example

group	1	2	3	4	5	isolated	Total
size	77	34	46	40	38	37	272

Formally, the cleaning step is expressed as:

Require: data set $D = y_1, y_2, ..., y_n$, $GC = \{GC_1, GC_2, ..., GC_{K'}\}$
Cleaning step
for $i = 1 \rightarrow K'$ **do**
  for $j = 1 \rightarrow n_i$ **do**
    Compute RD_j
  end for
  Compute $c_n(\delta)$
  $GC_i \leftarrow GC_i \setminus \{y_{ji} \in GC_i \mid RD_j > c_n(\delta), j = 1, 2, ..n_i\}$
end for
Output: $GC = \{GC_1, GC_2, ..., GC_{K'}\}$

3.4. Recombine step 1

The first step in the recombination stage is to order the K' groups such that the bigger partition is labelled as group 1, and the rest depending on how close (using the Mahalanobis distance) they are to the group 1, being "2" the closer, "3" the next, and so on. After ordering, we test for merging groups 1 and 2:

If $p(H_0) \leq \alpha$, we keep them as separated groups, and group 1 will stay as "candidate group". Now we test for merging groups 2 and 3, and so on. If $p(H_0) > \alpha$ We do not split the groups, we relabel the resulting merged group as group 1, and the remaining from 2 to K'-1

The process finishes when just one group remains, and in this case it is also assigned as a new candidate group.

In the example, the groups are merged as shown in Table 5:

Table 5: Test results of the first recombining step of SAGRA cluster to the geyser example ($\alpha = 0.01$)

test	1-2	12-3	123-4	4-5
p-value	0.038	1	0	0.003

So three groups are formed with the formers 1-2-3, 4 and 5, with size distribution given by Table 6.

Table 6: First recombining step of SAGRA cluster distribution of the geyser example

group	1	2	3	isolated	Total
size	157	40	38	37	272

Formally, the recombine step 1 is expressed as:

Require: data set $D = y_1, y_2, ..., y_n$, $GC = \{GC_1, GC_2, ..., GC_{K'}\}$

Recombine step 1.

$C \leftarrow \emptyset$

$C_1 \leftarrow \arg\max_{GC_k} |GC_k|, k \in 1, 2, ..., K'$

define $C_2, C_3, ...C'_K \mid D_M(C_2, C_1) > D_M(C_3, C_1) > ... > D_M(C'_K, C_1)$
$max \leftarrow K'$
for $k = 1 \rightarrow K' - 1$ **do**
  $G \leftarrow C_k \cup C_{k+1}$
  Compute in G $p(H_0), H_0 : \kappa = 1 \; vs \; H_1 : \kappa = 2$
  if $p(H_0) > \alpha$ **then**
    $C_k \leftarrow C_k \cup C_{k+1}$
    if $k < max - 1$ **then**
      $C_{k+1} \leftarrow C_{k+2}; C_{k+2} \leftarrow C_{k+3}; ...; C_{max-1} \leftarrow C_{max}$
      $C_{max} \leftarrow \emptyset$
      $max \leftarrow max - 1$
    end if
  end if
end for
Output: $C = \{C_1, C_2, ..., C_{K''}\}$

3.5. Recombine step 2

After we recombine the groups obtained by the splitting process, is still necessary to assign the isolated observations (i.e. discriminators and groups under the minimum size) to one of the candidate groups. This is done simply calculating the Mahalanobis distances (D_M) from each isolated point to all candidates and assigning it to the closer one.

In geyser data, the isolated points where assigned to the three groups as shown in 7:

Table 7: Second recombining step of SAGRA cluster distribution of the geyser example

group	1	2	3	Total
size	176	51	45	272

Formally we have:

Require: data set $D = y_1, y_2, ..., y_n$, $C = \{C_1, C_2, ..., C_{K''}\}$
Recombine step 2.

$D_{isolated} \leftarrow D \setminus C_1 \setminus C_2 ... \setminus C_{K''}$
for $i = 1 \rightarrow |D_{isolated}|$ **do**
$C_j \leftarrow C_j \cup y_i \mid C_j \leftarrow \arg\max_{C_k} D_M(C_k, y_i)$,
$j \in 1, 2, ..., K'', y_i \in D_{isolated}$
end for
Output: $C = \{C_1, C_2, ..., C_{K''}\}$

3.6. Recombine step 3

Finally, given that the incorporation of the isolated points increases the variability of the groups, a new merging process (first recombining step) is performed between the candidate groups leading to the final data configuration. The test results for our example is given by Table 8, while the final data distribution in the two final groups is shown in Table 9

Table 8: Test results of the third recombining step of SAGRA cluster for the geyser example

test	1-2	2-3
p-value	0	0.98

Table 9: Final SAGRA cluster distribution of the geyser example

group	1	2	Total
size	176	96	272

The graphical result of the SAGRA procedure applied to the geyser data is shown in the Figure 5(a) where we can observe that the two clusters are correctly separated, although no error ratios can be calculated because even it is clear that at least two groups are identified in the sample, there are no original labels.

Require: data set $D = y_1, y_2, ..., y_n$, $C = \{C_1, C_2, ..., C_{K''}\}$
Recombine step 3.
Apply Recombine step 1 to C to obtain Final Clusters.
Output: $FC = \{FC_1, FC_2, ..., FC_{K'''}\}$

3.7. Comparison with other algorithms

Because the SAGRA continues the work developed by Peña et al. (2004) is natural to compare our results with those obtained by the original SAR algorithm, which results are shown in Figure 5(b). Additionally, we will include in the comparison two benchmarking algorithms like k-means (MacQueen, 1967), presented in Figure 5(c) and M-clust (Fraley and Raftery, 1998), plotted in Figure 5(d). In the case of k-means we will set the number of groups as the original, two in the case of geyser data.

Regarding those algorithms, SAGRA shows similar results to k-means in detecting two groups, whereas SAR detects also the same two main groups and a group of isolated points between them, while M-clust split one of the main groups into two, leading to a three groups configuration.

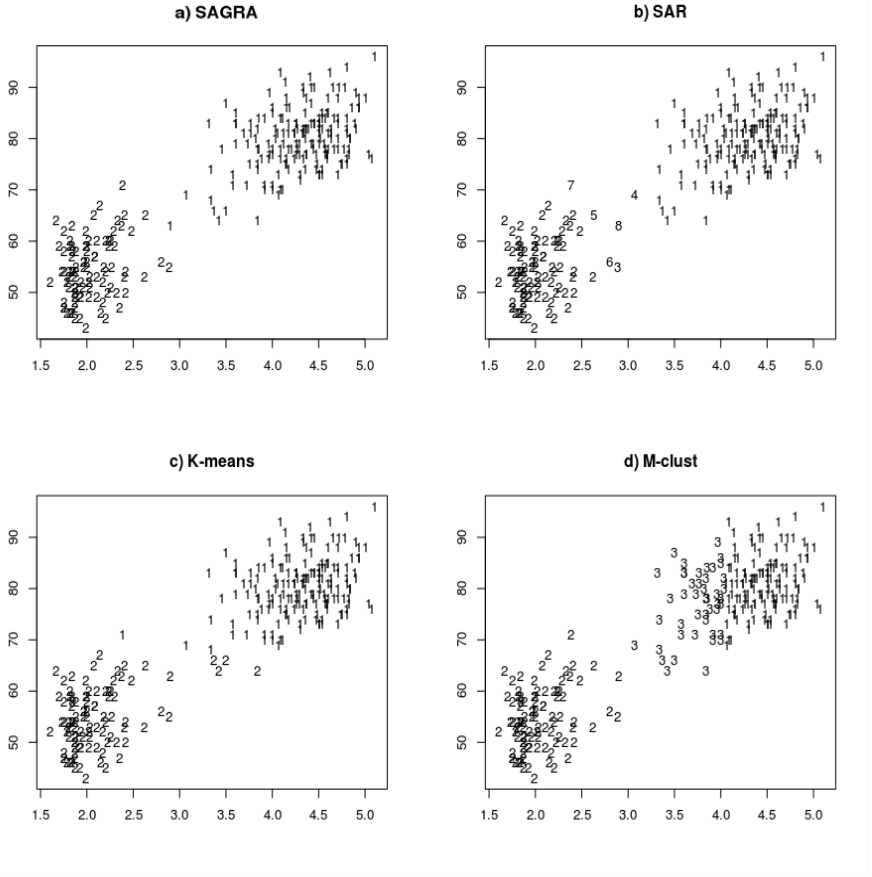

Figure 5: Comparison of cluster methods applied to the Old Faithful data set

3.8. Example: Four independent samples:

In this second example, four independent samples are generated with sizes $n_1 = n_2 = n_3 = n_4 = 100$, coming from a bivariate normal distribution with means $\mu_1 = (2, 2)$, $\mu_2 = (2, -2)$, $\mu_3 = (-2, -2)$, $\mu_4 = (-2, 2)$; and covariance matrices $\Sigma_1 = \begin{bmatrix} 0.25 & 0.15 \\ 0.15 & 0.25 \end{bmatrix}$, $\Sigma_2 = \begin{bmatrix} 0.25 & -0.15 \\ -0.15 & 0.25 \end{bmatrix}$, $\Sigma_3 = \begin{bmatrix} 0.25 & 0.15 \\ 0.15 & 0.25 \end{bmatrix}$, $\Sigma_4 = \begin{bmatrix} 0.25 & -0.15 \\ -0.15 & 0.25 \end{bmatrix}$.

In this way we have four well separated groups with different orientations as shown in Figure 6, whereas the graphical results of the clustering are in Figure 7. In this case, the modified SAR shows similar results to mclust, and

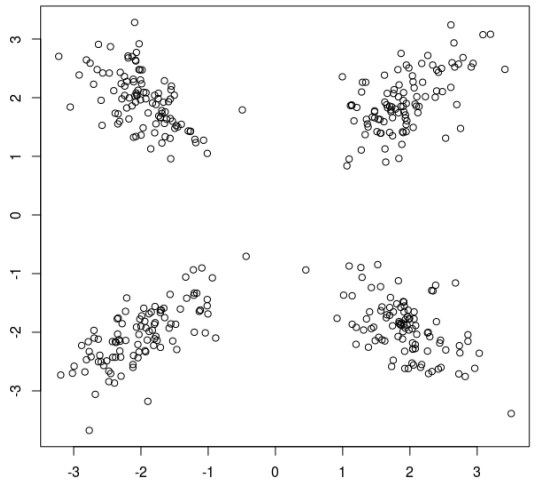

Figure 6: Four well separated normal samples example

slightly better than original SAR, which correctly classify the four groups but creates a 5th small group. K-means split one group into two, and forced by the number of groups set to 4, classify two groups as one.

4. Results

In order to generalize the previous examples and compare the results of the SAGRA algorithm with the other clustering procedures, we use classical measures to evaluate the quality of the output from such class of methods, based on the number of positive and negative decisions when classifying each data point from a data set.

To do so, we simulate data sets to have the original labels, allowing to compare the results from the clustering methods. Notice that generally in cluster analysis the labels are not available, so this comparison can be done only between two different clustering solutions.

Given the total number of pairs of observations $n(n-1)/2$ that can be formed from an original sample of size n, a true positive decision (TP) assigns two observations from the same class in the same cluster, and a true negative (TN) decision assigns two observations from different classes to different clusters.

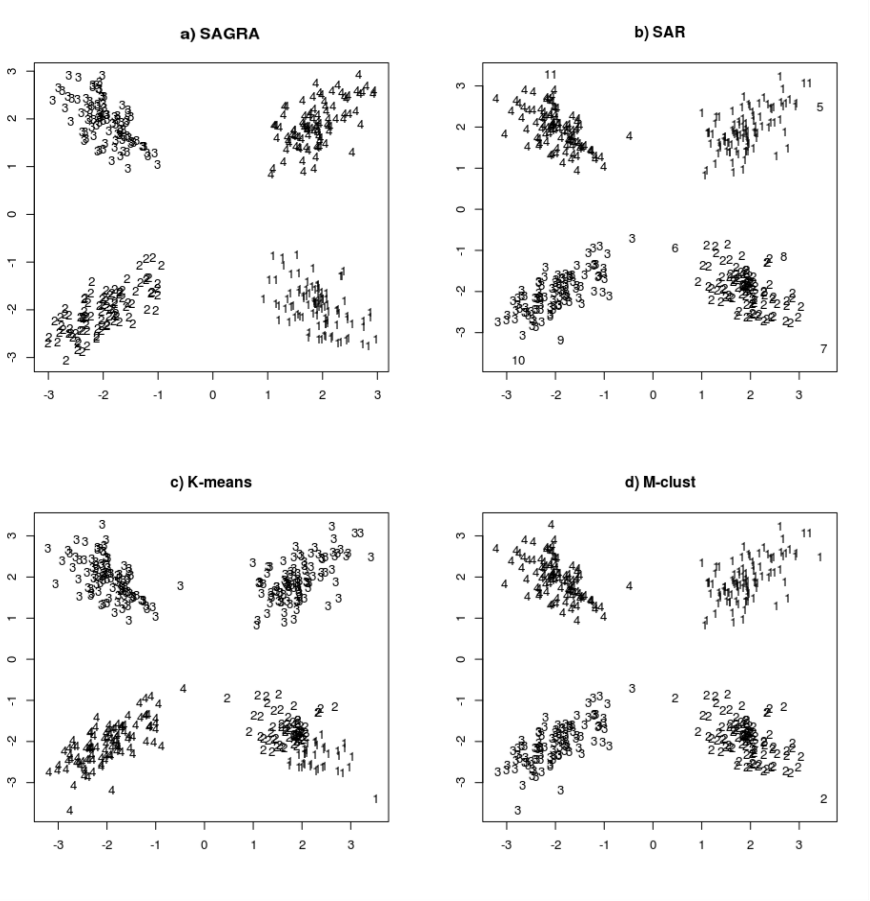

Figure 7: Comparison of cluster methods applied to four normal simulated samples

The errors of the algorithm can be defined in the same way, being a false positive decision (FP) if the algorithm assigns two data points from different classes in the same cluster, while a false negative (FN) decision assigns two observations from the same class in different clusters. The total counting of these four decisions are usually presented in a "Table of Confusion" as given by Table 1.

Table 10: Generic Table of Confusion

	Same cluster	Different clusters
Same class	TP	FN
Different classes	FP	TN

In base of those decisions, we compare the proposed SAGRA method with SAR, K-means and M-clust, using the following measures: Purity, Number of Groups, Rand Index, Adjusted Rand Index, and F_1.

Purity is the sum of the majority of observations assigned to each cluster by the algorithm over the total number of observations, and is a measure of how "pure" the clusters are, in the sense that they are formed only for elements from the same class. This measure will favour those solutions with many groups but whose elements belongs to the same class, for that reason we include the Number of Groups, so we can observe how close or far the cluster methods are from the original data.

Rand (1971) proposes an index to compute the percentage of correct decisions made by a cluster algorithm, defined by:

$$RI = \frac{TP + TN}{TP + FP + TN + FN} \tag{19}$$

A modification of the Rand Index (RI), called Adjusted Rand Index (ARI) was proposed by Hubert and Arabie (1985) to solve some issues from the RI, which expected value is not zero when two random partitions are compared, or that is higher when the number of groups increases. Then, the ARI take values from -1 to 1 and is expressed as:

$$ARI = \frac{\frac{n(n-1)}{2}(TP + FN) - [(TP + FN)(TP + FP) + (FP + TN)(FN + TN)]}{\left[\frac{n(n-1)}{2}\right]^2 - [(TP + FN)(TP + FP) + (FP + TN)(FN + TN)]} \tag{20}$$

The F_1 measure is a way to compare the precision and recall when comparing cluster results. Precision is the ratio between the TP over all pairs we assign to the same cluster $P = TP/(TP + FP)$, while Recall is the probability of assigning two elements to the same group given that they are from the same class, $R = TP/(TP + FN)$. Finally, the F_1 is defined as:

$$F_1 = \frac{2 * P * R}{P + R} \tag{21}$$

For all of this measures, the more similar the cluster result is respect to the real configuration, the bigger the index will be. The exception of this rule is the "Groups" measure, when the closer to the original number of groups is the best.

The SAGRA algorithm was coded and run under R framework. The code will be soon published in the authors web site, and is also available upon request.

The SAR algorithm was run via the `sarpt` function in Matlab described in Rodriguez (2002), the Mclust algorithm has been run with the R function `Mclust` with models "EII", "VII", "EEI", "VEI", "EVI", "VVI", "EEE", "EEV", "VEV", and "VVV" as a covariance structure, and the possible number of clusters is set to be between 1 and 8. The final configuration is selected by the BIC, as is detailed in Fraley and Raftery (1999). Finally, the K-means algorithm was also run under the R framework, using the function `cascadeKM`, from the `vegan` package where the rule to select the "K" number of clusters in the algorithm is the maximum of the Calinski criteria for $k = 1, ..., 8$ (see Calinski and Harabasz, 1974).

We test the procedures under four data configurations:

Case 1: Normal distribution

For this case we generate 100 random data sets, each consisting of four independent samples with sizes $n_1 = n_2 = n_3 = n_4 = 100$, coming from a bivariate normal distribution with means $\mu_1 = (1, 1)$, $\mu_2 = (1, -1)$, $\mu_3 = (-1, -1)$, $\mu_4 = (-1, 1)$; and covariance matrices $\Sigma_1 = \begin{bmatrix} 0.25 & 0.15 \\ 0.15 & 0.25 \end{bmatrix}$, $\Sigma_2 = \begin{bmatrix} 0.25 & -0.15 \\ -0.15 & 0.25 \end{bmatrix}$, $\Sigma_3 = \begin{bmatrix} 0.25 & 0.15 \\ 0.15 & 0.25 \end{bmatrix}$, $\Sigma_4 = \begin{bmatrix} 0.25 & -0.15 \\ -0.15 & 0.25 \end{bmatrix}$ respectively.

Case 2: Two correlated uniform samples

In a second scenario we generate 100 random data sets, each consisting of two independent samples with sizes $n_1 = n_2 = 500$, coming from a bivariate

uniform distribution with means $\mu_1 = (0, 0)$, $\mu_2 = (-0.5, 0)$. The correlation between the two variables on each sample is set to be $\rho = 0.9$.

Case 3: Three geometric uncorrelated uniform samples

Now we generate 100 random data sets, each consisting of three independent samples, with sizes $n_1 = n_2 = n_3 = 500$ on geometric shapes formed by uncorrelated uniform observations in the shape of one circle and two rectangles.

Case 4: Two half moons

Finally for the last case we generate 100 data sets, each consisting of two half moons, with sizes $n_1 = n_2 = 500$, from the R package `spa` (Culp, 2011). The two moons are oriented opposite to each other, so they cannot be linearly separated.

One of the samples of each dataset is shown in the Figure 8, and the results of the simulations are shown in Table 2.

5. Conclusions

The SAGRA (Split And Group Recombining Algorithm) method was developed based on a splitting and recombining methodologies in a similar way as the SAR algorithm proposed by Peña et al. (2004). In comparison with SAR itself and other classical cluster analysis procedures such as K-means and M-clust, SAGRA shows competitive results. We applied those methods over four different data configurations and we took five performance measures (Purity, Groups, RI, ARI, and F1).

SAGRA obtained similar results to M-clust under normally distributed data set (where M-clust tend to be optimal), and in general, better results than other methodologies in the rest of the cases over all the measures, with the exception of the detection of the real number of groups, where no methodology was exact for these four simulated data sets.

Some issues of the algorithm in comparison to other methods need to be considered. First, is necessary to fix two parameters, the minimum size for the splitting process and the critical value for the $p(H_0)$ in the recombining step. The minimum size determines one of the stopping rules in the algorithm, and is required to split the sample in such a way that 1) the groups are small enough to adequately separate the different classes and 2) the groups are big enough so they can be tested to be combined using a model approach using the Bayes factor. As reported in the description of the algorithm, an empirical use of a minimum size equivalent to the 5% of the total sample size

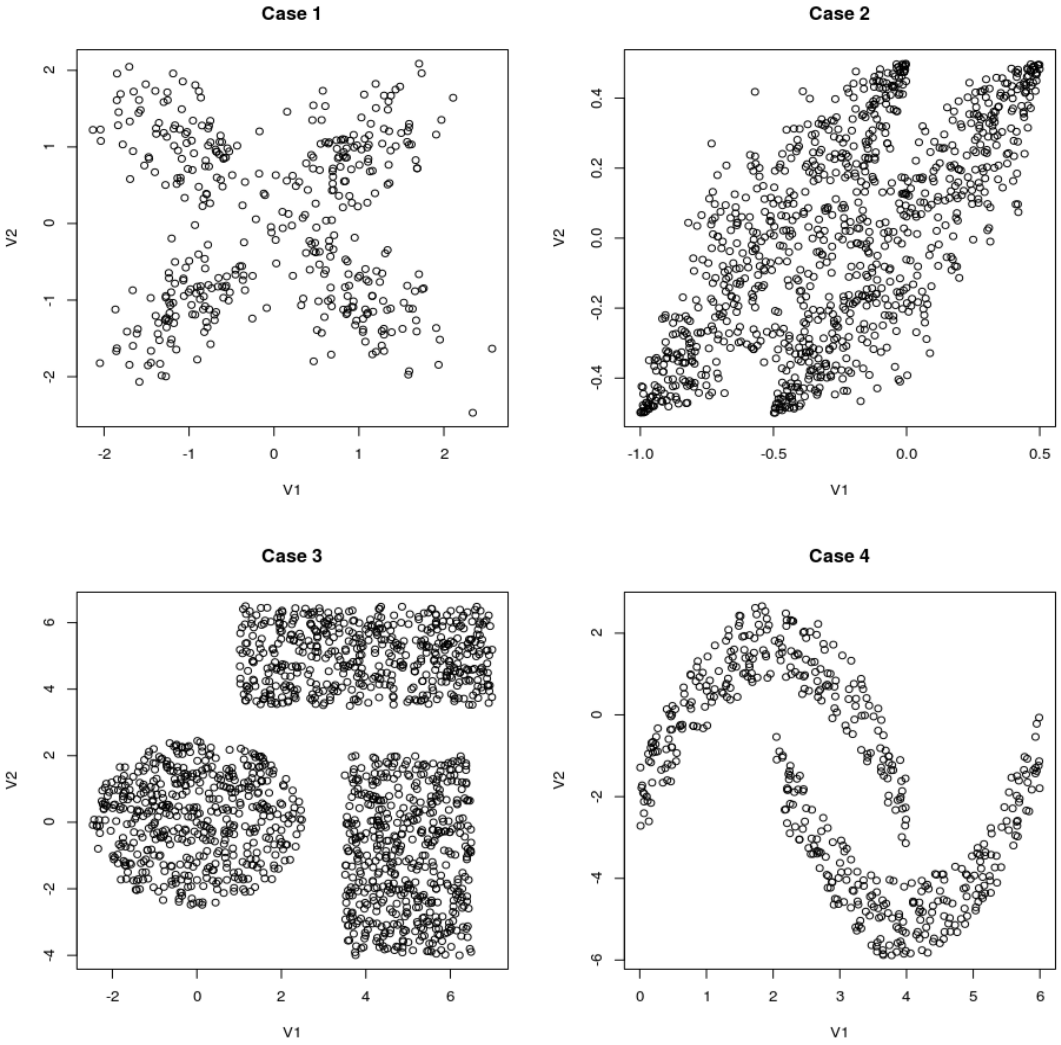

Figure 8: Four data configurations to test the performance of the SAGRA algorithm

adequately holds for these two conditions in a general context, although the number can depend on the complexity of the data set.

Regarding the critical value for the $p(H_0)$, is important to notice that we are comparing two different models, one where the data comes from the same distribution, and other when the data is generated by the structure implied in the partition. We choose the null hypothesis to be that where the data set

Table 11: Average estimated number of groups from 500 simulations for each case

	Criteria	SAGRA	SAR	K-means	M-clust
Four Normals	Purity	**0.96**	0.25	0.92	**0.96**
	Groups	4.02	1.37	5.83	**4.00**
	RI	**0.96**	0.25	0.90	**0.96**
	ARI	**0.89**	0	0.70	**0.89**
	F1	**0.92**	0.4	0.76	**0.92**
Two correlated uniforms	Purity	**0.96**	0.5	**0.96**	0.94
	Groups	5.08	**1.11**	7.97	8.13
	RI	**0.72**	0.5	0.61	0.63
	ARI	**0.44**	0	0.21	0.26
	F1	**0.62**	0.67	0.37	0.45
Three geometric uniforms	Purity	**1.00**	0.84	**1.00**	**1.00**
	Groups	3.86	**2.52**	5.39	7.55
	RI	**0.96**	0.89	0.88	0.81
	ARI	**0.90**	0.79	0.69	0.51
	F1	**0.93**	0.88	0.76	0.61
Two half moons	Purity	0.99	0.51	0.94	**1.00**
	Groups	4.76	**1.15**	4.10	6.43
	RI	0.75	0.51	**0.78**	0.67
	ARI	0.51	0.02	**0.56**	0.33
	F1	0.67	0.67	**0.71**	0.50

is coming from the same distribution, so it forms only one group, and we will split the sample only if there are strong evidence for that, choosing a value $p(H_0) < 0.01$ to split the data in the given partitions for our examples.

As advantages, the algorithm does not need to fix the number of groups, as k-means, or it is not necessary to compare different solutions as M-clust or the original SAR algorithm, both of them using the BIC criteria to choose the final data configuration. This is important since the BIC criteria is optimum to compare Normal distributions but tends to overestimate the real number of groups when the data depart from normality.

In summary, the main contribution of this research is a new clustering algorithm (SAGRA, Splitting and Group Recombining Algorithm) based on a splitting and recombining methodology using the discriminator function

and outlier cleaning for splitting, and Bayes factors for recombining. The obtained results show that the SAGRA algorithm is competitive with respect to the benchmarking algorithms in cluster analysis obtaining more than 95% of purity in each example, even when the four data configuration used to test the algorithms are quite different. Nevertheless, the algorithm can be modified in order to make flexible the normality assumption for the partitions in the Bayes factor. This can improve the results in cluster detection when data is not normally distributed, specially in terms of identify the proper number of groups, where the original SAR have better performance.

References

Aggarwal, C. C. (2013), *Outlier Analysis*, New York: Springer.

Atkinson, A. C. and Riani, M. (2007), "Exploratory Tools for Clustering Multivariate Data," *Computational Statistics & Data Analysis*, 52, 272–285.

Azzalini, A. and Bowman, A. W. (1990), "A Look at Some Data on the Old Faithful Geyser," *Applied Statistics*, 39, 357–365.

Baudry, J.-P., Raftery, A. E., Celeux, G., Lo, K., and Gottardo, R. (2010), "Combining Mixture Components for Clustering," *Journal of Computational and Graphical Statistics*, 9, 332–353.

Biernacki, C., Celeux, G., and Govaert, G. (2000), "Assessing a Mixture Model for Clustering with the Integrated Completed Likelihood," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22, 719–725.

Branch, J. W., Giannella, C., Szymanski, B., Wolff, R., and Kargupta, H. (2012), "In-Network Outlier Detection in Wireless Sensor Networks," *Knowledge and Information Systems*, 34, 23–54.

Breiman, L., Friedman, J., Olshen, R. A., and Stone, C. J. (1984), *Classification and Regression Trees*, Statistics/Probability Series, New York: Chapman and Hall.

Calinski, T. and Harabasz, J. (1974), "A Dendrite Method for Cluster Analysis," *Communications in Statistics Theory and Methods*, 3, 1–27.

Casella, G. and Fuentes, C. (2009), "Testing for the Existence of Clusters," *Statistics and Operations Research Transactions*, 33, 115–146.

Cho, H. Y., Oh, J. H., Kim, K. O., and Shim, J. S. (2013), "Outlier Detection and Missing Data Filling Methods for Coastal Water Temperature Data," *Proceedings of the 12th International Coastal Symposium (Plymouth, England), Journal of Coastal Research, Special Issue No. 65*, 1898–1903.

Culp, M. (2011), "spa: Semi-Supervised Semi-Parametric Graph-Based Estimation in R," *Journal of Statistical Software*, 40, 1–29.

Filzmoser, P., Garret, R., and Reimann, C. (2005), "Multivariate Outlier Detection in Exploration Geochemistry," *Computers & geosciences*, 31, 579–587.

Filzmoser, P. and Gschwandtner, M. (2013), "mvoutlier: Multivariate Outlier Detection Based on Robust Methods," *R package version 1.9.9*.

Fraiman, R., Ghattas, B., and Svarc, M. (2011), "Interpretable Clustering Using Unsupervised Binary Trees," *Arxiv preprint arXiv:1103.5339*, 1–22.

Fraley, C. and Raftery, A. (1998), "How Many Clusters? Which Clustering Method? Answers Via Model-Based Cluster Analysis," *The Computer Journal*, 41, 578–588.

Fraley, C. and Raftery, A. E. (1999), "MCLUST: Software for Model-Based Cluster and Discriminant Analysis," *Journal of Classification*, 16, 297–306.

— (2002), "Model-Based Clustering, Discriminant Analysis, and Density Estimation," *Journal of the American Statistical Association*, 97, 611–631.

Fukunaga, K. (1990), *Introduction to Statistical Pattern Recognition*, San Diego: Academic Press.

Geisser, S. (1964), "Posterior Odds for Multivariate Normal Classifications," *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 26, 69–76.

Hartigan, J. J. and Hartigan, P. P. (1985), "The DIP Test of Unimodality," *The Annals of Statistics*, 13, 70–84.

Hennig, C. (2010a), "Methods for Merging Gaussian Mixture Components," *Advances in Data Analysis and Classification*, 4, 3–34.

— (2010b), "Ridgeline Plot and Clusterwise Stability as Tools for Merging Gaussian Mixture Components," in *Classification as a Tool for Research*, eds. Locarek-Junge, H. and Weihs, C., Berlin: Springer, pp. 109–116.

Hodge, V. J. and Austin, J. (2004), "A Survey of Outlier Detection Methodologies," *Artificial Intelligence Review*, 22, 85–126.

Hubert, L. and Arabie, P. (1985), "Comparing Partitions," *Journal of Classification*, 2, 193–218.

Kothari, V., Wei, I., and Shankar, S. (2013), "Outlier Kinase Expression by RNA Sequencing as Targets for Precision Therapy," *Cancer Discovery*, 3, 280 –293.

Li, J. (2005), "Clustering Based on a Multilayer Mixture Model," *Journal of Computational and Graphical Statistics*, 14, 547–568.

MacQueen, J. (1967), "Some Methods for Classification and Analysis of Multivariate Observations," in *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, eds. Le Cam, L. M. and Neyman, J., California, USA, University of California Press, pp. 281–297.

Moll, V. (2012), "The Stirling Numbers of the Second Kind," in *Numbers and functions: from a classical-experimental mathematician point of view*, American Mathematical Society, pp. 191–209.

Pahuja, D. and Yadav, R. (2013), "Outlier Detection for Different Applications: Review," *International Journal of Engineering*, 2, 1–13.

Peña, D. and Prieto, F. J. (2001), "Multivariate Outlier Detection and Robust Covariance Matrix Estimation," *Technometrics*, 43, 286–310.

Peña, D., Rodriguez, J., and Tiao, G. (2004), "A General Partition Cluster Algorithm," in *COMPSTAT: Proceedings in Computational Statistics: 16th Symposium held in Prague, Czech Republic, 2004*, Springer, pp. 371–379.

Peña, D. and Tiao, G. C. (2006), "The SAR Procedure: A Diagnostic Analysis of Heterogeneous Data," Tech. rep., Universidad Carlos III de Madrid.

Pitman, J. (1996), "Some Developments of the Blackwell-MacQueen Urn Scheme," *Lecture Notes-Monograph Series*, 30, 245–267.

Popović, B., Janev, M., Pekar, D., Jakovljević, N., Gnjatović, M., Sečujski, M., and Delić, V. (2012), "A Novel Split-and-Merge Algorithm for Hierarchical Clustering of Gaussian Mixture Models," *Applied Intelligence*, 37, 377–389.

Rand, W. (1971), "Objective Criteria for the Evaluation of Clustering Methods," *Journal of the American Statistical Association*, 66, 846–850.

Rodriguez, J. (2002), "Contribuciones al Estudio de la Heterogeneidad y la Dependencia," Ph.D. thesis, Universidad Carlos III de Madrid.

Rousseeuw, P. J. (1985), "Multivariate Estimation with High Breakdown Point," in *Mathematical Statistics and Applications, Vol. B*, eds. Grossman, W., Pflug, G., Vincze, I., and Wertz, W., Dordrecht: Reidel, pp. 283 – 297.

Rousseeuw, P. J. and Driessen, K. V. (1999), "A Fast Algorithm for the Minimum Covariance Determinant Estimator," *Technometrics*, 41, 212 – 223.

Tantrum, J., Murua, A., and Stuetzle, W. (2003), "Assessment and Pruning of Hierarchical Model Based Clustering," in *Proceedings of the ninth SIGKDD*, New York, New York, USA: ACM Press, pp. 197–205.

Wei, Y. and McNicholas, P. D. (2012), "Mixture Model Averaging for Clustering and Classification," *arXiv preprint arXiv:1212.5760.*