Nearest-Neighbors Medians Clustering

Daniel Peña[1], Júlia Viladomat[2]* and Ruben Zamar[2]

[1]*Department of Statistics, Universidad Carlos III de Madrid, Spain*

[2]*Department of Statistics, UBC, Vancouver, Canada*

Received 28 October 2010; revised 15 March 2012; accepted 7 April 2012
DOI:10.1002/sam.11149
Published online 3 July 2012 in Wiley Online Library (wileyonlinelibrary.com).

Abstract: We propose a nonparametric cluster algorithm based on local medians. Each observation is substituted by its local median and this new observation moves toward the peaks and away from the valleys of the distribution. The process is repeated until each observation converges to a fixpoint. We obtain a partition of the sample based on the convergence points. Our algorithm determines the number of clusters and the partition of the observations given the proportion α of neighbors. A fast version of the algorithm where only a subset of the observations from the sample is processed is also proposed. A proof of the convergence from each point to its closest fixpoint and the existence and uniqueness of a fixpoint in a neighborhood of each mode is given for the univariate case. © 2012 Wiley Periodicals, Inc. Statistical Analysis and Data Mining 5: 349–362, 2012

Keywords: cluster analysis; local median; nearest neighbors; number of clusters

1. INTRODUCTION

Given a sample of p-dimensional observations drawn from a mixture of g populations, cluster analysis partitions the sample into homogeneous groups according to the populations that generate them. Several cluster algorithms, such as KMEANS [1] or its robust version Partitioning Around Medoids (PAM) [2], require the number of clusters to be specified by the user. Estimation of the number of clusters is one of the most difficult problems in cluster analysis and several approaches have been proposed to deal with this problem. One such approach is to obtain several partitions of the data for different values of g and choose the one that optimizes a given measure of the clusters strength [3]. For instance, MCLUST [4] uses the BIC criteria to choose the number of clusters. A second strategy that can be considered is the partition of the data into many small clusters and merge some of them in a second stage [5]. There are also approaches that extract one cluster at a time [6] and others that try to detect modes or bumps [7].

Recently, a new strategy for the estimation of g has appeared. The idea is to iteratively move the data points toward the cluster centers and to use the number of different limiting points as an estimate for the number of clusters. In this sense, gravitational clustering [8–11] assumes that the data points are particles of unit mass with zero velocity that move toward cluster centers as a result of gravitational forces. Furthermore, mean-shift clustering [12–17] uses kernel functions in density estimation to move data points toward denser areas.

In this paper we also present an algorithm that moves the observations toward their cluster centers, but using the nearest-neighbors approach [18]. In particular, we benefit from the robust properties of local medians and see that they have the ability to move toward the peaks and away from the valleys of the distribution. For each observation we iteratively calculate local medians and see that the sequence of medians converges to a neighborhood of a data mode. We propose a clustering algorithm, ATTRACTORS, that yields a partition of the sample based on the resulting convergence fixpoints. ATTRACTORS is a modification of a similar algorithm, CLUES, proposed by Wang et al. [19]. At each iteration both algorithms identify the neighbors of the target points. An important difference between the two procedures is that in CLUES at each step all the observations are globally updated toward the values of their respective local medians. In ATTRACTORS, on the other hand, we do not perform this global update and so the neighbors are always points from the original sample. The difference is essential in order to

Correspondence to: Júlia Viladomat
(juliavc@stanford.edu)

derive theoretical results because the repeated update of all the points in CLUES makes this procedure much more complex from mathematical and computational points of view. Using the mathematical simplicity of ATTRACTORS (compared with CLUES) we prove the convergence of each point to its closest fixpoint as well as the existence and uniqueness of a fixpoint in the neighborhood of each mode, for the univariate case (see details in Section 3). Our theoretical results shed some light on—and yield some tools for—the choice of the number of neighbors $[n\alpha]$, a key parameter for our algorithm. Specifically, our results link α to the size of the smallest cluster we wish to detect. We believe that choosing α in this way is more intuitive than choosing directly the number g of clusters, the approach taken by many clustering algorithms (see Section 4). Another important improvement over CLUES is that ATTRACTORS allows for a considerable gain in computational efficiency because we can restrict attention to a subset of observations drastically reducing the computational time. Section 5.1 addresses this issue. Finally, unlike CLUES, ATTRACTORS can be easily parallelized.

The rest of the paper is organized as follows. In Section 2 we present the relationship between local medians and cluster analysis. We derive some mathematical properties for the one-dimensional case in Section 3. Section 4 proposes a method to determine the key parameter α based on the theoretical results from Section 3. Section 5 presents the algorithm and introduce a fast modified version. In Sections 6 and 7 we study the performance of the algorithm through real examples and numerical simulations. Finally, we give our conclusions in Section 8. Our mathematical results are all proved in the Appendix.

2. LOCAL MEDIANS AND CLUSTER ANALYSIS

Let X be a p-dimensional random vector with density function f and support S.

DEFINITION 1: The α-nearest-neighbors median at $x \in \mathbb{R}^p$ is defined as $g_\alpha(x) = (m_1, \ldots, m_p)^{\mathrm{T}}$, where m_j is the median of the conditional distribution $X_j \mid X \in B_x$, with B_x being a ball around x such that $P(X \in B_x) = \alpha$.

Several definitions of multivariate median can be found in the literature. Here we use coordinate-wise median for computational ease. Performance and computational issues regarding other definitions of multivariate median are topics of future research interest.

DEFINITION 2: A fixpoint of g_α is any point $x \in S$ such that $g_\alpha(x) = x$.

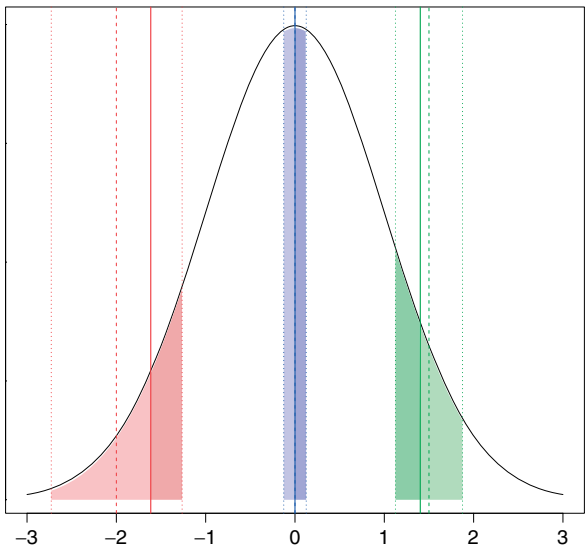

Fig. 1 The median (dotted lines) of the central point of an interval (dashed lines) moves toward the denser areas. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

If x is a fixpoint, the local median of X at x is x. If not, the local median has the property of moving toward the peaks and away from the valleys of f, because it is located at the denser region of B_x. Figure 1 illustrates this main idea when B_x is an interval ($p = 1$).

We will show in the next section that if we iterate this process defining $x_{k+1} = g_\alpha(x_k)$ for any starting value $x_0 \in \mathbb{R}^p$, the sequence $\{x_k\}$ converges to a fixpoint of g_α. If we apply this iteration to each point in $\mathbb{R}^p$, we obtain a partition of $\mathbb{R}^p$ based on where the sequences of local medians have converged to (fixpoints). This motivates the following iteration which is the core of ATTRACTORS, our clustering algorithm described in Section 5.

The iteration: Let $x_1, \ldots, x_n$ be a sample from the random vector X. For given $0 < \alpha < 1$ let $m = [\alpha n]$ be the number of neighbors. For each data point, the algorithm iterates as follows:

$$x_{k+1}^i = \hat{g}_\alpha(x_k^i),$$

starting from $x_0^i = x_i$, and where $\hat{g}_\alpha(x_k^i) = (\hat{m}_1, \ldots, \hat{m}_p)^{\mathrm{T}}$ is the m-nearest-neighbor median at x_k^i, and $\hat{m}_j$ is the median of the jth component of $x_{(1)}, \ldots, x_{(m)}$, the m observations from the sample that minimize the Euclidean distances $\|x_k^i - x_l\|$, $l = 1, \ldots, n$. The iteration stops when $x_{k+1}^i = x_k^i$ for $i = 1, \ldots, n$. This iteration yields a partition of the sample into as many clusters as fixpoints.

Figure 2 illustrates the local medians for a mixture of three normal distributions with means $-4, 0$ and 4 and variance 1. In Fig. 2(a) we show the density function f and

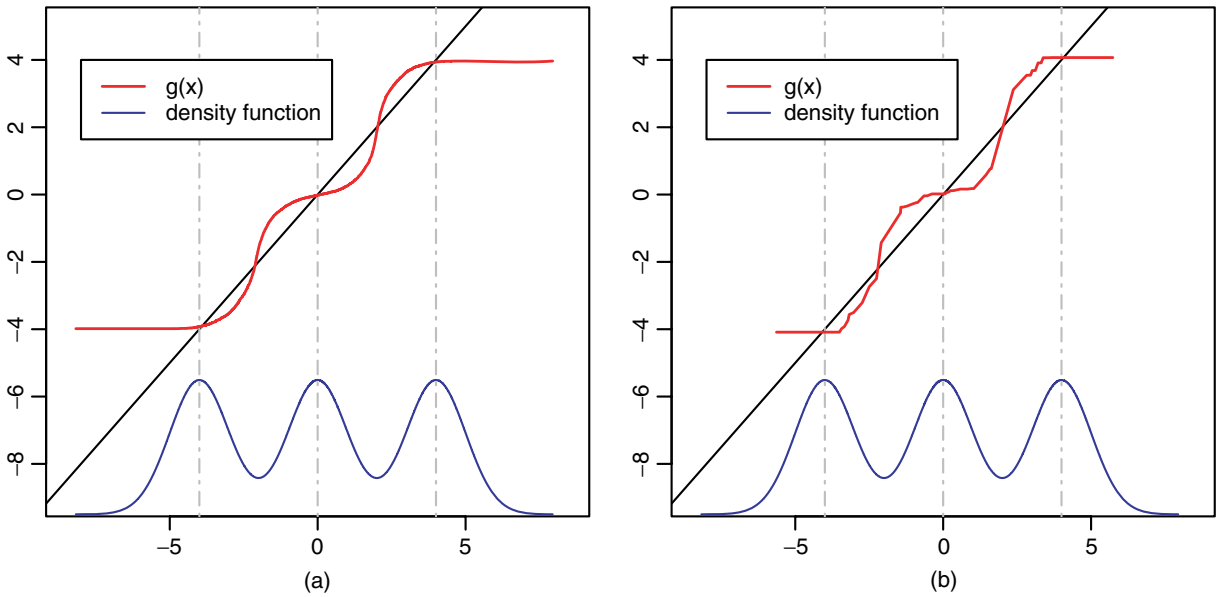

Fig. 2 Function g_α, $\hat{g}_\alpha$ and density function f for a mixture of three normal distributions with means $\mu_1 = -4$, $\mu_2 = 0$ and $\mu_3 = 4$. (a) g_α and f. (b) $\hat{g}_\alpha$. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

the local median function g_α with $\alpha = 1/3$. In Fig. 2(b) we display the estimated function $\hat{g}_\alpha$ from a random sample of size 100 drawn from the normal mixture distribution. The black line corresponds to the identity function $g(x) = x$ and thus every x for which $g_\alpha(x) = x$, corresponds to a fixpoint of $g_\alpha(x)$. In this example, the function g_α has five fixpoints, three of them (attractors) correspond to the three modes. Because the populations are symmetric, the fixpoints coincide with the modes. The other two fixpoints correspond to the two valleys of the mixture distribution and attract no points. Observe that all points in $(-\infty, v_1)$ converge to μ_1, the points in (v_1, v_2) converge to μ_2 and the ones in (v_2, ∞) converge to μ_3, where v_1 and v_2 are the two valleys. One can see that if we draw the sequence of local medians for any given x, it will stop at one of the three modes. For instance, the points in the extremes have already converged after the first iteration ($g_\alpha[(-\infty, \mu_1)] = \mu_1$ and $g_\alpha[(\mu_3, \infty)] = \mu_3$).

3. THEORETICAL RESULTS

The results in this section only apply to the univariate case. Extension to the multivariate case is nontrivial and may require a significant amount of original work. Nonetheless, the guidance and practical conclusions derived from this study of the univariate case are readily applicable to the general case.

We will prove the existence and uniqueness of a fixpoint near each mode and the convergence of each point to its closest fixpoint. These two results guarantee the identification of all the modes in a mixture of distributions.

Let X be a random variable with distribution function F and density function f with convex support S. Given $\alpha \in [0, 1]$, the local median $g_\alpha(x)$ of f at $x \in \mathbb{R}$ is the conditional median defined by the following equations:

$$F(g_\alpha(x)) - F(x - d_x) = \frac{\alpha}{2} \tag{1}$$

where d_x is such that

$$F(x + d_x) - F(x - d_x) = \alpha. \tag{2}$$

Substituting Eq. (2) in Eq. (1), $g_\alpha(x)$ can also be written as

$$g_\alpha(x) = F^{-1}\left[\frac{F(x + d_x) + F(x - d_x)}{2}\right].$$

Following Definition 2, if x is a fixpoint, the local median of f at x is x, the center of the interval. In Theorem 1 we prove that any density with convex support has at least one point with this property.

In the extreme case $\alpha = 1$, the local median of f is the global median, for any $x \in \mathbb{R}$. Therefore, the global median is the unique fixpoint of $g_\alpha(x)$ in this case. Naturally this case is of no interest to us and will not be further considered.

The following results are proved in the Appendix.

THEOREM 1: Let f be a density with convex support S, for $0 < \alpha < 1$, the function g_α has at least one fixpoint.

Theorem 2 below shows that any $x \in \mathbb{R}$ moves toward a fixpoint in the iteration $x_{k+1} = g_\alpha(x_k)$, with $x_0 = x$. Theorem 2 also shows where the sequence $\{x_k\}$ converges. If x_0 is located on a part of f with positive slope, the sequence $\{x_k\}$ converges to the first fixpoint larger than x_0. Similarly, if x_0 is located on a part of f with negative slope, $\{x_k\}$ converges to the first fixpoint smaller than x_0. In summary, the sequence escalates the density function toward the local mode.

THEOREM 2: Let f be a density with convex support S. Consider the iteration

$$x_{k+1} = g_\alpha(x_k).$$

Then, for any starting value $x_0 \in \mathbb{R}$, and for $0 < \alpha < 1$, the sequence $\{x_k\}$ converges to a fixpoint of g_α. In particular, if $x_0 < g_\alpha(x_0)$, $\{x_k\}$ converges to the smallest fixpoint greater than x_0. If $x_0 > g_\alpha(x_0)$, $\{x_k\}$ converges to the greatest fixpoint smaller than x_0.

The next theorem states that, if the distribution is unimodal, the corresponding local median function g_α has only one fixpoint for any $0 < \alpha < 1$.

THEOREM 3: Let f with convex support S be a strictly unimodal density, then, for $0 < \alpha < 1$, the function g_α of f has a unique fixpoint.

The following Corollaries shed light on the actual location of the fixpoints. The smaller the value of α the closer the fixpoint is to the corresponding mode.

COROLLARY 1: Let x_m be the mode of f, then $|F(x^*) - F(x_m)| \leq \frac{\alpha}{2}$, where x^* is the fixpoint.

COROLLARY 2: If $\alpha \to 0$ then $x^* \to x_m$.

DEFINITION 3: x_m is a (δ_1, δ_2)-mode if it is a mode and f is strictly unimodal in the interval $[F^{-1}(y_m - \delta_1), F^{-1}(y_m + \delta_2)]$, where $y_m = F(x_m)$ and $\delta_1, \delta_2 > 0$.

Theorem 4 and its corollary are the main results in this section: for small enough α, there exist a unique fixpoint in the neighborhood of every mode of f. More precisely, if f is strictly unimodal in an interval of weight $\delta_1 + \delta_2$, for any $\alpha \leq \min(\delta_1, \delta_2)$ the identification of the population that induces the mode is guaranteed. Any $x_0 \in \left[F^{-1}(y_m - \delta_1 + \alpha/2), F^{-1}(y_m + \delta_2 - \alpha/2)\right]$ will be attracted by a fixpoint x^*, which assures the existence of a mode in its proximity. Therefore, any population in f characterized by a (δ_i, δ_j)-mode with $\alpha \leq \min(\delta_i, \delta_j)$ will be found. Theorem 4, thus, gives some guidance for the usage of our algorithm.

THEOREM 4: Let x_m be a (δ_1, δ_2)-mode, then, for any $\alpha \leq \min(\delta_1, \delta_2)$, there exists a fixpoint $x^* \in \left(F^{-1}(y_m - \alpha/2), F^{-1}(y_m + \alpha/2)\right)$ and it is the only fixpoint in the interval $[F^{-1}(y_m - \delta_1 + \alpha/2), F^{-1}(y_m + \delta_2 - \alpha/2)]$.

COROLLARY 3: Following Theorems 2 and 4, for any starting value $x_0 \in \left[F^{-1}(y_m - \delta_1 + \alpha/2), F^{-1}(y_m + \delta_2 - \alpha/2)\right]$, the sequence $\{x_k\}$ converges to x^*.

4. THE CHOICE OF α

In practice, α must be chosen by the user. In principle, choosing α sufficiently small guarantees the identification of all the clusters represented by modes, at the population level (Theorem 4 proves this in the univariate setting). However, for finite samples, small values of α could result in spurious fixpoints. It may happen—as observed in our numerical study—that in situations where x is not a fixpoint, just by chance the same number of neighbors can be found on both sides of every entry of x. This would cause the iteration to prematurely stop. These unwanted fixpoints are more likely to occur when α is too small. On the other hand, if α is too large, some interesting clusters may go undetected. Consequently, the choice of α is a trade-off.

Our approach is to set α so that any cluster that represents at least a proportion q of the dataset is identified. Therefore, one must decide *a priori* the size q of the smallest clusters which we would like to detect. Setting α in this way, all clusters of size larger than q should be detected. In other words, using the notation in Theorem 4, we wish to uncover (by identifying the corresponding fixpoint) any (δ_1, δ_2)-mode representing a population of size q. In this context $q = \delta_1 + \delta_2$, and so α has to satisfy the inequality

$$\alpha \leq q - \max(\delta_1, \delta_2).$$

In practice, we generally do not know δ_1 and δ_2, and so we suggest the following procedure:

1. If there is evidence that the clusters are approximately symmetric, set $\alpha = q/2$.

2. If not, hoping that the clusters are not completely left- or right-skewed, set $\alpha = q/3$.

Suppose we wish to detect any cluster with minimal size q. According to the results in Section 3, α should be chosen smaller than the values given in Table 1. It should be noted that these values (chosen so that $\alpha \leq \min(\delta_1, \delta_2)$) are very conservative. Good performances can still be obtained using values of α below these bounds. For instance, in the case

Table 1. Maximum values of α to be able to uncover (δ_1, δ_2)-modes representing a population of size q

q	Symmetric	Not symmetric
0.05	0.025	0.017
0.10	0.05	0.033
0.20	0.10	0.067
0.30	0.15	0.1
0.40	0.20	0.133

of Fig. 2, the size of each component of the mixture is 1/3, and the corresponding δ_1 and δ_2 are all equal to 1/6 for each (symmetric) population. To draw the graph, we have used $\alpha = 1/3$, which does not satisfy the requirements of Table 1. In spite of that all three clusters were well identified.

In view of the above comments, we recommend the use of a relatively small value for α (i.e., between 5 and 20%). The algorithms ATTRACTORS and Fast-ATTRACTORS described in the next section have a second phase to eliminate unwanted fixpoints. Spurious fixpoints are not difficult to identify because they attract just a handful of observations. In this second phase, all fixpoints attracting less than $[\alpha/3n]$ are eliminated, and the observations converging to them are reassigned to the closest remaining fixpoint. In addition, we consider a final step where we merge any pair of clusters if their means are too close in terms of the Mahalanobis distances.

In summary, we believe that choosing α having in mind the size of the smallest cluster one wishes to detect is rather natural and often more appealing than choosing the actual number of clusters. For example, for a sample size of 5000, one may wish to detect any cluster with at least 100 observations, but we may not know anything about the number of clusters, which ranges from 1 to 5000. Suppose, following this example, that there are six clusters of different sizes and the two largest clusters are very close. If we run the algorithm KMEANS with four groups, for example, it will likely merge these two large groups. On the other hand, ATTRACTORS with a small α would identify both groups, and later on, during the merging phase it may consider (but not force) the merging of these two groups. Since the merging decision is based on the Mahalanobis distance between the cluster means it is likely that these two clusters will remain separated.

5. THE *ATTRACTORS* AND *FAST-ATTRACTORS* ALGORITHMS

Let $x_1, \ldots, x_n$ be a sample. The following steps constitute the ATTRACTORS algorithm. Steps 1–3 are the main steps. Steps 4 and 5 are needed to eliminate and merge unwanted spurious clusters.

1. Choose α, the proportion of neighbors and set the number of neighbors m equal to $[\alpha n]$ (here [ ] means integer part).

2. For each observation x_i, $i = 1, \ldots, n$:

   (a) Set $x_0^i = x_i$ and for $k > 0$.

   (i) Calculate the local median at x_k^i, $x_k^i = \hat{g}_\alpha(x_{k-1}^i)$.

   (ii) If $x_k^i \neq x_{k-1}^i$ set $k = k + 1$ and return to (i). Otherwise set $\phi(x_i) = x_k^i$, the fixpoint to which the sequence $\{x_k^i\}$ has converged.

3. Let $x_1^*, \ldots, x_g^*$ be the elements of $\bigcup_{i=1}^n \{\phi(x_i)\}$. For each $t = 1, \ldots, g$, set $G_t = \{x_i \mid \phi(x_i) = x_t^*\}$. That is, G_t is the set of observations attracted by the fixpoint x_t^*.

4. For each $j = 1, \ldots, g$, discard x_j^* if $|G_j| < G_{\text{low}}$, where in general $|A|$ equals the number of elements in the set A and $G_{\text{low}} = [\alpha/3n]$. In this case, update the number of fixpoints, g, and reassign the elements of G_j to the closest cluster G_t. To determine the closest cluster G_t minimize the Mahalanobis distances $\text{MD}(\overline{x}_{G_j}, \overline{x}_{G_t}, S_{G_t})$ over $t = 1, \ldots, g$. Replace x_t^* by the weighted mean between x_j^* and x_t^*.

5. For each pair of clusters (j, t), $j = 1, \ldots, g$, $t > j$, calculate the Mahalanobis distance $\text{MD}(\overline{x}_{G_j}, \overline{x}_{G_t}, S_{j,t})$, where $S_{j,t}$ is the covariance matrix for the largest of the two clusters.
   Sort the distances by ascending order and consider the pair (j, t) with minimum distance. If $\text{MD}(\overline{x}_{G_j}, \overline{x}_{G_t}, S_{G_j}) < \chi^2_{0.9}$, merge the groups G_j and G_t, replace x_j^* by the weighted mean of x_j^* and x_t^* and iterate 5. Otherwise stop.

The last step of the algorithm uses a measure of distance between groups to decide whether to merge them or not. For every pair of groups, it calculates the Mahalanobis distance between their means, using the covariance matrix of the largest one. After all the distances are calculated, it merges the pair with minimum distance if that distance is smaller than a chi-squared threshold. It then proceeds to recalculate the distances, repeating the procedure until the smallest distance between groups is large enough.

5.1. Improving the Computational Efficiency

ATTRACTORS determines the neighbors and calculates local medians several times for each observation until

convergence. When n is large, this step can be time consuming. Therefore, we propose a modified version of the algorithm, Fast-ATTRACTORS, where only a subset of 'sampled observations' is considered. In this paper the sampled observations are chosen randomly, but other approaches could also be considered. A key issue is to decide the size n_{sub} for the set of sampled observations. Let x^* be a fixpoint attracting a proportion $q > 0$ of the data points. Thus, the probability of a randomly chosen data point to converge to x^* is q. Therefore, the probability of a consecutive sampled observations not converging to x^* is $(1-q)^a$. If a tends to ∞, $(1-q)^a$ tends to 0. Hence there exist N such that for any $a > N$ the probability $(1-q)^a$ is arbitrarily small. Therefore, if after sampling N consecutive observations none of them has converged to a new fixpoint x^* we can assume such x^* does not exist. We set $\gamma = (1-q)^N$ to be very small and so determine $N = \log(\gamma)/\log(1-q)$, where q is the minimum size for a fixpoint to be considered of interest, in the sense that we do not mind missing fixpoints attracting less than a proportion q of points (see Section 4). The procedure starts sampling observations and marking to which fixpoint they have converged using a counter to keep track of the number of consecutive observations that converge to old fixpoints. Whenever an observation converges to a *new fixpoint* (a fixpoint appears for the first time) we set the counter to zero. If we find N consecutive observations converging to 'old' fixpoints, that is, if the counter reaches the value N, we stop sampling. Each non-sampled observation is then assigned its closest fixpoint. Note that the number of sampled observations will be n_{sub}.

Depending on the values of q, γ and the sample size n, we may encounter n_{sub} to be larger than n. In this case, all observations are treated and we experience no improvement in computational efficiency. However, this only happens for small datasets, where we do not have any problem to begin with. On the other hand, when n is large, $n - n_{\text{sub}}$ also tends to be large and the efficiency gains are significant.

Finally, an attractive by-product of Fast-ATTRACTORS is that, since not all observations are sampled, fewer spurious fixpoints are found (observed in our numerical experiments).

Fast-Attractors *algorithm:* Let $x_1, \ldots, x_n$ be a sample. The following steps implement the fast version of the algorithm.

1. Set γ to a very small value and choose q to be the maximum size for a cluster. Choose α, the proportion of neighbors. Set $N = \frac{\log(\gamma)}{\log(1-q)}$, $m = [\alpha n]$ to be the number of neighbors, $s = 0$ to be the counter and $i = 1$. Order the n observations randomly.

2. While $s < N$ and $i \leq n$ repeat the following:

   (a) Let $x_0^i = x_i$ and $k = 0$.

      (i) Calculate the local median at x_k^i, $x_{k+1}^i = \hat{g}_\alpha(x_k^i)$.

      (ii) If $x_k^i \neq x_{k+1}^i$ set $k = k+1$ and return to (i). Otherwise $\phi(x_i) = x_k^i$ is the fixpoint where the sequence $\{x_k^i\}$ converges.

      (iii) If $\phi(x_i) \in \Phi$ then $s = s+1$. Otherwise set $s = 0$ and $\Phi = \Phi \bigcup \{\phi(x_i)\}$. Set $i = i+1$.

3. Set $n_{\text{sub}} = i - 1$ and let $x_1^*, \ldots, x_g^*$ be the elements of Φ. For each $j = 1 : g$, define the group $G_j = \{x_i \mid \phi(x_i) = x_j^*\}$ as the set of observations attracted by the fixpoint x_j^*, where $i = 1 : n_{\text{sub}}$.

4. For each $i = n_{\text{sub}} + 1 : n$, assign x_i to the cluster G_j, where j is such that the Euclidean distance $\|x_i - \overline{x}_{G_j}\|$ is minimized, for $j = 1 : g$.

5. Apply additional steps 4 and 5 of the previous ATTRACTORS algorithm.

Note that the algorithm does not need to choose q and α independently, and therefore only requires one input parameter.

6. EXAMPLES

We start by illustrating the behavior of ATTRACTORS on two well-known examples from the literature.

The Ruspini dataset—described by Ruspini [20]—is a two-dimensional example consisting on 75 observations divided into four well-separated clusters. Figure 3(a) shows the true partition. Figure 3(b)–3(d) shows the first, second and third iterations of ATTRACTORS with $\alpha = 0.2$. The gray points represent the original observations and the colored points represent the corresponding local medians. Figure 3(d) shows that the 75 sequences have all converged to four different fixpoints, perfectly identifying the four clusters. Notice that in this case the additional steps of the algorithm to avoid spurious fixpoints are not needed.

The Iris dataset—described by Fisher [21]—consists of 50 flowers from each of three species: *Iris setosa*, *Iris versicolor* and *Iris virginica*. The four variables are the length and the width of the sepal and petal, respectively. One of the species is easily separable from the other two, which tend to overlap and therefore are harder to separate. Figure 4(a) displays the observations in the

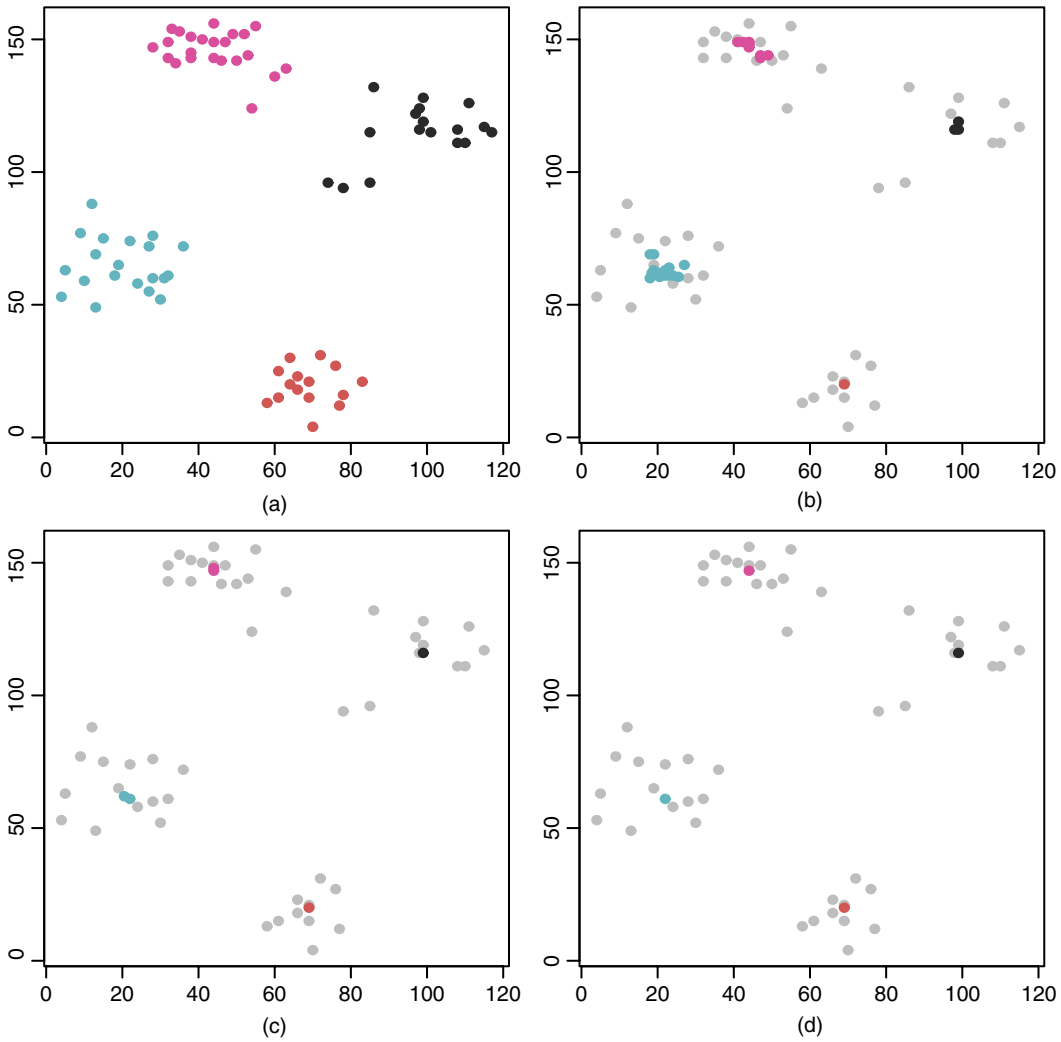

Fig. 3 Ruspini data and the local medians (colored points) after three iterations of ATTRACTORS with $\alpha = 0.2$. (a) Original observations; (b) 1st iteration: 27 different local medians; (c) 2nd iteration: 6 different local medians and (d) 3rd iteration: 4 fixpoints. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

space spanned by the variables 'sepal-width' and 'petal-length'. We apply ATTRACTORS as well as the algorithms MCLUST, CLUES, HIERARCHICAL, KMEANS and MEANSHIFT to these data (we give details on the packages used and their specifications at the beginning of the next section (Section 7)). ATTRACTORS performs very well, with only three flowers being misclassified (see Fig. 4(b)). MCLUST assumes that the sample comes from a mixture of elliptical distributions and estimates the parameters following a model-based clustering approach. It repeats the process for different number of clusters, choosing the number that maximizes the BIC criteria. Figure 4(c) shows the MCLUST results. The algorithm does not perform well, merging the overlapping clusters. CLUES obtains partitions of the sample for different number of clusters and chooses the best, according to a measure of clusters strength. Notice that although MCLUST and CLUES are very different clustering procedures, they have similar approaches for determining the number of clusters. The implementation of CLUES in R allows for the choice between the Silhouette index [2] and the Calinski and Harabasz [22] index. From our experiments, we have found that the Silhouette index outperforms the CH-index most of the time. This behavior is also observed with the Iris dataset, where 15 flowers are misclassified using Silhouette, whereas the CH-index gives the same results as MCLUST, merging the two overlapping groups. HIERARCHICAL with *complete* linkage finds four clusters instead of three; with *single* linkage finds two clusters; with *average* linkage it finds the three groups but misclassifies 14 flowers. MEANSHIFT finds two groups (merging the two overlapping clusters). Finally, KMEANS identify the 3 clusters but misclassifies 16 observations.

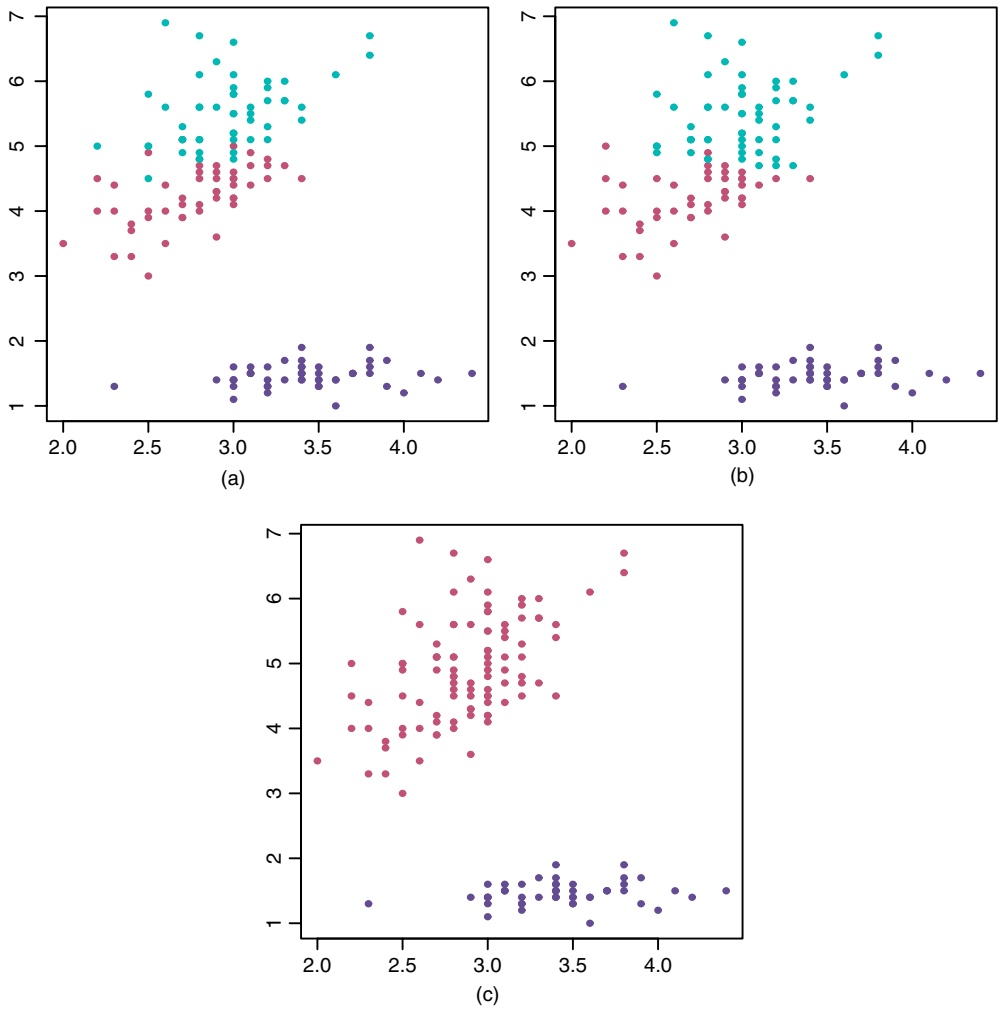

Fig. 4 Iris data in the space spanned by the variables 'sepal-width' and 'petal-length'. Results obtained by MCLUST and ATTRACTORS ($\alpha = 0.1$). (a) True clusters, (b) ATTRACTORS and (c) MCLUST. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Finally, we use the simulated data in Fig. 5 ($n = 1000$) to illustrate the behavior of Fast-ATTRACTORS. ATTRACTORS with $\alpha = 0.10$ returns no misclassification error. On the other hand, Fast-ATTRACTORS with $q = 0.1$ and $\gamma = 0.001$, misclassifies 14 observations. The algorithm samples only $n_{\text{sub}} = 160$ observations, 1/6 of the whole sample, considerably reducing the computational time. In this case, since $q = 0.1$ and $\gamma = 0.001$, the stop-sampling parameter is $N = 66$ (required number of consecutive observations not revealing a new fixpoint).

7. NUMERICAL RESULTS

In this section we investigate the properties of ATTRACTORS using Monte Carlo simulations. We run 100 replications for each sampling situation and consider two main settings. First we generate samples from mixtures of g multivariate normal distributions ($g = 2, 4, 8$) with different means and scatter matrices. The considered dimensions are $p = 4, 8, 15$. The number of observations in each cluster is determined randomly, but ensuring that the total sample size is equal to $n = 100p$ and that each cluster contains a minimum of $p + 1$ observations. The means for the normal distributions are chosen at random as values from a multivariate normal distribution with mean zero and covariance matrix $f\sqrt{p}I$. The factor f is selected so that the probability of overlapping between groups is roughly equal to 1%, see Table 1 in ref. 23. The covariance matrices are different for each cluster and randomly generated using the formula $S = UDU^{\mathrm{T}}$, where U is a random orthogonal matrix and D is a diagonal matrix, its diagonal elements are independent uniform random variables on $[10^{-3}, 5\sqrt{p}]$.

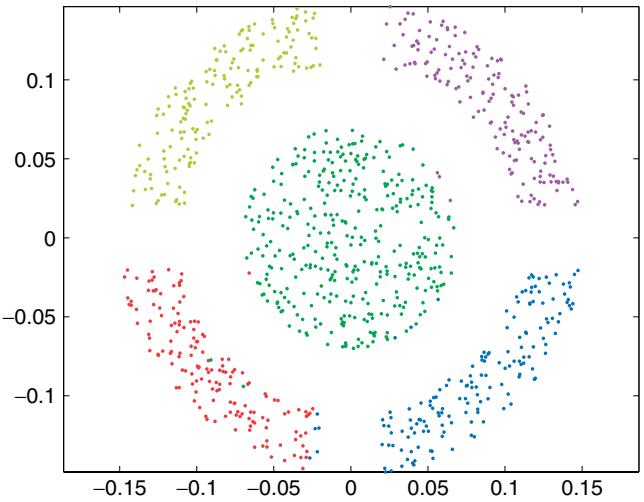

Fig. 5 Partition of the dataset using fast-ATTRACTORS algorithm with $\alpha = 0.1$. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

We also consider mixtures of non-normal distributions. In this case, the clusters are generated using independent Student's-t random variables with 2 degrees of freedom. Each variable is multiplied by the factor $2f/\sqrt{p}$. See ref. 23 for details on the scaling factor f. The clusters now are non-elliptical and star shaped. The cluster sizes and centroids are randomly selected, as in the normal case.

We compare the following clustering algorithms:

- ATTRACTORS, with $\alpha = 0.05$. We also consider other values of α (see below) but $\alpha = 0.05$ has consistently given the best results in our simulations.
- MCLUST, implemented by the R-package *mclust*.
- CLUES, implemented by the R-package *clues* [27].
- HIERARCHICAL, hierarchical agglomerative clustering, implemented by the R-package *hclust*.
- KMEANS, implemented by the R-package *kmeans*.
- MEANSHIFT, moves data points toward denser areas in the dataset(such as CLUES and ATTRACTORS). Implemented by the R-package Local Principal Curve Methods (LPCM).
- KURTOSIS, which uses projections that optimize the kurtosis coefficient to identify the clusters [23]). We use the Matlab implementation provided by the authors.

As implemented in *clues*, the algorithm CLUES can use the silhouette index [2] or the Calinski and Harabasz [22] index to determine the final number of clusters. We report the results using both procedures. Also, MEANSHIFT implementation has a built-in function (select.self.coverage) to select the kernel bandwidth for its density estimator. This function returns, for the given data, the best option for the bandwidth. Details are found in the study by Einbeck [24]. On the other hand, HIERARCHICAL and KMEANS need the number of clusters k to be specified by the user. Hence we apply the algorithms for different values of k, ranging from 2 to 14, and use the Calinski and Harabasz [22] index to decide the number of clusters. Milligan and Cooper [25] compare several measures of clusters strength and conclude that the best performance is obtained by the Calinski and Harabasz [22] index, which is defined as $[\mathrm{tr}(B)(n-k)]/[\mathrm{tr}(W)(k-1)]$, where B and W are the between and pooled within cluster sum of squares. In addition to that, HIERARCHICAL requires the choice of a dissimilarity measure. We report the results for the three most commonly used measures: *single, complete and average linkage*. Finally, we run MCLUST for several number of clusters as well (1–14) and the Mclust function itself chooses the one that maximizes the BIC criteria.

We wish to assess the performance of the different methods regarding (i) their ability to estimate the number of clusters and (ii) their ability to find the clusters themselves (clusters strength). We use the Hubert and Arabie's [26] adjusted Rand index to measure the clusters strength, as suggested by a Referee. Similar conclusions are achieved using the percentage of misclassified observations. The adjusted Rand index ranges between 0 and 1, with 1 corresponding to a perfect match between the estimated and true partitions. We also report the percentage of samples for which the estimated and the true number of clusters coincide.

Table 2 gives our results for 'clusters strength' and Table 3 gives our results for 'number of clusters'. Each entry in these tables is an average over 100 replications. Looking at Table 2, notice that ATTRACTORS performs very well under elliptical and non-elliptical distributions, showing robustness against different cluster shapes. MCLUST does very well in the normal case—as expected because MCLUST was designed to estimate mixtures of elliptical distributions. However, MCLUST's performance deteriorates in the Student's-t case (also as expected). HIERARCHICAL, MEANSHIFT and KURTOSIS perform very well in the normal mixture case. For mixtures of Student's-t distributions their performance considerably deteriorates, especially for large p. The results for KMEANS and CLUES are relatively weak when compared with the top performers in the normal case. In the Student's-t case they share the second best performance after ATTRACTORS. Similarly, KURTOSIS performs very well in the normal case but not so well

Table 2. Hubert and Arabie adjusted Rand index.

	$p = 4$			$p = 8$			$p = 15$		
Normal mixtures	$g = 2$	$g = 4$	$g = 8$	$g = 2$	$g = 4$	$g = 8$	$g = 2$	$g = 4$	$g = 8$
ATTRACTORS	1.00	0.99	0.99	0.99	0.99	0.99	0.98	0.99	0.99
MCLUST	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99	0.99
KMEANS	1.00	0.75	0.76	1.00	0.77	0.73	1.00	0.73	0.73
HC_{average}	1.00	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00
HC_{complete}	1.00	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00
HC_{single}	1.00	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00
MEANSHIFT	0.99	0.99	0.99	1.00	1.00	1.00	1.00	1.00	1.00
CLUES_{CH}	0.72	0.80	0.79	0.78	0.81	0.79	0.77	0.84	0.77
$\text{CLUES}_{\text{Silhouette}}$	0.71	0.78	0.78	0.77	0.84	0.85	0.77	0.86	0.85
KURTOSIS	0.93	0.96	0.97	0.94	0.98	0.98	0.96	0.98	0.99
Student's-t mixtures	$g = 2$	$g = 4$	$g = 8$	$g = 2$	$g = 4$	$g = 8$	$g = 2$	$g = 4$	$g = 8$
ATTRACTORS	0.99	0.99	0.98	0.98	0.99	0.98	0.99	0.98	0.98
MCLUST	0.20	0.37	0.49	0.16	0.34	0.44	0.13	0.32	0.43
KMEANS	0.97	0.81	0.84	0.92	0.86	0.87	0.93	0.84	0.89
HC_{average}	0.97	0.95	0.96	0.90	0.91	0.88	0.04	0.18	0.15
HC_{complete}	0.97	0.95	0.97	0.92	0.94	0.95	0.70	0.81	0.70
HC_{single}	0.97	0.91	0.91	0.79	0.82	0.69	0.05	0.13	0.06
MEANSHIFT	0.92	0.85	0.89	0.85	0.82	0.86	0.50	0.45	0.39
CLUES_{CH}	0.74	0.78	0.78	0.76	0.82	0.85	0.73	0.83	0.82
$\text{CLUES}_{\text{Silhouette}}$	0.74	0.82	0.82	0.75	0.84	0.87	0.72	0.84	0.87
Kurtosis	0.57	0.74	0.83	0.47	0.65	0.75	0.40	0.58	0.65

Table 3. Percentage of times the estimated and true number of cluster coincide.

	$p = 4$			$p = 8$			$p = 15$		
Normal mixtures	$g = 2$	$g = 4$	$g = 8$	$g = 2$	$g = 4$	$g = 8$	$g = 2$	$g = 4$	$g = 8$
ATTRACTORS	100	91	59	99	91	54	98	92	52
MCLUST	99	99	95	98	97	99	97	98	99
KMEANS	100	39	13	100	40	9	100	34	5
HC_{average}	100	99	98	100	100	100	100	100	100
HC_{complete}	100	99	98	100	100	100	100	100	100
HC_{single}	100	99	98	100	100	100	100	100	100
MEANSHIFT	100	98	91	100	100	100	100	100	100
CLUES_{CH}	65	24	17	70	27	10	72	30	15
$\text{CLUES}_{\text{Silhouette}}$	75	19	7	76	20	8	78	31	7
Student's-t mixtures	$g = 2$	$g = 4$	$g = 8$	$g = 2$	$g = 4$	$g = 8$	$g = 2$	$g = 4$	$g = 8$
ATTRACTORS	100	92	60	99	91	60	100	82	57
MCLUST	0	0	0	0	0	1	0	0	2
KMEANS	85	18	9	77	37	8	69	28	10
HC_{average}	42	15	15	9	5	1	80	1	0
HC_{complete}	53	22	14	20	8	5	20	6	0
HC_{single}	29	11	9	19	2	2	76	1	2
MEANSHIFT	15	21	14	2	9	2	1	7	1
CLUES_{CH}	67	24	15	70	29	15	62	27	15
$\text{CLUES}_{\text{Silhouette}}$	73	17	7	76	25	8	72	29	13

in the Student's-t case. The results for CLUES are stable but somewhat weaker, with misclassification rates varying between 9 and 18% for all the considered cases.

Table 3 gives the percentage of samples (replicates) where the estimated and true number of clusters coincide. The results are consistent with those of Table 2, with all methods, except for KMEANS and CLUES doing very well for the normal case but rather poorly in the non-elliptical case. We notice that when g is large, ATTRACTORS has more difficulty finding the correct number of clusters.

Theorem 4 linking the value of α to the size of the smallest detectable cluster suggests that ATTRACTORS should be able to detect clusters with more than 10% of the datapoints. When the mixture has 8 components, the weight of each component on average is 12.5% but since the actual number of observations in each cluster is determined randomly some clusters might contain less than 10% of the data. On the other hand, the average adjusted Rand indexes shown in Table 2 are not affected by this, since the missed clusters are small in size and have little influence on the performance measure.

An important observation is that all the methods included in our simulation study are considerably more computationally intensive than ATTRACTORS because they must evaluate several partitions to find the optimal number of clusters (neighbors, in the case of MEANSHIFT).

Finally, to study the sensitivity of ATTRACTORS to the choice of α we run our algorithm with four different values of α: 0.05, 0.1, 0.2 and 0.3. The results are displayed in Table 4. The performance is uniformly best for $\alpha = 0.05$ and quite stable in the range $0.05 \leq \alpha \leq 0.10$. But it considerably deteriorates for larger values of α.

8. CONCLUSIONS

In summary, ATTRACTORS is a modification of CLUES that provides a robust, computationally efficient and scalable approach to clustering when the number of groups is unknown. Precisely,

1. ATTRACTORS is robust because it does not make any assumptions regarding the cluster shapes and uses robust coordinate-wise medians to move the data points to denser areas in the dataset.

2. ATTRACTORS is mathematically simple and computationally efficient because it does not update all the points at each iteration. Moreover, since ATTRACTORS does not estimate the number of clusters, it has considerable computational advantage over other clustering methods such as KMEANS, MCLUST HIERARCHICAL and MEANSHIFT clustering, which must be run for several cluster sizes to select the 'optimal' one.

3. ATTRACTORS is scalable with respect to the sample size because it has a fast option, *Fast*-ATTRACTORS, that samples a relatively small fraction of the dataset. ATTRACTORS is also scalable with respect to the data dimension because coordinate-wise medians are computationally linear in p.

The main challenges facing ATTRACTORS are the selection of the appropriate neighborhood size α and the appearance of spurious fixpoints in some applications. Specifically,

1. ATTRACTORS does not estimate the number of clusters. Instead, it is aimed at finding all the clusters larger than a certain threshold (i.e., clusters representing at least 5% of the data). This is achieved, in principle, by setting an appropriate value for the neighborhood size α, which is ATTRACTORS only input parameter. We have some partial theoretical results to guide the choice of this key parameter. On the basis of our results and experimental experience, we recommend to use $\alpha = 0.05$, as a practical rule of thumb. But this issue deserves further study and will be the topic of future research.

2. The main challenge facing ATTRACTORS is the appearance of spurious fixpoints. We give a partial solution to this problem by merging small and close clusters (steps 4 and 5 in our algorithm). This point also deserves further research.

Table 4. Hubert and Arabie adjusted Rand index for different values of α.

	$p = 4$			$p = 8$			$p = 15$		
Normal mixtures	$g = 2$	$g = 4$	$g = 8$	$g = 2$	$g = 4$	$g = 8$	$g = 2$	$g = 4$	$g = 8$
$\alpha = 0.05$	1	0.99	0.99	0.99	0.99	0.99	0.98	0.99	0.99
$\alpha = 0.1$	0.98	0.98	0.95	0.95	0.99	0.93	0.93	0.99	0.94
$\alpha = 0.2$	0.93	0.90	0.51	0.94	0.91	0.53	0.87	0.88	0.50
$\alpha = 0.3$	0.85	0.67	0.28	0.91	0.66	0.30	0.80	0.62	0.23
Student's-t mixtures	$g = 2$	$g = 4$	$g = 8$	$g = 2$	$g = 4$	$g = 8$	$g = 2$	$g = 4$	$g = 8$
$\alpha = 0.05$	0.99	0.99	0.98	0.98	0.99	0.98	0.99	0.98	0.98
$\alpha = 0.1$	0.98	0.97	0.92	0.97	0.98	0.92	0.97	0.97	0.90
$\alpha = 0.2$	0.86	0.90	0.45	0.92	0.87	0.34	0.92	0.83	0.16
$\alpha = 0.3$	0.79	0.66	0.24	0.81	0.57	0.13	0.74	0.47	0.02

APPENDIX

PROOF OF THEOREM 1

Proof: From Eq. (1) we have

$$F(g_\alpha(x)) = \frac{\alpha}{2} + F(x - d_x) \geq \frac{\alpha}{2}$$

Similarly, from Eqs. (1) and (2)

$$F(g_\alpha(x)) = F(x + d_x) - \frac{\alpha}{2} \leq 1 - \frac{\alpha}{2}$$

Thus, g_α is bounded by

$$F^{-1}\left(\frac{\alpha}{2}\right) \leq g_\alpha(x) \leq F^{-1}\left(1 - \frac{\alpha}{2}\right). \tag{A.1}$$

Therefore,

$$g_\alpha(x) > x, \text{ for } x < F^{-1}\left(\frac{\alpha}{2}\right)$$
$$\text{and } g_\alpha(x) < x, \text{ for } x > F^{-1}\left(1 - \frac{\alpha}{2}\right)$$

Since F and F^{-1} are continuous, g_α is continuous and thus there exists an $x^* \in (F^{-1}(\frac{\alpha}{2}), F^{-1}(1 - \frac{\alpha}{2}))$ such that $g_\alpha(x^*) = x^*$. ■

PROOF OF THEOREM 2

Proof: In order to prove that g_α is non-decreasing we want to show that $g_\alpha(x) \geq g_\alpha(y)$ if $x > y$. Due to the monotonicity of F^{-1}, it is sufficient to prove that $F(x + d_x) \geq F(y + d_y)$ and $F(x - d_x) \geq F(y - d_y)$. Again, due to the monotonicity of F, it is enough to show

$$\begin{aligned} x + d_x &\geq y + d_y \\ x - d_x &\geq y - d_y. \end{aligned} \tag{A.2}$$

Let us suppose the contrary, $x + d_x < y + d_y$, then $d_x < d_y$ and so $x - d_x < y - d_y$. Therefore

$$\alpha = F(x + d_x) - F(x - d_x) < F(y + d_y) - F(y - d_y) = \alpha, \tag{A.3}$$

which is a contradiction. The proof for the second part of Eq. (A.2) is analogous. The inequality in Eq. (A.3) is strict because it can only be equal if both $F(x + d_x) = F(y + d_y)$ and $F(x - d_x) = F(y - d_y)$, which can happen if the four points are not in S, and that is only possible for the excluded case $\alpha = 1$.
Consider first $x_0 < g_\alpha(x_0) = x_1$, then, since g_α is non-decreasing, $g_\alpha(x_0) \leq g_\alpha(x_1)$. Thus,

$$x_0 < g_\alpha(x_0) = x_1 \leq g_\alpha(x_1) = x_2 \leq \ldots \leq g_\alpha(x_{k-1}) = x_k \leq \ldots$$

since the sequence $\{x_k\}$ is non-decreasing and bounded (see (A.1)), there exists x^* such that $\lim_{k\to\infty} x_k = x^*$. Moreover, x^* is a fixpoint:

$$x^* = \lim_{k\to\infty} x_{k+1} = \lim_{k\to\infty} g_\alpha(x_k) = g_\alpha(\lim_{k\to\infty} x_k) = g_\alpha(x^*)$$

Also, for $x \in (x_k, x_{k+1})$, $g_\alpha(x) \geq g_\alpha(x_k) = x_{k+1} > x$, which means that there are no fixpoints in (x_k, x_{k+1}). Therefore the fixpoint x^* is the smallest fixpoint greater than x_0.

Analogously, if $x_0 > g_\alpha(x_0)$, $\{x_k\}$ converges to the greatest fixpoint smaller than x_0.
If $x_0 = g_\alpha(x_0)$, x_0 is already a fixpoint. ■

PROOF OF THEOREM 3

Proof: In Theorem 1 we proved the existence of at least one fixpoint, for any f. In this proof we deal with its uniqueness for f unimodal.
Suppose there exist two fixpoints $x_1, x_2 \in \mathbb{R}$ such that $x_1 < x_2$. Assume that, without loss of generality, $f(x_1) < f(x_2)$. Otherwise consider the random variable $Y = -X$ with density function $f_Y(x) = f(-x)$ instead.
Let d_1 and d_2 be such that $F(x_1 + d_1) - F(x_1) = F(x_1) - F(x_1 - d_1) = F(x_2 + d_2) - F(x_2) = F(x_2) - F(x_2 - d_2) = \frac{\alpha}{2}$.
Note that $x_1 + d_1 < x_2 + d_2$, otherwise $(x_2, x_2 + d_2) \subset (x_1, x_1 + d_1)$ and, since the integrals of $f(x)$ on these intervals are $\frac{\alpha}{2}$, it is a contradiction because S is a convex support.
When f is a unimodal density

$$f(x) > \min\{f(a), f(b)\}, \text{ for any } a < x < b. \tag{A.4}$$

The following results hold too,

$$f(x) < f(x_1), \text{ for any } x < x_1 \tag{A.5}$$

$$f(x) > f(x_1), \text{ for any } x \in (x_1, x_2) \tag{A.6}$$

the expression (A.6) is due to (A.4).
Observe that

$$f(x_1 + d_1) < f(x_1). \tag{A.7}$$

Indeed, since

$$\alpha/2 = \int_{x_1 - d_1}^{x_1} f(x)dx < d_1 f(x_1),$$

because of (A.5), and

$$\alpha/2 = \int_{x_1}^{x_1 + d_1} f(x)dx > d_1 \min\{f(x_1), f(x_1 + d_1)\},$$

using Eq. (A.4), and we obtain that $\min\{f(x_1), f(x_1 + d_1)\} < f(x_1)$ which leads to (A.7).
This result implies that $x_2 < x_1 + d_1$, otherwise $x_1 < x_1 + d_1 < x_2$, and we know that $f(x_2) > f(x_1) > f(x_1 + d_1)$, which contradicts (A.4).
Therefore, we established the following order

$$x_1 < x_2 < x_1 + d_1 < x_2 + d_2.$$

We will see now that $d_1 > d_2$. In effect,

$$\alpha/2 = \int_{x_1 - d_1}^{x_1} f(x)dx = \int_{x_2 - d_2}^{x_2} f(x)dx,$$

and the values of $f(x)$ in the second integral are larger than in the first, because the expressions (A.5) and (A.6) hold, so the interval of integration should be shorter. Thus, the interval (x_1^+, x_2^+), where $x_1^+ = x_1 + d_1$ and $x_2^+ = x_2 + d_2$, is shorter than (x_1, x_2) because $x_2^+ - x_1^+ = (x_2 - x_1) - (d_1 - d_2) < x_2 - x_1$.

Finally,

$$F(x_2) - F(x_1) > (x_2 - x_1) f(x_1) > (x_2^+ - x_1^+) f(x_1)$$
$$> (x_2^+ - x_1^+) \max_{x \in (x_1^+, x_2^+)} f(x) > F(x_2^+) - F(x_1^+).$$

The first inequality is due to Eq. (A.6), the second due to $d_1 > d_2$, and the third inequality is because $f(x_1) > \max_{x \in (x_1^+, x_2^+)} f(x)$, which is true since (A.7) and the fact that f is strictly decreasing after x_1^+ because the mode of f is in (x_1, x_1^+).

This result leads to a contradiction because $F(x_2) - F(x_1) = F(x_1^+) - F(x_1) - (F(x_1^+) - F(x_2)) = \alpha/2 - (F(x_1^+) - F(x_2)) = F(x_2^+) - F(x_2) - (F(x_1^+) - F(x_2)) = F(x_2^+) - F(x_1^+)$, therefore $x_1 = x_2$ and we have shown that it is not possible to have two distinct fixpoints x_1 and x_2. Therefore, for any unimodal distribution, the function g_α has one and only one fixpoint (the existence was already proved in Theorem 1). ■

PROOF OF COROLLARY 1

Proof: Since x^* is a fixpoint, d_{x^*} is such that

$$F(x^* + d_{x^*}) - F(x^*) = F(x^*) - F(x^* - d_{x^*}) = \frac{\alpha}{2} \qquad \text{(A.8)}$$

Then, x_m must be in $(x^* - d_{x^*}, x^* + d_{x^*})$, otherwise the density is strictly monotonous and the two integrals in Eq. (A.8) can not be equal. Therefore, $|F(x^*) - F(x_m)| \leq \frac{\alpha}{2}$. ■

PROOF OF COROLLARY 2

Proof: From the previous proof,

$$|x_\alpha^* - x_m| < d_{x^*} = F\left(x_\alpha^* + \frac{\alpha}{2}\right) - F\left(x_\alpha^*\right).$$

Since F is continuous, $d_{x^*} \to 0$ as $\alpha \to 0$. Therefore $|x_\alpha^* - x_m| \to 0$ as well. ■

PROOF OF THEOREM 4

Proof: In order to prove the existence of a fixpoint in the interval $\left(F^{-1}(y_m - \frac{\alpha}{2}), F^{-1}(y_m + \frac{\alpha}{2})\right)$, we define

$$\delta_x^- = x - F^{-1}\left(F(x) - \frac{\alpha}{2}\right)$$
$$\delta_x^+ = F^{-1}\left(F(x) + \frac{\alpha}{2}\right) - x$$

which implies that $F(x + \delta_x^+) - F(x) = F(x) - F(x - \delta_x^-) = \frac{\alpha}{2}$. If x is a fixpoint, then $\delta_x^- = \delta_x^+$.

Let $x_l = F^{-1}(y_m - \frac{\alpha}{2})$ be on the left of the mode, then $\delta_{x_l}^- > \delta_{x_l}^+$ because f increases in $\left(F^{-1}(y_m - \delta_1), x_m\right)$. Let also $x_r = F^{-1}(y_m + \frac{\alpha}{2})$, on the right of the mode, then $\delta_{x_r}^- < \delta_{x_r}^+$ because f decreases in $\left(x_m, F^{-1}(y_m + \delta_2)\right)$. Note that $\delta_{x_l}^-$ and $\delta_{x_r}^+$ are contained in $\left[F^{-1}(y_m - \delta_1), F^{-1}(y_m + \delta_2)\right]$, so that proper monotonicity is in place. Therefore, since $\delta_x^+ - \delta_x^-$ is a continuous function of x, because F and F^{-1} are continuous in $\left[F^{-1}(y_m - \frac{\alpha}{2}), F^{-1}(y_m + \frac{\alpha}{2})\right]$, there exist an $x^* \in \left(F^{-1}(y_m - \frac{\alpha}{2}), F^{-1}(y_m + \frac{\alpha}{2})\right)$ such that $\delta_{x^*}^+ = \delta_{x^*}^-$, which implies that x^* is a fixpoint.

Regarding the uniqueness of the fixpoint in $\left[F^{-1}(y_m - \delta_1 + \frac{\alpha}{2}), F^{-1}(y_m + \delta_2 - \frac{\alpha}{2})\right]$, we refer to the proof of Theorem 3. However, we should mention a number of things that changed now. Since we are proving the uniqueness of the fixpoint on a finite interval, we start assuming that x_1 and x_2 are two different fixpoints in $\left[F^{-1}(y_m - \delta_1 + \frac{\alpha}{2}), F^{-1}(y_m + \delta_2 - \frac{\alpha}{2})\right]$. Also, inequalities in Eqs. (A.4) and (A.5) should be restricted to the interval of interest, so that $f(x) > \min\{f(a), f(b)\}$, for any $F^{-1}(y_m - \delta_1) \leq a < x < b \leq F^{-1}(y_m + \delta_2)$, and $f(x) < f(x_1)$, for any $x \in [F^{-1}(y_m - \delta_1), x_1)$. The rest of the proof is exactly the same. We can conclude now that the unique fixpoint in $[F^{-1}(y_m - \delta_1 + \frac{\alpha}{2}), F^{-1}(y_m + \delta_2 - \frac{\alpha}{2})]$ is located in $\left(F^{-1}(y_m - \frac{\alpha}{2}), F^{-1}(y_m + \frac{\alpha}{2})\right)$. ■

REFERENCES

[1] J. A. Hartigan and M. A. Wong, A k-means clustering algorithm, J R Stat Soc [Ser C] 28 (1979), 100–108.

[2] L. Kaufman and P. J. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, New York, John Wiley, 1990.

[3] R. Tibshirani, G. Walther, and T. Hastie, Estimating the number of clusters in a data set via the gap statistic, J R Stat Soc [Ser B] 63 (2001), 411–423.

[4] C. Fraley and A. E. Raftery, Mclust: Software for model-based cluster analysis, J Classif 16 (1999), 297–306.

[5] H. Frigui and R. Krishnapuram, A robust competitive clustering algorithm with applications in computer vision, IEEE Trans Pattern Anal Mach Intell 21 (1999), 450–465.

[6] X. Zhung, Y. Huang, K. Palaniappan, and J. S. Lee, Gaussian mixture modelling, decomposition and applications, IEEE Trans Signal Process 5 (1996), 1293–1302.

[7] M. Y. Cheng and P. Hall, Calibrating the excess mass and dip tests of modality, J R Stat Soc [Ser B] 60 (1998), 579–589.

[8] W. E. Wright, Gravitational clustering, Pattern Recognit 9 (1977), 151–166.

[9] S. Kundu, Gravitational clustering: a new approach based on the spatial distribution of the points, Pattern Recognit 32 (1999), 1149–1160.

[10] Y. Sato, An autonomous clustering technique, In Data Analysis, Classification, and Related Methods, A. L. H. Kiers, J. P. Rasson, P. J. E. Groenen, and M. Schader, eds. Berlin, Springer, 2000.

[11] J. H. Wang and J. D. Rau, VQ-agglomeration: a novel approach to clustering, IEE Proc Vis Image Signal Process 148 (2001), 36–44.

[12] K. Fukunaga and L. D. Hostetler, The estimation of the gradient of a density function, with applications in pattern recognition, IEEE Trans Inform Theory 21 (1975), 32–40.

[13] Y. Cheng, Mean shift, mode seeking, and clustering, IEEE Trans Pattern Anal Mach Intell 17 (1995), 790–799.

[14] D. Comaniciu and P. Meer, Mean shift analysis and applications, In Proceedings of the Seventh International Conference on Computer Vision, 1999, 1197–1203.

[15] D. Comaniciu and P. Meer, Real-time tracking of non-rigid objects using mean shift, IEEE Conf Comput Vis Pattern Recognit 2 (2000), 142–149.

[16] D. Comaniciu and P. Meer, The variable bandwidth mean shift and data-driven scale selection, Proc 8th Int Conf Comput Vis 1 (2001), 438–445.

[17] D. Comaniciu and P. Meer, Mean shift: a robust approach toward feature space analysis, IEEE Trans Pattern Anal Mach Intell 24 (2002), 603–619.
[18] Y. P. Mack and M. Rosenblatt, Multivariate *k*-nearest neighbour density estimates, J Multivariate Anal 9 (1979), 1–15.
[19] X. Wang, W. Qiu, and R. Zamar, CLUES: A non-parametric clustering method based on local shrinking, Comput Stat Data Anal 52 (2007), 286–298.
[20] E. H. Ruspini, Numerical methods for fuzzy clustering, Inf Sci 2 (1970), 319–350.
[21] R. A. Fisher, The use of multiple measurements in taxonomic problems, Ann Eugenic 7 (1936), 179–188.
[22] R. B. Calinski and J. Harabasz, A dendrite method for cluster analysis, Comm Stat 3 (1974), 1–27.
[23] D. Peña and F. J. Prieto, Cluster identification using projections, J Am Stat Assoc 96 (2001), 1433–1445.
[24] J. Einbeck, Bandwidth selection for mean-shift based unsupervised learning techniques: a unified approach via self-coverage, J Pattern Recognit Res 2 (2011), 175–192.
[25] G. W. Milligan and M. C. Cooper, An examination of procedures for determining the number of clusters in a dataset, Psychometrika 50 (1985), 159–179.
[26] L. Hubert and P. Arabie, Comparing partitions, J Classif 2 (1985), 193–218.
[27] F. Chang, W. Qiu, R. Zamar, R. Lazarus, and X. Wang, clues: an R package for nonparametric clustering based on local shrinking, J Stat Softw 33 (2010), 1–16.