A game-inspired algorithm for marginal and global clustering

A game-inspired algorithm for marginal and global clustering

Miguel de Carvalho,[a,b,*] Gabriel Martos[b], and Andrej Svetlošák[a]

[a] School of Mathematics, University of Edinburgh, United Kingdom

[b] Department of Mathematics, CIDMA, Universidade de Aveiro, Portugal

[c] Universidad Torcuato Di Tella, Buenos Aires, Argentina

*Corresponding author at: School of Mathematics, University of Edinburgh, James Clerk Maxwell Building, Edinburgh, EH9 3FD, United Kingdom.

E-mail address: *miguel.decarvalho@ed.ac.uk* (M. de Carvalho)

1 Introduction

Clustering is an unsupervised learning approach for the task of partitioning data into meaningful subsets. The huge literature on cluster analysis is difficult to survey in a few sentences, but a concise description of well-known approaches is offered by [1–3]. True clusters, can be regarded in a number of different ways, each offering unique insights into the underlying patterns and relationships [4]. Examples of mainstream methods for clustering data include model-based (i.e., via mixture models), similarity-based (i.e., via $\mathcal{K}$-means and $\mathcal{K}$-medoids), and hierarchical clustering (i.e., clustering via dendograms).

Model-based clustering is a fast-evolving and intradisciplinary research topic as can be seen from the recent papers of [5, 6], the survey papers of [7–9], and the Handbook on Mixture Analysis [10]. Despite decades of development in similarity-based clustering, $\mathcal{K}$-means algorithms also remains in widespread use, with refined versions continually emerging [11, 12].

In this paper we propose a novel game-inspired method for cluster analysis that lies at the interface of model-based and similarity-based clustering. The proposed approach aims to benefit from the flexibility and soundness of clustering via mixture models, while attempting to mitigate Pitfalls 1 and 2 below. In words, Pitfall 1 refers to an often overlooked aspect of model-based clustering: it usually leads to the same number of clusters for each margin—an assumption that may not align with practical applications. Pitfall 2 constrains the applicability of model-based clustering in high-dimensional data settings.

1.1 Pitfall 1: The single $\mathcal{K}$ problem

The idea of thinking of a cluster as a component of a mixture model has a long tradition in cluster analysis, that has its roots in Tiedeman's work in 1955 [8]. Despite the resilience and flexibility of this paradigm, it is often unnoticed that multivariate model-based clustering may induce the same number of clusters on each margin. For many applied contexts of interest it is however unnatural to believe that all margins should have exactly the same number of components—and hence the same number of marginal clusters. To appreciate this issue, let's revisit the Gaussian finite-mixture model,

$$f(\boldsymbol{x}) = \sum_{k=1}^{\mathcal{K}} \pi_k \phi_d(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k), \quad \boldsymbol{x} = (x_1, \ldots, x_d), \tag{1}$$

where and $(\pi_1, \ldots, \pi_{\mathcal{K}})$ are mixture weights, $\phi_d(\boldsymbol{x}; \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$ is the density function of a d-dimensional multivariate Normal distribution with mean $\boldsymbol{\mu}_k = (\mu_{k,1}, \ldots, \mu_{k,d})$ and variance-covariance matrix

$\mathbf{\Sigma}_k$, with diagonal elements $(\sigma^2_{k,1}, \ldots, \sigma^2_{k,d})$. The marginal distributions stemming from (1) are

$$f_j(x) = \sum_{k=1}^{\mathcal{K}} \pi_k \phi(x; \mu_{k,j}, \sigma^2_{k,j}), \tag{2}$$

for $j = 1, \ldots, d$. As can be seen from (2), model-based clustering as in (1) implies that all margins have $\mathcal{K}$ clusters per margin, except if $\mu_{k,j} = \mu_{k',j}$ and $\sigma_{k,j} = \sigma_{k',j}$ for some $k' \neq k$. Since in practice it is challenging to learn from data if this (i.e., $\mu_{k,j} = \mu_{k',j}$ and $\sigma_{k,j} = \sigma_{k',j}$) holds exactly, we will refer to this challenge as the single $\mathcal{K}$ problem.

1.2 Pitfall 2: Curse of dimensionality

The Gaussian mixture model in (1) has $(\mathcal{K}-1)+\mathcal{K}d+\mathcal{K}d(d+1)/2$ parameters, and hence the number of parameters increases quadratically with d. This shortcoming is well known to limit the scope of application of model-based clustering on high-dimensional data [13]. Some approaches have been developed with the aim of providing a more parsimonious specification, and hence as byproduct this paper will also contribute to that literature. A key paper on parsimonious model-based clustering is that of [14] who suggest a latent Gaussian model that can be regarded as a mixture of factor models.

1.3 Main Contributions

Motivated by these two pitfalls, the main contributions of this paper are as follows:

- We pioneer the development of a model-based solution for the single $\mathcal{K}$ problem outlined in (2), by specifying an individual finite mixture model for each of the margins, but making no assumptions on the joint distribution. The sample space is then partitioned via a strategy game-inspired algorithm, which can be used for clustering data, both marginally as well as in a multivariate fashion.

- We develop a computationally appealing and partially parallelizable model-based approach that bypasses the need to learn about $\mathcal{K}d(d+1)/2$ parameters used in the covariance matrices $\mathbf{\Sigma}_1, \ldots, \mathbf{\Sigma}_{\mathcal{K}}$ required for a 'full' (joint) Gaussian model-based clustering approach.

- The proposed data-driven approach for partitioning the sample space, automatically sieves regions that only have a residual amount of mass—via a minimum entry-level requirement that is specified by the user or set in a data-driven manner. In addition, we assess numerically the proposed methodologies and ascertain the reliability of their clustering performance in a battery of numerical experiments.

- As a byproduct, the proposed method contributes to the literature on game-inspired clustering approaches such as those of [15, 16]. As will be shown below the proposed approach differs, however significantly from that of the previous paper—both in terms of scope (the focus of Bulò and Pelillo is on hypergraph clustering) as well as on the specificities of the game underlying the proposed clustering approach.

1.4 Structure and organization

The remainder of this paper unfolds as follows. In Section 2 we introduce the probabilistic framework underlying the partition of the sample space, which will be the building block of the proposed clustering approach to be introduced in Section 3. Section 4 outlines a conceptualization of a variant of the proposed partitioning approach by reinterpreting it as a strategy game. Experiments with artificial and real data are conducted in Sections 5 and 6, respectively. Final observations and closing remarks are given in Section 7.

2 Reign-and-Conquer partitioning

2.1 The probabilistic framework

A key goal in this section is to devise a partition of the sample space of the joint distribution that is meaningful in a sense to be made more clear below. The proposed framework entails three steps. To streamline the presentation we first focus on the bivariate setting. Comments on the multivariate extension are given in Section 2.2, and Section 4 outlines a game-theoretical variant of the proposed approach. This section does not yet consider data nor estimation, it rather focuses on a probabilistic setup for partitioning a sample space; comments on learning from data based on the principles below are given in Section 3. Here and below, no assumption whatsoever is made on the joint density, and we model each margin using a mixture model. Keeping in mind that any density can be approximated by a mixture of Normals, given enough components, the latter assumption is relatively mild.

Step 1: Margins **(Model-Based Clustering)**

Let $X \sim f_X$ and $Y \sim f_Y$, where

$$f_X(x \mid K_1, \boldsymbol{\Theta}) = \sum_{k=1}^{K_1} \pi_k \, p(x \mid \boldsymbol{\theta}_k), \qquad f_Y(y \mid K_2, \boldsymbol{\Psi}) = \sum_{k=1}^{K_2} \omega_k \, q(y \mid \boldsymbol{\psi}_k). \tag{3}$$

Here, p and q are density functions, with parameters $\boldsymbol{\Theta} = (\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{K_1})$ and $\boldsymbol{\Psi} = (\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_{K_2})$ and $(\pi_1, \ldots, \pi_{K_1})$ and $(\omega_1, \ldots, \omega_{K_2})$ are mixture weights; in addition K_1 and K_2 are the number of clusters respectively associated with the margins X and Y.

Step 2: Reign **(Similarity-Based Joint Protocluster Allocation)**

We first divide the sample space of (X, Y), to be denoted by Ω, via a partition that is based on the set of all marginal cluster means

$$\begin{cases} \mu_X = \{\mu_X^{(1)}, \ldots, \mu_X^{(K_1)}\}, \\ \mu_Y = \{\mu_Y^{(1)}, \ldots, \mu_Y^{(K_2)}\}, \end{cases} \quad \text{where} \quad \begin{cases} \mu_X^{(i)} = \mathrm{E}(X \mid \boldsymbol{\theta}_i) = \int x\, p(x \mid \boldsymbol{\theta}_i)\, \mathrm{d}x, \\ \mu_Y^{(i)} = \mathrm{E}(Y \mid \boldsymbol{\psi}_i) = \int y\, q(y \mid \boldsymbol{\psi}_i)\, \mathrm{d}y. \end{cases} \tag{4}$$

Specifically, to each point $(\mu_X^{(i)}, \mu_Y^{(j)})$ in the Cartesian product

$$\mu_X \times \mu_Y = \{(\mu_X^{(1)}, \mu_Y^{(1)}), \ldots, (\mu_X^{(K_1)}, \mu_Y^{(K_2)})\}, \tag{5}$$

corresponds a Voronoi cell $A_{i,j}$ for $i = 1, \ldots, K_1$ and $j = 1, \ldots, K_2$. We refer to the Voronoi cells $A_{1,1}, \ldots, A_{K_1,K_2}$ as *protoclusters*, as they define a first partition of Ω, and call the sites of $\mu_X \times \mu_Y$ *protocluster centers*.

Step 3: Conquer **(Final Joint Cluster Allocation)**

After dividing Ω we conquer. That is, Step 3 identifies low density protoclusters to be conquered by high density regions, hence refining the naive partition of Ω from Step 2. To avoid including in the resulting partition regions that have a residual amount of mass, a minimum entry-level requirement is chosen to which we refer to as the sieve size $u \in [0, 1]$. Let

$$D_u \equiv \{(i, j) : P(A_{i,j}) \leqslant u\}, \tag{6}$$

be the indices of the protoclusters that have low mass and that hence will be conquered for a given sieve size. The final sample space partition corresponds to the Voronoi cells $B_{i,j}$ associated with the protocluster centers of the conquerors, i.e., $(\mu_X^{(i)}, \mu_Y^{(j)})$ with $(i, j) \in D_u^c$. To assess how the number of final clusters depends on the sieve size, we define the *conquering function* as

$$C(u) = |D_u^c| = K_1 K_2 - |D_u|, \tag{7}$$

where $|D_u|$ denotes the cardinality and D_u^c is the complement of the set D_u. It follows from (6) that although the Reign-and-Conquer partitioning does not involve covariance matrices, it still includes information on the relationship between X and Y through $P(A_{i,j})$.

Example 1 illustrates the main concepts and ideas of the sample space partitioning approach discussed above.

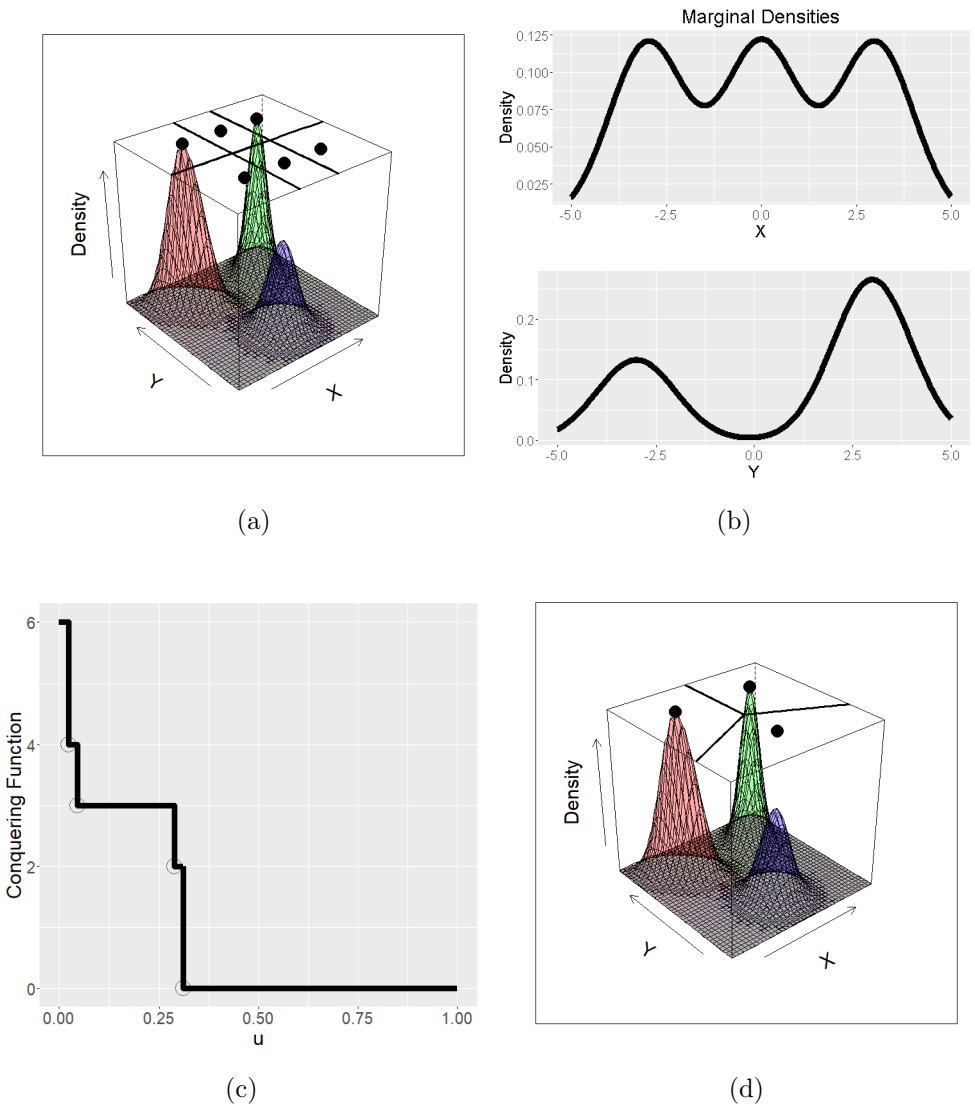


Figure 1: Reign-and-Conquer partitioning for Example 1. (a) protoclusters, protocluster centers (•), and joint density. (b) Marginal densities. (c) Conquering function. (d) The Voronoi cells of the conquerors for $u = 0.1$.

Example 1 (Reign-and-Conquer partitioning on a mixture of 3 bivariate Normal distributions)**.** In Figure 1 (a) we depict a mixture of 3 bivariate Normal distributions as in Equation (1), with means $\boldsymbol{\mu}_1 = (-3, 3)$, $\boldsymbol{\mu}_2 = (3, 3)$, $\boldsymbol{\mu}_3 = (0, -3)$, mixing probabilities $\pi_1 = \pi_2 = \pi_3 = 1/3$, and covariance matrices:

$$\boldsymbol{\Sigma}_1 = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}, \quad \boldsymbol{\Sigma}_2 = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}, \quad \boldsymbol{\Sigma}_3 = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

This set-up leads to a different number of clusters per margin, as can be seen in Figure 1 (a) and (b). Specifically, in the Y–margin there are two clusters with centers in $\mu_Y^{(1)} = 3$ and $\mu_Y^{(2)} = -3$ (i.e., $K_2 = 2$), while in the X–margin there are 3 clusters with centers in $\mu_X^{(1)} = 3$, $\mu_X^{(2)} = 0$ and

$\mu_X^{(3)} = -3$ (i.e., $K_1 = 3$). In Figure 1 (c), we also display the conquering function, and in Figure 1 (d) we depict the Voronoi cells of the conquerors corresponding to the sieve size $u = 0.1$. If we were to regard protoclusters from Step 2 as 'territories, what the Reign-and-Conquer Partition does at Step 3 is conquer low-density cells, allowing the dominant mass regions to annex them.

2.2 d-dimensional extension and theoretical properties

The approach from Section 2.1 extends naturally to a d-dimensional context as follows. For the margins, we now consider $X_1 \sim f_1, \ldots, X_d \sim f_d$ with

$$f_j(x \mid K_j, \boldsymbol{\Theta}_j) = \sum_{k=1}^{K_j} \pi_{k,j}\, p_j(x \mid \boldsymbol{\theta}_k), \tag{8}$$

where the notation in (8) extends that in (3), with $j = 1, \ldots, d$. In particular, (8) implies that the first margin (X_1) has K_1 clusters, that the second margin has K_2 clusters, and so on. The partition underlying the divide step is now formed by the Voronoi tesselation $\{A_{\mathbf{i}} : \mathbf{i} \in I\}$, with $\mathbf{i} = (i_1, \ldots, i_d)$, $I = \{1, \ldots, K_1\} \times \cdots \times \{1, \ldots, K_d\}$, and where the $A_{\mathbf{i}}$ cell corresponds to the protocluster center $(\mu_{X_1}^{(i_1)}, \ldots, \mu_{X_d}^{(i_d)})$, with $\mu_{X_j}^{(k)} = \mathrm{E}(X_j \,|\, \boldsymbol{\theta}_k) = \int x\, p_j(x \,|\, \boldsymbol{\theta}_k)\, \mathrm{d}x$. The final clusters yield from the conquering step correspond to the Voronoi cell $B_{\mathbf{i}}$ associated with the protocluster centers of the conquerors, i.e., $(\mu_{X_1}^{(i_1)}, \ldots, \mu_{X_d}^{(i_d)})$ with $\mathbf{i} \in D_u^c$, where

$$D_u \equiv \{\mathbf{i} \in I : P(A_{\mathbf{i}}) \leqslant u\}, \tag{9}$$

for $u \in [0, 1]$. Similarly as in Section 2.1, (9) implies Reign-and-Conquer partitioning does not involve covariance matrices, it still includes information on the relationships between variables through $P(A_{\mathbf{i}})$.

In the d-dimensional setting the conquering function is more generally defined as

$$C(u) = |D_u^c| = K_1 \times \cdots \times K_d - |D_u|. \tag{10}$$

The conquering function is characterized by a set of properties summarized in the following theorem.

Theorem 1. *The conquering function, $C(u)$ as defined in (10), obeys the following properties:*

a) It is nonincreasing.

b) It is continuous from the left.

c) It is bounded below by $C(1) = 0$ and above by $C(0) = \prod_{j=1}^{d} K_j$.

d) It integrates to one, i.e., $\int_0^1 C(u)\, \mathrm{d}u = 1$.

Proof. See Appendix A. □

It can be noticed that the conquering function from Example 1 verifies all the claims of Theorem 1 as can be seen from Figure 1 (c). In addition to Theorem 1 it can be shown that the conquering function is a step function that has a finite number of steps provided that $K_1, \ldots, K_d$ are finite. See Appendix B.

3 Learning from data

3.1 The Reign-and-Conquer algorithm

We now devise an algorithm based on the probabilistic framework from Section 2. The goal is to allocate observations in a dataset $\{\mathbf{x}_l\}_{l=1}^n$, with $\mathbf{x}_l = (x_{l,1}, \ldots, x_{l,d})^{\mathrm{T}}$ in $\mathbb{R}^d$, into a set of meaningful classes—both in terms of the margins as well as the joint. Using the notation from Section 2.2, we introduce the RC (Reign-and-Conquer) clustering algorithm.

Algorithm 1 RC (Reign-and-Conquer) Clustering

Step 1. Margins: Fit the jth marginal density in (8) using $\{x_{1,j}, \ldots, x_{n,j}\}$, for $j = 1, \ldots, d$, so to learn about $\{(K_j, \mu_{X_j}^{(1)}, \ldots, \mu_{X_j}^{(K_j)})\}_{j=1}^d$.

Step 2. Reign: Learn about the protoclusters $\{A_{\mathbf{i}} : \mathbf{i} \in I\}$ of $\{(\mu_{X_1}^{(i_1)}, \ldots, \mu_{X_d}^{(i_d)}) : \mathbf{i} \in I\}$.

Step 3. Conquer: Learn about the Voronoi cells of the conquerors, $\{B_{\mathbf{i}} : \mathbf{i} \in D_u^c\}$, i.e. that of $\{(\mu_{X_1}^{(i_1)}, \ldots, \mu_{X_d}^{(i_d)}) : \mathbf{i} \in D_u^c\}$, and allocate the lth observation to a cluster using the encoder

$$\mathrm{Enc}(l) = \underset{(i_1,\ldots,i_d)}{\arg\min} \|\mathbf{x}_l - (\mu_{X_1}^{(i_1)}, \ldots, \mu_{X_d}^{(i_d)})\|^2. \tag{11}$$

If we were to regard protoclusters from Step 2 as 'territories,' what RC does at Step 3 is to let the mass dominant regions conquer the low density ones. The RC algorithm warrants some further comments:

- **Step 1. Margins**: To learn about $\{(K_j, \mu_{X_j}^{(1)}, \ldots, \mu_{X_j}^{(K_j)})\}_{j=1}^d$ several approaches can be taken. We use the non-local prior for mixtures approach of [17], but alternatively one could use, for example, RJ MCMC (Reversible Jump Markov Chain Monte Carlo) [18]. Determining K_j is a well-studied yet open problem, and an overview of the literature in this can be found in [19], [20], and [21].

- **Step 2. Reign**: To compute the protoclusters $\{A_{\mathbf{i}} : \mathbf{i} \in I\}$ of $\{(\mu_{X_1}^{(i_1)}, \ldots, \mu_{X_d}^{(i_d)}) : \mathbf{i} \in I\}$, we resort to the cluster centers from Step 1.

- **Step 3. Conquer**: To learn about the cells of the conquerors, we need to learn about $D_u = \{\mathbf{i} \in I : P(A_\mathbf{i}) \leqslant u\}$—and this implies estimating $P(A_\mathbf{i})$. Several approaches can be taken, and here we opt for the simplest one—the maximum likelihood estimator (MLE). To avoid burdening the notation, we introduce the MLE on the bivariate case, but the details extend easily to the multivariate setting. Let $\{\mathbf{x}_l\}_{l=1}^n = \{(x_{l,1}, x_{l,2})\}_{l=1}^n$ and note that the number of points falling on the protoclusters, $n_{i,j} = |\{\mathbf{x}_l \in A_{i,j}\}_{l=1}^n|$ is Multinomial distributed, that is,

$$\mathbf{n} \sim \text{Multinomial}(\mathbf{p}), \tag{12}$$

where $\mathbf{n} = (n_{1,1}, \ldots, n_{K_1,1}, \ldots, n_{1,K_2}, \ldots, n_{K_1,K_2})$ and $\mathbf{p} = (p_{1,1}, \ldots, p_{K_1,1}, \ldots, p_{1,K_1}, \ldots, p_{K_1,K_2})$, with $p_{i,j} = P(A_{i,j})$. Hence, the MLE is $\hat{\mathbf{p}} = \mathbf{n}/n$ and full Bayesian inference can also be easily conducted.[†] Following the principles from Section 2, this estimate implies conquering protoclusters centered at (μ_k, μ_l), for which

$$\frac{n_{k,l}}{n} < u, \quad \text{for } u \in (0, 1]. \tag{13}$$

The estimated regions of the conquerors $B_{k,l}$ are obtained by the Voronoi tessellation on the remaining (μ_k, μ_l) so that $(k, l) \notin D_u \equiv \{(i, j) : P(A_{i,j}) < u\}$.

The RC algorithm combines the paradigms of model-based clustering and similarity-based clustering. Indeed, Step 1 consists of a marginal model-based clustering approach. In addition, just as in similarity-based clustering methods, such as k-means, Step 3 entails an encoder [1], which determines to which cluster observation $\mathbf{x}_l$ belongs to. In theory the Euclidean norm in (11) could be replaced by any preferred norm (e.g., Mahalanobis). Yet, as we show in the supplementary material there are compelling reasons to prefer the Euclidean norm over the Mahalanobis norm.

3.2 Implementation and computing

Some comments on implementation and computing are in order. As mentioned earlier, to avoid including in the resulting partition of the sample space regions that have a residual amount of mass, a minimum entry-level requirement $u \in [0, 1]$ should be set by the user. That value might be set at

[†]Bayesian inference can be conducted by assuming a Dirichlet prior over the unit simplex on $\mathbb{R}^{K_1 K_2}$, i.e., $\mathbf{p} \sim \text{Dirichlet}(\boldsymbol{\alpha})$, where $\boldsymbol{\alpha} = (\alpha_{1,1}, \ldots, \alpha_{K_1,1}, \ldots, \alpha_{1,K_1}, \ldots, \alpha_{K_1,K_2})$. Dirichlet–Multinomial conjugacy then implies that posterior inferences can be obtained from $\mathbf{p} \mid \mathbf{n} \sim \text{Dirichlet}(\boldsymbol{\alpha} + \mathbf{n})$; the above-mentioned MLE corresponds to the MAP (Maximum a Posteriori) with $\boldsymbol{\alpha} = \mathbf{1}_{K_1 K_2}$. Finally, another alternative would be to specify a model for the joint distribution. Yet, given that we only need to learn about the $p_{i,j}$, and since we prefer to avoid specifying a copula that may not accurately describe the true joint distribution, we opt for the above-described likelihood-based approaches.

a fixed low level (say, $u = 0.1$), so that all resulting clusters have at least that mass. Alternatively, data-driven approaches for setting u, based on the fitted conquering function, are also explored in Section 5.

Step 1 of the RC algorithm can be parallelized into d cores, and so to speed up the computations parallel computing was implemented with the R package `parallel` [22]. To fit marginal densities in Step 1, we consider Normal mixture models with non-local priors via the `R` package `mombf` [23]. Other mixture models capable of internally determining the number of components could also have been considered for Step 1. Examples of such models include repulsive point processes [24, 25] and group-sort-fuse procedures [26]. We opt for non-local priors as they are designed to enforce parsimony by penalizing mixtures with a redundant number of components, and they bypass the need for complicated algorithms such as RJ MCMC.

Hence, for each margin in (8) we consider a non-local prior for its mixture model. To ease notation we describe the prior for a general margin with K components. For a mixture with K components, the non-local prior consists of a penalized inverse-Gamma prior given by

$$\mathrm{p}(K, \Theta) = \mathrm{penalty}(K, \Theta) \times \prod_{j=1}^{K} N(\mu_j \mid 0, a\, H_\sigma)\, \mathrm{IG}(\sigma_j \mid b, c).$$

Here, IG is the inverse-Gamma distribution, (a, b, c) are positive hyperparameters, H_σ is the harmonic mean of the scale components (i.e., $H_\sigma = 1/\{K^{-1}\sum_{j=1}^{K}\sigma_j^{-1}\}$), and

$$\mathrm{penalty}(K, \Theta) = \frac{\prod_{1 \leqslant i < j \leqslant K}\{(\mu_i - \mu_j)^2/(aH_\sigma)\}}{\prod_{j=1}^{K}\Gamma(j+1)}, \tag{14}$$

where Γ is the gamma function. As can be seen from (14), the penalized inverse-Gamma prior diminishes the penalty function to zero as two component means approximate each other, thereby lowering the prior probability of models with redundant components and effectively penalizing them. Finally, we place a symmetric uniform Dirichlet prior on the weights of the mixture components.

Steps 2 and 3 of the RC algorithm involve the computation of Voronoi tesselations from out of $a = K_1 \times \cdots \times K_d$ protocluster centers and from $a - |D_u|$ protocluster centers of the conquerors. While parallel algorithms could have been employed also for higher-performance computation of Steps 2 and 3 [27] we have opted for a simple implementation that only parallelizes Step 1.

We close this section with a simple yet important comment. While Step 3 of the RC algorithm leads to multivariate clustering of $\{\mathbf{x}_l\}_{l=1}^{n}$, marginal clustering can be made directly from Step 1 via the posterior probabilities

$$\hat{Z}_{l,k,j} = \frac{\pi_{k,j} p_j(x_{l,j}|\boldsymbol{\theta}_k)}{\sum_{k=1}^{K_j}\pi_{k,j} p_j(x_{l,j}|\boldsymbol{\theta}_k)}. \tag{15}$$

Note that $\hat{Z}_{l,k,j} \in [0,1]$ estimates the cluster membership labels of the lth observation on the jth margin, which are defined as $Z_{l,k,j} = 1$ if the lth observation on the jth margin $x_{l,j}$ belongs to the kth component, or $Z_{l,k,j} = 0$ otherwise.

4 An outline of a game theory conceptualization

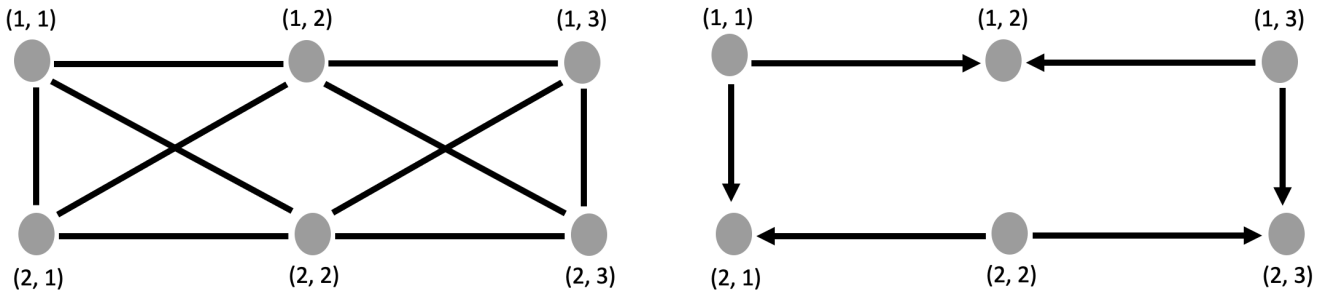
4.1 A game of thrones—starting point

This section outlines an alternative way to look into the sample space partitioning approach from Section 2 as a game. To streamline the presentation we focus on the bivariate case; the extension to the multivariate case is a matter of adjusting notation. More specifically, the game to be considered starts at Step 2 of Reign-and-Conquer Partitioning (Section 2), players are to be understood as K_1K_2 'Kings' owning the protocluster 'territories' ($\{A_{i,j}\}$) and who decide whether or not they will attack their neighbors. To make matters concrete, think of Figure 1 (a) as representing the protocluster 'territories' of $K_1K_2 = 6$ Kings, who have to decide whether or not they attempt to conquer the territories of their neighbors. If a territory is attacked by two Kings, they might have to share the conquered territory.

The neighboring structure of players can be represented via a $K_1K_2 \times K_1K_2$ adjacency matrix $\mathbf{M}$, and it can be visualized using a (undirected) graph $\mathscr{G} = (N, \mathscr{E})$, where $\mathscr{E}$ is a set of edges representing a neighboring relation. The outcome of the game is an element in S (to be defined in Section 4.2), and it can be visualized with a directed graph $\mathcal{G} = (N, \mathcal{E})$, where $\mathcal{E}$ is a set of directed edges or arrows representing attacks. To build intuition surrounding these ideas and concepts, let's revisit Example 1. Figure 2 (a) depicts the graph corresponding to the neighboring structure of the $K_1K_2 = 6$ Kings. The corresponding adjacency matrix is

$$\mathbf{M} = \begin{pmatrix} 0 & 1 & 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 1 \\ 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix}.$$

The directed graph in Figure 2 (b) depicts an example of attack decisions compatible with the outcome from Figure 1 (d). Indeed, for example, we can think of the outcome in Figure 1 (d) as the consequence of players $(1,1)$ and $(2,2)$ attacking player $(2,1)$ and sharing the conquered territory, and so on.



(a) (b)

Figure 2: Revisiting Example 1. (a) Neighboring structure corresponding to Figure 1 (a). In (b) is depicted a directed graph with an instance of attack decisions compatible with the outcome from Figure 1 (d). In both charts the nodes represent players ('Kings') (1, 1), ..., (2, 3).

4.2 Representation, equilibrium, and open challenges

Below, an 'attack' is denoted with a '1', and 'not to attack' with a '0'. The (normal form) game of interest is given by the triple $G = (N, \{S_i\}_{i\in N}, \{U_i\}_{i\in N})$, where:

- $N = \{(i,j) : i = 1,\dots,K_1,\ j = 1,\dots,K_2\}$ is the set of players ('Kings').
- $S_{i,j}$ is the pure set of strategies of King (i,j),

  $$S_{i,j} = \{\text{who to attack, keeping in mind that only neighbors can be attacked}\} \subseteq \{0,1\}^{K_1K_2},$$

  and $S = \underset{(i,j)\in N}{\times} S_{i,j}$ is the set of all vectors of strategies, where '$\times$' is the Cartesian product.
- $U_{i,j}(\mathbf{s})$ is the payoff of King (i,j), with $\mathbf{s} = (\mathbf{s}_{i,j})_{(i,j)\in N}$, with $U_{i,j} : S \to \mathbb{R}$.

By construction, the strategy set of each player is finite and hence this is said to be is a finite game. While a Nash equilibrium for this game may not exist over pure strategies, an equilibrium will exist over mixed strategies. A mixed strategy for player (i,j) is a distribution over their set of pure strategies $S_{i,j}$, that is

$$\mathscr{S}_{i,j} = \left\{\sigma_{i,j} : S_{i,j} \to [0,1] : \sum_{\mathbf{s}_{i,j}\in S_{i,j}} \sigma_{i,j}(\mathbf{s}_{i,j}) = 1\right\}.$$

The celebrated Nash theorem, recalled below for completeness, ensures that the game of interest has at least one equilibrium in mixed strategies.

Theorem 2. *Every finite game in strategic form, that has a finite number of players, has a Nash equilibrium in mixed strategies.*

Proof. See, for example, Maschler et al. [28, Section 5]. □

Conceptually speaking, the approach above endows Step 3 with a much broader range of possibilities on how to partition the sample space Ω. First, there are numerous ways in which the 'incentives' (utility functions) can be set, and in particular they can mimic the ones from Section 2. A refinement of Step 3 based on the principles outlined above is as follows: a) Compute a Nash equilibrium; b) Derive the cells of the conquerors resulting from such equilibrium. In terms of a) we note that computation of Nash equilibria is nontrivial in general, but it can be conducted using simplicial subdivision [29], a Newton method known as Govindan–Wilson algorithm [30], search methods [31], among other. Keeping in mind the computational motivation of the paper, in the numerical experiments to be reported below we focus on the computationally appealing approach from Section 3—that bypass the need for computing Nash equilibria in Step 3—but we aim to revisit the numerical performance of this game-theoretical variant of the proposed methods in future research.

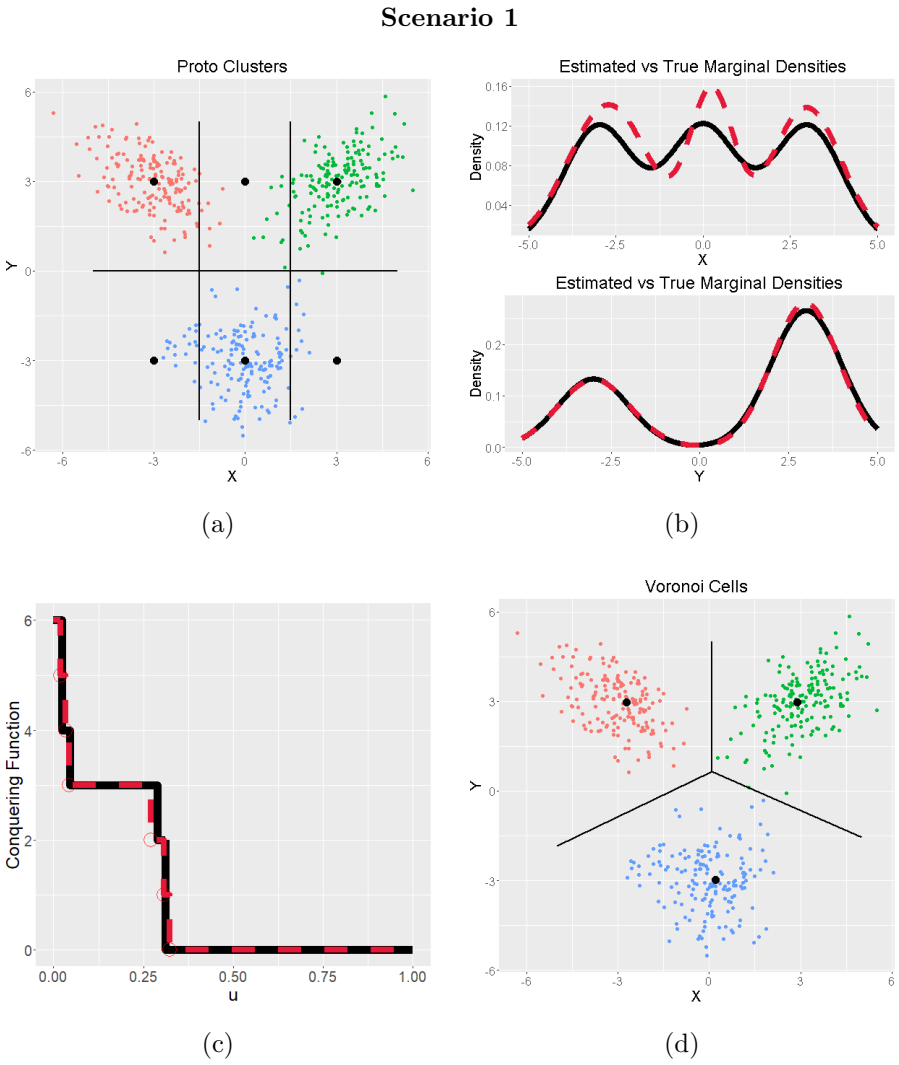


Figure 3: One shot experiments for Scenario 1. (a) Simulated data and protoclusters (Step 1). (b) Estimated (dashed) vs true (solid) marginal densities. (c) Estimated (dashed) vs true (solid) conquering functions. (d) Voronoi cells of the conquerors for $u = 0.1$ (Steps 2 and 3).

5 Numerical experiments on artificial data

5.1 Simulation setup and one shot experiments

In this section we study the performance of the proposed methods via numerical experiments. An exhaustive Monte Carlo simulation study will be presented in Section 5.2. The Monte Carlo simulation will assess two data-driven approaches for setting the sieve size based on the conquering function as well as the strategy of setting a fixed low sieve size (e.g., $u = 0.1$). The data-driven approaches based on the conquering function will be called throughout as the *plateau* (u at which the longest plateau of $C(u)$ ends) and the *edge* (u at which the largest jump on $C(u)$ occurs); see Appendix B for technical details.

Pitfalls 1–2 from Section 1 motivated us to design the following simulation scenarios:

- **Scenario 1**: Data are drawn from a mixture of $\mathcal{K} = 3$ bivariate Normal distributions with weights, mean vectors, and covariance matrices as in Example 1. To study the clustering performance as the sample size increases, we consider $n \in \{50, 100, 250, 500, 1000\}$. In Figure 3 we depict a one shot example of Reign-and-Conquer algorithm corresponding to a sample of size $n = 500$ and $u = 0.1$; as can be seen in Figure 3 (d), the proposed method suitably partitions the multivariate data.

- **Scenario 2**: Data are drawn from a mixture of $\mathcal{K} = 3$ Clayton copulas [32, Chapter 4.2] with margins: $f_X(x) = \phi(x; -5, 4^2)/2 + \phi(x; 3, 4^2)/2$, and $f_Y(y) = \phi(x; -5, 1)/3 + \phi(x; 2.5, 1)/3 + \phi(x; 5, 1)/3$. Trivially, the joint distribution does not obey (1), and the number of clusters per margin is different ($K_1 = 2$, $K_2 = 3$). Here, we also consider sample sizes $n \in \{50, 100, 250, 500, 1000\}$, and in Figure 4 we depict the outcome of a one shot experiment with $n = 500$. As can be seen in Figure 4 (d), the proposed method suitably partitions the multivariate data.

- **Scenario 3**: Data are drawn from a mixture of d-variate Normal distributions in dimensions $d \in \{5, 10, 15, 20\}$ (moderate high dimensional data). In this scenario, the sample sizes and the number of clusters depend on d in the following way: $n = \lfloor 10d^{3/2} \rfloor$ and $\mathcal{K}_d \equiv \mathcal{K} = \texttt{round}(\sqrt{d+1})$, where $\lfloor \cdot \rfloor$ and $\texttt{round}()$ denotes the the floor and round functions respectively. The covariance matrices and mixing probabilities are specified as $\mathbf{\Sigma}_k = \mathbf{I}_d$ and $\pi_k = 1/d$, for $k = 1, \ldots, \mathcal{K}$, whereas the mean vectors $\mu_k = (\mu_1^{(k)}, \ldots, \mu_d^{(k)})$ are sparsely defined: $\mu_i^{(k)} = 0$ for $i \neq k$ and $\mu_k^{(k)} = d/\sqrt{2}$. This scenario leads to several identical marginal distributions, and the mean vector components are constrained to be equidistant $\|\boldsymbol{\mu}_i - \boldsymbol{\mu}_j\|_2 = d$ for all $j \neq i$; therefore the separation between clusters increases linearly with the number of dimensions.

Motivated by the challenges of clustering data in the presence of skewness and outliers [33, 34], we present in the online supplementary material a series of Monte Carlo experiments exploring additional simulation scenarios. Beyond these, we also examine a scenario with imbalanced mixing proportions.

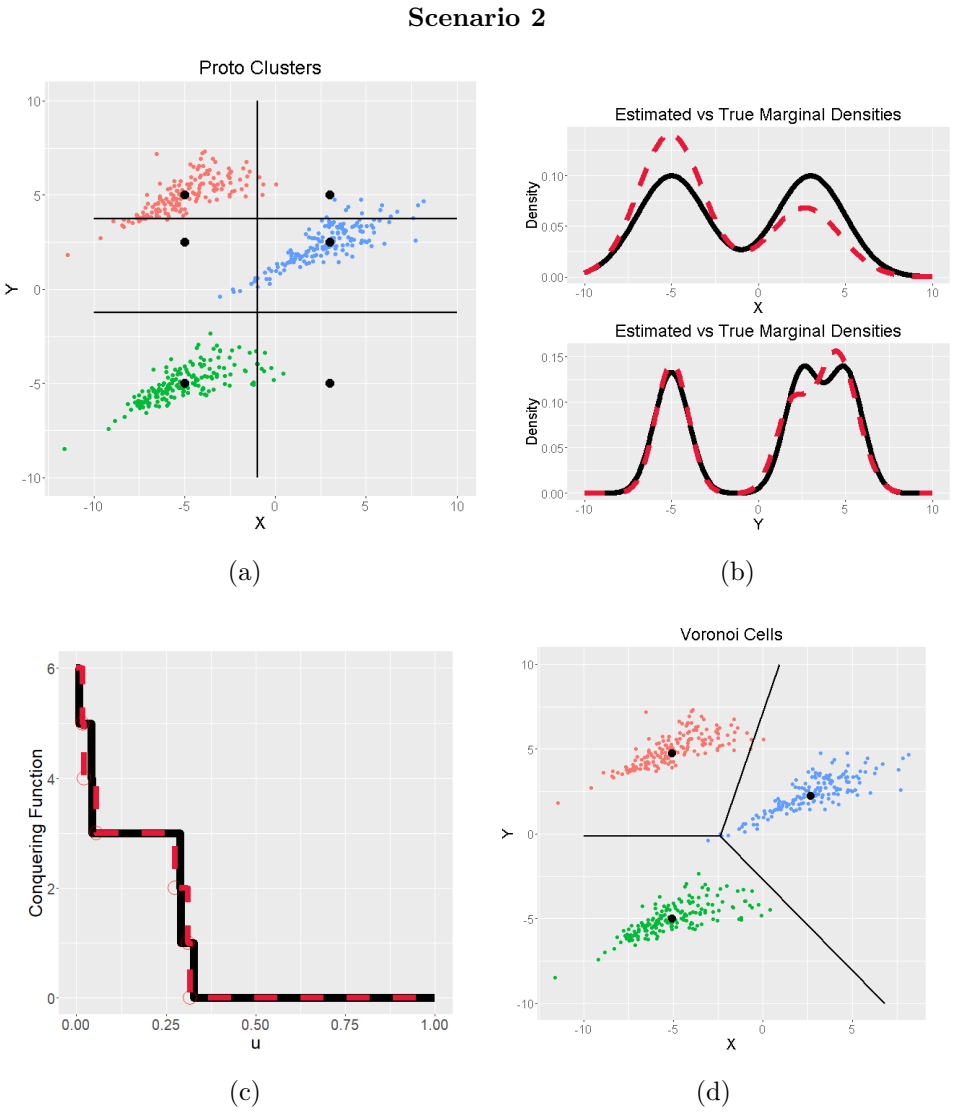


Figure 4: One shot experiments for Scenario 2. (a) Simulated data and protocluster (Step 1). (b) Estimated (dashed) vs true (solid) marginal densities. (c) Estimated (dashed) vs true (solid) conquering functions. (d) Voronoi cells of the conquerors for $u = 0.1$ (Steps 2 and 3).

5.2 Monte Carlo simulation study

To assess the performance of the proposed clustering approach, for Scenarios 1–3 we redo the previous one shot analysis $M = 1000$ times so to estimate the following clustering agreement metrics: Rand and Adjusted Rand Index (RI and ARI respectively), Jaccard Index (JI), and Fowlkes–Mallows Index (FMI); see [35] and references therein.

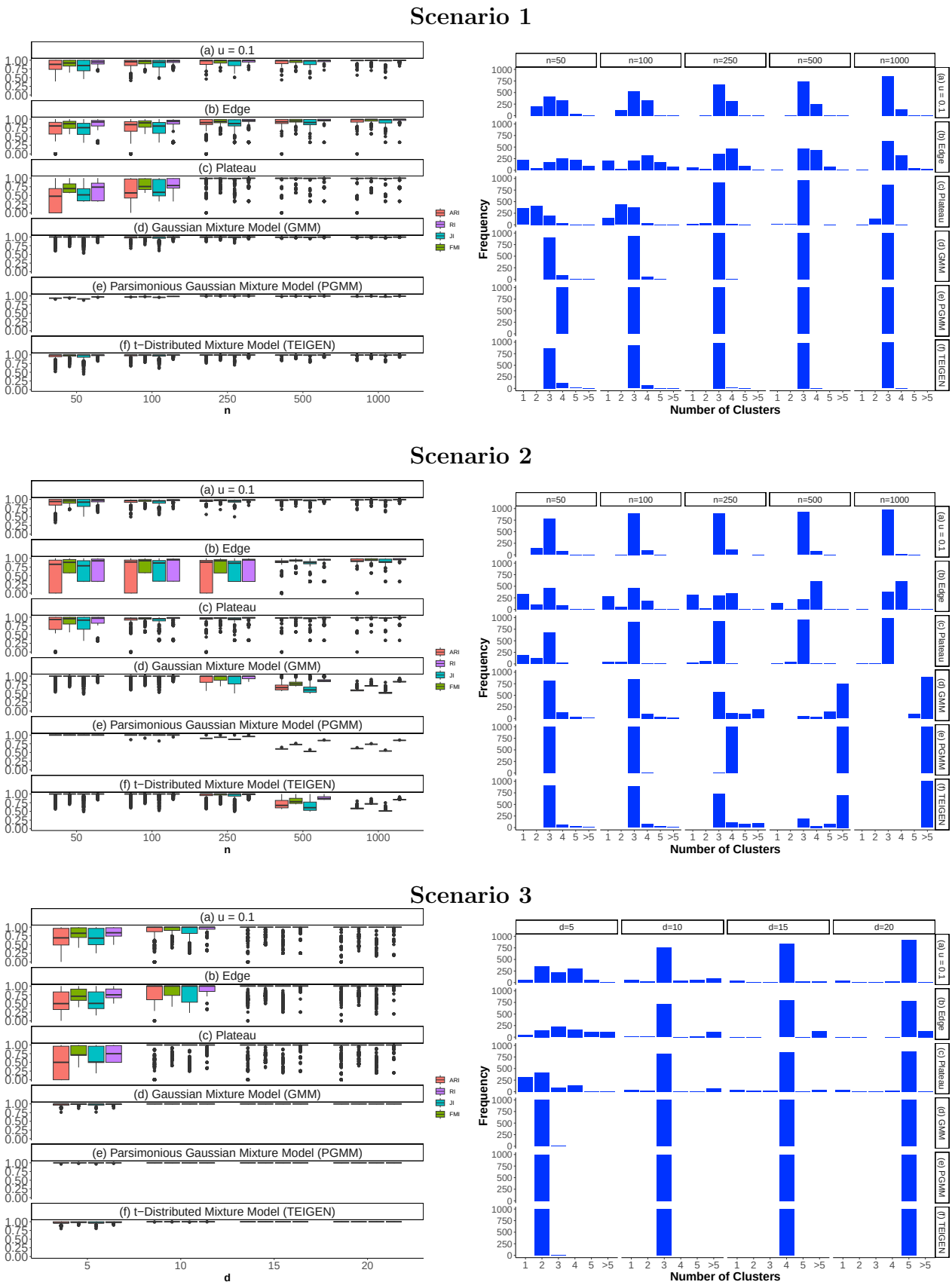


Figure 5: Monte Carlo simulation study: (Left) Performance metrics (ARI, RI, JI, FMI) (Right) Empirical distribution on the number of detected clusters.

We also report the empirical distribution of the number of clusters detected by the proposed

method, over different conquering strategies (i.e., fixed u, plateau, and edge), along with the same outputs for the GMM (Gaussian Mixture Model) in (1), PGMM (Parsimonious Gaussian Mixture Model) [14] and TEIGEN (t–Distributed Mixture Model) [36, 37]. For these approaches, model selection was conducted using the BIC. Model selection clearly involves more than just choosing the value of $\mathcal{K}$, which in this case ranges from 1 to 6 across all scenarios. In particular, for GMM and TEIGEN, BIC is used to define a suitable covariance decomposition, while for PGMM, BIC is used to determine the number of latent components (in the simulations we allow for the maximum number of latent components by setting the option `relax = TRUE` in the corresponding package). In Figure 5 (top and middle), it can be seen that for Scenarios 1–2, as the sample size increases the performance metrics increase on average for all conquering strategies. Interestingly, for Scenario 1, the edge conquering strategy works better than the plateau for relatively small sample sizes. Conversely, in Scenario 2, the plateau yields on average better results than the edge for small sample sizes. In Scenarios 1–2, fixing a sieve size of $u = 0.1$, produces accurate clustering results on average even for small sample sizes. In addition, as the sample sizes increases the proposed method identifies most frequently the correct number of clusters $\mathcal{K} = 3$ for Scenarios 1–2—both when $u = 0.1$ as well as when u is set using the plateau.

In Figure 5 (bottom) we present the performance of the proposed method for Scenario 3. As can be seen in Figure 5 (bottom–left), with an increasing number of dimensions, on average, the proposed method presents better agreement metrics (recall that in Scenario 3 cluster separation grows linearly with data dimension). In Figure 5 (bottom–right), it can be seen that as the sample size increases, the proposed method most frequently captures the true number of clusters $\mathcal{K}_d \in \{2, 3, 4, 5\}$ for dimensions $d \in \{5, 10, 15, 20\}$ respectively.

Some final comments on the comparison of the Reign-and-Conquer clustering against GMM, PGMM, and TEIGEN are in order. In Scenario 1 the data are simulated from a low-dimensional Gaussian mixture model, and hence perhaps not surprisingly GMM outperforms the proposed approach. Still, the performance of the proposed approach is remarkable especially as we make no assumption on the joint. In addition, Reign-and-Conquer outperforms GMM, PGMM, and TEIGEN in Scenario 2. Reign-and-Conquer's superior performance in Scenario 2 stems from its flexibility in adjusting to a different number of cluster per margin. Finally, the RC algorithm has a comparable performance to the alternative methods over Scenario 3.

6 Real data illustrations

Banknotes

The first dataset to be analyzed with the proposed methods contains $p = 6$ measurements (in millimetres) made on 100 genuine and 100 counterfeit old-Swiss 1000-franc bank notes [38, pp. 5–8]. The data include bill length, left and right edge widths, bottom and top margin widths, and diagonal width, and are available from the `mclust` `R` package [39].

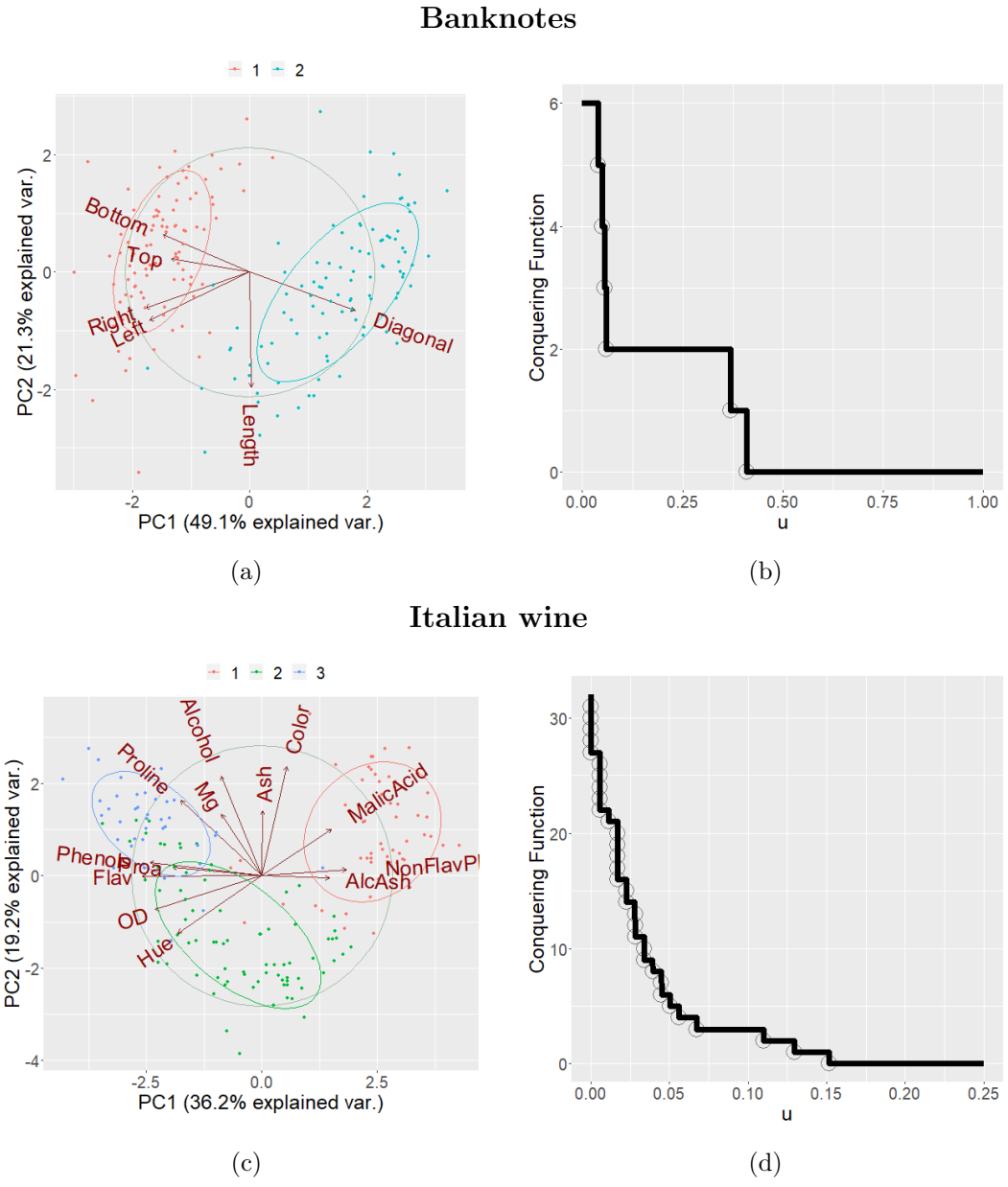


Figure 6: (a) Biplot banknote data set, the color key corresponds to the cluster labels obtained for $u = 0.36$ (Plateau). (b) Conquering function estimate. (c) Biplot of wine data set, the color key corresponds to the cluster labels obtained for $u = 0.105$ (Plateau). (d) Conquering function estimate (restricted to $[0, 0.25]$ for visualization purposes).

Banknotes

Method	Cluster	Counterfeit	Genuine
RC	1	99	0
	2	1	100
GMM	1	16	2
	2	0	98
	3	84	0
PGMM	1	0	74
	2	0	25
	3	74	0
	4	11	0
	5	15	1
TEIGEN	1	0	75
	2	0	24
	3	85	0
	4	15	1

Italian wine

Method	Cluster	Barbera	Grignolino	Barolo
RC	1	42	9	5
	2	1	56	16
	3	5	7	37
GMM	1	56	2	0
	2	3	65	0
	3	0	4	48
PGMM	1	0	40	0
	2	0	0	59
	3	0	28	0
	4	48	3	0
TEIGEN	1	0	26	0
	2	0	33	2
	3	27	0	0
	4	21	3	0
	5	0	4	31
	6	0	2	26
	7	0	3	0

Rice

Method	Cluster	Cammeo	Osmancik
RC	1	184	2062
	2	1446	118
GMM	1	4	612
	2	2	465
	3	370	305
	4	413	7
	5	28	151
	6	644	57
	7	101	43
	8	68	540
PGMM	1	4	622
	2	492	39
	3	1013	222
	4	121	1297
TEIGEN	1	726	113
	2	34	892
	3	533	32
	4	11	645
	5	326	498

Iris

Method	Cluster	Setosa	Versicolor	Virginica
RC	1	0	0	50
	2	39	8	3
	3	33	17	0
GMM	1	50	0	0
	2	0	50	50
PGMM	1	0	50	0
	2	47	0	3
	3	0	0	50
TEIGEN	1	50	0	0
	2	0	50	50

Table 1: Confusion matrices. NOTE: RC = Reign-and-Conquer, GMM = Gaussian Mixture Model, PGMM = Parsimonious Gaussian Mixture Model, and TEIGEN = t-Distributed Mixture Models.

In Figure 6 (a) we depict a biplot of the first two principal components of the data, along with the corresponding clustering yield by the proposed Reign-and-Conquer clustering. The sieve size was set using the plateau conquering strategy that corresponds to the best average results on the simulations setting in Scenario 3. In Figure 6 (b) we depict the fitted conquering function, from where it can be seen that the plateau consists of $u = 0.36$. As can be noticed from the confusion matrix in Table 1, Reign-and-Conquer does an excellent job classifying counterfeit as well as genuine data. The GMM analysis combined with BIC suggests that there could be three clusters. The latter analysis is not least interesting from a forensic viewpoint, as it suggests that there might be two clusters of counterfeit banknotes. Finally, we note that the number of marginal clusters obtained using Reign-and-Conquer (i.e., $K_1, \ldots, K_6$) ranges from 1–3 clusters.

Italian wine

The second dataset on which the proposed approach is illustrated contains the results of a chemical analysis of wines grown in Italy, derived from three cultivars (Barbera, Grignolino, and Barolo); the data are available from [40]. The chemical analysis includes the measurement of $d = 13$ continuous variables (such as alcohol, malic acid, ash, etc) on $n = 178$ instances. Similarly to the banknote data illustration, in Figure 6 (c) we depict a biplot to represent the first two principal components of the data, along with the corresponding clustering yield by the proposed method. The sieve size was set using the plateau conquering strategy that corresponds to the best average results on the simulations setting in Scenario 3. In Figure 6 (d) we depict the fitted conquering function, the plateau consists of $u = 0.105$. As can be seen from the confusion matrix in Table 1, Reign-and-Conquer learns about the 'right' number of cultivars and the obtained clusters have a resemblance with the cultivars. The GMM analysis combined with BIC would offer another interesting outlook, suggesting that two of these clusters are so similar that they should perhaps be merged. Finally, we note that the number of marginal clusters (i.e., $K_1, \ldots, K_{13}$) is 1 for 8 of the dimensions, and 2 for the remaining dimensions.

Wholesale customers

The Wholesale database is available from UCI Machine Learning repository (https://archive.ics.uci.edu/dataset/292/wholesale+customers) and includes the annual spending corresponding to 6 categories (fresh, milk, grocery, frozen, detergents paper, and delicatessen) of $n = 440$ clients of a wholesale distributor in Portugal. These data lacks labels to identify which group each client belongs to. Thus, in Table 2 we report for each clustering method the number of groups iden-

tified in the data, the ratio between the size of the biggest and the smallest cluster and 4 internal evaluation metrics [41], namely: Davies–Bouldin (DB), Hubert–Levine (HL), Silhouette (S), and Calinski–Harabasz (CH) indexes. Note that smaller values of DB and HL, along with larger values of the S and CH, indicate better internal evaluations.

Wholesale

Method	Clusters	Ratio	Davies–Bouldin	Hubert–Levine	Silhouette	Calinski–Harabasz
RC (Plateau)	2	1.168	3.196	0.114	0.127	59.552
GMM	7	4.889	2.099	0.068	0.032	50.897
PGMM	2	4.641	2.209	0.084	0.437	88.490
TEIGEN	3	1.750	3.336	0.105	0.130	51.608

Rice

Method	Clusters	Ratio	Davies–Bouldin	Hubert–Levine	Silhouette	Calinski–Harabasz
RC (Plateau)	2	1.401	1.034	0.224	0.410	3456.480
GMM	8	4.831	4.126	0.198	0.018	609.082
PGMM	4	2.670	2.740	0.195	0.112	1156.104
TEIGEN	5	1.639	3.305	0.176	0.046	781.018

Iris

Method	Clusters	Ratio	Davies–Bouldin	Hubert–Levine	Silhouette	Calinski–Harabasz
RC (Plateau)	3	2.8	1.145	0.277	0.367	164.261
GMM	2	2.8	0.682	0.270	0.581	251.352
PGMM	3	1.1	1.231	0.263	0.327	187.448
TEIGEN	2	2.8	0.682	0.270	0.581	251.352

Table 2: Internal agreement metrics for different clustering methods.

As can be seen in Table 2, the metrics of TEIGEN and the Reign-and-Conquer algorithm are fairly in line; in particular, TEIGEN and Reign-and-Conquer produce balanced cluster solutions (relatively small ratios) compared to PGMM and GMM. Since smaller values of DB and HL indicate better internal evaluation, from the point of view of these indexes GMM outperforms the remainder approaches. Yet, PGMM accounts for the largest S–index, followed by TEIGEN and Reign-and-Conquer. PGMM also maximizes the CH–index.

Finally, the RC algorithm not only identifies two global clusters but it also suggests varying subgroups within each margin: fresh ($K_1 = 2$), milk ($K_2 = 2$), grocery ($K_3 = 2$), frozen ($K_4 = 3$), detergents paper ($K_5 = 2$), and delicatessen ($K_6 = 3$).

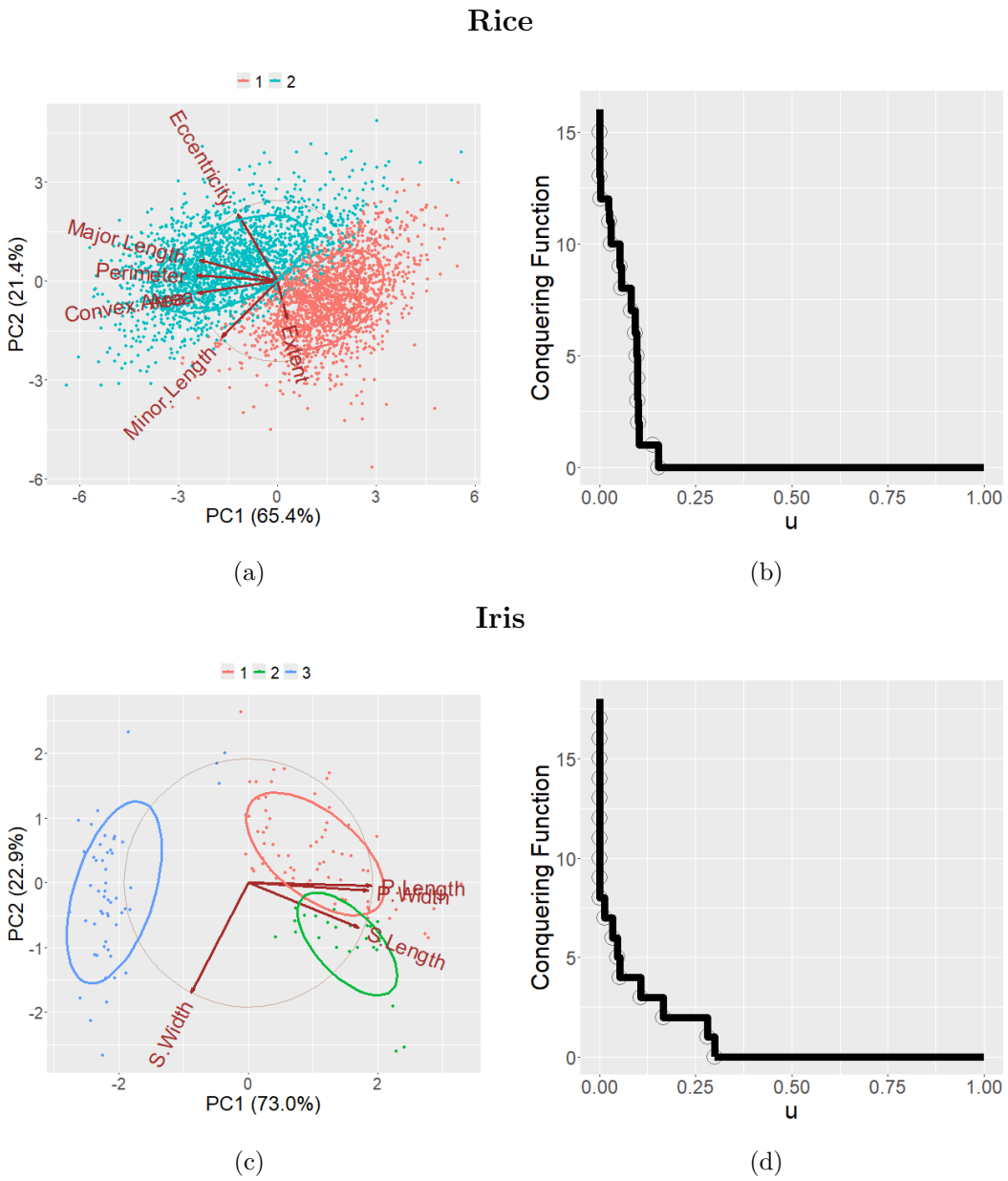


Figure 7: (a) Biplot Rice data set, the color key corresponds to the cluster labels obtained for $u = 0.10$ (Plateau). (b) Conquering function estimate. (c) Biplot of Iris data set, the color key corresponds to the cluster labels obtained for $u = 0.16$ (Plateau). (d) Conquering function estimate.

Rice image data

The Rice data set [42] is available from the UCI Machine Learning repository (https://archive.ics.uci.edu/dataset/545/rice+cammeo+and+osmancik). It contains information on $n = 3810$ images of rice grains corresponding to two species (Cammeo and Osmancik). The data are already preprocessed and 7 morphological features were obtained for each grain image. In Table 1, Reign-and-Conquer, along with Plateau conquering, is used to learn about the 'right' number of species, and the obtained clusters have a resemblance with the true groups in the data. The GMM, PGMM,

and TEIGEN analysis combined with BIC offer another interesting outlook, suggesting that $\mathcal{K} > 2$. Interestingly, we note that the number of marginal clusters (i.e., $K_1, \ldots, K_7$) is 2 for 4 of the dimensions, and 3 for the remaining 3 dimensions. Finally, Table 2 shows that Reign-and-Conquer accounts for the largest S and CH indexes and the smallest DB index, while TEIGEN achieves the smallest HL metric (followed by PGMM and GMM).

Iris data

The Iris data set consists of $n = 50$ samples from each of three species of Iris (setosa, versicolor, and virginica) and $p = 4$ measurements (in centimeters): length and width of the sepals and petals. The data are available from the UCI Machine Learning repository (https://archive.ics.uci.edu/dataset/53/iris).

In Figure 7 panel (a) and (c) we depict a biplot of the first two principal components of Rice and Iris data respectively, along with the corresponding clustering yield by the proposed Reign-and-Conquer clustering, while in panel (b) and (d) illustrates the fitted conquering function, from where it can be seen that the plateau consists of $u = 0.10$ for Rice and $u = 0.16$ for Iris.

As can be seen in the confusion matrices in Table 1, Reign-and-Conquer does a similar job to PGMM classifying Iris data, while GMM and TEIGEN suggest that there could be two clusters in data. Finally, Table 2 shows that only Reign-and-Conquer and PGMM accounts for the correct number of clusters in data, nevertheless PGMM accounts for the best indexes among all methods (followed by Reign-and-Conquer).

7 Discussion

This paper devises an strategy game-inspired algorithm for unsupervised learning that be used for clustering data, both jointly as well as marginally. A clear strength of the proposed approach is its resilience to the issues outlined as Pitfalls 1 and 2 in Section 1. Unlike traditional model-based clustering, it allows for varying numbers of clusters across margins and is better suited for moderate- to high-dimensional data analysis. Another advantage is the fact that the herein proposed clustering approach only specifies a model for the margins but leaves the joint unspecified, and hence it is partially parallelizable. To our knowledge, our paper takes the lead in identifying Pitfall 1 and is the first to propose a solution to this overlooked issue.

While the obtained numerical evidence indicates a satisfactory performance of the proposed

method under a variety of situations, there is still room for improvement, open problems to be addressed as well as opportunities for future research. First, the geometry of the boundaries of the final kingdoms (i.e., the Voronoi cells of the conquerors) could perhaps be bended so to better adapt to the structure of the data, to offer more flexibility to the partitioning of the sample space, and ultimately to improve clustering results. Second, the game-theoretical variant from Section 4 opens a world of opportunities on ways to set the 'incentives' to conquer, via an utility function, to explored in a follow-up paper. Thirdly, as noted by a referee, Step 3 leads to a 'hard' partition. Propagating uncertainty in a Bayesian fashion, starting from Step 1, appears to be a natural approach to mitigate this. Yet, several challenges would need to be addressed to facilitate this approach; perhaps the most significant challenge involves the need for averaging posterior Voronoi tessellations in the final analysis, which poses some interesting conceptual and numerical challenges. Finally, while here the focus has been on unsupervised learning, the potential of related strategy-game inspired approaches for supervised learning would seem natural.

Appendix

Appendix A: Proof of Theorem 1

Before getting started with the proofs we lay the groundwork. The proof of Theorem 1 uses the following representation of the conquering function

$$C(u) = \prod_{j=1}^{d} K_j - \sum_{\mathbf{i} \in I} 1_{D_u}(\mathbf{i}), \tag{16}$$

which follows directly from (10). Here, 1_A is the indicator of set A and in the proof we will make use of some of its well-known properties [43], such as

$$\limsup_{n\to\infty} 1_{A_n} = 1_{\limsup_{n\to\infty} A_n}, \qquad \liminf_{n\to\infty} 1_{A_n} = 1_{\liminf_{n\to\infty} A_n}. \tag{17}$$

Since (17) holds for both lim sup and lim inf it follows that $\lim_{n\to\infty} 1_{A_n} = 1_{\lim_{n\to\infty} A_n}$. Recall in addition that if $\{A_n\}$ is an nondecreasing sequence of sets, then its limit is the infinite union, that is

$$A_n \subseteq A_{n+1} \Longrightarrow \lim_{n\to\infty} A_n = \bigcup_{n=1}^{\infty} A_n. \tag{18}$$

See, for instance, Resnick [43, Proposition 1.4.1]. Finally, the proof of Claim d) in Theorem 1 will make use of the Lebesgue measure over the unit interval, $\lambda([a,b]) = b - a$, for $[a,b] \subseteq [0,1]$.

Proof of Theorem 1.

a) Consider $(u,v) \in [0,1]^2$ such that $u \leqslant v$. Then, whenever $P(A_{\mathbf{i}}) \leqslant u$ it follows that $P(A_{\mathbf{i}}) \leqslant v$; or in other words $D_u \subseteq D_v$, which in turn implies that $1_{D_u}(\mathbf{i}) \leqslant 1_{D_v}(\mathbf{i})$. Hence,

$$-\sum_{\mathbf{i}\in I} 1_{D_u}(\mathbf{i}) \geqslant -\sum_{\mathbf{i}\in I} 1_{D_v}(\mathbf{i}) \Longrightarrow \prod_{j=1}^{d} K_j - \sum_{\mathbf{i}\in I} 1_{D_u}(\mathbf{i}) \geqslant \prod_{j=1}^{d} K_j - \sum_{\mathbf{i}\in I} 1_{D_v}(\mathbf{i})$$
$$\Longrightarrow C(u) \geqslant C(v),$$

from where the final result follows. □

b) First note that the proof of Claim a) implies that D_u is a nondecreasing, in the sense $D_u \subseteq D_v$, for any $u \leqslant v$ with $(u, v) \in [0, 1]^2$. Next, consider an arbitrary $u \in [0, 1]$ and a sequence u_n such that $u_n \to u$, with $u_n \leqslant u$ for every $n \in \mathbb{N}$. Then, for a sufficiently large n it holds that $u_n \leqslant u_{n+1}$ which in turn implies that $D_{u_n} \subseteq D_{u_{n+1}}$. This, along with (18) and the fact that D_u is nondecreasing, implies that $\lim_{n\to\infty} D_{u_n} = \bigcup_{n=1}^{\infty} D_{u_n} = D_u$. Finally, (16) and (17) then yield that

$$\lim_{n\to\infty} C(u_n) = \prod_{j=1}^{d} K_j - \sum_{\mathbf{i}\in I} \lim_{n\to\infty} 1_{D_n}(\mathbf{i}) = \prod_{j=1}^{d} K_j - \sum_{\mathbf{i}\in I} 1_{\lim_{n\to\infty} D_n(\mathbf{i})} = \prod_{j=1}^{d} K_j - \sum_{\mathbf{i}\in I} 1_{D_u(\mathbf{i})} = C(u),$$

which concludes the proof. □

c) Trivially, since Claim a) shows that $C(u)$ is nonincreasing it follows that for every $u \in [0, 1]$,

$$C(u) \geqslant C(1) = \prod_{j=1}^{d} K_j - |D_1| = 0,$$

where the final equality is a consequence of the fact that $|D_1| = |\{\mathbf{i} : P(A_\mathbf{i}) \leqslant 1\}| = |\{A_\mathbf{i} : \mathbf{i} \in I\}| = \prod_{j=1}^{d} K_j$. The final result then follows from (16) and (17) by noting that for every $u \in [0, 1]$,

$$C(u) \leqslant C(0) = \prod_{j=1}^{d} K_j - |D_0| = \prod_{j=1}^{d} K_j,$$

since $|D_0| = |\{\mathbf{i} : P(A_{i,j}) \leqslant 0\}| = |\varnothing| = 0$. □

d) First note that,

$$1_{D_u^c}(\mathbf{i}) = \begin{cases} 1, & \mathbf{i} \in D_u^c, \\ 0, & \mathbf{i} \in D_u, \end{cases} = \begin{cases} 1, & 0 \leqslant u < P(A_\mathbf{i}), \\ 0, & \text{otherwise.} \end{cases} \tag{19}$$

Next, observe that $|D_u^c| = \sum_{\mathbf{i}\in I} 1_{D_u^c}(\mathbf{i})$ which along with (19) yields

$$\int_0^1 C(u)\,\mathrm{d}u = \int_0^1 |D_u^c|\,\mathrm{d}u = \sum_{\mathbf{i}\in I} \int_0^1 1_{D_u^c}(\mathbf{i})\,\mathrm{d}u = \sum_{\mathbf{i}\in I} \int_{[0,P(A_\mathbf{i})]} \mathrm{d}u = \sum_{\mathbf{i}\in I} \lambda([0, P(A_\mathbf{i})]) = \sum_{\mathbf{i}\in I} P(A_\mathbf{i}) = 1,$$

which concludes the proof. □

Appendix B: Step function representation, plateau, and edge

This appendix shows formally that the conquering function is a step function with a finite number of steps (provided that $K_1, \ldots, K_d$ are finite), and it uses that representation so to formally define the plateau and the edge. As a consequence of (16) and of (19) in Appendix A it holds that

$$C(u) = \prod_{j=1}^{d} K_j - \sum_{\mathbf{i}\in I} 1_{D_u}(\mathbf{i}) = \prod_{j=1}^{d} K_j - \sum_{\mathbf{i}\in I} 1_{[0,P(A_\mathbf{i}))}(u) = \sum_{\mathbf{i}\in I} \{1 - 1_{[0,P(A_\mathbf{i}))}(u)\} = \sum_{\mathbf{i}\in I} 1_{[P(A_\mathbf{i}),1]}(u), \tag{20}$$

where the final equality follows from the well-known property of the indicator, $1 - 1_B = 1_{B^c}$. Hence, Equation (20) shows that $C(u)$ is a step function with a maximum of $|I| = \prod_{j=1}^{d} K_j$ steps. Given this representation, it follows that

$$\text{plateau} = \sup\left\{u : C(u) = \max_{\mathbf{i}\in I}[P(A_\mathbf{i}^c)]\right\}, \qquad \text{edge} = \arg\max_u \{C(u) - C(u^+)\}.$$

In words, the plateau is the value at which the longest plateau of $C(u)$ ends, and the edge is the value at which the largest jump on $C(u)$ occurs.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

We thank the Editor, the Associate Editor, and three reviewers for insightful comments and constructive feedback. The research was partially funded by the Royal Society of Edinburgh, Abrdn, and CIDMA (under the FCT grants https://doi.org/10.54499/UIDB/04106/2020 and https://doi.org/10.54499/UIDP/04106/2020).

References

[1] T. Hastie, R. Tibshirani, J. Friedman. The Elements of Statistical Learning. Springer, New York, 2nd edition, 2009.

[2] B. Everitt, S. Landau, M. Leese, D. Stahl. Cluster Analysis. Chichester, UK: Wiley, 2011.

[3] R. S. King. Cluster Analysis and Data Mining: An Introduction. Mercury Learning & Information, Dulles, VA, 2014.

[4] C. Hennig. What are the true clusters? Pattern Recognit. Lett. 64 (2015) 53–62.

[5] V. Melnykov, S. Sarkar, Y. Melnykov. On finite mixture modeling and model-based clustering of directed weighted multilayer networks. Pattern Recognit. 112 (2021) 107641.

[6] V. Melnykov, Y. Wang. Conditional mixture modeling and model-based clustering. Pattern Recognit. 133 (2023) 108994.

[7] V. Melnykov, R. Maitra. Finite mixture models and model-based clustering. Statist. Surveys 4 (2010) 80–116.

[8] P. D. McNicholas. Model-based clustering. J. Classification 33 (3) (2016) 331–373.

[9] I. Claire Gormley, T. Brendan Murphy, A.E. Raftery. Model-based clustering. Annual Review Statist. Appl. (10) (2023).

[10] S. Fruhwirth-Schnatter, G. Celeux, C.P. Robert. Handbook of Mixture Analysis. Chapman & Hall/CRC, Boca Raton, 2019.

[11] S. Huang, Z. Kang, Z. Xu, Q. Liu. Robust deep k-means: An effective and simple method for data clustering. Pattern Recognit. 117 (2021) 107996.

[12] H. Hu, J. Liu, X. Zhang, M. Fang. An effective and adaptable k-means algorithm for big data cluster analysis. Pattern Recognit. 139 (2023) 109404.

[13] C. Bouveyron, C. Brunet-Saumard. Model-based clustering of high-dimensional data: A review. Comput. Statist. & Data Analysis 71 (2014) 52–78.

[14] D.P. McNicholas, T.B. Murphy, Parsimonious Gaussian mixture models. Statist. Comput. 18 (2008) 285–296.

[15] S. Bulò, M. Pelillo. A game-theoretic approach to hypergraph clustering. Advances Neural Inf. Process. Systems 22 (2009).

[16] J. Hou, M. Pelillo, H. Yuan. Hypergraph matching via game-theoretic hypergraph clustering. Pattern Recognit. 125 (2022) 108526.

[17] J. Fúquene, M. Steel, D. Rossell. On choosing mixture components via non-local priors. J. Royal Statist. Soc., Ser. B 81 (5) (2019) 809–837.

[18] P. J. Green. Reversible jump Markov chain Monte Carlo computation and Bayesian model determination. Biometrika 82 (4) (1995) 711–732.

[19] S. Richardson, P. J. Green. On Bayesian analysis of mixtures with an unknown number of components (with discussion). J. Royal Statist. Soc., Ser. B 59 (4) (1997) 731–792.

[20] C. Fraley, A. E. Raftery. Model-based clustering, discriminant analysis, density estimation. J. Amer. Statist. Assoc. 97 (458) (2002) 611–631.

[21] J.-P. Baudry, A. E. Raftery, G. Celeux, K. Lo, R. Gottardo. Combining mixture components for clustering. J. Comput. Graph. Statist. 19 (2) (2010) 332–353.

[22] R Development Core Team. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, 2022.

[23] D. Rossell. Bayesian model selection and averaging with `mombf`. Unpublished manuscript (2018).

[24] F. Xie, Y. Xu. Bayesian repulsive Gaussian mixture model. Journal of the American Statistical Association, 529 (2019) 187–203.

[25] M. Beraha, R. Argiento, J. Møller, G. Alessandra. MCMC computations for Bayesian mixture models using repulsive point processes. Journal of Computational and Graphical Statistics, 31(2) (2022) 422–435.

[26] T. Manole, A. Khalili. Estimating the number of components in finite mixture models via the group-sort-fuse procedure. The Annals of Statistics, 49(6) (2021) 3043–3069.

[27] T. Peterka, D. Morozov, C. Phillips. High-performance computation of distributed-memory parallel 3d Voronoi and Delaunay tessellation, in SC'14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 997–1007. IEEE, 2014.

[28] M. Maschler, S. Zamir, E. Solan. Game Theory. Cambridge University Press, Cambridge, MA, 2020.

[29] G. van der Laan, A.J. Talman, L. Van der Heyden. Simplicial variable dimension algorithms for solving the nonlinear complementarity problem on a product of unit simplices using a general labelling. Math. Operations Res. 12 (3) (1987) 377–397.

[30] S. Govindan and R. Wilson. A global Newton method to compute Nash equilibria. J. Econ. Theo. 110 (1) (2003) 65–86.

[31] R. Porter, E. Nudelman, Y. Shoham. Simple search methods for finding a Nash equilibrium. G. Econ. Behav. 63 (2) (2008) 642–662.

[32] R.B. Nelsen. An Introduction to Copulas. Springer, New York, 2007.

[33] A. Punzo, A. Mazza, P. D. McNicholas, ContaminatedMixt: An R package for fitting parsimonious mixtures of multivariate contaminated normal distributions. J. Statist. Soft. 85 (2018) 1–25.

[34] V. Melnykov, Y. Wang, Y. Melnykov, F. Torti, D. Perrotta, M. Riani, On simulating skewed and cluster-weighted data for studying performance of clustering algorithms. J. Comput. Graph. Statist. (2023) 1–7.

[35] D. Pfitzner, R. Leibbrandt, D. Powers. Characterization and evaluation of similarity measures for pairs of

clusterings. Knowledge Inf. Sys. 19 (3) (2009) 361–394.

[36] J.L. Andrews, P.D. McNicholas. Model-based clustering, classification, and discriminant analysis via mixtures of multivariate t-distributions: the t EIGEN family. Statist. Comput. 22 (2012) 1021–1029.

[37] J.L. Andrews, J.R. Wickins, N.M. Boers, P.D. McNicholas. teigen: An R package for model-based clustering and classification via the multivariate t distribution. J. Statist. Soft. 83 (2018) 1–32.

[38] B. Flury. Multivariate Statistics: A Practical Approach. Chapman & Hall, Boca Raton FL, 1988.

[39] C. Fraley, A. E. Raftery, L. Scrucca, T. Brendan Murphy, M. Fop, Package 'mclust', 2012.

[40] B. Vandeginste. Parvus: An extendable package of programs for data exploration, classification and correlation. J. Chemometrics 4 (2) (1990) 191–193.

[41] M. Charrad, N. Ghazzali, V. Boiteau, A. Niknafs. `NbClust`: an R package for determining the relevant number of clusters in a data set. J. Statist. Soft. 61 (2014) 1–36.

[42] I. Cinar, M. Koklu. Classification of rice varieties using Artificial Intelligence methods. Int. J. Int. Sys. Appl. Eng. 7 (2019) 188–194.

[43] S. Resnick. A Probability Path. Springer, New York, 2019.

Miguel de Carvalho is the Chair of Statistical Data Science at the University of Edinburgh. Miguel's research and trajectory have been recognized with a variety of awards, including the Lindley Prize from the International Society of Bayesian Analysis (ISBA). His research interests are diverse, spanning applied statistics, biostatistics, econometrics, risk analysis, and statistics of extremes. He is an Elected Fellow of the International Statistical Institute and has served as an Associate Editor for several leading statistical journals, including *Journal of the American Statistical Association*, *The Annals of Applied Statistics*, and *The American Statistician*.

Gabriel Martos Venturini is Associate Professor of Statistics at Universidad Torcuato Di Tella in Argentina. His research lies broadly in statistics and machine learning, especially in the context of high-dimensional data.

Andrej Svetlošák is post-doctoral researcher at the School of Mathematics of the University of Edinburgh. His research primarily focuses on nonparametric and unsupervised machine learning methods. His work primarily focuses on Financial Innovation and Open Banking-driven technologies, but extends beyond these areas, including applications in AI and large language models.