Pattern Recognition 173 (2026) 112909

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

CRB-NCE: An adaptable cohesion rule-based approach to number of clusters estimation

J. Tinguaro Rodríguez [a,b,*], Xabier Gonzalez-Garcia [c], Daniel Gómez [d,e], Humberto Bustince [c]

[a] Department of Statistics and Operations Research, Complutense University of Madrid, Spain
[b] Institute of Interdisciplinary Mathematics, Complutense University of Madrid, Spain
[c] Department of Statistics, Computer Science and Mathematics, Public University of Navarra, Spain
[d] Department of Statistics and Data Science, Complutense University of Madrid, Spain
[e] Institute of Statistics and Data Science, Complutense University of Madrid, Spain

ARTICLE INFO

Keywords:
Cluster analysis
Number of clusters estimation
Cluster cohesion measures

ABSTRACT

Accurate number-of-clusters estimation (NCE) is a central task in many clustering applications, particularly for prototype-based k-centers methods like k-Means, which require the number of clusters k to be specified in advance. This paper presents CRB-NCE, a general cluster cohesion rule-based framework for NCE integrating three main innovations: (i) the introduction of tail ratios to reliably identify decelerations in sequences of cohesion measures, (ii) a threshold-based rule system supporting accurate NCE, and (iii) an optimization-driven approach to learn these thresholds from synthetic datasets with controlled clustering complexity. Two cohesion measures are considered: inertia (SSE) and a new, scale-invariant metric called the mean coverage index. CRB-NCE is mainly applied to derive general-purpose NCE methods, but, most importantly, it also provides an adaptable framework that enables producing specialized procedures with enhanced performance under specific conditions, such as particular clustering algorithms or overlapping cluster structures. Extensive evaluations on synthetic Gaussian datasets (both standard and high-dimensional), clustering benchmarks, and real-world datasets show that CRB-NCE methods consistently achieve robust and competitive NCE performance with efficient runtimes compared to a broad baseline of internal clustering validity indices and other NCE methods.

1. Introduction

Clustering is a central tool in many exploratory data analysis and machine learning applications, as the ability of clustering algorithms to divide data into homogeneous groups can be extremely useful in preprocessing, dimensionality reduction, or establishing target categories for applied analyses. In this sense, a key distinction exists in cluster analysis between *realistic* clustering, which seeks to identify the actual groups structure inherent in the data (e.g. biomolecular categories [1], phylogenetic ensembles [2], mental disease subtypes [3], brain tissues [4], etc.), and *constructive* clustering, which creates clusters for interpretive or computational convenience, regardless of whether they reflect inherent divisions [5].

Thus, especially in a realistic context and once the data have been found to be clusterable [6], the goal of cluster analysis is to produce a coherent partition with as many clusters as there are inherent groups in the data, such that the clusters effectively capture those groups [7]. Achieving this relies heavily on accurate number of clusters estimation (NCE), as most clustering algorithms yield varying numbers of clusters k depending on hyperparameter choices. This is particularly true for prototype-based k-centers algorithms, such as k-Means and its variants, which require k to be prespecified. As a result, NCE often becomes a core part of the clustering process itself and remains a challenging, active research area (see e.g. [8–10]). This continued interest is largely driven by the widespread use of k-centers algorithms across diverse domains (see e.g. [11–13]), a trend that may be reinforcing due to their scalability [14,15].

1.1. Related work

The most widely adopted approach to the NCE task proceeds by producing a sequence of candidate partitions through a preselected clustering algorithm to explore the search space for the k parameter. The value of k that optimizes a reference internal cluster validity index (ICVI [16]) computed on this sequence is then taken as the estimated number of clusters. Most ICVIs combine diverse models or measures of cluster

* Corresponding author.
E-mail address: jtrodrig@ucm.es (J.T. Rodríguez).

https://doi.org/10.1016/j.patcog.2025.112909
Received 29 April 2025; Received in revised form 27 October 2025; Accepted 8 December 2025
Available online 13 December 2025

cohesion (also called compacity) and separation, typically associating clustering quality with tightly grouped, spherical, and isolated clusters. This association has led some authors to note that using ICVIs for NCE implicitly assumes that the best-quality partition must simultaneously exhibit the correct number of clusters [17]. However, several studies suggest that such coincidence occurs only exceptionally [18,19]. Moreover, some modern ICVIs considering only cohesion information seem to achieve better NCE performance than those also using separation information (see e.g. [8]), suggesting that cohesion may be more relevant for NCE than separation.

There are other approaches to the NCE task [20]. One is based on clustering stability [21], where an index is computed for each k in a given search space, assessing the similarity between partitions obtained from different non-disjoint subsamples of the data. The compared partitions should be most similar for the correct k, since for other values different random substructures would be merged or split across partitions. Although this approach has the virtue of not assuming any notion of what a quality clustering should be-as ICVIs do-it is far more computationally demanding. Another approach to NCE uses correlation-based indices, under the idea that distances or dissimilarities between objects should correlate with those between clusters for the correct number of clusters [22]. Other approaches, such as the elbow method [8] or VAT [23], tend to be subjective or rely on visual inspection and are not considered here.

1.2. Contributions

This paper presents a general cohesion rule-based procedure for number of clusters estimation (referred to as CRB-NCE, or CRB) that integrates several novel components: First, it introduces tail ratios to weigh differences in a cluster cohesion measure sequence, improving the robustness of related ratios such as that in [24]. The cohesion measures considered are inertia (SSE) and the mean coverage index, the latter also proposed here as a scale-invariant alternative for modeling cohesion, following ideas from [25–27]. Second, the CRB procedure itself, which relies on a small set of threshold-triggered rules to exploit tail ratios for NCE purposes, building to some extent on the decision criterion in [8] (see also Eq. 9). Particular NCE procedures arise from any choice of cohesion measure and threshold values. Third, an optimization-based approach is proposed to learn these thresholds from synthetic datasets. Gaussian data spanning a wide range of clustering complexity conditions are used to obtain general-purpose CRB-NCE procedures. However, specialized CRB-NCE procedures with enhanced performance under particular conditions or algorithms can be derived by tailoring the learning data and obtaining adapted thresholds.

An extensive computational study was conducted to compare CRB procedures with a baseline of NCE methods in the context of prototype-based clustering, first on synthetic Gaussian data similar to those used to fit CRB's thresholds, and then on benchmark, real-world, and high-dimensional synthetic Gaussian datasets. An additional experiment illustrates the performance gains enabled by specialized CRB procedures. The results show that CRB methods achieve competitive and robust performance across all scenarios[1].

Alongside its accuracy and robustness, the above mentioned adaptability is arguably CRB's most significant feature. For example, in datasets with partially overlapping clusters, ICVIs often lose accuracy as cohesion and separation trends are distorted by ambiguous boundaries. In such cases, CRB's ability to be tailored to specific conditions can enable improved NCE performance, as illustrated in Section 5.4. Thus, CRB-NCE differs in nature from ICVIs: its rule-based structure is not designed to measure clustering quality, but rather to accurately estimate the number of clusters (the task for which it is explicitly trained), potentially under highly specific conditions.

This paper is structured as follows: Notation and the baseline ICVIs are introduced in Section 2. Coverage indices are formulated in Section 3. The CRB method is exposed in Section 4. Section 5 describes the computational study and its results. Conclusions are shed in Section 6.

2. Cluster cohesion and separation in ICVIs

Along this work, objects $X = \{x_1, \dots, x_N\}$ to be clustered are assumed to lie in a D-dimensional continuous space, i.e. $x_i \in \mathbb{R}^D, i = 1, \dots, N$. A prototype-based clustering algorithm then produces a clustering or partition $P_k = \{C_1, \dots, C_k\}$, where $\emptyset \neq C_j \subseteq X, j = 1, \dots, k$. Centroids $p_j \in \mathbb{R}^D$ are obtained by averaging each cluster C_j's elements. The Euclidian metric in $\mathbb{R}^D$ is denoted by d. Symbols $p_{(i)}$ and $N_{(i)}$ are used to respectively denote the centroid and the size of the cluster $C_{(i)}$ to which object x_i belongs.

Next, the definition of the benchmark ICVIs for the computational study of Section 5 is recalled[2], emphasizing their reliance on cluster cohesion (usually assessed through its negative notion of cluster dispersion) and separation components, respectively denoted by $Disp$ and Sep. Typical application of an ICVI F to the NCE task departs from a sequence of partitions $P(X) = (P_k)_{k=\underline{k}}^{\overline{k}}$ provided by a clustering algorithm, where $\underline{k} \geq 1$ and $\overline{k} \leq N$ respectively denote the initial and final number of clusters of the partitions in the sequence. Then, a sequence $S(F) = (F_k)_{k=\underline{k}}^{\overline{k}}$ of ICVI values is obtained by applying F to each partition in $P(X)$. Finally, a certain decision criterion is applied on $S(F)$ to produce the estimated number of clusters $\hat{K} \in [\underline{k}, \overline{k}]$.

Inertia or sum of squared errors (SSE), the loss function of the k-Means algorithm,

$$SSE(P_k) = SSE_k = \sum_{i=1}^{N} d(x_i, p_{(i)})^2, \tag{1}$$

is possibly the most extended cluster cohesion measure, acting as the $Disp$ component of many ICVIs. The Calinski-Harabasz (CH, [29]) index is then defined as the ratio of between-clusters dispersion and within-cluster dispersion,

$$CH_k = \frac{Sep_k/(k-1)}{Disp_k/(N-k)}, \tag{2}$$

where $1 < k < N$, $Disp_k = SSE_k$ and $Sep_k = \sum_{j=1}^{k} N_j d(p_j, p)^2$, with p denoting the mean of X. The decision criterion is given by $\hat{K} = \arg\max_k S(CH)$. The Davies-Bouldin (DB, [30]) index consist of the average similarity between each cluster and its most similar one,

$$DB_k = \frac{1}{k} \sum_{j=1}^{k} \max_{l \neq j} \frac{Disp_j + Disp_l}{Sep_{j,l}}, \tag{3}$$

where $k > 1$ and $Disp_h = \frac{1}{N_h} \sum_{x_i \in C_h} d(x_i, p_h)$, $Sep_{j,l} = d(p_j, p_l)$, and $\hat{K} = \arg\min_k S(DB)$. The Silhouette Coefficient (SC, [31]) is a ratio of clusters' cohesion and separation

$$SC_k = \frac{1}{N} \sum_{i=1}^{N} \frac{Sep_k(i) - Disp_k(i)}{\max\{Sep_k(i), Disp_k(i)\}}, \tag{4}$$

where $Sep_k(i) = \min_{C_h \neq C_{(i)}} \frac{1}{N_h} \sum_{x_j \in C_h} d(x_i, x_j)$ and $Disp_k(i) = \frac{1}{N_{(i)}-1} \sum_{x_j \in C_{(i)}/j \neq i} d(x_i, x_j)$, for each $x_i \in X$ and $1 < k < N$. It is

[1] Detailed results of this study are presented in the attached Supplementary Materials file, containing Tables SM1-SM18. Besides, the code allowing to replicate the generation of synthetic data, the fitting process of the presented CRB-NCE methods, and the results of this study is available at the paper's GitHub https://github.com/sir-xabier/crb-nce.

[2] Two other baseline NCE methods used in the computational study of Section 5 are not described here due to their more complex and lengthy definition and because they do not lie within the cluster cohesion-separation framework applied here: one is a stability-based method, reval [28], and the other is a correlation-based index, NCI [10].

also $\hat{K} = \arg\max_k S(SC)$. The Tang-Sun (TS, [32]) index, a variation of the Xie-Beni index [33], measures a ratio of intra-class similarity and inter-class differences,

$$TS_k = \frac{Disp_k + \frac{1}{k(k-1)} \sum_{j=1}^{k} \sum_{l \neq j} Sep_{j,l}^2}{1/k + \min_{j \neq l} Sep_{j,l}^2}, \tag{5}$$

for $k > 1$, where again $Sep_{j,l} = d(p_j, p_l)$ and $Disp_k = SSE_k$, and $\hat{K} = \arg\min_k S(TS)$. A more modern variant of Xie-Beni is given by the Triple Center Relation (TCR) index [34], defined as

$$TCR_k = \frac{\frac{1}{N} Disp_k}{\frac{N}{k-1} \sum_{j=1}^{k} d(p_j, p)^2 \cdot \frac{1}{k(k-1)} \sum_{j=1}^{k} \sum_{l \neq j} Sep_{j,l}^2 \cdot \min_{j \neq l} Sep_{j,l}^2}, \tag{6}$$

where $k > 1$, $Disp_k = SSE_k$, $Sep_{j,l} = d(p_j, p_l)$ and $\hat{K} = \arg\min_k S(TCR)$.

The five previous indices feature both cluster cohesion and separation components and are thus associated with the measurement of clustering quality. The following ICVIs instead consider only a cohesion component and are either focused on model selection purposes or were directly proposed for the NCE task. The Bayesian information criterion (BIC, [35,36]) is a complexity-penalized form of the log-likelihood of the present partition,

$$BIC_k = \sum_{j=1}^{k} N_j \left(\ln \frac{N_j}{N} - \frac{D}{2} \left(\ln \frac{2\pi Disp_k}{D(N-k)} + 1 \right) \right), \tag{7}$$

where $k < N$, $Disp_k = SSE_k$, and $\hat{K} = \arg\max_k S(BIC)$. The Curvature (CV, [24]) method uses ratios of successive differences of inertia,

$$CV_k = \frac{Disp_{k-1} - Disp_k}{Disp_k - Disp_{k+1}}, \tag{8}$$

with $Disp = SSE$ and $\hat{K} = \arg\max_k S(CV)$, $1 < k < N$. Finally, the Variance Last Reduction (VLR, [8]) index compares clusters standard deviation to that of a uniform distribution with the smallest variance so far observed,

$$VLR_k = \sqrt{\frac{Disp_k}{\frac{N-k}{k^{2/D}} \min_{j=1,\ldots,k-1} \frac{j^{2/D}}{N-j} Disp_j}}, \tag{9}$$

for $1 < k < N$ and $VLR_1 = \gamma$, where $Disp_k = SSE_k$ and $\gamma < 1$ is a threshold such that $\hat{K} = \max\{k | VLR_k \leq \gamma\}$ ($\gamma = 0.99$ is suggested in [8]).

3. Coverage indices

Coverage indices are introduced here as an alternative to inertia for measuring cohesion, avoiding both its negative description in terms of dispersion and its scale-dependent behavior. This will also later enable to check the robustness of the proposed CRB-NCE method under variations of the underlying cohesion measure.

The intuition behind the notion of coverage used here is that the degree to which a cluster $C_j \in P_k$ covers or represents the objects $x_i \in X$ with $C_j = C_{(i)}$ may vary across those objects. Some will be more central to the cluster, while outlying objects may also exist. Following [27], coverage is seen as a gradable property, measured in degrees rather than as a binary condition, i.e. the key question is not whether an object is covered, but how much it is covered. Thus, coverage is to be assessed on its own scale, which can be taken as any closed interval $I = [a, b] \subset \mathbb{R}$, with $a < b$, where b denotes maximum or full coverage and a minimum or complete lack of coverage. Intermediate values in (a, b) then reflect partial coverage levels.

A definition of coverage in the context of prototype-based clustering is proposed next[3]. Here, centroids are regarded as the source of cluster coverage, so the closer an object is to its centroid, the higher its coverage level. This implies an inverse relationship between the distance $d(x_i, p_{(i)})$ and the coverage level of x_i. An exponentially decaying function is suitable for modeling coverage levels that diminish rapidly as x_i moves away from $p_{(i)}$, so that objects in well-separated clusters are scarcely covered by other clusters' centroids. Thus, the coverage level $u_i \in I$ of an object $x_i \in X$ by its cluster $C_{(i)}$ is defined as

$$u_i = a + (b-a) e^{-r\left(d(x_i, p_{(i)})/s\right)^m}, \tag{10}$$

where r, s, and m are positive parameters that provide flexibility in modulating the exponential decay. The scale parameter s defines the unit of distance in terms of d; the amplitude parameter r sets the coverage level $u_i = a + (b-a)e^{-r}$ for unit distance $d(x_i, p_{(i)}) = s$; and the exponent m adjusts the shape of the decay for $0 < d(x_i, p_{(i)})/s < 1$. In practice, setting s as the length of a major diagonal of the smallest D-dimensional hypercube $H \subset \mathbb{R}^D$ containing X together with $r = 2 \ln 10$ gives good results[4], leading to a coverage level of $u_i = a + (b-a) \cdot 10^{-2}$ at distance s for any choice of m. Hence, regarding the choice of s, if data is min-max normalized, then $H = [0, 1]^D$ and $s = \sqrt{D}$; under standardization, $H = [-2.5, 2.5]^D$ can be assumed, and then $s = 5\sqrt{D}$. In this work, we explore values $m \in \{1, 2\}$, with which the right-hand side of Eq. 10 becomes related to the mountain function introduced by Yager in [25] for density estimation when selecting prototypes from a grid (using $m = 1$), and later applied in [26] (with $m = 2$) in the context of subtractive clustering.

Next, the production of a coverage index of a partition P_k from the coverage levels u_i of all $x_i \in X$ is addressed. This index aims to reflect how well the objects in X are *globally* covered by the clusters in P_k. We then first focus on providing a comprehensive framework capturing the essential features any such *global* index should verify.

Let $\mathbf{u} = (u_1, \ldots, u_N) \in I^N$ be referred to as the coverage vector, and let $\mathbf{a}$ (resp. $\mathbf{b}$) denote an N-dimensional vector with all its components equal to a (resp. b). The task of producing a coverage index can be stated as that of defining a suitable operator $F : I^N \to I$ mapping $\mathbf{u}$ into an index value $F(\mathbf{u}) \in I$. Any operator F verifying the following six conditions will be referred to as a coverage index (CI):

(CI1) F is continuous.
(CI2) F is non-decreasing.
(CI3) F is symmetric.
(CI4) $\min(\mathbf{u}) \leq F(\mathbf{u}) \leq \max(\mathbf{u})$, for all $\mathbf{u} \in I^N$.
(CI5) $F(\mathbf{u}) = a$ if and only if $\mathbf{u} = \mathbf{a}$.
(CI6) $F(\mathbf{u}) = b$ if and only if $\mathbf{u} = \mathbf{b}$.

Conditions CI1-CI3 provide a sound general behavior in terms of smoothness, monotonicity, and symmetry. Specifically, CI1 prevents abrupt changes in the index from small variations in $\mathbf{u}$; CI2 ensures that increasing any coverage level u_i cannot decrease the index; and CI3 guarantees invariance under permutations of objects in X, making the index independent of object ordering. This last condition rules out non-ordered weighted means, whose weights are tied to specific positions

[3] Any specific coverage model should be linked to the structural features of the clustering method that produces the assessed partition. For instance, in prototype-based methods (such as k-centers algorithms), it makes sense to define coverage in terms of the proximity of x_i to its cluster's prototype, i.e. the centroid $p_{(i)}$, in terms of the same distance d used by the method. In distribution-based methods (e.g., Gaussian mixtures), this proximity would be more appropriately measured via the Mahalanobis distance associated with the cluster's covariance matrix. In graph- or density-based methods (e.g., spectral clustering or DBSCAN), coverage may instead be better modeled through graph-theoretic notions [7], such as an object's centrality within its cluster's subgraph.

[4] The value $r = 2 \ln 10$ is chosen to associate fractional powers of 10^{-2} with the corresponding fractions of distance s, based on the idea that the maximum plausible distance in the standardized/normalized data, i.e., s, has to correspond to a small covering level-above a by only a proportion of 10^{-2} of the length $(b-a)$ of the covering interval $I = [a, b]$. In this way, for example, the distances $s, s/2, s/3$, etc., are associated with the proportions $10^{-2}, 10^{-1}, 10^{-2/3}$, etc.

of the coverage vector. Condition CI4 imposes the averaging nature of the index, avoiding extreme conjunctive and disjunctive behaviors. Finally, CI5 and CI6 define the index's boundary conditions, respectively reserving the extreme valuations $F(\mathbf{u}) = a$ and $F(\mathbf{u}) = b$ for the cases in which all objects have minimum or maximum coverage (i.e., $\mathbf{u} = \mathbf{a}$ or $\mathbf{u} = \mathbf{b}$, respectively). These two conditions also impose a compensative behavior, forcing the index to balance all information in $\mathbf{u}$, thus excluding operators such as the median, which would output a (resp. b) as soon as $\lfloor N/2 \rfloor + 1$ entries equal a (resp. b).

Although various specific CIs can be defined under this framework, possibly the most natural choices are those associated to standard means, such as the arithmetic, harmonic, or geometric mean. In this paper, the use of the arithmetic mean is explored due to its simplicity and familiarity. Thus, the *mean coverage index* (MCI) of the data X with respect to partition P_k is defined as

$$MCI_k = \frac{1}{N}\sum_{i=1}^{N} u_i. \tag{11}$$

It is straightforward to verify that the MCI satisfies all conditions (CI1)-(CI6).

Clearly, a partition in which objects lie closer to their respective cluster centroids, i.e. one with more cohesive clusters, is associated with a higher MCI value. The MCI can then be interpreted as a cohesion measure that assesses cluster cohesion positively, rather than through its negative notion of dispersion, as inertia does. Moreover, although MCI and inertia are negatively correlated, MCI offers a distinct perspective on cohesion due to its formulation based on exponential decay, as well as its bounded and scale-invariant nature. For instance, when all objects coincide with their centroids, MCI reaches its maximum value b, whereas inertia drops to 0. As objects move away from centroids, MCI approaches a, while inertia grows unbounded. Moreover, adding new objects to X, even if equally well-covered as those already present, causes inertia to increase indefinitely, whereas MCI remains stable.

4. A cohesion rule-based procedure for NCE

A sequence of cohesion measurements usually exhibits a monotonic behavior as the number of clusters k of the partitions increases. For this reason, cohesion measures such as SSE or MCI are not directly applicable to the NCE task on their own, i.e. without being further processed into more suitable indices, as the ICVIs presented in Section 2 do. In this section, we introduce the proposed cohesion rule-based method for number of cluster estimation (CRB-NCE, or simply CRB). This denomination reflects two key features of the method, which are described below.

Firstly, and similarly to the BIC, CV, and VLR indices (see Eqs. 7-9), the CRB method relies only on a cluster cohesion component, without incorporating a cluster separation one. In fact, CRB can be viewed as an extension or refinement of the CV method (see Eq. 8), as it also uses ratios of differences between successive cohesion measurements. However, unlike CV, CRB additionally incorporates second-order differences and considers two ratios simultaneously, rather than just one. Moreover, it employs a more sophisticated type of ratio, referred to as a *tail ratio*. A tail ratio essentially compares the k-th element in a sequence of differences to the remaining h-th elements, for $h > k$. As will be shown below, tail ratios provide more robust and precise evidence for determining the correct number of clusters K.

Secondly, CRB uses a rule-based decision criterion to estimate K from the sequence of tail ratios. To some extent, this resembles the decision criterion used in VLR (see text below Eq. 9), in that it employs thresholds (rather than sequence maxima or minima as most ICVIs do) to determine the estimation $\hat{K}$. Thresholds are useful, if not essential, given that CRB operates with two ratios. Furthermore, as will be illustrated in Section 5.4, thresholding enables fine-tuning of the CRB method to better adapt it to the particular characteristics of the data on which the NCE task is being performed.

Finally, let us remark that CRB is applicable to any monotonic cohesion measure: the examples in this section illustrate the method using the proposed MCI, but we will also apply CRB to SSE sequences in the computational study of Section 5.

4.1. Tail ratios

Let $S(F) = (F_k)_{k=\underline{k}}^{\bar{k}}$ be a sequence of values of a monotonic cohesion measure F. The first order difference of $S(F)$ is denoted by $\Delta_k^1 = F_{k+1} - F_k$, with $k = 1, \ldots, \bar{k} - 1$. If F is increasing, it is $\Delta_k^1 \geq 0$ for all k. If F is decreasing, for convenience we switch the sign of the sequence Δ^1 so that it is positive, i.e. assign $\Delta_k^1 = -\Delta_k^1$ for all k. The second order difference is then computed from Δ^1 as $\Delta_k^2 = \Delta_{k+1}^1 - \Delta_k^1, k = 1, \ldots, \bar{k} - 2$. A sharp drop in both Δ^1 and Δ^2 at a given k indicates a deceleration in $S(F)$ and suggests that P_{k+1} yields comparatively more cohesive clusters than previous partitions. The ratio of successive differences, defined as $R_k^q = \Delta_k^q / \Delta_{k+1}^q$, is useful for identifying such drops in $\Delta^q, q = 1, 2$: the larger the ratio, the sharper the associated drop. This is precisely the intuition behind the CV index, which is defined as $CV_k = R_k^1$ using $F = SSE$, as shown in Eq. 8. It is therefore natural to consider estimating the number of clusters K as $\hat{K} = k + 1$, where k maximizes R_k^q (see e.g. the decision criterion of the CV index below Eq. 8).

Although this line of reasoning is basically sound, observation of diagrams such as those in Figs. 1 and 2(c1) reveals some patterns that may be useful in improving the estimation of K:

1. Relatively high values of the ratios R^q appear by chance, since the differences Δ_k^q fluctuate quite randomly as k grows above K due to the existence of random substructures.
2. The magnitudes $|\Delta_k^q|$ for $k \geq K$ tend to remain below a certain bound, considerably smaller than the magnitudes observed for $k < K$.
3. Magnitudes $|\Delta_k^q|$ similar to those attained just before a random drop tend to be observed again shortly after.
4. It is unlikely that a random sharp drop occurs in both Δ^1 and Δ^2 for the same $k > K$.

Item 1 above entails that the argmax of ratios R^q may lack robustness as an estimator of K, especially when $\bar{k}$ is quite greater than K (see Fig. 1 as well as the behavior of CV in the $\bar{k}$ panel of Fig. 3 and Table SM2). Items 2 and 3 point out that comparing a certain Δ_k^q with all Δ_h^q such that $h > k$ can bring in a more robust behavior (see Fig. 1). Item 4 suggests that simultaneously considering ratios on both differences can further enhance the efficacy of an estimator (see Fig. 2).

To leverage these observations, we define the first and second order *tail ratios* as

$$TR\Delta_k^1 = \frac{\Delta_k^1}{\max_{h=k+1,\ldots,\bar{k}-1} \Delta_h^1}, \quad k = 1, \ldots, \bar{k} - 3, \tag{12}$$

$$TR\Delta_k^2 = \frac{\Delta_k^2}{\min_{h=k+1,\ldots,\bar{k}-2} \Delta_h^2}, \quad k = 1, \ldots, \bar{k} - 3, \tag{13}$$

in such a way that the tail ratio $TR\Delta_k^q$ weighs the difference Δ_k^q against the remaining values Δ_h^q in its right tail, i.e., for $h > k$. In the case of $TR\Delta^1$, each Δ_k^1 is compared to the maximum value in its tail. A high value of $TR\Delta_k^1$ indicates both that a sharp drop in Δ^1 is about to occur and that the magnitude $|\Delta_k^1|$ prior to the drop is not reached again. For $TR\Delta^2$, a similar comparison is made between Δ_k^2 and its tail, but here the minimum value in the tail is used, since a deceleration in the growth of the sequence $S(F)$ is associated with a negative second-order difference. Therefore, a high positive value of $TR\Delta_k^2$ typically signals that $S(F)$ is decelerating at $k + 1$ and that no deceleration of similar magnitude occurs afterward.

Hence, a high value of either $TR\Delta_k^1$ or $TR\Delta_k^2$ suggests that $k + 1$ may be the correct number of clusters. Preliminary experimental results indicate that $TR\Delta^1$ tends to be more reliable than $TR\Delta^2$ (see left panel of Fig. 1), and thus a natural first choice for an estimator of K would be $\hat{K} =$

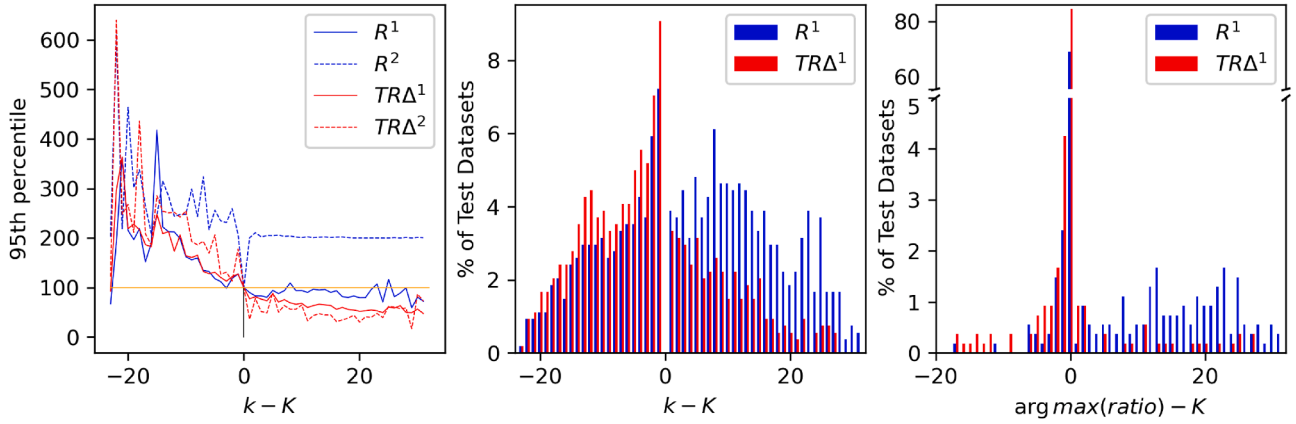

Fig. 1. Example 1: Comparison of the behavior of ratios R^1, R^2 and tail ratios $TR\Delta^1, TR\Delta^2$, using $F = MCI$ and the synthetic Gaussian test data described in Section 5.1 (540 datasets, each with a different number of clusters K, and corresponding sequences $S(F)$ of length $\bar{k} = 35$ obtained via k-Means). In this example, the ratios are set to 100 at $k = K$ and scaled proportionally for $k \neq K$. **Left**: 95th percentile (interpolation tends to occur at both extremes) of ratio values for each difference $k - K$ across the test datasets. Ratios tend to exceed their value at K for $k < K$, specially for the second-order ratios R^2 and $TR\Delta^2$. For $k > K$, however, tail ratios consistently produce smaller values than 100 and also than the corresponding ratios R^1 and R^2. **Center**: Percentage of test datasets for which each ratio's value at $k - K$ exceeds 100. $TR\Delta^1$ shows a slightly greater tendency than R^1 to underestimate K, whereas R^1 shows a much stronger tendency to overestimate. **Right**: Percentage of test datasets for which the ratios' argmax estimator is triggered at each $k - K$. The estimator is considerably more accurate for $TR\Delta^1$ than for R^1, note the value at $k - K = 0$ on the top broken axis.

$am^1 + 1$, where $am^1 = \arg\max_k TR\Delta^1_k$. We refer to this estimator as the *default estimator* (DE). When am^1 coincides with $am^2 = \arg\max_k TR\Delta^2_k$, the DE gains additional support. Let us refer to the equality $am^1 = am^2$ as the *combined argmax* (CA) condition. However, when CA does not hold, due to the DE's tendency to underestimation (see Fig. 1) it is possible that alternative values $k \neq am^1$ for which $TR\Delta^1_k$ is relatively high provide a more accurate estimate $\hat{K} = k + 1$ than DE. We refer to this third case as the *alternative estimator* (AE). The next section addresses how to combine and operationalize these three options DE, CA, and AE to produce an improved estimator of K.

4.2. CRB Method: Exploiting tail ratios for NCE

The CRB method is presented here using the running example in Fig. 2 to facilitate understanding and illustrate its features. These examples use $F = \text{MCI}$ as the base cohesion measure to also illustrate it, but completely similar patterns would have been reached by using $F = SSE$ instead.

Thus, consider a 2-D dataset X such as that shown in Fig. 2(a1), with $K = 4$ well-separated clusters. The k-Means algorithm is applied on X after standardization, producing a partition P_k with k clusters. A coverage vector $\mathbf{u}_k$ is obtained from P_k by using Eq. 10 with $I = [0, 1]$ and $m = 1$. $MCI_k = MCI(\mathbf{u}_k)$ then provides the mean coverage of X by partition P_k. A sequence $S(MCI) = (MCI_k)_{k=\underline{k}}^{\bar{k}}$ is produced by varying k between $\underline{k} = 1$ and a prefixed $\bar{k}$. Figs. 2(b1)-(d1) respectively show the resulting monotonically increasing sequence $S(MCI)$ for $\bar{k} = 20$, the first and second order differences of $S(MCI)$, and the corresponding first and second order tail ratios sequences obtained by Eqs. 12 and 13. Particularly, notice that both $TR\Delta^1_k$ and $TR\Delta^2_k$ reach their maximum at $k = am^1 = am^2 = 3$. Thus, the CA condition holds, and the estimation provided by the DE, $\hat{K} = am^1 + 1 = 4$, is correct. This combination of CA and DE (to be referred to as CA + DE) tends to be highly accurate in datasets with well-separated clusters.

However, the more the clusters overlap, the less likely the CA condition holds, and the more likely that the DE underestimates K (see Figs. 1 and 4). To see how to address this issue, consider first the example in Fig. 2(a2)-(b2): Although due to 2 pairs of overlapping clusters the dataset might appear to contain 6 groups, it actually consists of $K = 8$ clusters (see the actual centers signaled by red crosses). Notice that the CA condition does not hold in this case, and $TR\Delta^1$ reaches its maximum at $k = 5$, so the DE would now produce $\hat{K} = 6 \neq K$. Nevertheless, $TR\Delta^1$ attains a relatively high value at $k = 7$, and thus a correct estimation would instead be obtained if the AE is triggered at $k = 7$ providing $\hat{K} = k + 1 = 8 = K$. Consider also the example in Fig. 2(c2)-(d2): Again, although the dataset may seem to contain just 2 groups, it actually features $K = 5$ quite overlapped clusters. In this case, both tail ratios reach their maximum at $am^1 = am^2 = 1$, so the CA conditions holds and CA + DE would produce $\hat{K} = am^1 + 1 = 2 \neq K$. However, the value of $TR\Delta^1_{am^1}$ is much lower than in Fig. 2(d1), and $TR\Delta^1$ attains a relatively high value later at $k = 4$, and thus a correct estimation would instead be obtained if the AE is triggered and provides $\hat{K} = k + 1 = 5 = K$. These examples suggest that a more demanding condition should be added to the CA one, allowing AE to be triggered instead of CA + DE when either CA or this more exigent condition is not satisfied. Moreover, a further criterion is required to determine when the "$TR\Delta^1$ attains a relatively high value" premise of the AE is met.

Therefore, rather than relying on a single estimator to derive $\hat{K}$, it may be more effective to combine multiple indicators through carefully defined conditions. This is the approach in the design of the proposed CRB method. In particular, it involves setting thresholds on $TR\Delta^1$ values to establish minimum evidence levels required to activate each estimator. As discussed above, higher $TR\Delta^1_k$ values indicate sharper and more definitive decelerations in the cohesion sequence, and can thus be interpreted as stronger evidence for the corresponding estimation $\hat{K} = k + 1$.

The CRB method is detailed in Algorithm 1. It defines a three-step estimation procedure based on two thresholds, δ_1 and δ_2. First, if the CA condition holds and the evidence level $TR\Delta^1_{am^1} > \delta_1$ is met (solid red line in Fig. 2), the DE $\hat{K} = am^1 + 1$ is returned immediately. Second, if CA does not hold or the evidence is insufficient, the method looks for the largest k such that $TR\Delta^1_k > \delta_2$ (dotted red line in Fig. 2). If such a k exists, the AE is used and $\hat{K} = k + 1$ is returned. Finally, if no k satisfies the AE condition, the DE $\hat{K} = am^1 + 1$ is returned (see Fig. 2(a3)-(b3)).

The computational or time complexity of Algorithm 1 is practically negligible, and similar in any case to the application of any of the ICVIs exposed in Section 2. Specifically, the computation of the differences Δ^1 and Δ^2, as well as that of their right-tail maxima and minima can be accomplished in a single loop with complexity order $O(\bar{k})$. The same applies to the computation of both tail ratios and the positions of their

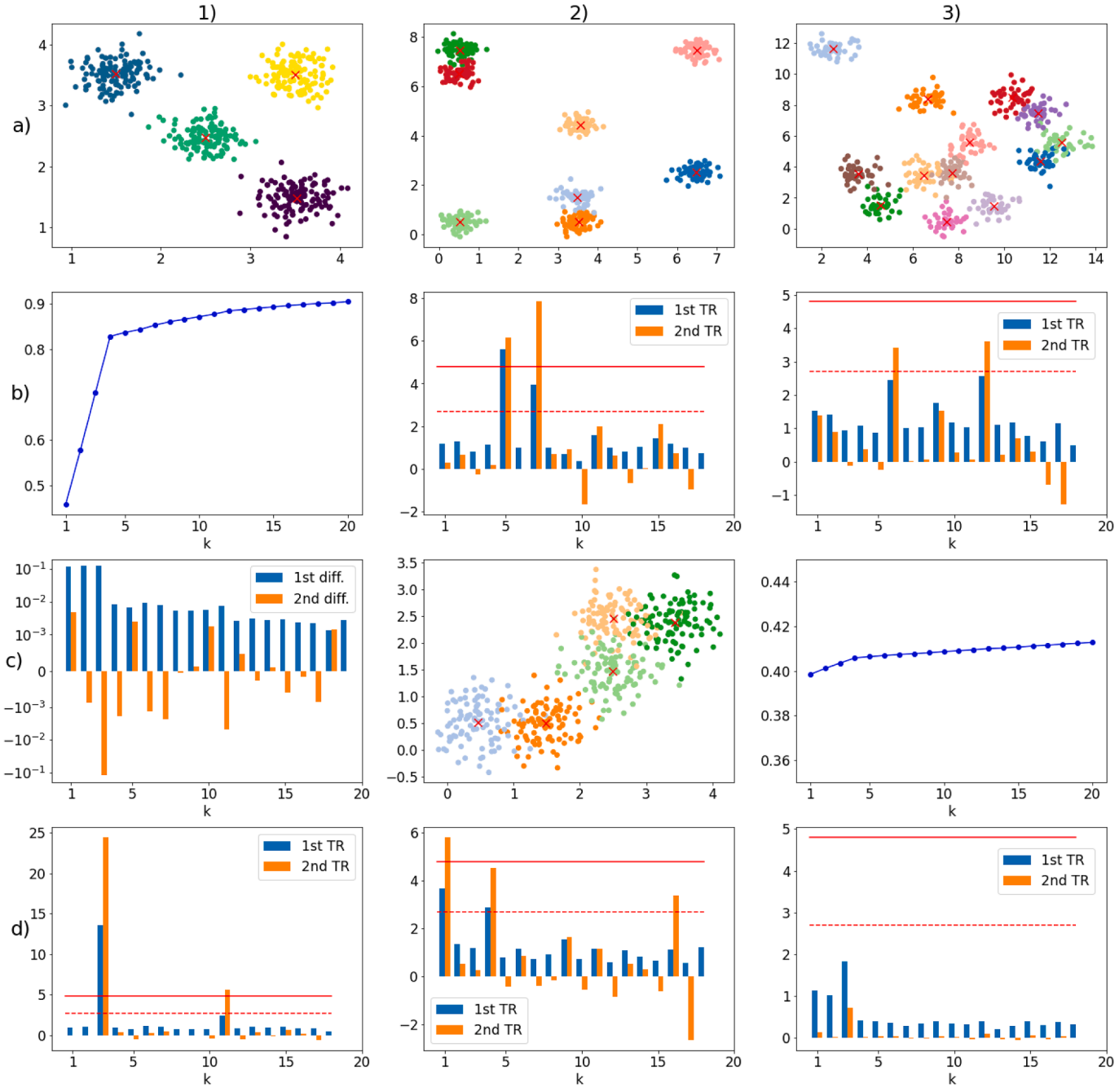

Fig. 2. Example 2: Application of tail ratios to the NCE task using Algorithm 1. 2-D synthetic Gaussian blob datasets were generated as described in Section 4.3.2, with blob centers marked by red crosses (blobs correspond to colors). The data is clustered using the k-Means algorithm with $\bar{k} = 20$. The solid and dotted red lines respectively represent the CRB-MCI thresholds δ_1, δ_2 in Table 2. **Left**: (a1) Dataset with $K = 4, Std = 0.15$; (b1) Sequence $S(MCI)$; (c1) First- and second-order differences of $S(MCI)$, note the logarithmic scale on the y-axis; (d1) Tail ratio sequences $TR\Delta^1$ and $TR\Delta^2$. CA + DE is triggered, producing $\hat{K} = 4$. **Center**: (a2)-(b2) Dataset with $K = 8, Std = 0.23$; CA does not hold, and thus CA + DE is not triggered despite $TR\Delta^1_{am^1} > \delta_1$. Instead, AE is activated, yielding $\hat{K} = 8$; (c2)-(d2) Dataset with $K = 5, Std = 0.34$; CA + DE is not triggered because $TR\Delta^1_1 < \delta_1$. AE then produces $\hat{K} = 5$. **Right**: (a3)-(b3) Dataset with $K = 13, Std = 0.5$; CA holds, but neither CA + DE nor AE are triggered. DE is used instead, producing $\hat{K} = 13$; (c3)-(d3) Flat $S(MCI)$ of a dataset with $D = 500, K = 4, Std = 0.15$; $TR\Delta^1$ values are low, so only DE can be used and yields $\hat{K} = 4$ in this case. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

maxima. When conditions for the application of CA + DE does not hold, the consideration of AE again requires $O(\bar{k})$ operations. Thus, the complexity of Algorithm 1 is just $O(\bar{k})$. Besides, when $F = SSE$ or $F = MCI$, the computation of the cohesion sequence $S(F)$ requires $O(ND\bar{k} + N\bar{k}) = O(N\bar{k}D)$ operations to, first, obtaining the distance (in a D-dimensional space) between each of the N objects x_i and the associated centroid $p_{(i)}$ for each of the $\bar{k}$ partitions in the sequence $P(X) = (P_k)_{k=1}^{\bar{k}}$, and secondly computing F_k (which requires the addition of N values) for each $k = 1, \ldots, \bar{k}$. Finally, obtaining $P(X)$ requires $O(\bar{k}C)$ operations, where C denotes the complexity of the considered clustering algorithm. Typically, the complexity of this last step dominates the whole process, but this does not produce any correlative complexity difference between ICVIs and CRB, as ICVIs also require the computation of sequences $P(X)$.

Algorithm 1: Cohesion Rule-Based (CRB) procedure for NCE.

Input:Sequence $S(F)$, thresholds $\delta_1, \delta_2 > 0$, boolean *F_is_decreasing* **Output**:$\hat{K}$ Compute sequence $\Delta^1(F)$, the first-order difference of $S(F)$ **if** *F_is_decreasing* **then** $\Delta^1(F) = -\Delta^1(F)$;
Compute sequence $\Delta^2(F)$, the first-order difference of $\Delta^1(F)$
Compute sequences $TR\Delta^1$ and $TR\Delta^2$ using Eqs. (12) and (13)
Compute $am^1 = \arg\max TR\Delta^1$ and $am^2 = \arg\max TR\Delta^2$ Assign $\hat{K} = am^1 + 1$ (DE) **if** $am^1 = am^2$ ***and*** $TR\Delta^1_{am^1} > \delta_1$ **then return** $\hat{K}$ (CA + DE);
else
  for $k = 1$ **to** $\bar{k} - 4$ **do**
    if $TR\Delta^1_k > \delta_2$ **then** $\hat{K} = k + 1$ (AE);
  return $\hat{K}$

4.3. Optimizing CRB thresholds

The CRB procedure in Algorithm 1 provides flexibility to apply different estimators of K in view of the available evidence, as determined by the thresholds δ_1 and δ_2. Therefore, carefully choosing the value of these thresholds is paramount for the effectiveness of CRB. To this aim, here an evolutionary optimization approach have been applied based on synthetic data.

4.3.1. Optimization procedure

The specifics of the optimization process are as follows: A training and a validation sample, each with T synthetic datasets $D_t, t = 1, \ldots, T$, are generated to feed a genetic algorithm. Each D_t features a known number of clusters $K_t > 1$. Next, for each t, a clustering algorithm allowing to specify the desired number of clusters k to be obtained is run on D_t, varying k from $\underline{k} = 1$ to a given upper limit $\bar{k}$. This produces a sequence of partitions $(P^t_k)_{k=1}^{\bar{k}}$. A cohesion measure F is then computed on each P^t_k, yielding a cohesion sequence $S_t(F) = (F^t_k)_{k=1}^{\bar{k}}$. $S_t(F)$ can then be input into Algorithm 1, producing for any given choice of δ_1 and δ_2 an estimation $\hat{K}_t$, which can be compared with K_t. The genetic algorithm thus evolves a population of individuals composed of real-coded chromosomes with components δ_1 and δ_2. The individuals' fitness is evaluated by a combination of accuracy (Acc) and root mean squared error ($RMSE$),

$$f(\delta_1, \delta_2) = Acc - \alpha \cdot RMSE = \frac{1}{T}\sum_{t=1}^{T}(\hat{K}_t = K_t) - \alpha\left(\frac{1}{T}\sum_{t=1}^{T}(\hat{K}_t - K_t)^2\right)^{1/2} \tag{14}$$

where $(\hat{K}_t = K_t) = 1$ if $\hat{K}_t = K_t$ and 0 otherwise. Parameter α determines the fitness balance between Acc and $RMSE$. A population of pop individuals is randomly initialized and evolved through cxBlend crossover, Gaussian mutation, and SPEA2 selection according to individuals' fitness in the training sample. To avoid overfitting and local minima, all individuals remaining after selection are evaluated in the validation sample, the mean validation fitness is computed, and the population is randomly restarted each time this mean validation fitness does not improve for W consecutive iterations, or when convergence is declared due to the standard deviation of the selected individuals' training fitness being lower than a certain tolerance ε. The algorithm finishes after R restarts or n_iter iterations. The returned solution is given by the individual with the best validation fitness throughout the whole process. In this work, we have used $\alpha = 10^{-7}$, crossover and mutation probabilities of 0.3 and 0.9, respectively, $W = 5, \varepsilon = 10^{-10}$, $R = 30$, and $n\_iter = 500$, although many other configurations were also tried[5].

[5] Specifically, we chose α small enough that the $RMSE$ component only serves to break ties among individuals with the same Acc. As described in Section 4.3.2, the training and validation samples comprise 2430 sequences; thus, $\alpha = 10^{-7}$ ensures the desired effect, since $1/2430 \approx 4E-4$ and $RMSE < 1E2$. Similarly, $\varepsilon = 10^{-10}$ and $W = 5$ provided sufficient exploration of the current optimum's neighborhood before restarts, while $pop = 500$, $R = 30$, and $n\_iter = 500$ offered a good balance between exploration and execution time. These parameters were tuned until the training and validation results stabilized across different random seeds.

Table 1
Levels of complexity factors used to define the scenarios for synthetic dataset generation. Each scenario corresponds to a combination of one level from each factor, for a total of $3^3 = 27$ scenarios.

Factor	K	D	Std
Levels	2–5, 6–9, 10–25	2, 3–9, 10–50	0.1-0.19, 0.2-0.29, 0.3-0.5

4.3.2. Synthetic data generation

Before describing the details of the synthetic data generation procedure, let us remark that the thresholds δ_1 and δ_2 resulting from this optimization process are those that enable the correct estimation of the actual number of clusters of the greatest possible amount of training and validation datasets. In this sense, it is important to stress that the CRB procedure using these thresholds is optimized for the context or conditions defined by the features of the training and validation data fed to the genetic algorithm. Therefore, in order to deliver a CRB procedure that can adequately perform the NCE task in different, general conditions, the characteristics of the data with which it is fitted have to be equally general.

Consequently, the data used to feed the genetic algorithm is composed of synthetic datasets featuring a mix of conditions in terms of the complexity factors number of clusters K, data dimensionality D, and level of cluster overlap, which is controlled through the clusters' standard deviation Std. A total of $27 = 3^3$ complexity scenarios are obtained by crossing the 3 levels defined for each of the 3 factors. These levels correspond to the ranges shown in Table 1. 10 datasets are generated in each scenario, for a total of 270 datasets, thus gathering an extensive sample of clustering problems with different characteristics and complexity.

The generation of the datasets in these 27 scenarios proceeds as follows: For each dataset of a given scenario, values of K and D are randomly drawn from the corresponding ranges shown in Table 1. The parameter Std is however chosen from its range in a uniformly increasing, nonrandom manner. For instance, the first dataset generated in a scenario for which $Std \in [0.1, 0.19]$ has $Std = 0.1$, the second $Std = 0.11$, and so on until the tenth, that has $Std = 0.19$. Then, the centers or means of the K clusters are located at the center of K hypercubes randomly selected without replacement from a grid of K^D unit-length hypercubes. For instance, for $K = 3$ and $D = 2$ this grid is composed of the hypercubes $[0, 1]^2, [0, 1] \times [1, 2], [0, 1] \times [2, 3], [1, 2] \times [0, 1], \ldots, [2, 3]^2$. Then, Std times the minimum distance between any pair of the K centers is taken as the standard deviation of all the clusters' isotropic Gaussian distributions, from which approximately $500/K$ instances are drawn for each of the K clusters, such that each dataset is composed of $N = 500$ instances. This procedure guarantees separated centers so that the complexity of the NCE task due to the level of clusters overlap can be modulated through the Std parameter[6] (see Fig. 2 for illustrative examples of the overlap levels in the generated datasets).

This process is applied to generate independent training and validation samples with $T = 270$ datasets each. Three clustering algorithms are then run on each dataset of both samples to generate sequences of

[6] As is well known, the probability that a value x randomly drawn from a $N(\mu, \sigma)$ distribution lies within the interval $|x - \mu| \leq 2\sigma$ is about 95 %. Therefore, since the distance between the closest centers is defined as one Std unit, clusters will tend to be well-separated when $Std < 0.2$; moderately overlapped when $0.2 \leq Std < 0.3$; and quite overlapped when $0.3 \leq Std < 0.5$, though not to the point of becoming fully indistinguishable.

Table 2
Optimized thresholds δ_1 and δ_2 of the CRB procedure for different cohesion measures.

F	δ_1	δ_2
SSE	10.61968002	2.557468209
MCI	4.851226028	2.724030522
MCI2	10.21417873	2.533375136

partitions $(P_k^t)_{k=1}^{\bar{k}}, t = 1, \ldots, T$: k-Means, Agglomerative Clustering and k-Medoids, applied using their default *scikit-learn* [39] and *scikit-learn-extra* implementations. For each k, the clusterings provided by k-Means and k-Medoids are those with the best value of inertia among 10 k-Means++ initializations, which are shared by both algorithms. All datasets are standardized before the application of the clustering algorithms.

Moreover, three choices of the upper limit $\bar{k}$ are used to also allow adaptation to the effect of different sequence lengths on NCE performance: Since the learning datasets' number of clusters ranges from $K = 2$ to $K = 25$, a first choice is a fixed value of $\bar{k} = 35$, that covers all datasets and allows a moderate upper tail above the actual K for those datasets with a higher K. To consider the effect of extending this tail, a second choice sets $\bar{k} = 50$. Finally, an adaptive or variable sequence length setting, referred to as $\bar{k} = Var$, is tried, in which for each dataset $\bar{k}$ depends on K as follows: if $K \leq 5$, then $\bar{k} = 15$; else, if $K \leq 9$, then $\bar{k} = 25$; otherwise $\bar{k} = 35$. This *Var* setting reflects a situation in which some approximate *a priori* information exists on the actual K.

As a result, both the training and validation sample consist of 2430 (270 datasets × 3 clustering algorithms × 3 upper limits $\bar{k}$) different sequences $(P_k^t)_{k=1}^{\bar{k}}$. Three cohesion measures F are then applied on these partitions to produce sequences $S(F)$ that can be input into Algorithm 1: SSE, MCI using $m = 1$ in Eq. 10, and MCI using $m = 2$ (referred to as MCI2). The boolean $F\_is\_decreasing$ is set to *False* for MCI and MCI2, and *True* for SSE. Finally, 3 runs of the exposed genetic algorithm are carried out, one for each of the considered cohesion measures. The resulting thresholds δ_1, δ_2 of the corresponding CRB procedures are shown in Table 2.

5. Computational study

This study focuses on assessing the performance of the proposed CRB procedures at the NCE task, comparing it to that of a baseline of NCE methods. To this aim, three experiments are carried out: A first one analyzes CRB methods' performance on similar conditions to those considered in optimizing their thresholds, thus using an extensive test sample composed of Gaussian blobs synthetic datasets. The second experiment instead evaluates the performance of CRB methods on real-world and clustering benchmark synthetic datasets, thus studying their generalization ability to different conditions from those to which they were fitted. A third experiment is devoted to compare the performance of the CRB and the baseline methods on high-dimensional data. Finally, a fourth experiment analyzes whether the performance of specialized CRB procedures can improve on that of the base, reference CRB procedures.

5.1. Synthetic gaussian datasets

In this experiment, CRB and baseline ICVIs are applied to estimate the actual number of clusters K of synthetic Gaussian data with similar conditions to the training and validation samples on which CRB's thresholds were optimized. To this aim, a test sample is built through the same steps described in Section 4.3.2, with the only difference that now the number of observations N of the datasets is not fixed, but varies in 2 levels: $N = 500$ and $N = 10000$. Thus, a total of 54 complexity scenarios are obtained crossing the levels of factors K, D and Std, given in Table 1, and those of N. As above, 10 Gaussian blobs datasets are generated in each scenario, for a total of $T = 540$ datasets. Of course, a different random seed is used so that a test sample totally independent from the training and validation samples described above is obtained.

Sequences of partitions with 3 different lengths $\bar{k} = Var, 35, 50$ are then obtained by applying the Agglomerative Clustering, k-Means, and k-Medoids algorithms on this test sample, with the same considerations as in Section 4.3.2. Notice that when using the Euclidean distance, these algorithms tend to provide spherical clusters that are more or less well-represented by their centroids, although differences on the resulting clusterings and such a goodness of representation are expected due to the particularities of each algorithm. This allows studying the performance of NCE methods under different conditions of prototype validity of the clusters' centroids, while still providing approximately spherical solutions matching the methods' assumptions. We consider only shallow clustering algorithms since the application of deep algorithms (see e.g. [37]) to generate the sequence of partitions would not produce any difference in the comparative assessment of the NCE methods' performance in relation with shallow algorithms.

Thus, a total of 4860 (540 datasets × 3 algorithms × 3 lengths) partition sequences are obtained, on which NCE methods are applied to estimate K. The considered baseline ICVIs are the 8 indices exposed in Section 2 plus the correlation-based index NCI [10]. Three CRB estimators are applied using Algorithm 1, corresponding to the cohesion measures $F = SSE, MCI, MCI2$ and the respective thresholds in Table 2. The application of any of these methods produces an estimation $\hat{K}$ for each sequence. The estimators' performance is then assessed by measuring the percentage of sequences %Acc for which $\hat{K}$ coincides with the actual number of clusters K.

Table 3 shows the resulting %Acc across the 4860 sequences of the test sample. Due to this relatively high number of sequences and the variety of conditions under which they are produced, this %Acc should constitute a considerably robust estimator of each method's NCE performance. Then, a first observation is that the 3 CRB methods, i.e. SSE, MCI and MCI2, obtain a rather clear superior performance to that of the baseline ICVIs: CRB methods correctly estimate K for at least 4 % more sequences in absolute terms than BIC, the best performing baseline ICVI. In turn, MCI leads within CRB methods, with roughly a 2 % improvement on SSE and MCI2, and a 6 % on BIC, which corresponds to 308 correctly estimated sequences more. Besides, let us stress that tail ratios-based CRB methods attain around a 10 % improvement on the R^1 ratio-based CV estimator, further supporting the performance difference exposed in Fig. 1. Finally, notice also that the six best performing methods in Table 3, i.e. BIC, CV, VLR, SSE, MCI and MCI2, do only consider a cluster cohesion component, in contrast with CH, DB, SC, TS and TCR, which also incorporate a separation one.

It is possible to assess the statistical significance of these seemingly different performances by applying nonparametric statistical tests (see [38]). To this aim, the 4860 test sequences are grouped into the 54 complexity scenarios from which the corresponding test datasets were drawn, with 90 sequences (10 datasets × 3 algorithms × 3 sequence lengths) in each scenario. In this way, scenarios can be regarded as blocks of similar experimental conditions, so that the omnibus Friedman test[7], a non-parametric equivalent of two-way ANOVA, can be applied

[7] The Friedman test, as well as the subsequent post-hoc Holm's test, is based on translating the %Acc performances of the estimators at each scenario into ranks, in such a way that the worst performing index obtains rank 1, the second-worst gets rank 2, and so on until the best performing one is reached, which would obtain rank 10 in the absence of ties. Average ranks are assigned in case of ties (e.g. if exactly two estimators are tied as the best ones in a given scenario, both get rank 9.5 at that scenario). Next, the mean rank across all scenarios is computed for each estimator. The null hypothesis of similar performance of the compared methods, under which the corresponding mean ranks should tend to be equal, can then be tested using the Iman-Davenport test statistic and the associated p-value.

Table 3
Aggregated NCE performance on the synthetic Gaussian dataset test sample for the three CRB methods (SSE, MCI, and MCI2) and the baseline ICVIs, expressed as percent accuracy %Acc over the 4860 test sequences. Best result is **bolded**.

CH	DB	SC	TS	BIC	CV	VLR	TCR	NCI	SSE	MCI	MCI2
61.81	36.32	44.63	30.12	72.24	67.63	68.87	26.73	41.44	76.56	**78.58**	76.65

Table 4
Analysis of ranks and %Acc performances on the 54 test scenarios spanning different complexity factors conditions (see Table SM1 of the Supplementary Materials file for the detailed %Acc results by scenario). The associated non-parametric Friedman test is highly significant (F-distributed Iman-Davenport statistic is 78.50 with 11 and 583 d.f., $p < 1.11\text{E-}16$). Best results per row are **bolded**.

	CH	DB	SC	TS	BIC	CV	VLR	TCR	NCI	SSE	MCI	MCI2
Mean Rank	6.85	3.79	5.26	2.83	8.76	7.29	7.95	2.33	3.91	9.25	**10.4**	9.42
# Top Rank	18	2	5	0	20	6	18	0	0	20	**35**	21
# Top Rank (No Tie)	4	0	0	0	2	1	2	0	0	0	**10**	1
Median %Acc.	78.3	27.2	36.7	21.7	78.3	72.8	80	18.9	36.1	85.6	**90**	85
%Acc. IQR	74.1	50.8	57.5	48.6	49.2	47.8	54.2	45.9	42.2	**40**	41.7	**40**

using the NCE methods as treatments, and the scenario-level %Acc of each method (across the corresponding 90 sequences) as responses.

As Table 4 shows, the result of the Friedman test is highly significant ($p \approx 0$) and allows concluding that there are performance differences between the 12 compared methods. Indeed, the three CRB methods obtain the highest mean ranks, and particularly MCI achieves the top performance in 35 of the 54 scenarios, with no ties in 10 of them, and attains a %Acc greater or equal than 90 % in at least half of the scenarios. For comparison, BIC, the best performing baseline ICVI in Table 3, achieves the top position in 20 scenarios, leading outright in only 2 of them and obtaining a median %Acc of 78 %. Interestingly, CH leads with no ties in 4 scenarios and obtains the same median as BIC, despite its mean rank being quite lower than that of BIC. This suggests that quite abrupt differences in the performance of CH occur along the different complexity conditions, which is corroborated by its %Acc IQR, by far the greatest among all methods. On the other extreme, SSE and MCI2 exhibit the lowest %Acc ICR, pointing to a robust performance under different complexity conditions. Moreover, as above, methods considering both cohesion and separation components (SC, CH, DB, TS, TCR) obtain lower mean ranks than those only including a cohesion component (CRB, BIC, VLR, CV).

Post-hoc tests based on the above mean ranks are then conducted to check for significant pairwise differences in relation to MCI's performance, using the Holm's adjustment of p-values to control the experimentwise Type I error probability. Table 5 shows the result of these tests, allowing us to conclude that, at least in the context of Gaussian data, MCI achieves a significantly better NCE performance than all baseline ICVIs at significance level $\alpha = 0.01$, except for BIC, for which significance is achived with $\alpha = 0.1$. Differences between MCI and the other 2 CRB methods (SSE and MCI2) are not significant.

Table 5
Statistics Z, p-values and Holm's adjusted p-values for the pairwise comparisons of MCI versus the rest of considered NCE estimators, using average ranks computed by %Acc (see Table 4). Both p-values and adjusted p-values are superscripted when significant at the significance levels $0.1^{*}, 0.05^{**}$ and 0.01^{***}.

Comparison	Z	p-value	Adj. p-value
MCI vs TCR	11.57	0^{***}	0^{***}
MCI vs TS	10.85	0^{***}	0^{***}
MCI vs DB	9.47	0^{***}	0^{***}
MCI vs NCI	9.30	0^{***}	0^{***}
MCI vs SC	7.35	$1.95\text{E}{-13}^{***}$	$1.36\text{E}{-12}^{***}$
MCI vs CH	5.06	$4.25\text{E}{-07}^{***}$	$4.70\text{E}{-05}^{***}$
MCI vs CV	4.43	$9.41\text{E}{-06}^{***}$	$4.71\text{E}{-05}^{***}$
MCI vs VLR	3.46	$.00052^{***}$	$.00209^{***}$
MCI vs BIC	2.31	$.02097^{**}$	$.06291^{*}$
MCI vs SSE	1.60	0.10931	0.21862
MCI vs MCI2	1.36	0.17349	0.21863

As just mentioned, some differences seem to exist in the robustness of the compared methods performance under variations of the test datasets complexity conditions. Fig. 3 allows delving into this matter with more detail, taking advantage of the factorial design of the test data. Several observations arise from these factor-level results:

- Sequence lengths $\bar{k}$ seem to have little or no influence on the methods performance, except for NCI, TCR and CV. This last suffers a clear decline as $\bar{k}$ grows, further exposing the lack of robustness of the R^1 ratio mentioned in Fig. 1. Instead, CRB methods based on tail ratios present a stable behavior.
- The varying validity of centroids as cluster representatives, which depends on the choice of clustering algorithm, affects the methods performance in quite different ways. On one hand, all methods except CH obtain their worst result with k-Medoids, although the drop observed at this algorithm for CRB methods, CV and NCI is more pronounced. On the other hand, all methods obtain their best result with k-Means, except CH, and NCI and CV, which respectively obtain it with k-Medoids and Agglomerative Clustering.
- The data dimensionality D strongly impacts all methods' performance, very unevenly in this case: CRB methods, BIC and CV slightly improve when going from $D = 2$ to $D \in [3, 9]$, and then worsen in higher dimensional datasets; however, TS, TCR, DB and SC instead show a consistent improvement as D grows, while CH, NCI and VLR show the opposite trend, specially CH, which goes from being the best estimator for 2D data to being the second-worst one (after NCI) when $D \in [10, 50]$.
- The effect of the number of clusters K is less pronounced, although the performance of some methods such as TS, DB and SC clearly deteriorates as K grows. CRB methods show a slight decline instead, and a distinct lead for low K. Remarkably, the performance of NCI consistently improves with K.
- An increase in dataset sizes N tends to positively affect all methods, particularly CRB ones, with MCI, CV and TS being the most benefited estimators.
- Clusters overlap appears as the single most influential complexity factor: all methods performance worsens sharply as Std grows, although that of SSE, MCI2, TS and to some extent also DB seems to deteriorate at a slower rate. Notably, CRB methods, specially SSE and MCI2, stand out for their leading ability to estimate K in the presence of highly overlapped clusters.

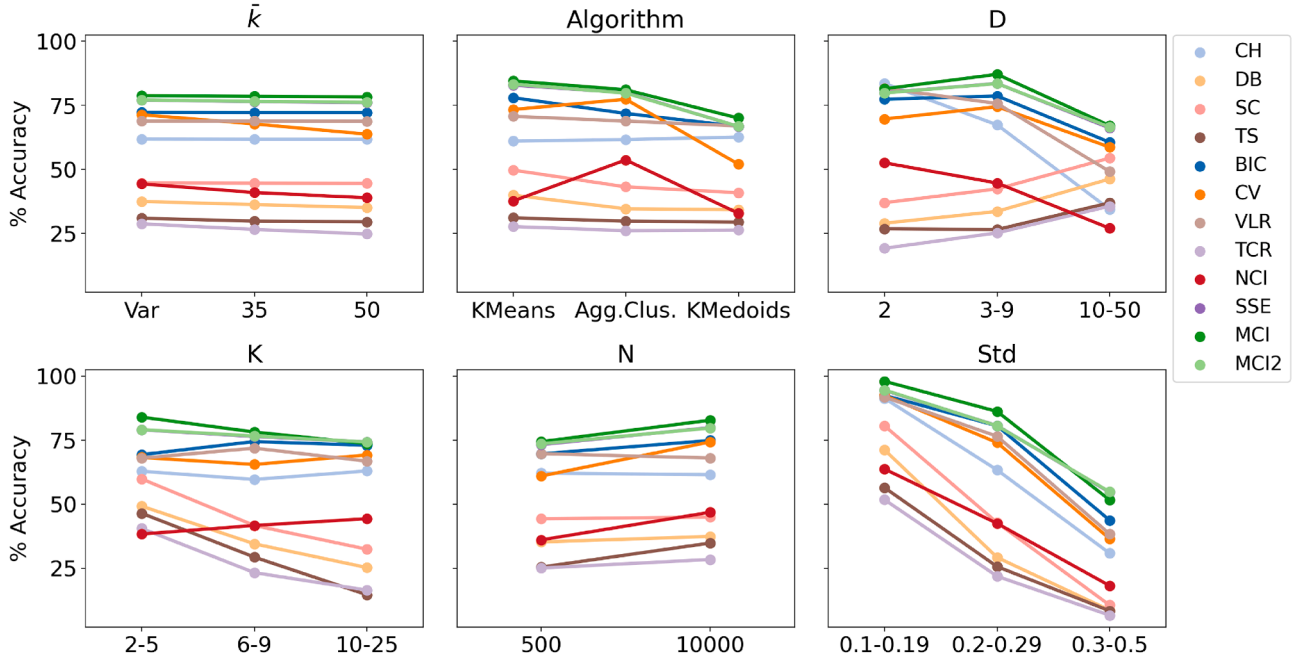

Fig. 3. %Acc performances of the NCE methods by level of the complexity factors data dimensionality D, number of clusters K, cluster size N, clusters overlap Std, clustering algorithm and sequence length $\bar{k}$. Results across 1620 (4860 / 3 levels) sequences are averaged for each level. SSE results overlap with those of MCI2. Except for $D = 2$, CRB methods (SSE, MCI, MCI2) outperform all baseline ICVIs in all considered conditions. Detailed results can be seen in Tables SM2-SM7 of the Supplementary Materials file.

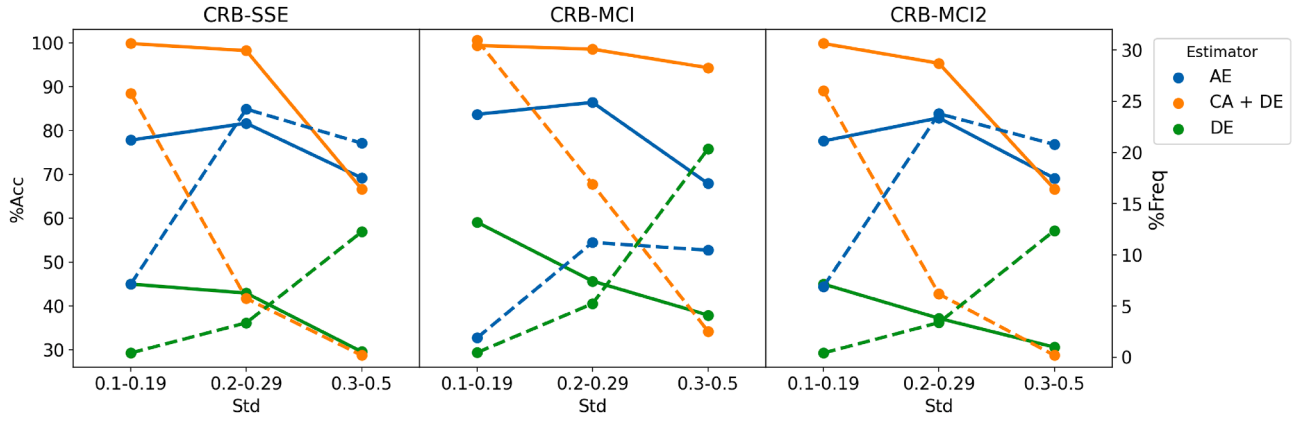

Fig. 4. Test sample NCE performance conditional to activation (solid lines, left axis) and frequency of activation (dashed lines, right axis) of the three estimators CA + DE, AE and DE combined in Algorithm 1 for the CRB methods using either SSE, MCI or MCI2 as base cohesion measure, across varying cluster overlap levels Std.

- CRB methods outperform all baseline ICVIs at virtually all levels of all complexity factors.

Furthermore, the performance of CRB appears to be quite robust also across different base cohesion measures F. Although MCI achieves somewhat better results than SSE and MCI2 in this Gaussian test sample, the performance variation between the former and the latter ones is rather small. These differences can nonetheless be attributed to the diverse activation frequencies and accuracy rates attained, for each F, by the three CRB estimators CA + DE, AE and DE, as shown in Fig. 4. The highly accurate CA-DE activates more frequently in the case of MCI, while SSE and MCI2 rely more on the moderately accurate AE, even in the case of highly overlapped clusters. In turn, for high Std MCI compensates the smaller activation rate of the AE resorting to the less accurate DE and keeping an excellent accuracy of CA-DE, which explains its only slightly inferior performance for overlapped clusters in comparison to SSE and MCI2.

5.2. Real-world and synthetic benchmark datasets

The test data used above replicate the characteristics of the data employed to optimize the CRB thresholds δ_1 and δ_2. Now, real-world and clustering benchmark synthetic datasets fully unrelated to those training conditions are considered as test data, while we still apply the CRB methods derived from the thresholds in Table 2. The $T = 40$ datasets that compose this new test sample are presented in Table 6. Synthetic datasets in this sample span a wide range of cluster structure complexity conditions: anisotropy, noisy data, imbalanced clusters, varying densities, presence of outliers and shapes. Certain datasets combine several of these conditions at the same time.

Regarding the configuration of this experiment, the same considerations as in the previous section apply, with three exceptions: First, now only the k-Means algorithm is used to partition the data, since the effect of the different algorithms has already been assessed, and most ICVIS obtained their best results with k-Means; furthermore, the consideration of a new NCE baseline method, see below, also influences this choice.

Table 6
Real-world and synthetic clustering benchmark datasets, with their number of clusters K, dimensionality D and size N. Superscripts in the *Dataset* column indicates the dataset origin: 1 = [39]; 2 = [40]; 3 = [41]. Column *Type* refers to the typology of the datasets in terms of their clusters' structure complexity condition: A = anisotropy; D = varying density; I = size imbalance; N = noise; O = outliers; R = real-world; S = shapes.

Dataset	K	D	N	Type	Dataset	K	D	N	Type
aniso[1]	3	2	500	A	**aml28**[3]	5	2	804	O, I
varied[1]	3	2	500	D	**balance**[2]	3	4	625	R
cure-t0-2000n-2D[3]	3	2	2000	D, I	**breast**[2]	2	9	277	R
dpb[3]	6	2	4000	D, I	**bupa**[2]	2	6	345	R
dpc[3]	6	2	1000	D, I	**cleveland**[2]	5	13	297	R
2d-10c[3]	9	2	2990	D, I, A	**digits**[1]	10	64	1797	R
2d-4c[3]	4	2	1261	D, I, A	**ecoli**[2]	8	7	336	R
2d-4c-no4[3]	4	2	863	D, I, A	**glass**[2]	7	9	214	R
2d-4c-no9[3]	4	2	876	D, I, A	**ionosphere**[2]	2	33	351	R
sizes1[3]	4	2	1000	I	**iris**[2]	3	4	150	R
sizes2[3]	4	2	1000	I	**led7digit**[2]	10	7	500	R
sizes3[3]	4	2	1000	I	**pima**[2]	2	8	768	R
sizes4[3]	4	2	1000	I	**sonar**[2]	2	60	208	R
sizes5[3]	4	2	1000	I	**vowel**[2]	11	13	990	R
2d-3c-no123[3]	3	2	715	I, A	**wdbc**[2]	2	30	569	R
zelnik2[3]	2	2	303	N	**wine**[2]	3	13	178	R
zelnik4[3]	4	2	622	N	**yeast**[2]	10	8	1484	R
cure-t2-4k[3]	5	2	4200	N, D	**olivetti-faces**[1]	40	4096	400	R
cluto-t8.8k[3]	8	2	8000	N, D, I, S	**circles**[1]	2	2	500	S
2d-20c-no0[3]	20	2	1517	O, D, I, A	**moons**[1]	2	2	500	S

Table 7
NCE performance by clustering structure complexity condition, measured by %Acc across those datasets in Table 6 featuring each condition (the number of averaged results is shown between parentheses next to each condition label). Global %Acc and RMSE across all datasets is reported at the bottom part of the table. *Time* refers to average run time expressed in seconds per dataset, including the $\bar{k} = 50$ executions of k-Means. Best results per row are **bolded**.

Condition	reval	CH	DB	SC	TS	BIC	CV	VLR	TCR	NCI	SSE	MCI	MCI2
Anisotropy (6)	50	0	**66.7**	**66.7**	16.7	0	16.7	0	33.3	0	50	33.3	50
Density (11)	36.4	0	45.5	36.4	9.1	0	0	0	18.2	0	**54.5**	27.3	**54.5**
Imbalance (16)	31.3	12.5	**62.5**	50	12.5	37.5	6.3	12.5	37.5	0	50	25	50
Noise (4)	0	0	25	0	0	0	0	0	0	0	25	**50**	**50**
Outliers (2)	0	0	**50**	0	0	**50**	0	0	0	0	0	0	0
Shapes (3)	**33.3**	0	0	0	0	0	0	0	0	**33.3**	**33.3**	**33.3**	**33.3**
Real-World (17)	29.4	35.3	11.8	35.3	**41.2**	0	11.8	0	5.9	11.8	**41.2**	29.4	29.4
%Acc	35	20	35	42.5	27.5	15	10	5	22.5	7.5	**47.5**	27.5	40
RMSE	**4.6**	23.4	24.7	12.6	9.1	23.2	24.7	25.1	20.2	26.8	5.7	12.3	4.7
Time	3343	1.16	1.57	4.05	1.5	1.1	**1.07**	1.08	1.1	6.39	**1.07**	1.13	1.15

Second, only sequences of length $\bar{k} = 50$ are formed now, as the effect of this parameter has been shown to be scarcely relevant, at least for CRB methods, and the inclusion of a dataset with $K = 40$ clusters makes it unfeasible to consider $\bar{k} = Var$ or $\bar{k} = 35$ sequence lengths. Third, an stability-based NCE method, reval [28], is added to the previous benchmark of ICVIs. This method could not be used in the previous experiment because its available implementation does not enable the usage of k-Medoids, and also because its time complexity made it unfeasible to recurrently apply it on datasets with $N = 10000$.

Aggregated results of the ICVIs, reval, and CRB methods in this test sample-both by cluster structure condition and globally-are presented in Table 7. Detailed results by dataset are provided in Table SM8. Remarkably, some indices that performed poorly on the previous Gaussian data, such as DB, SC, and TS, now show leading results under diverse conditions including anisotropy, imbalanced clusters, and real-world data, as well as relatively moderate (DB, TS) or high (SC) global %Acc rates. Conversely, cohesion-based ICVIs that performed well on Gaussian data, such as BIC, CV, and VLR, now yield rather poor results. Notably, this trend does not apply to the CRB methods-especially SSE and MCI2-which instead exhibit consistently good performance across almost all conditions, achieving leading results for several condition such as varying densities, noisy clusters, complex shapes and real-world data. In particular, SSE attains the best global %Acc rate, correctly identifying the actual number of clusters for almost half of the datasets in the sample, despite its complexity and broad range of conditions. Nonetheless, Table 7 also highlights potential limitations of the CRB methods, particularly for data with imbalanced clusters or clusters containing outliers. In the latter case, however, the results may be inconclusive due to the small number of averaged instances (just 2). Furthermore, as indicated by the global RMSE values, even when incorrect, CRB methods tend to produce estimates of $\hat{K}$ much closer to the actual K than most methods, which often overestimate it substantially (see also Table SM8). Indeed, only reval achieves a smaller RMSE than SSE and MCI2, although the difference with the latter is negligible, while reval attains a lower global %Acc and requires roughly three orders of magnitude longer execution times. In this regard, the observed runtimes of the CRB methods are relatively lower than, or comparable to, those of most ICVIs, consistent with the discussion of the computational complexity of Algorithm 1 in Section 4.2.

5.3. High-dimensional synthetic gaussian data

In this section, we explore the performance of CRB methods relative to the baseline under high-dimensional (HD) conditions. To assess the HD capabilities of the NCE methods using controlled data suitable for clustering-and consistent with the isotropy assumption of both ICVIs

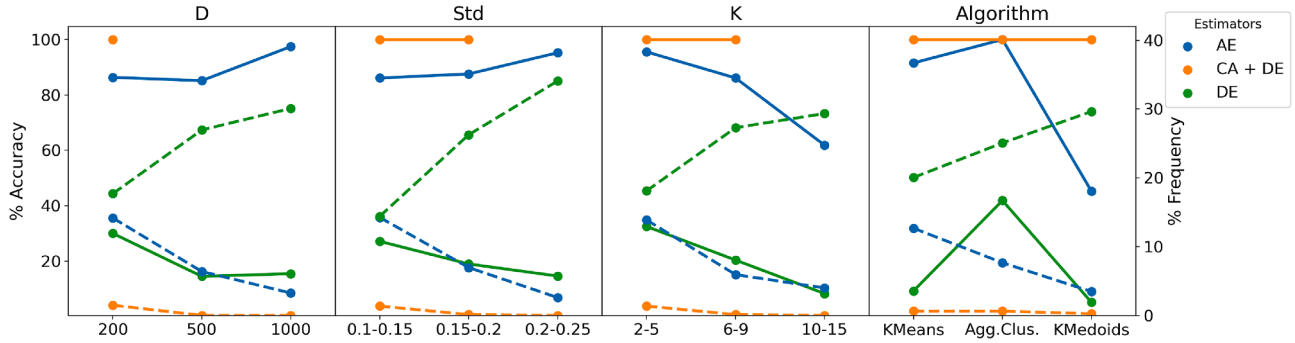

Fig. 5. Accuracy rates (solid lines, left axis) and frequency of activation (dashed lines, right axis) of the CRB-MCI2 estimators on the HD test sample across levels of dimensionality D, cluster overlap Std, number of clusters K and clustering algorithm.

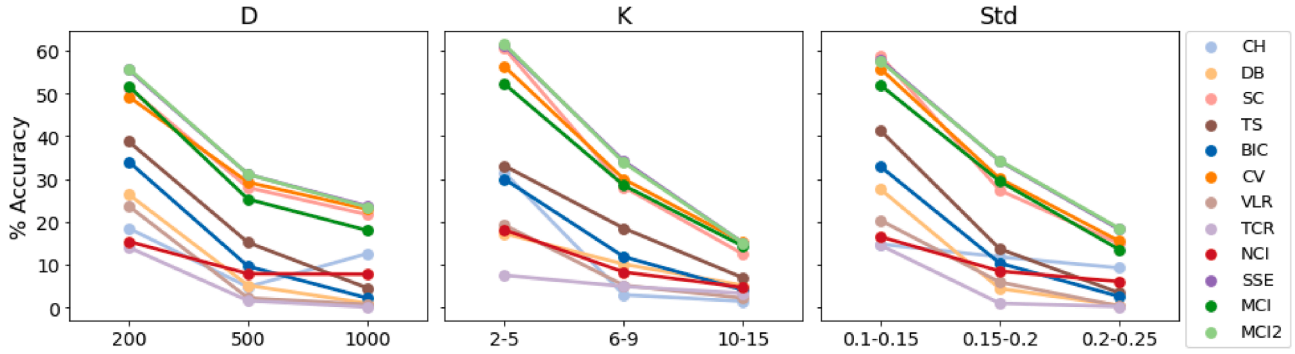

Fig. 6. High-dimensional test %Acc performances of NCE methods by level of the complexity factors data dimensionality D, number of clusters K and clusters overlap Std. Results across 810 (2430 / 3 levels) sequences are averaged for each level. Results of SSE and MCI2 superpose, and outperform all baseline ICVIs in nearly all considered conditions. Detailed results can be seen in Tables SM9-SM13 of the Supplementary Materials file.

and clustering algorithms-we generate a new test sample from synthetic Gaussian blob datasets following the same procedure as in Section 5.1, with two exceptions: 1) The ranges of the complexity factors in Table 1 are modified as follows: the third level of the K factor is now 10-15 instead of 10-25; Std uniformly varies over the ranges 0.1-0.145, 0.15-0.195, and 0.2-0.25; and, most importantly, the datasets' dimensionality D is fixed at $D = 200, 500$, and 1000. 2) Only $N = 500$ datasets are used. As before, 10 datasets are generated for each of the 27 scenarios resulting from all combinations of the three factors, yielding a total of $T = 270$ datasets and 2430 test sequences $(P_k^t)_{k=1}^{\bar{k}}$ after applying k-Means, Agglomerative Clustering, and k-Medoids with $\bar{k} = \text{Var}, 35, 50$. It is worth noting that this test sample constitutes a highly complex NCE benchmark due to the absence of dimensionality reduction or data representation techniques, and because the clustering algorithms used are known to struggle in producing cohesive partitions under HD conditions. Scenarios with high values of D, K, and Std pose particularly challenging conditions.

The baseline of NCE methods for this experiment consists of the same 9 ICVIs as in Section 5.1, since the available implementation of reval does not support the use of the k-Medoids algorithm. Due to their reliance on cohesion and/or separation components, all baseline methods can be expected to find strong difficulties in the exposed test sample because of the uniformization of distances caused by the curse of dimensionality. CRB methods are no exception. As shown in Figs. 2(c3)-(d3), under HD conditions the base cohesion sequences $S(F)$ used by CRB tend to be nearly flat across the partitions produced by the clustering algorithm. Consequently, $TR\Delta^1$ values are uniformly low, making it difficult to trigger the CA + DE and AE and reducing the accuracy of the DE. This situation worsens as dimensionality increases, or as the number of clusters and their overlap level grow, as shown in Fig. 5. It is worth emphasizing that CRB's performance degradation is also related to the impact of dimensionality on clustering algorithms, and variations in that impact influence the severity of this degradation. Indeed, the same figure shows that $TR\Delta^1$ sequences with lower values (and thus lower AE activation frequency) but maxima more aligned with the true number of clusters K (and thus more accurate AE and DE estimates) are obtained with Agglomerative Clustering than with k-Means, while k-Medoids produces even lower $TR\Delta^1$ values and the weakest correlations between its maxima and K.

Table 8 shows that, consistent with these observations and the difficulty of this HD NCE benchmark, all considered methods achieve rather modest performance in absolute terms, with global %Acc rates not exceeding 40 %. Although most methods-except CH, TCR, and NCI-show competitive results in the least complex condition ($D = 200, K \leq 5, Std < 0.15$), performance tends to drop sharply as dimensionality, number of clusters, or overlap increase, due to the compounding effects of the curse of dimensionality and inter-cluster blending. Nonetheless, while the performance of CRB methods also tends to decline, SSE and MCI2 stand out for their consistency and resilience across increasing complexity, whereas all other methods-except perhaps SC-degrade more rapidly. Indeed, as shown in Fig. 6, both SSE and MCI2 consistently rank among the top performers across all scenarios, particularly for medium ranges of K and moderate to severe overlap, demonstrating robustness compared to most baseline methods.

5.4. Specialized CRB procedures

The experiment described in this section aims to explore and illustrate the possibility of devising specialized CRB procedures, that is, specifically fitted to operate under particular conditions rather than under general ones. As mentioned in Section 4.3.2, the scope of CRB is determined by the characteristics of the training and validation samples used to optimize the thresholds δ_1 and δ_2. Therefore, specialized CRB procedures are to be derived by carrying out such optimization on data with specific conditions. These more specific conditions may allow the optimization process to find thresholds that provide a better

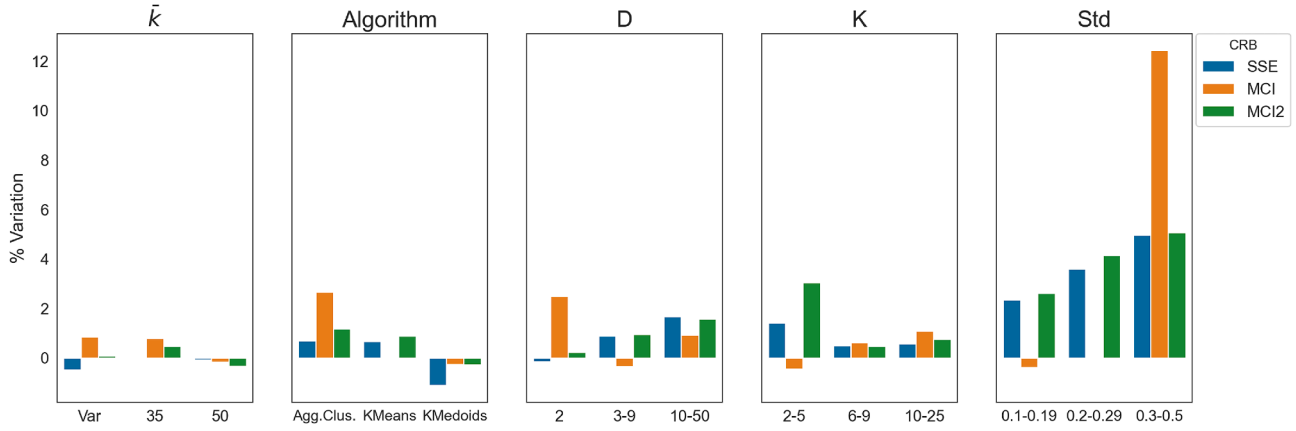

Fig. 7. Test performance improvement rates enabled by specialized CRB procedures fitted only considering specific sequence lengths $\bar{k}$, clustering algorithms and levels of the complexity factors D, K and Std. The reported %Variation is obtained as the relative difference between the %Acc of specialized (Sp) and reference (Ref) CRB procedures, i.e. %Variation $= 100 \cdot (\%\text{Acc}_{Sp} - \%\text{Acc}_{Ref})/\%\text{Acc}_{Ref}$. Most rates are positive, thus indicating the potential of specialized procedures to enhance the performance of reference procedures under specific conditions. Detailed results can be found in Tables SM14-SM18 at the Supplementary Materials file.

Table 8
Detailed %Acc performance of the NCE methods across the 27 complexity scenarios of the high-dimensional synthetic Gaussian test sample. Each scenario is described by its unique combination of levels or ranges of the complexity factors number of clusters K, data dimensionality D and clusters overlap Std. At the top part of the table, each %Acc value averages performance across 90 sequences. Global %Acc at the bottom part presents the overall accuracy across the whole test sample (2430 sequences). Best results per row are **bolded**.

D	K	Std	CH	DB	SC	TS	BIC	CV	VLR	TCR	NCI	SSE	MCI	MCI2
200	2–5	0.1-.15	46.7	84.4	**96.7**	93.3	93.3	85.6	90	46.7	32.2	90	91.1	90
200	2–5	0.15-.2	50	25.6	76.7	61.1	70	71.1	53.3	7.8	25.6	82.2	**83.3**	**83.3**
200	2–5	0.2-.25	36.7	4.4	62.2	18.9	23.3	53.3	3.3	0	18.9	65.6	55.6	**66.7**
200	6–9	0.1-.15	20	68.9	**83.3**	81.1	73.3	78.9	46.7	41.1	28.9	80	76.7	80
200	6–9	0.15-.2	0	12.2	56.7	34.4	10	57.8	0	1.1	13.3	**67.8**	58.9	**67.8**
200	6–9	0.2-.25	0	0	20	10	0	20	0	0	6.7	**38.9**	23.33	**38.9**
200	10–15	0.1-.15	13.3	42.2	60	46.7	36.7	53.3	20	30	13.3	57.8	**62.2**	57.8
200	10–15	0.15-.2	0	0	6.7	3.3	0	**13.3**	0	0	0	**13.3**	**13.3**	**13.3**
200	10–15	0.2-.25	0	0	0	0	0	**10**	0	0	0	4.4	1.1	4.4
500	2–5	0.1-.15	10	30	**80**	65.6	50	70	20	12.2	20	73.3	64.4	73.3
500	2–5	0.15-.2	13.33	2.22	46.7	18.9	13.3	53.3	0	0	10	**63.3**	45.6	**63.3**
500	2–5	0.2-.25	16.67	0	23.3	2.2	0	23.3	0	1.1	8.9	**24.4**	21.1	**24.4**
500	6–9	0.1-.15	3.33	10	53.3	34.4	23.3	53.3	0	1.1	6.7	**54.4**	46.7	**54.4**
500	6–9	0.15-.2	0	0	13.3	3.3	0	16.7	0	0	5.6	**24.4**	14.4	**24.4**
500	6–9	0.2-.25	0	0	0	0	0	**6.7**	0	0	3.3	0	0	0
500	10–15	0.1-.15	0	4.4	35.6	12.2	0	**40**	0	0	10	36.7	30	36.7
500	10–15	0.15-.2	0	0	0	0	0	0	0	0	3.3	0	**5.6**	0
500	10–15	0.2-.25	0	0	0	0	0	0	0	0	**3.3**	**3.3**	0	**3.3**
1000	2–5	0.1-.15	36.7	8.9	**83.3**	35.6	20	72.2	6.7	0	25.6	74.4	51.1	74.4
1000	2–5	0.15-.2	43.3	0	46.7	2.2	0	**51.1**	0	0	13.3	48.9	37.8	48.9
1000	2–5	0.2-.25	**30**	0	28.9	0	0	26.7	0	0	8.9	27.8	20	28.9
1000	6–9	0.1-.15	3.3	0	26.7	3.3	0	28.9	0	0	6.7	**34.4**	28.9	31.1
1000	6–9	0.15-.2	0	0	0	0	0	**7.8**	0	0	2.2	**7.8**	6.7	**7.8**
1000	6–9	0.2-.25	0	0	0	0	0	0	0	**1.1**	**1.1**	0	**1.1**	0
1000	10–15	0.1-.15	0	0	10	0	0	**20**	0	0	5.6	**20**	16.7	**20**
1000	10–15	0.15-.2	0	0	0	0	0	0	0	0	**3.3**	0	0	0
1000	10–15	0.2-.25	0	0	0	0	0	0	0	0	**3.3**	0	0	0
Global %Acc			12	10.9	33.7	19.5	15.3	33.8	8.9	5.3	10.4	**36.8**	31.7	**36.8**

fitness, once these are freed from the need of establishing a trade-off between accurately estimating K in sequences with a certain condition, say a high Std, and in sequences with a different condition, say a low Std. However, it may also be the case that the balance found in general conditions cannot be improved under those more specific conditions, or that such an improvement leads to overfit.

The experiment proceeds by repeatedly applying the optimization process exposed in Section 4.3.1, each time using the training and validation samples described in Section 4.3.2 filtered by specific levels of factors K, D, Std, clustering algorithm, and $\bar{k}$. For example, optimization is run first on the 810 sequences with $K \in [2, 5]$, then with $K \in [6, 9]$, $K \in [10, 25]$, $D = 2$, etc. Each run yields a pair of specialized thresholds δ_1, δ_2, and the corresponding specialized CRB procedure is then applied on the test sample used in Section 5.1, similarly restricted to the 1620 sequences (now with both $N = 500$ and $N = 10000$ datasets) meeting the same filter. In all other matters this experiment follows the setup in Section 5.1. Finally, the resulting %Acc is computed and compared to that obtained on the same sample using the general CRB thresholds from Table 2 (i.e., %Acc values shown in Fig. 3 and Tables SM2-SM7).

The results of this experiment are presented in Fig. 7. Notice that most of the reported relative %Acc variations between the specialized and reference CRB procedures are positive. This indicates that specialized CRB procedures generally perform better than reference CRB ones under the specific conditions to which the former were fitted. However, this improvement is uneven across the different factors and levels.

Indeed, in some cases, the variation rates are quite small or even negative, either signaling conditions already prioritized in the balance of the general optimization, so that little or no room for improvement could be found, or for which the improvement found leads to overfit. For instance, sequence lengths $\bar{k}$ were assessed in Section 5.1 as having little impact on CRB methods' performance, and thus it is plausible that specialization on certain $\bar{k}$ can only render a small improvement, if any. On the other extreme, clusters overlap, determined by factor Std, was assessed as highly influential, and thus specialization may provide relevant improvements. These examples allow explaining the behavior of variation rates in the respective cases of Fig. 7. In the case of Std, particularly, the differences in the rates behavior between SSE and MCI2, on the one hand, and MCI, on the other hand, can be attributed to the different trade-offs reached when fitting the respective base procedures: a balanced one in the case of SSE and MCI2, and one prioritizing sequences with $Std \in [0.1, 0.29]$ in the case of MCI.

6. Conclusions

Several conclusions can be drawn from this work. Firstly, the notion of covering and its quantification through covering indexes provide an operative, positively defined, and scale-invariant alternative to SSE for measuring cluster cohesion. Moreover, tail ratios offer more robust and accurate indicators of the correct number of clusters than standard difference ratios. In turn, the CRB procedure, which layers three different estimators within a rule-based framework, enables the effective exploitation of tail ratios, further enhancing their benefits. Importantly, CRB thresholds optimized on synthetic data spanning a wide spectrum of dataset and cluster conditions yield robust and competitive NCE performance that generalizes beyond the training conditions to substantially different settings and across various cohesion measures. Indeed, while the results presented in Section 5.1 show that CRB methods outperform, with statistical significance, all baseline ICVIs across the range of conditions used in their training set, the results in Section 5.2 demonstrate that CRB methods can successfully generalize to markedly different scenarios, achieving competitive and consistent performance on both real-world and synthetic data across a wide range of cluster structure complexities not included in their training set. To some extent, this also extends to HD data, where CRB performance deteriorates but still surpasses that of most baseline ICVIs, as shown in Section 5.3. Beyond this expected decline under HD conditions, the experimental results also reveal that CRB methods may show limitations or reduced accuracy in certain clustering contexts, such as anisotropic data, imbalanced clusters, and datasets with outliers. Finally, possibly the most relevant and promising contribution of the CRB framework is its ability to produce specialized NCE procedures tailored to specific clustering scenarios, e.g. highly overlapped clusters, as evidenced by the results in Section 5.4.

The CRB-NCE approach can be further developed in several ways. Future work to this aim shall address its extension to more general clustering contexts (e.g. non-Gaussian data, non-prototype-based algorithms), the consideration of more general cohesion measures (such as coverage indices based on MEOWA operators [42]) and optimization mechanisms (e.g. neural networks, particularly rule-oriented ones such as ANFIS [43]), the development of additional rules or estimators to be incorporated into Algorithm 1, the exploration and exploitation of specialized CRB procedures, and even also the adaptation of the CRB approach to other clustering tasks, such as clusterability [6] or cluster quality analysis [17–19].

Data availability

All the code and synthetic data generation can be replicated through the paper's GitHub, linked at the paper itself. Other data come from repositories, equally ackowledged in the paper.

CRediT authorship contribution statement

J. Tinguaro Rodríguez: Writing - original draft, Visualization, Validation, Supervision, Software, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization; **Xabier Gonzalez-Garcia:** Writing - review & editing, Writing - original draft, Visualization, Validation, Software, Investigation; **Daniel Gómez:** Writing - review & editing, Supervision, Project administration, Investigation, Funding acquisition, Formal analysis, Conceptualization; **Humberto Bustince:** Writing - review & editing, Resources, Project administration, Methodology, Conceptualization.

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.

Acknowledgements

This work has been supported by projects PID2021-122905NB-C21, PID2022-136627NB-I00 and PID2024-155289NB-I00 of the Government of Spain.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at 10.1016/j.patcog.2025.112909.

References

[1] J. Wu, W. Xue, G.A. Voth, K-Means clustering coarse-graining (KMC-CG): a next generation methodology for determining optimal coarse-grained mappings of large biomolecules, J. Chem. Theory Comput. 19 (23) (2023) 8987–8997.
[2] R.N. Mcarthur, A.N. Zehmakan, M.A. Charleston, Y. Lin, G. Huttley, Spectral cluster supertree: fast and statistically robust merging of rooted phylogenetic trees, Front. Mol. Biosci. 11 (2024) 1432495.
[3] V. Mandelli, I. Landi, E.M. Busuoli, E. Courchesne, K. Pierce, M.V. Lombardo, Prognostic early snapshot stratification of autism based on adaptive functioning, Nat. Ment. Health. 1 (5) (2023) 327–336.
[4] C. Singh, S.K. Ranade, D. Kaur, A. Bala, A kernelized-bias-corrected fuzzy C-means approach with moment domain filtering for segmenting brain magnetic resonance images, Soft. Comput. 28 (3) (2024) 1909–1933.
[5] C. Hennig, What are the true clusters?, Patt. Recogn. Lett. 64 (2015) 53–62.
[6] A. Adolfsson, M. Ackerman, N.C. Brownstein, To cluster, or not to cluster: an analysis of clusterability methods, Patt. Recogn. 88 (2019) 13–26.
[7] N.R. Pal, J. Biswas, Cluster validation using graph theoretic concepts, Patt. Recogn. 30 (1997) 847–857.
[8] E. Schubert, Stop using the elbow criterion for k-means and how to choose the number of clusters instead, ACM SIGKDD Explor. Newslett. 25 (1) (2023) 26–42.
[9] A. Rykov, R.C.D. Amorim, V. Makarenkov, B. Mirkin, Inertia-based indices to determine the number of clusters in K-Means: an experimental evaluation, IEEE Access 12 (2024) 11761–11773.
[10] N. Wiroonsri, Clustering performance analysis using a new correlation-based cluster validity index, Patt. Recogn. 145 (2004) 109910.
[11] L. Guo, J. Zhan, Z. Xu, J.C.R. Alcantud, A consensus measure-based three-way clustering method for fuzzy large group decision making, Inf. Sci. 632 (2023) 144–163.
[12] A. Georgakis, D. Gatziolis, G. Stamatellos, A primer on clustering of forest management units for reliable design-based direct estimates and model-based small area estimation, Forests 14 (10) (1994) 2023.
[13] C.X. Gao, D. Dwyer, Y. Zhu, C.L. Smith, L. Du, K.M. Filia, …S. M. Cotton, An overview of clustering methods with guidelines for application in mental health research, Psychiatry Res. 327 (2023) 115265.
[14] M.A. Mahdi, K.M. Hosny, I. Elhenawy, Scalable clustering algorithms for big data: a review, IEEE Access 9 (2021) 80015–80027.
[15] R. Mussabayev, N. Mladenovic, B. Jarboui, R. Mussabayev, How to use K-means for big data clustering?, Patt. Recogn. 137 (2023) 109269.
[16] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J.M. Pérez, I. Perona, An extensive comparative study of cluster validity indices, Patt. Recogn. 46 (1) (2013) 243–256.
[17] I. Gurrutxaga, J. Muguerza, O. Arbelaitz, J.M. Pérez, J.I. Martín, Towards a standard methodology to evaluate internal cluster validity indices, Patt. Recogn. Lett. 32 (3) (2011) 505–515.
[18] L. Vendramin, R.J. Campello, E.R. Hruschka, Relative clustering validity criteria: a comparative overview, Stat. Anal. Data Min. 3 (4) (2010) 209–235.
[19] Z. Botta-Dukát, A new approach for evaluating internal cluster validation indices, arxiv:2308.03894, 2023.

[20] S. Xu, X. Qiao, L. Zhu, Y. Zhang, C. Xue, Reviews on determining the number of clusters, Appl. Math. Inf. Sci. 10 (4) (2016) 1493–1512.
[21] T. Liu, H. Yu, R.H. Blair, Stability estimation for unsupervised clustering: a review, Wiley Interdiscip. Rev. Comput. Stat. 14 (6) (2022) 1575.
[22] M. Popescu, J.C. Bezdek, T.C. Havens, J.M. Keller, A cluster validity framework based on induced partition dissimilarity, IEEE Trans. Cybern. 43 (1) (2012) 308–320.
[23] D. Kumar, J.C. Bezdek, Visual approaches for exploratory data analysis: a survey of the visual assessment of clustering tendency (VAT) family of algorithms, IEEE Syst. Man Cybern. Mag. 6 (2) (2020) 10–48.
[24] Y. Zhang, J. Mandziuk, H.C. Quek, W. Goh, Curvature-based method for determining the number of clusters, Inf. Sci. 415 (2017) 414–428.
[25] R.R. Yager, D.P. Filev, Approximate clustering via the mountain method, IEEE Trans. Syst. Man Cybern 24 (8) (1994) 1279–1284.
[26] S. Chiu, Method and software for extracting fuzzy classification rules by subtractive clustering, in: Proc. North American Fuzzy Inf. Proc, 1996, pp. 461–465.
[27] A.D. Amo, J. Montero, G. Biging, V. Cutello, Fuzzy classification systems, Eur. J. Oper. Res. 156 (2) (2004) 495–507.
[28] I. Landi, V. Mandelli, M.V. Lombardo, reval: a python package to determine best clustering solutions with stability-based relative clustering validation, Patterns 2 (2021) 100228.
[29] T. Caliński, J. Harabasz, A dendrite method for cluster analysis, Commun. Stat.-Theory Methods 3 (1974) 1–27.
[30] D.L. Davies, D.W. Bouldin, A cluster separation measure, IEEE Trans. Pattern Anal. Mach. Intell 1 (2) (1979) 224–227.
[31] P.J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, Comput. Appl. Math. 20 (1987) 53–65.
[32] Y. Tang, F. Sun, Z. Sun, Improved validation index for fuzzy clustering, in: Proc. of the 2005 American Control Conference, 2005, pp. 1120–1125.
[33] X. Xie, G. Beni, A validity measure for fuzzy clustering, IEEE Trans. Pattern Anal. Mach. Intell. 13 (8) (1991) 841–847.
[34] Y. Tang, J. Huang, W. Pedrycz, B. Li, F. Ren, A fuzzy clustering validity index induced by triple center relation, IEEE Trans. Cybern. 53 (8) (2023) 5024–5036.
[35] D. Pelleg, A.W. Moore, X-Means: extending k-means with efficient estimation of the number of clusters, in: Int. Conf. Machine Learning, 2000, pp. 727–734.
[36] A. Foglia, B. Hancock, Notes on bayesian information criterion calculation for x-means clustering, 2012.
[37] Y. Ren, J. Pu, Z. Yang, J. Xu, G. Li, X. Pu, …L. He, Deep clustering: a comprehensive survey, IEEE Trans. Neural Networks Learn. Syst. 36 (4) (2024) 5858–5878.
[38] S. García, A. Fernández, J. Luengo, F. Herrera, Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: experimental analysis of power, Inf. Sci. 180 (10) (2010) 2044–2064.
[39] F. Pedregosa, et al., Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830.
[40] J. Derrac, S. Garcia, L. Sanchez, F. Herrera, Keel data-mining software tool: data set repository, integration of algorithms and experimental analysis framework, J. Mult. Valued Logic Soft. Comput. 17 (2015) 255–287.
[41] T. Barton, Clustering Benchmarks, https://github.com/deric/clusteringbenchmark.
[42] R.R. Yager, Families of OWA operators, Fuzzy Sets. Syst. 55 (1993) 255–271.
[43] J.S. Jang, ANFIS: Adaptive-network-based fuzzy inference system, IEEE Trans. Syst. Man Cybern. 23 (3) (1993) 665–685.