IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 16 October 2023, accepted 19 December 2023, date of publication 8 January 2024, date of current version 25 January 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3350791

RESEARCH ARTICLE

Inertia-Based Indices to Determine the Number of Clusters in K-Means: An Experimental Evaluation

ANDREI RYKOV[1], RENATO CORDEIRO DE AMORIM[2], VLADIMIR MAKARENKOV[3,4], AND BORIS MIRKIN[1,5]

[1]Department of Data Analysis and Machine Intelligence, National Research University Higher School of Economics, 101000 Moscow, Russia
[2]Computer Science and Electrical Engineering Department, University of Essex, CO4 3SQ Wivenhoe, U.K.
[3]Département d'informatique, Université du Québec à Montréal, Montreal, QC H3C 3P8, Canada
[4]Mila—Quebec AI Institute, Montreal, QC H2S 3H1, Canada
[5]Department of Computer Science and Information Systems, Birkbeck, University of London, WC1E 7HX London, U.K.

Corresponding author: Renato Cordeiro De Amorim (r.amorim@essex.ac.uk)

ABSTRACT This paper gives an experimentally supported review and comparison of several indices based on the conventional K-means inertia criterion for determining the number of clusters, K, in datasets, using the popular Silhouette width index as a benchmark. Our experiments involve a novel version of the Elbow index, defined using values of K two or three steps apart. We also discuss alternative ways of computing the inertia and summarizing its values. Even though there are no overall winners in our experiments, some of our results are very conclusive and can be used as a guide for indices determining the number of clusters in K-means.

INDEX TERMS K-means, number of clusters, inertia, elbow method, Calinski-Harabasz index, Hartigan rule.

I. INTRODUCTION

This paper computationally explores a popular approach for choosing the right number of clusters, K, in K-means clustering. We computationally review the use of cluster validation indices based on the inertia, i.e. a square-error criterion of the conventional K-means method in Equation (1). Also, we bring forth a set of novel uses of these indices. They are as follows: (a) use of the Euclidean distance rather than the squared Euclidean distance in criterion (1) of K-means; (b) summarization of a set of inertia values resulting from multiple runs of K-means using the mean, rather than the minimum (or maximum) value. Another novelty is an explicit use of a set of three Elbow indices defined by the step size.

The interested reader is referred to numerous reviews on the problem of determining the right number of clusters according to the structure of the dataset under investigation (see, for example, [1], [2], [3], [4]). No comprehensive solution to the problem has been found so far. The stream of work on the subject does not run dry though; just the opposite: see, for example, some recent papers such as [5], [6], [7], [8], [9], [10], and [11], which may be considered as a support for this opinion.

Among the best performers, one frequently encounters cluster validity indices based on the inertia. Especially frequent are mentions of the Calinski-Harabasz index, the Elbow index (including a very successful Curvature index by Zhang et al. [12], see further on for more details), and the Hartigan rule. However, to the best of our knowledge, no research specifically focused on comparison of inertia-based indices has been conducted so far.

Our major interest is testing the data consisting of intermix among clusters at which the use of a particular index may be more advantageous. Therefore, we consider three types of datasets at which cluster intermix can be analyzed. Two of them involve an explicit and controllable intermix parameter: the first is a generator of "synthetic" cluster structures at which the intermix is represented by a so-called squeezing parameter, and the second is based on the within-cluster dispersion. The third data type, a set of relevant real-world datasets from the UCI repository, has no explicit intermix parameters; moreover, the extent of association between the features and the ground-truth partition is not very clear and may be too weak to get discovered within our approach.

The associate editor coordinating the review of this manuscript and approving it for publication was Md. Abdur Razzaque.

II. K-MEANS CLUSTERING

A. K-MEANS CRITERION AND RELATED QUANTITIES

K-means clustering is the most popular method in multivariate data sciences, especially in machine learning, data mining, and quantitative psychology. Given an entity-to-feature data matrix $Y = (y_{iv})$ $(i \in I, v = 1, \ldots, V)$, a run of the algorithm produces a partition S of I, in K nonintersecting groups (clusters) S_k, $S = \{S_k\}$, with V-dimensional centers $c_k = (c_{kv})$ with $k = 1, 2, \ldots, K$. Given an initial set of centers $c = \{c_k\}$, the K-means algorithm, in a batch version, works in iterations to alternatingly minimize the square error criterion, also referred to as inertia, $D(K)$, defined as follows:

$$D(K) = \sum_{k=1}^{K} \sum_{i \in S_k} d(y_i, c_k)^2, \tag{1}$$

where $S = \{S_k\}$ is the K-cluster partition, c_k is the center of cluster S_k, y_i is the i-th row of matrix Y, and $d(y_i, c_k)^2 = \sum_{v=1}^{V}(y_{iv} - c_{kv})^2$ is the squared Euclidean distance.

Due to its quadratic format, the inertia can be considered within a Pythagorean decomposition of the data scatter $T = \sum_{i,v} y_{iv}^2$:

$$T = D(K) + F(K), \tag{2}$$

where $F(K) = \sum_{k=1}^{K} N_k \langle c_k, c_k \rangle$ and N_k is the cardinality of S_k.

Let us develop an explicit expression used in criterion (1):

$$\begin{aligned} D(K) &= \sum_{k=1}^{K} \sum_{i \in S_k} \sum_{v=1}^{V} (y_{iv} - c_{kv})^2 \\ &= \sum_{k=1}^{K} \sum_{i \in S_k} \sum_{v=1}^{V} (y_{iv}^2 - 2y_{iv}c_{kv} + c_{kv}^2) \\ &= \sum_{i=1}^{N} \sum_{v=1}^{V} y_{iv}^2 - \sum_{k=1}^{K} N_k \langle c_k, c_k \rangle = T - F(K), \end{aligned}$$

which proves Equation (2).

In the case of a data matrix Y being centered, so that the grand mean vector $g = (g_v)$ has been subtracted from each row of Y, Equation (2) is well-known in the theory of analysis of variance, ANOVA (see, for example, [13] Eq. 36.2.1). In that case, T is obviously equal to the inertia of the dataset as is, $T = D(1)$. The value $D(K)$ does not depend on the grand mean location. It is usually referred to as the within-group sum of squares and denoted as $SSW = D(K)$, whereas the right-hand part of Equation (2) is referred to as the between-group sum of squares SSB. The SSB obviously depends on the position of the grand mean g. However, this dependence has nothing to do with the partition S. Let us put the subtracted grand mean in SSB explicitly:

$$SSB = \sum_{k=1}^{K} N_k \langle c_k - g, c_k - g \rangle = F(K) - N\langle g, g \rangle. \tag{3}$$

One can see from Equation (3) that $F(K)$ and SSB differ by a constant, $N\langle g, g \rangle$, which is equal to $F(1)$ and does not depend on the partition. Therefore,

$$SSB = F(K) - F(1) = D(1) - D(K). \tag{4}$$

The right part of Equation (4) shows that the between-group of squares is a drop in value of the inertia when moving from $k = 1$ to K clusters.

B. K-MEANS ALGORITHM AND ITS RANDOM SWAP VERSION

Given a set of centers c, the K-means algorithm searches for an optimal S by assigning to cluster S_k the entities that are nearest to c_k. Then, given $S = \{S_k\}$, the algorithm finds each cluster centroid c_k as the center of gravity of S_k (within-cluster means) with $k = 1, 2, \ldots, K$. If the new centers differ from those in the previous iteration, a new iteration of K-means is carried out using the updated centers. The K-means algorithm is intuitive and converges fast; centers can be used as interpretation vehicles – these are its main advantages.

In this study, we also use a version of K-means with random "mutations", allegedly helping to reach a deeper minimum of the criterion in (1). This version, referred to as Random Swap by [14], sometimes randomly changes one of the centers for one of the entities. In our computations, random center swaps occurred every 60 iterations (or after convergence), so that the number of swaps was reaching 30 (per execution).

One of the main disadvantages of K-means is the need in pre-specifying the number of clusters K, and initial centers $c = (c_k)$, with $k = 1, 2, \ldots, K$.

C. INITIALIZATION OF CLUSTER CENTERS

The problem of determining the "right" number of clusters K will be treated at length further on. Here, we describe two options for initialization of cluster centers. One option consists of the use of the very popular K-means++ algorithm [15]. According to this approach, the first center is a randomly chosen entity. The general step: having a subset of centers c already selected, define the distance to c, for every entity out of c, as the minimum distance to all entities in c. Assign to each of the entities a probability proportional to its distance to c. Choose the next center randomly according to the specified rule. Another algorithm, referred to as a version of MaxMin algorithm from [16], can be considered as a deterministic version of the K-means++. According to this approach, the very first center is defined as a randomly selected entity. Every next center is chosen so that it is maximally distant from already selected centers c. The distance between an entity and c is defined as the minimum distance to all centers in c.

D. CHOOSING THE RIGHT NUMBER OF CLUSTERS: A REVIEW

Unfortunately, the minimal inertia can give no lead to the problem of selecting the "right" number of clusters K in K-means because it decreases monotonically as a function of K

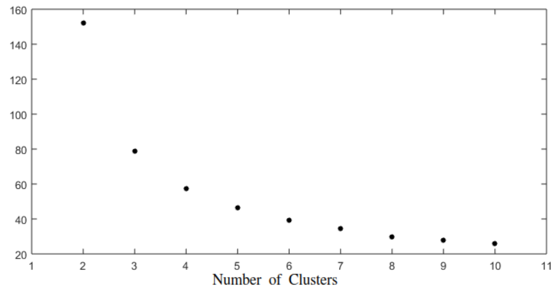

FIGURE 1. Minimum values of inertia $D(K)$ for different numbers of clusters $K = 2, 3, \ldots, 10$, each of them obtained after carrying out the K-means++ initialization and 100 random starts on the popular Iris data set from the Irvine repository.

(see, for example, Fig. 1). Moreover, no universal solution to the problem of finding a rule for determining the right number of clusters has been found so far (see, for example, the recent works [5], [6], [7], [8], [9]).

In the absence of outside information, two approaches for choosing the optimal number of clusters can be distinguished: (i) pre-analysis of a set of potential centroids in data, and (ii) post-processing multiple runs of K-means with random initializations at different values of K [1]. The approach (i) can be exemplified by the so-called affinity propagation algorithm [17] and iterated anomalous clustering [18], [19], [20]. The latter is based on the complementary K-means criterion (3) that can be maximized one-by-one.

This paper falls into the area of approach (ii) which can be pursued by using different strategies, including those mentioned in [1]:

- Resampling: choosing K according to the similarity of clustering results on randomly perturbed or sampled data;
- Combining multiple clusterings: choosing K according to stability of multiple clustering results at different values of K;
- Variance-, or inertia-, based approach: using some extensions of criterion (1) that should provide extreme values at a correct K.

Among the best performers, one frequently encounters cluster validity indices based on the inertia. Especially frequent are mentions of the Calinski-Harabasz index [21], the Elbow index (including a rather successful Curvature index by [12]), and the Hartigan rule [22]. Less known are the Xu index [23] and the WB (Within-Between) index [24].

The goal of our study is to present a comprehensive experimental testing of these indices, in the presence of a benchmark, especially for highly intermixed clusters, as well as their versions based on different ways of summarizing the results of multiple K-means runs at random initializations. Specifically, we consider that not only the minimum of the obtained inertia values should be taken into account, but their mean as well. Also, we maintain that the inertia can be computed by using both: conventional squared Euclidean distance and the (non-squared) Euclidean distance between the centers of clusters and their elements. Another novelty of this paper concerns different explanations of the elbow concept. Introduced somewhat vaguely in a speech by [25], the concept of elbow admits different interpretations depending on the numbers of clusters considered before and after the current number of clusters K. In particular, we focus on the three following indices: Elbow1, Elbow2, and Elbow3. Elbow1 is computed as the ratio of the differences between the value of inertia at K and one-step away values at $K-1$ and $K+1$, whereas Elbow2 is computed by using the differences between the inertia values at K and two-steps away values at $K-2$ and $K+2$. The Elbow3 is computed the same way, except that it compares the current inertia with three-steps away inertia values. It is worth noting that different step parameters used in the numerator and denominator of the Elbow formula, such as, for instance, difference between inertia values at $K-1$ and K in the denominator, or between inertia values at K and $K+2$ in the numerator, lead to inferior results.

III. INERTIA-BASED INDICES FOR DETERMINING THE RIGHT NUMBER OF CLUSTERS

A. ELBOW CRITERIA

Since the alternatingly mininized inertia criterion, $D(K)$ in Equation (1), cannot be used for determining the right number of clusters as is, one should look for a marked drop in the values of $D(K)$. The earliest reference to this idea, from an inauguration speech, tells us this: "Intuitively, it seems that a sudden marked flattening of the curve at any point should identify a distinctively "right" value of K" [25]. Currently, this is formulated as the idea of the largest "elbow" in the shape of the function $D(K)$ (see Fig. 1), or the largest drop from $D(K-1)$ to $D(K)$ relative to the drop from $D(K)$ to $D(K+1)$:

$$EL1(K) = \frac{D(K-1) - D(K)}{D(K) - D(K+1)}. \tag{5}$$

We should point out that the $EL1(K)$ index is closely related to the so-called Curvature index $C(K)$ that is a discrete approximation of a derivative-based measure of curvature in the function relating the number of clusters K to the minimum inertia criterion $D(K)$ [12]. The value of $C(K)$ is equivalently expressed by the following equation [12]: $C(K) = \frac{D(K-1)-D(K)}{D(K)-D(K+1)} - 1$, so that, obviously, $EL1(K) = C(K) + 1$. Since in several studies, $C(K)$ showed superiority over some popular metrics for choosing the true number of clusters [12], we included it in our experiments.

In our view, the elbow concept should not be restricted to one-step differences only. Thus, we define Elbow2, in a similar way:

$$EL2(K) = \frac{D(K-2) - D(K)}{D(K) - D(K+2)}, \tag{6}$$

as well as Elbow3, an index based on three-step differences:

$$EL3(K) = \frac{D(K-3) - D(K)}{D(K) - D(K+3)}. \tag{7}$$

Of course, the rule for selecting the right value of K is the same for $EL1$, $EL2$, and $EL3$ - it corresponds to the maximum values of these indices. We also tested "intermediate" Elbow metrics by swapping either numerators or denominators in Equations (5) and (6), but those versions appeared inferior in our experiments and, thus, are omitted from this narrative. In contrast, we did not expect the index $EL3$ to have any relevance. However, it showed interesting results in our experiments and, therefore, is included in the exposition.

B. CALINSKI-HARABASZ INDEX

The Calinski and Harabasz index, CH(K) [21], is defined by the ratio of the between-group dispersion and the within-group dispersion, i.e. SSB and SSW, in the manner of the ANOVA F-criterion:

$$CH(K) = \frac{SSB(K)/(K-1)}{SSW(K)/(N-K)}, \tag{8}$$

assuming that SSB has $K-1$ degrees of freedom, whereas SSW has $N-K$ degrees of freedom. By using Equations (2) and (4), this is easily converted into the relative cumulative drop in the $D(K)$ value after K steps, weighted by a constant value:

$$CH(K) = \frac{D(1)-D(K)}{D(K)} \times \frac{N-K}{K-1}. \tag{9}$$

Allegedly, $CH(K)$ is supposed to reach its maximum at the right number of clusters K.

C. HARTIGAN RULE

This rule works as follows. Let us step-by step increase K by one starting from $K = 1$ and compute the corresponding values of $H(K)$, below. The very first K at which $H(K)$ decreases to 10 is taken as the right K; this rule is referred to as the "rule of thumb" by [22].

$$H(K) = \left(\frac{D(K)}{D(K+1)} - 1\right) \times (N-K-1). \tag{10}$$

D. WB INDEX

The WB index [24] can be considered a rescaled reciprocal of the CH index and defined as follows:

$$WB(K) = K\frac{SSW(K)}{SSB(K)} = \frac{KD(K)}{D(1)-D(K)}. \tag{11}$$

Obviously, $WB(K) = \frac{K(K-1)}{(N-K)CH(K)}$. In contrast to the CH index, the WB index reaches its minimum at the right number of clusters, K.

E. XU INDEX

The Xu index [23] has been derived from a logarithmic formula for divergence under a version of a Gaussian mixture model. Its formula is the following:

$$XU(K) = V\log\left(\sqrt{\frac{D(K)}{VN^2}}\right) + \log K, \tag{12}$$

where V is the dimensionality of the data, N is the number of entities, and K is the number of clusters. It should reach its minimum at the right number of clusters K.

F. FOUR WAYS OF USING THE CLUSTER VALIDITY INDICES

To apply any of the cluster validity indices described above, one usually uses the following strategy. First, specify a range of K values, say $K = 2, \ldots, 25$. Then, for each K from the pre-defined range, run K-means many times, with random K entities taken as initial centers each time, and consider the obtained local minimum inertia values as a proxy to the global minimum value of inertia $D(K)$ for a given dataset. After this, Equations (5) to (12) can be applied to compute the best K values according to the minimum (or maximum) of the corresponding index.

Besides this conventional use of Equations (5) to (12), we will try some other, less conventional ways. Specifically, we will consider, on par with Equation (1), an unconventional Equation (13) below for computing $D(K)$ (not for running K-means, though):

$$DE = \sum_{k=1}^{K}\sum_{i\in S_k} d(y_i, c_k), \tag{13}$$

where $d(y_i, c_k) = \sqrt{\sum_{v=1}^{V}(y_{iv} - c_{kv})^2}$ is the Euclidean distance, and not its squared form used in Equation (1).

Also, when running through a range of K-values for computing a proxy $D(K)$ to the minimum inertia value over multiple runs of K-means at a given K, we will consider not only the minimum, but the average value of the respective D-values as well.

Therefore, in the follow-up experiments, every series of multiple runs of K-means over a given dataset results in four values of every index depending on the way of computing the final value of $D(K)$: (a) the minimum of conventional inertia values in Equation (1) (MinC); (b) the minimum of non-squared Euclidean distances in Equation (13) (MinE); (c) the mean of conventional inertia values in Equation (1) (MeanC); (d) the mean of non-squared Euclidean distances in Equation (13) (MeanE). This will allow us to test whether any less conventional way of using cluster validity indices and of computing inertia may be beneficial.

G. SILHOUETTE WIDTH INDEX

Silhouette width [26] is certainly one of the most popular cluster validity indices not based on the inertia formula. Given a dataset and a partition of entities S, the Silhouette width, $s(i)$ with $i \in I$, shows the degree of correspondence between the entity i and the partition S. Let us first define the average distance from i to its cluster S_k, $i \in S_k$:

$$a(i) = \frac{1}{|S_k| - 1}\sum_{j\in S_k, j\neq i} d(y_i, y_j), \tag{14}$$

and to a nearest cluster to which i does not belong:

$$b(i) = \min_{S_k : i \notin S_k} \{\frac{1}{|S_k|} \sum_{j \in S_k} d(y_i, y_j)\}. \tag{15}$$

The Silhouette width $s(i)$ of any entity $i \in I$ is defined then as the relative difference between $a(i)$ and $b(i)$:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}. \tag{16}$$

The average Silhouette width value $s(S) = \frac{1}{N} \sum_{i \in I} s(i)$ shows the extent of consistency in partition S: the closer the value of $s(S)$ to its maximum value, i.e. the unity, the better. The maximum value of $s(S)$ over K, $SW(K)$, corresponds to the right number of clusters. We are going to use this index as a reasonable match to the inertia-based indices.

IV. EXPERIMENTAL SETTINGS

A. INDICES UNDER COMPARISON

The following cluster validity indices were used in our simulations to select the right number of clusters:

1) SW (Silhouette width),
2) CH (Calinski-Harabasz),
3) HR (Hartigan),
4) WB (Within-Between),
5) XU (Xu),
6) EL1/EK2/EK3 (three Elbow indices differing by step size).

The three common cluster validity indices SW, CH, and HR have been tested in various experimental conditions by several authors (see, for example, [12], [18], [27], [28], [29]). In our experiences, the SW index, used as benchmark, was frequently superior over CH and HR rules. The WB and XU indices have been tested by [24] and found to be competitive against existing cluster validity indices.

In contrast to some existing literature, we consider three explanations of the concept of elbow in the curve presenting inertia values for different numbers of clusters (see an example in Fig. 1): one, Elbow conventional, based on a one-step scale, the others, Elbow2 and Elbow3, are unconventionally based on two-step and three-step scales, respectively. Besides, we measure the Silhouette width (SW) for each of the resulting partitions, using it as a popular benchmark. As we mentioned above, each of these indices is tested with four different options depending on how the final value of $D(K)$ is computed (see Section III-F for more details). This gives us four options for each of the indices: MinC, MeanC, MinE, and MeanE.

B. K-MEANS SETTINGS

On each considered dataset, standardized with what we call range normalization (for each feature, its grand mean is subtracted from its values, with a follow-up division by its range, the difference between maximum and minimum values), we run K-means, from initial centers found with a randomized MaxMin algorithm [16], 50 times for each K from 2 to 31, and then process the 50 results, as described above. Also, we apply Random Swap every 60 iterations (or after convergence) to randomly change one of the centers for one of the entities (swap step) and continue K-means iterations. Random Swap applies 30 swaps per execution.

C. DATA FOR COMPUTATIONAL EXPERIMENTS

To test and compare the indices above, we need an ensemble of datasets in which a hidden partition is known to us, so that we could compare the partition hidden in data table with that found by K-means using the number of clusters K advised by the cluster validity index under consideration. In our experiments, we use both real-world and synthetic datasets.

1) SYNTHETIC DATA GENERATOR

We generate synthetic data for clustering as a set of Gaussian clusters that may be intermixed to a degree controlled by one parameter only. Our data generator follows that described by [30]. It produces an $N \times V$ data matrix Y with a prespecified number K^* of Gaussian clusters using the following steps:

1) First, we specify the number of entities, N, the number of features, V, and the number of clusters, K^*. Also, we define the minimum cluster size, n, which may be needed rather large – to operate with probabilistic data models. The product K^*n must be less than N, so that the remaining $N - K^*n$ entities could be randomly distributed among the K^* clusters. Moreover, the number of cluster elements, N_k, can be specified for each cluster to be generated.
2) For each cluster k, $k = 1, 2, \ldots, K^*$, its center is generated as a V-dimensional vector c_k of components, following a uniform distribution over the interval $(\alpha - 1, 1 - \alpha)$, where α is a user-defined real, $0 < \alpha < 1$, to characterize the intermix of generated clusters. The larger the parameter α, the nearer to each other the centers and the more intermixed the elements of different clusters. This means that the value of α refers to the level of squeezing the data points (see this effect illustrated in Fig. 2).
3) To generate k-th cluster elements, we first generate a vector of its standard deviations $s_k = (s_{kv})$, following a uniform distribution from the interval $[0.05, 0.10]$, with $v = 1, 2, \ldots, V$. Then, we generate N_k V-dimensional vectors whose v-th components are taken from a Gaussian distribution $N(0, s_{kv})$ with 0 expectation and standard deviation s_{kv}, and add each of them to the center c_k. In this way, we obtain a matrix Y_k of dimension $N_k \times V$ of elements of cluster k.
4) To obtain an $N \times V$ data matrix Y, we combine the matrices Y_k together, so that entities from the same cluster sit together, and a hidden partition of the matrix row indices is known to the user.

2) REAL-WORLD DATASETS

We picked up eleven popular datasets of varying complexity from the celebrated UCI Machine Learning Repository [31].

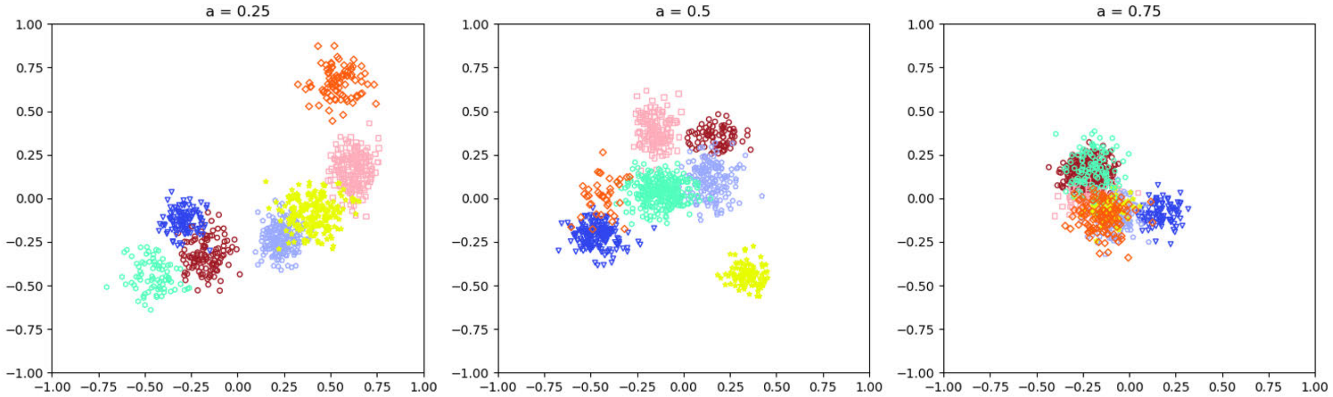

FIGURE 2. A synthetic dataset with 7 differently colored Gaussian clusters at different squeeze parameter values, $\alpha = 0.25$, on the left, $\alpha = 0.5$, on the center, and $\alpha = 0.75$, on the right ($N = 2,500$, $V = 15$). Clusters are more intermixed at the last picture because of the larger α.

Table 1 gives a short description of them. Unlike the synthetic datasets, relations between the features and ground-truth partition are not clear-cut and, quite possibly, may be very complex in some of the real-world datasets.

3) OVERLAPPING DATASETS FROM THE LITERATURE

We use "benchmark" datasets from [32], which are specifically designed to model various levels of cluster oddities. These sets are:

1) **Sets S**: 4 sets of size $5,000 \times 2$ with 15 equally-sized Gaussian spherical, and partly truncated, clusters of varying intermix. Clusters in the set S4 are highly intermixed.
2) **Sets G2** contain two Gaussian clusters at fixed locations, each with 1,024 points. Intermix is created by augmenting the standard deviation from 10 to 100. The data dimensions vary from 2 to 1,024 (the dimensions are of a 2^n form with $n = 1, 2, \ldots, 10$).
3) **Unbalanced sets of data** are created for eight clusters in two groups. The first three clusters are dense with 2,000 points each, whereas five other clusters are sparse with 100 points each.

Datasets (a) and (b) model various situations of intermix among clusters, whereas datasets (c) have rather well-separated, but highly unbalanced (in sizes), groups. They are available at: http://cs.uef.fi/sipu/datasets.

D. EVALUATION OF RESULTS

We use three conventional metrics for evaluation of the quality of reproduction of the partitions hidden in data by a clustering algorithm. They are as follows:

- MARE, Mean Absolute Relative Error in the number of clusters. We define absolute relative error in the number of clusters as the ratio $\frac{|K^*-K|}{|K^*|}$, where K^* is the number of clusters in the ground-truth partition and K is the number of clusters obtained by the algorithm under consideration. If there are R runs of the algorithm at different initializations, then we use the average of the R values, i.e. the mean absolute relative error:

$$MARE(K^*) = \frac{1}{R}\sum_{r=1}^{R}\frac{|K^*-K_r|}{K^*}. \tag{17}$$

- ARI, Adjusted Rand Index [33], a measure of similarity between two partitions, based on the number of pairs of entities that are consistent in the partitions, that is, either belong to the same cluster, or to different clusters, in both partitions. The maximum value of ARI is 1; it is reached only if the partitions coincide. (18), as shown at the bottom of the next page.
In the above, A and B are two partitions of the entity set with K_A and K_B elements, respectively; a_k and b_m are the cardinalities of parts A and B, respectively; n_{km} are the frequencies in the joint AB distribution; $\binom{n}{2}$ is a binomial term equal to $n(n-1)/2$.
- NMI, Normalized Mutual Information between two partitions according to a definition from [34]:

$$NMI(A, B) = 2 \times \frac{H(A) + H(B) - H(A, B)}{H(A) + H(B)}, \tag{19}$$

where p_k is the proportion of entities in k-th part of partition $A(k = 1, 2, \ldots, K)$, $H(A) = -\sum_{k=1}^{K} p_k \log(p_k)$ is the entropy of A, and $H(AB)$ is the entropy of the joint distribution AB.

TABLE 1. Characteristics of datasets from UCI machine learning repository used in our experiments.

Data set	#features	#observations	#clusters
E.coli	7	336	8
Glass	9	214	7
Ionosphere	34	351	2
Iris	4	150	3
Optical Digits	64	5,620	10
Pima Indians Diabetes	8	768	2
Segmentation (image)	19	2,310	7
Wine	13	178	3
Wisconsin Breast Cancer (diag.)	34	198	2
Wisconsin Breast Cancer (prog.)	32	569	2
Zoo	17	101	7

The MARE evaluates the performance of algorithms in the recovery of the number of clusters, whereas ARI and NMI evaluate the quality of recovery of the clusters themselves.

V. RESULTS OF EXPERIMENTS

Before reporting the obtained results, we remark that the Random Swap algorithm on top of K-means has hardly inflicted any changes in the results. The differences between results provided by the two implementations, i.e. with and without Random Swap, were quite minor even when they did occur. Moreover, sometimes it is the "naked" K-means which demonstrated superior results over Random Swap, as, for instance, for the WBC diagnosis dataset from the UCI repository. The XU index led to good results with K-means alone (ARI = 0.96 between the ground truth partition and that found by the algorithm), whereas the maximum ARI value was only 0.34 after the application of Random Swap. This phenomenon perhaps can be explained by the nature of our experiments which involve a number of random initializations, probably making an effect similar to that of random swaps. Therefore, in our description, we decided to skip the results found by using Random Swap; these results, as well as all obtained experimental results, are available in our GitHub repository: https://github.com/glendawur/indices_kmeans.

In the remainder of this section, we describe our experimental results for each of the three main settings: (a) synthetic datasets with a controllable cluster intermix; (b) real-world datasets from the UCI repository, and (c) specifically designed datasets from the literature, as explained in Section IV-C3.

A. RESULTS FOR SYNTHETIC DATA

In our experiments, we generated datasets with $N = 2,500$ entities (rows), $V = 15$ or $V = 50$ features, and $K^* = 7$, or 15, or 21 clusters. In this way, we can observe cases at which the number of clusters is smaller than the number of features versus cases at which, in contrast, the number of clusters is greater than the number of features. In our experience, some clustering algorithms might lead to different clustering effects depending on the relation between these two numbers. We considered the values of the squeeze parameter at three levels: $\alpha = 0.5, 0.75$, and 0.85. Therefore, we had $2 \times 3 \times 3 = 18$ combinations of parameters V, K^*, and α. For each of them we generated 30 datasets and ran K-means for each K from 2 to 31. The ARI values obtained for various configurations of the squeeze parameter α and the true number of clusters K^* are presented in Table 2 (for the space dimension $V = 15$) and Table 3 (for the space dimension $V = 50$).

Observations on the results in Tables 2 and 3:

1) First, one cannot help but notice that the ARI values monotonically weaken with the growth of the intermix level. Also, we note that the growth of the space dimension positively affects the results of every index. Of course, this relates to the way the synthetic datasets have been generated, so that features do not much differ in relevance to the clusters.
2) Rather unexpectedly, the indices appear to differ with respect to the ways the results of multiple runs of K-means are processed. Two of them, CH and WB, are best with the conventional squared Euclidean distance. Moreover, taking the average rather than optimal results leads to better cluster recovery for these two indices. Two other indices, HR and XU, produce better results with the Euclidean distance. In general, they are also better off with the averaging option, although at a greater cluster intermix ($\alpha = 0.85$), taking the optimal solution leads to better results. As to the elbow-based indices EL1, EL2, and EL3, they seem to work equally well with both the Euclidean distance and its squared form. Using the mean post-processing results seems beneficial for each of them, again with a caveat that at a greater intermix using the optimal solution is slightly better.
3) Among all the indices under consideration, SW is an overall winner, especially at the larger space dimension ($V = 50$). The only rival capable of getting superior results, at $V = 15$, is the XU index. At greater cluster intermixes, $\alpha = 0.75$ or 0.85, with optimal values and the Euclidean distance, XU frequently outperforms SW, as can be seen in Table 2.
4) Among the inertia-based indices, EL2 and EL3 used in the conventional mode (with the squared Euclidean distance), as well as the XU index with the Euclidean distance, mentioned above, both with the optimal post-processing option, seem the most balanced. The CH and WB indices, both in the conventional mode, work quite well if clusters are not much intermixed; however, with $\alpha = 0.85$ both become irrelevant. The HR index is good at $V = 50$; however, at $V = 15$, it is hopeless for small numbers of clusters, although it gets better for larger cluster numbers.

For the sake of space, we skip over presenting the results obtained with the NMI criterion. They have an overall pattern similar to those of ARI, although in a somewhat blurred way, and we refer an interested reader to our GitHub repository: https://github.com/glendawur/indices_kmeans.

Errors in the number of clusters generally follow the patterns of ARI values. Therefore, we present

$$ARI(A, B) = \frac{\binom{N}{2} \times \sum_{k=1}^{K_A} \sum_{m=1}^{K_B} \binom{n_{km}}{2} - \sum_{k=1}^{K_A} \binom{a_k}{2} \sum_{m=1}^{K_B} \binom{b_m}{2}}{\frac{1}{2} \times \binom{N}{2} \times \left[\sum_{k=1}^{K_A} \binom{a_k}{2} + \sum_{m=1}^{K_B}\right] - \sum_{k=1}^{K_A} \binom{a_k}{2} \sum_{m=1}^{K_B} \binom{b_m}{2}}. \qquad (18)$$

TABLE 2. Adjusted Rand Index values followed by their standard deviations obtained on synthetic datasets with 15 dimensions. Each of the inertia-based indices was applied at four different modes, combining (a) two different distances: Euclidean distance and the conventional squared Euclidean distance; and (b) using the average index value (mean) or the optimal one (max/min). An upper row contains the values of the "independent" benchmark index SW (Silhouette width). The maximum ARI values are highlighted with bold font column-wise.

			$\alpha = 0.5$			$\alpha = 0.75$			$\alpha = 0.85$		
ind	dist	Ag/K^*	7	15	21	7	15	21	7	15	21
SW	-	-	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.98/0.05	**1.0/0.01**	0.98/0.03	0.88/0.15	**0.88/0.07**	0.88/0.03
CH	Eucl	mean	0.44/0.14	0.14/0.03	0.1/0.01	0.4/0.1	0.15/0.03	0.09/0.02	0.38/0.1	0.14/0.02	0.08/0.01
		max	0.42/0.1	0.14/0.03	0.1/0.01	0.4/0.1	0.15/0.03	0.09/0.02	0.38/0.1	0.14/0.02	0.08/0.01
	Conv	mean	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.84/0.17	0.82/0.31	0.66/0.42	0.48/0.17	0.15/0.06	0.08/0.01
		max	1.0/0.0	1.0/0.0	1.0/0.0	0.84/0.17	0.82/0.31	0.69/0.41	0.48/0.17	0.15/0.06	0.08/0.01
HR	Eucl	mean	0.35/0.07	0.77/0.07	0.97/0.05	0.37/0.07	0.78/0.09	0.89/0.05	0.37/0.07	0.69/0.09	0.83/0.04
		min	0.35/0.08	0.75/0.07	0.92/0.06	0.36/0.08	0.76/0.08	0.91/0.06	0.36/0.07	0.68/0.09	0.84/0.05
	Conv	mean	0.0*/0.0	0.09*/0.21	0.08*/0.25	0.01*/0.05	0.12*/0.24	0.28*/0.4	0.0*/0.0	0.0*/0.0	0.0*/0.0
		min	0.03*/0.09	0.19*/0.28	0.21*/0.35	0.02*/0.06	0.12*/0.23	0.08*/0.24	0.0*/0.03	0.05*/0.16	0.02*/0.13
WB	Eucl	mean	0.87/0.17	0.57/0.36	0.12/0.04	0.52/0.16	0.18/0.08	0.11/0.02	0.43/0.13	0.16/0.06	0.09/0.02
		min	0.88/0.17	0.58/0.37	0.12/0.04	0.52/0.16	0.16/0.06	0.1/0.02	0.43/0.13	0.16/0.06	0.09/0.02
	Conv	mean	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.96/0.06	0.99/0.02	0.97/0.02	0.83/0.12	0.55/0.12	0.31/0.1
		min	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.96/0.06	0.99/0.02	0.98/0.02	0.83/0.12	0.55/0.12	0.31/0.1
XU	Eucl	mean	0.91/0.14	0.99/0.05	0.99/0.03	0.96/0.1	0.94/0.07	0.94/0.05	**0.93/0.07**	0.87/0.14	0.89/0.03
		min	0.89/0.14	0.99/0.05	0.99/0.03	0.96/0.1	0.98/0.06	0.99/0.03	**0.93/0.07**	0.87/0.14	**0.9/0.02**
	Conv	mean	0.23/0.04	0.55/0.05	0.76/0.04	0.24/0.05	0.55/0.05	0.76/0.04	0.23/0.05	0.49/0.06	0.68/0.03
		min	0.23/0.04	0.55/0.05	0.76/0.04	0.24/0.05	0.55/0.05	0.75/0.04	0.23/0.05	0.49/0.06	0.68/0.04
EL1	Eucl	mean	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.98/0.06	0.97/0.1	0.93/0.07	0.91/0.09	0.7/0.21	0.72/0.24
		min	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.95/0.16	0.96/0.14	**1.0/0.0**	0.87/0.2	0.69/0.23	0.82/0.2
	Conv	mean	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.99/0.06	0.98/0.05	0.94/0.06	0.9/0.11	0.67/0.23	0.67/0.29
		min	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.99/0.06	0.98/0.09	**1.0/0.0**	0.9/0.11	0.74/0.23	0.86/0.16
EL2	Eucl	mean	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.98/0.05	0.98/0.05	0.96/0.05	0.9/0.12	0.74/0.19	0.67/0.28
		min	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.99/0.04	0.99/0.04	**1.0/0.0**	0.9/0.12	0.74/0.21	0.84/0.18
	Conv	mean	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.99/0.04	0.98/0.05	0.96/0.05	0.91/0.1	0.76/0.2	0.64/0.29
		min	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.99/0.04	0.99/0.04	**1.0/0.0**	0.91/0.1	0.77/0.21	0.84/0.18
EL3	Eucl	mean	0.98/0.07	**1.0/0.0**	**1.0/0.0**	0.97/0.08	0.97/0.06	0.96/0.04	0.92/0.1	0.75/0.18	0.77/0.22
		min	0.99/0.04	**1.0/0.0**	**1.0/0.0**	0.99/0.06	0.99/0.04	**1.0/0.01**	0.92/0.1	0.73/0.19	0.87/0.14
	Conv	mean	0.98/0.07	**1.0/0.0**	**1.0/0.0**	0.99/0.06	0.97/0.06	0.96/0.05	0.92/0.1	0.77/0.17	0.78/0.2
		min	0.99/0.04	**1.0/0.0**	**1.0/0.0**	**1.0/0.01**	**1.0/0.0**	**1.0/0.0**	0.92/0.1	0.81/0.17	0.85/0.17

TABLE 3. Adjusted Rand Index values followed by their standard deviations obtained on synthetic datasets with 50 dimensions. Each of the inertia-based indices was applied at four different modes, combining (a) two different distances: Euclidean distance and the conventional squared Euclidean distance; and (b) using the average index value (mean) or the optimal one (max/min). An upper row contains the values of the "independent" benchmark index SW (Silhouette width). The maximum ARI values are highlighted with bold font column-wise.

			$\alpha = 0.5$			$\alpha = 0.75$			$\alpha = 0.85$		
ind	dist	Ag/K^*	7	15	21	7	15	21	7	15	21
SW			**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.01**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**
CH	Eucl	mean	0.8/0.27	0.1/0.04	0.05/0.02	0.34/0.12	0.09/0.05	0.06/0.03	0.37/0.14	0.09/0.04	0.06/0.03
		min	0.38/0.12	0.1/0.04	0.05/0.02	0.33/0.12	0.09/0.05	0.06/0.03	0.37/0.14	0.09/0.04	0.06/0.03
	Conv	mean	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.98/0.08	0.98/0.13	**1.0/0.0**	0.75/0.27	0.57/0.39	0.51/0.46
		min	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.96/0.11	0.95/0.19	**1.0/0.0**	0.59/0.24	0.29/0.26	0.14/0.17
HR	Eucl	mean	0.92/0.13	**1.0/0.0**	**1.0/0.0**	0.97/0.08	**1.0/0.0**	**1.0/0.0**	0.98/0.07	0.98/0.04	0.98/0.03
		min	0.75/0.15	**1.0/0.0**	**1.0/0.0**	0.81/0.14	0.98/0.07	**1.0/0.0**	0.81/0.15	0.99/0.03	**1.0/0.0**
	Conv	mean	0.55/0.11	0.98/0.06	**1.0/0.0**	0.64/0.13	0.98/0.05	**1.0/0.0**	0.65/0.14	0.92/0.06	0.95/0.03
		min	0.55/0.11	0.89/0.09	0.98/0.03	0.61/0.08	0.92/0.1	0.99/0.03	0.61/0.09	0.9/0.08	**1.0/0.02**
WB	Eucl	mean	0.97/0.07	**1.0/0.0**	**1.0/0.0**	0.85/0.19	0.26/0.22	0.1/0.05	0.64/0.2	0.18/0.09	0.1/0.05
		min	0.93/0.11	0.9/0.26	0.75/0.42	0.63/0.17	0.11/0.06	0.06/0.04	0.38/0.14	0.1/0.05	0.06/0.03
	Conv	mean	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.97/0.1	0.94/0.16	0.99/0.02
		min	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.01**	**1.0/0.0**	**1.0/0.0**	0.92/0.1	0.83/0.21	0.97/0.04
XU	Eucl	mean	0.95/0.12	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.95/0.05	0.94/0.04
		min	0.81/0.18	0.93/0.08	0.98/0.03	0.93/0.14	0.94/0.1	0.98/0.05	0.9/0.16	0.96/0.06	0.99/0.04
	Conv	mean	0.26/0.05	0.6/0.05	0.81/0.03	0.27/0.04	0.6/0.06	0.81/0.03	0.27/0.05	0.61/0.04	0.81/0.02
		min	0.26/0.05	0.6/0.05	0.8/0.03	0.27/0.04	0.6/0.07	0.81/0.03	0.27/0.05	0.61/0.04	0.81/0.03
EL1	Eucl	mean	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.88/0.27	0.98/0.05	0.97/0.04
		min	0.91/0.24	0.99/0.04	**1.0/0.0**	0.83/0.3	0.96/0.14	0.99/0.03	0.67/0.36	0.9/0.16	0.99/0.04
	Conv	mean	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.93/0.23	0.98/0.05	0.97/0.04
		min	0.93/0.21	**1.0/0.0**	**1.0/0.0**	0.84/0.3	0.99/0.08	0.99/0.03	0.62/0.36	0.93/0.15	**1.0/0.0**
EL2	Eucl	mean	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.99/0.06	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.98/0.03	0.97/0.03
		min	0.97/0.1	**1.0/0.0**	**1.0/0.0**	0.95/0.15	0.98/0.07	**1.0/0.02**	0.96/0.16	0.99/0.07	**1.0/0.0**
	Conv	mean	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.99/0.06	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.98/0.03	0.97/0.03
		min	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.98/0.07	**1.0/0.0**	0.96/0.16	0.99/0.07	**1.0/0.0**
EL3	Eucl	mean	0.99/0.04	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.97/0.05	0.97/0.03
		min	0.93/0.12	0.99/0.03	**1.0/0.02**	0.98/0.07	0.98/0.07	**1.0/0.02**	0.96/0.13	0.99/0.03	**1.0/0.0**
	Conv	mean	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	**1.0/0.0**	0.97/0.05	0.97/0.03
		min	0.95/0.11	0.99/0.02	**1.0/0.0**	0.99/0.04	0.98/0.07	**1.0/0.02**	0.96/0.11	0.99/0.04	**1.0/0.0**

here only MARE values at $V = 15$ and skip the intermediate cluster squeezing level $\alpha = 0.75$ (see Table 4).

B. RESULTS FOR THE UCI REPOSITORY DATASETS

Tables 5 and 6 report, respectively, the ARI and MARE values for eight of the eleven Irvine repository datasets considered in

TABLE 4. Average mean absolute relative error (MARE) in the number of clusters followed by its standard deviation for synthetic datasets of 15 dimensions.

		α	$\alpha = 0.5$			$\alpha = 0.85$		
ind	dist	Ag/K^*	7	15	21	7	15	21
SW	-	-	**0.0/0.0**	**0.0/0.01**	**0.0/0.01**	0.18/0.21	0.14/0.13	0.09/0.06
CH	Eucl	mean	0.58/0.25	0.87/0.0	0.9/0.0	0.71/0.0	0.87/0.0	0.9/0.0
		min	0.71/0.03	0.87/0.0	0.9/0.0	0.71/0.0	0.87/0.0	0.9/0.0
	Conv	mean	**0.0/0.03**	**0.0/0.0**	**0.0/0.0**	0.62/0.14	0.84/0.05	0.9/0.0
		min	**0.0/0.03**	**0.0/0.0**	**0.0/0.0**	0.65/0.12	0.86/0.03	0.9/0.0
HR	Eucl	mean	1.11/0.37	0.04/0.06	**0.0/0.0**	1.02/0.23	0.22/0.08	0.09/0.06
		min	0.95/0.38	0.16/0.08	0.03/0.03	0.97/0.24	0.17/0.08	**0.03/0.03**
	Conv	mean	2.42*/0.89	0.83*/0.25	0.29*/0.13	2.75*/0.68	0.79*/0.26	0.36*/0.09
		min	2.08*/0.75	0.49*/0.2	0.23*/0.1	2.28*/0.62	0.62*/0.21	0.26*/0.13
WB	Eucl	mean	0.09/0.18	0.09/0.24	0.74/0.34	0.66/0.07	0.85/0.03	0.9/0.0
		min	0.29/0.27	0.5/0.41	0.9/0.01	0.7/0.05	0.86/0.02	0.9/0.0
	Conv	mean	**0.0/0.0**	**0.0/0.0**	**0.0/0.0**	0.37/0.17	0.64/0.12	0.75/0.08
		min	**0.0/0.0**	**0.0/0.0**	**0.0/0.0**	0.34/0.17	0.61/0.14	0.75/0.08
XU	Eucl	mean	**0.0/0.03**	**0.0/0.0**	**0.0/0.0**	0.17/0.18	0.14/0.16	0.1/0.16
		min	0.07/0.1	**0.0/0.02**	**0.0/0.01**	0.13/0.16	**0.13/0.18**	**0.03/0.03**
	Conv	mean	3.24/0.12	0.62/0.48	0.03/0.09	3.16/0.57	0.94/0.11	0.41/0.03
		min	3.12/0.25	0.84/0.27	0.27/0.13	3.13/0.25	0.84/0.21	0.35/0.08
EL1	Eucl	mean	0.1/0.52	**0.0/0.0**	**0.0/0.0**	1.11/1.06	0.43/0.29	0.16/0.12
		min	0.77/1.07	0.15/0.31	0.04/0.11	1.7/1.14	0.57/0.26	0.14/0.14
	Conv	mean	0.1/0.52	**0.0/0.0**	**0.0/0.0**	0.96/1.04	0.38/0.27	0.17/0.12
		min	0.51/1.06	0.07/0.21	0.03/0.09	1.64/1.24	0.47/0.32	0.12/0.13
EL2	Eucl	mean	**0.0/0.0**	**0.0/0.0**	**0.0/0.0**	0.17/0.17	0.35/0.28	0.3/0.31
		min	0.02/0.05	**0.0/0.0**	**0.0/0.01**	0.34/0.59	0.27/0.3	0.1/0.18
	Conv	mean	**0.0/0.0**	**0.0/0.0**	**0.0/0.0**	0.16/0.17	0.4/0.28	0.3/0.29
		min	0.01/0.04	**0.0/0.0**	**0.0/0.01**	0.35/0.69	0.2/0.27	0.06/0.1
EL3	Eucl	mean	**0.0/0.0**	**0.0/0.0**	**0.0/0.0**	0.13/0.11	0.34/0.24	0.33/0.3
		min	0.03/0.06	**0.0/0.01**	0.01/0.02	**0.1/0.12**	0.26/0.29	0.06/0.14
	Conv	mean	**0.0/0.0**	**0.0/0.0**	**0.0/0.0**	0.12/0.11	0.34/0.25	0.37/0.32
		min	0.02/0.05	**0.0/0.0**	**0.0/0.01**	**0.1/0.12**	0.21/0.29	0.06/0.14

TABLE 5. Adjusted rand index (ARI) results, followed by their standard deviations, obtained for the real-world UCI repository datasets.

index	dist	agg	e-coli	iris	optdigits	segment	wbc dia	wbc pro	wine	zoo
SW	-	-	0.21/0.3	0.57/0.0	**0.67/0.02**	0.1/0.0	**0.96/0.0**	**0.7/0.01**	**0.9/0.0**	0.28/0.01
CH	Eucl	mean	0.65/0.1	0.57/0.0	0.13/0.0	0.12/0.05	**0.96/0.0**	**0.7/0.01**	0.37/0.0	0.45/0.0
		min	0.47/0.11	0.57/0.0	0.13/0.0	0.1/0.0	**0.96/0.0**	**0.7/0.01**	0.37/0.0	0.45/0.0
	Conv	mean	**0.69/0.0**	0.57/0.0	0.13/0.0	0.31/0.0	**0.96/0.0**	**0.7/0.01**	0.39/0.09	0.57/0.15
		min	0.68/0.06	**0.72/0.0**	0.13/0.0	0.48/0.0	**0.96/0.0**	**0.7/0.01**	0.37/0.0	0.45/0.0
HR	Eucl	mean	0.47/0.06	0.47/0.0	0.0*/0.0	0.07*/0.15	0.4/0.01	0.29/0.06	**0.9/0.0**	**0.87/0.01**
		min	0.54/0.13	0.42/0.0	0.02*/0.09	0.39/0.04	0.41/0.01	0.27/0.02	**0.9/0.0**	**0.87/0.01**
	Conv	mean	0.31/0.04	0.36/0.01	0.0*/0.0	0.02*/0.09	0.34/0.02	0.2/0.01	0.69/0.03	0.73/0.1
		min	0.42/0.04	0.36/0.0	0.0*/0.0	0.33*/0.11	0.35/0.04	0.19/0.01	0.65/0.04	0.69/0.0
WB	Eucl	mean	**0.69/0.0**	0.57/0.0	0.13/0.0	0.31/0.0	**0.96/0.0**	**0.7/0.01**	0.37/0.0	0.51/0.12
		min	0.47/0.11	0.57/0.0	0.13/0.0	0.1/0.0	0.96/0.0	**0.7/0.01**	0.37/0.0	0.45/0.0
	Conv	mean	0.66/0.07	0.47/0.05	0.39/0.06	0.48/0.01	**0.96/0.0**	**0.7/0.01**	0.85/0.0	0.29/0.0
		min	**0.69/0.0**	**0.72/0.0**	0.42/0.0	0.48/0.0	**0.96/0.0**	**0.7/0.01**	0.85/0.0	0.29/0.0
XU	Eucl	mean	0.19/0.01	0.68/0.06	0.46/0.02	0.34/0.02	**0.96/0.0**	0.12/0.0	0.18/0.01	0.28/0.01
		min	0.38/0.22	**0.72/0.0**	0.46/0.02	0.34/0.02	**0.96/0.0**	0.12/0.0	0.18/0.01	0.28/0.01
	Conv	mean	0.19/0.01	0.17/0.01	0.47/0.02	0.34/0.02	0.32/0.01	0.12/0.01	0.18/0.01	0.29/0.0
		min	0.19/0.01	0.17/0.01	0.47/0.02	0.34/0.02	0.32/0.01	0.12/0.01	0.18/0.01	0.29/0.0
EL1	Eucl	mean	0.47/0.22	0.55/0.13	0.58/0.09	0.31/0.0	0.33/0.03	0.27/0.11	**0.9/0.0**	0.44/0.2
		min	0.52/0.15	0.54/0.24	0.55/0.09	0.43/0.05	0.34/0.06	0.17/0.05	0.74/0.29	0.49/0.13
	Conv	mean	0.55/0.22	0.64/0.15	0.56/0.11	0.32/0.01	0.46/0.22	0.22/0.1	0.85/0.0	0.52/0.24
		min	0.53/0.14	0.64/0.19	0.57/0.05	0.45/0.05	0.46/0.2	0.15/0.02	0.48/0.31	0.45/0.14
EL2	Eucl	mean	**0.69/0.0**	0.56/0.07	0.61/0.06	0.52/0.01	0.35/0.13	0.39/0.0	0.73/0.01	0.8/0.19
		min	0.66/0.08	0.62/0.0	0.45/0.06	0.45/0.05	0.31/0.05	0.17/0.05	0.73/0.01	0.42/0.14
	Conv	mean	**0.69/0.0**	0.61/0.03	0.5/0.14	**0.53/0.01**	0.62/0.24	0.6/0.01	0.75/0.0	0.8/0.1
		min	**0.69/0.0**	0.61/0.0	0.49/0.09	0.47/0.03	0.72/0.18	0.21/0.06	0.73/0.1	0.76/0.09
EL3	Eucl	mean	0.68/0.0	0.47/0.0	0.63/0.04	0.48/0.0	20.38/0.17	0.24/0.03	0.65/0.02	0.62/0.04
		min	0.68/0.05	0.47/0.0	0.5/0.01	0.44/0.06	0.37/0.05	0.25/0.07	0.65/0.02	0.47/0.16
	Conv	mean	0.68/0.0	0.46/0.0	0.5/0.14	0.48/0.01	0.8/0.0	0.3/0.06	0.67/0.0	0.85/0.03
		min	**0.69/0.0**	0.46/0.0	0.51/0.02	0.48/0.04	0.73/0.15	0.28/0.07	0.67/0.0	0.78/0.1

our study (see Table 1). Datasets Ionosphere, Pima Indian and Glass were excluded because none of the presented indices provided any reasonable value for them: the maxima of the ARI values for these datasets were 0.20, 0.06, and 0.36, respectively. This should be attributed to the fact that features in these datasets are not quite indicative of the ground truth partitions – which is also true for many other real-world datasets.

The ARI values presented in Table 5 are much more diverse than those reported in Tables 2 and 3. The datasets differ, and so differ the results. SW is not an overall winner anymore, although it does win on four out of eight datasets: Optical digits, both Wisconsin breast cancer, diagnosis and prognosis, and Wine. Two other indices that win on four datasets are CH and WB, in their conventional mode with optimal choices. The ''winning'' datasets are the same for these two

TABLE 6. Mean absolute relative error (MARE) number of clusters recovery results obtained for the real-world UCI repository datasets.

index	dist	agg	e-coli	iris	optdigits	segment	wbc dia	wbc pro	wine	zoo
SW	-	-	0.65/0.17	0.33/0.0	**0.01/0.03**	0.71/0.0	**0.0/0.0**	**0.0/0.0**	**0.0/0.0**	3.2/0.14
CH	Eucl	mean	0.4/0.08	0.33/0.0	0.8/0.0	0.7/0.04	**0.0/0.0**	**0.0/0.0**	0.33/0.0	0.71/0.0
		min	0.63/0.09	0.33/0.0	0.8/0.0	0.71/0.0	**0.0/0.0**	**0.0/0.0**	0.33/0.0	0.71/0.0
	Conv	mean	0.26/0.05	0.33/0.0	0.8/0.0	0.57/0.0	**0.0/0.0**	**0.0/0.0**	0.32/0.06	0.65/0.08
		min	0.44/0.09	**0.0/0.0**	0.8/0.0	0.43/0.0	**0.0/0.0**	**0.0/0.0**	0.33/0.0	0.71/0.0
HR	Eucl	mean	0.17/0.12	0.67/0.0	2.0*/0.0	3.15*/0.32	3.77/0.69	2.5/0.96	**0.0/0.0**	0.43/0.0
		min	0.57/0.06	1.0/0.0	1.99*/0.04	1.43/0.79	3.43/0.9	2.47/0.35	**0.0/0.0**	0.43/0.0
	Conv	mean	1.14/0.24	2.59/0.17	2.0*/0.0	3.26*/0.12	9.85/1.42	5.1/0.61	0.61/0.13	0.14/0.11
		min	0.52/0.24	2.67/0.0	2.0*/0.0	1.96*/0.74	8.37/3.06	5.45/0.33	0.74/0.14	0.0/0.0
WB	Eucl	mean	0.27/0.04	0.33/0.0	0.8/0.0	0.57/0.0	**0.0/0.0**	**0.0/0.0**	0.33/0.0	0.68/0.06
		min	0.63/0.09	0.33/0.0	0.8/0.0	0.71/0.0	**0.0/0.0**	**0.0/0.0**	0.33/0.0	0.71/0.0
	Conv	mean	**0.16/0.09**	0.64/0.12	0.45/0.11	0.29/0.0	**0.0/0.0**	**0.0/0.0**	**0.0/0.0**	3.24/0.09
		min	0.36/0.13	**0.0/0.0**	0.4/0.0	0.43/0.0	**0.0/0.0**	**0.0/0.0**	**0.0/0.0**	3.25/0.07
XU	Eucl	mean	2.72/0.05	0.08/0.14	2.0/0.0	3.29/0.0	**0.0/0.0**	14.0/0.0	9.0/0.0	3.29/0.0
		min	1.61/1.14	**0.0/0.0**	2.0/0.0	3.21/0.13	**0.0/0.0**	13.92/0.19	9.0/0.0	3.29/0.0
	Conv	mean	2.74/0.03	9.0/0.0	2.0/0.0	3.29/0.0	13.85/0.3	14.0/0.0	9.0/0.0	3.29/0.0
		min	2.74/0.03	9.0/0.0	2.0/0.0	3.28/0.04	13.95/0.15	13.95/0.15	9.0/0.0	3.29/0.0
EL1	Eucl	mean	0.97/0.8	0.6/0.85	0.58/0.57	0.65/0.44	9.43/2.81	4.6/4.18	**0.0/0.0**	1.98/1.21
		min	0.66/0.39	2.23/3.32	1.07/0.73	1.16/0.94	7.37/3.76	7.87/3.59	1.39/2.65	1.05/1.05
	Conv	mean	0.81/0.87	0.6/1.85	0.6/0.53	0.72/0.58	8.98/4.97	5.75/3.35	**0.0/0.0**	1.65/1.25
		min	0.55/0.1	1.06/2.74	0.91/0.72	0.9/0.81	5.55/3.89	9.58/2.25	3.66/3.32	1.35/1.01
EL2	Eucl	mean	0.26/0.05	0.48/0.17	0.17/0.2	**0.1/0.06**	9.63/2.88	1.0/0.0	0.33/0.0	0.77/0.89
		min	0.35/0.14	0.33/0.0	0.4/0.27	1.03/0.9	7.22/2.65	7.78/3.39	0.33/0.0	1.38/0.9
	Conv	mean	0.25/0.06	0.34/0.06	0.27/0.22	0.13/0.04	5.33/5.45	1.0/0.0	0.33/0.0	0.51/0.47
		min	0.36/0.13	0.33/0.0	0.29/0.15	0.54/0.46	1.82/1.51	5.73/2.91	0.56/1.22	0.33/0.16
EL3	Eucl	mean	0.25/0.0	0.67/0.0	0.07/0.06	0.23/0.12	8.88/3.11	3.4/0.53	0.67/0.0	0.29/0.0
		min	0.34/0.07	0.67/0.0	0.3/0.02	0.87/0.99	5.0/2.32	3.72/2.79	0.67/0.0	1.15/0.92
	Conv	mean	0.25/0.02	0.67/0.0	0.28/0.23	0.28/0.05	1.5/0.0	2.33/1.04	0.67/0.0	0.29/0.0
		min	0.35/0.06	0.67/0.0	0.26/0.06	0.39/0.51	1.95/1.03	2.93/1.83	0.67/0.0	0.2/0.12

TABLE 7. ARI results, followed by their standard deviations, obtained for benchmark datasets (S and unbalance).

		Dataset	S ($K^* = 15$)				Unbalance ($K^* = 8$)
Ind	dist	Agg/var	1.0	2.0	3.0	4.0	
SW	-	-	**0.99/0.0**	**0.93/0.0**	**0.7/0.0**	**0.62/0.0**	**1.0/0.0**
CH	Eucl	mean	0.13/0.0	0.13/0.0	0.11/0.0	0.1/0.0	0.12/0.0
		min	0.13/0.0	0.13/0.0	0.11/0.0	0.1/0.0	0.82/0.36
	Conv	mean	0.94/0.01	0.89/0.01	**0.7/0.01**	0.61/0.01	**1.0/0.0**
		min	**0.99/0.0**	**0.93/0.0**	**0.7/0.0**	0.61/0.0	0.81/0.21
HR	Eucl	mean	0.5*/0.47	0.06*/0.22	0.0*/0.0	0.0*/0.0	0.64*/0.19
		min	0.0*/0.0	0.0*/0.0	0.0*/0.0	0.0*/0.0	0.37*/0.31
	Conv	mean	0.5*/0.47	0.0*/0.0	0.0*/0.0	0.0*/0.0	0.51*/0.27
		min	0.0*/0.0	0.0*/0.0	0.0*/0.0	0.0*/0.0	0.29*/0.32
WB	Eucl	mean	0.97/0.0	0.32/0.0	0.11/0.0	0.19/0.0	**1.0/0.0**
		min	**0.99/0.0**	0.23/0.0	0.11/0.0	0.19/0.0	**1.0/0.0**
	Conv	mean	0.93/0.01	0.87/0.01	0.69/0.01	0.6/0.01	**1.0/0.0**
		min	**0.99/0.0**	**0.93/0.0**	**0.7/0.0**	0.61/0.0	0.6/0.08
XU	Eucl	mean	0.13/0.0	0.13/0.0	0.11/0.0	0.1/0.0	0.12/0.0
		min	0.13/0.0	0.13/0.0	0.11/0.0	0.1/0.0	**1.0/0.0**
	Conv	mean	0.93/0.01	0.87/0.01	**0.7/0.01**	0.61/0.01	**1.0/0.0**
		min	**0.99/0.0**	**0.93/0.0**	**0.7/0.0**	0.61/0.0	0.6/0.08
EL1	Eucl	mean	0.95/0.03	0.85/0.15	0.42/0.2	0.47/0.17	0.98/0.08
		min	**0.99/0.0**	**0.93/0.0**	**0.7/0.0**	0.6/0.03	0.97/0.1
	Conv	mean	0.95/0.03	0.38/0.18	0.28/0.0	0.19/0.0	0.94/0.14
		min	**0.99/0.0**	**0.93/0.0**	**0.7/0.0**	0.22/0.09	0.97/0.1
EL2	Eucl	mean	0.96/0.01	0.42/0.22	0.29/0.08	0.27/0.0	**1.0/0.0**
		min	**0.99/0.0**	**0.93/0.0**	**0.7/0.0**	**0.62/0.0**	0.88/0.18
	Conv	mean	0.79/0.28	0.32/0.0	0.28/0.0	0.27/0.0	0.98/0.07
		min	**0.99/0.0**	**0.93/0.0**	0.33/0.15	0.27/0.0	0.7/0.17
EL3	Eucl	mean	0.96/0.01	0.88/0.02	0.52/0.18	0.31/0.0	**1.0/0.0**
		min	**0.99/0.0**	**0.93/0.0**	**0.7/0.0**	**0.62/0.0**	0.99/0.07
	Conv	mean	0.96/0.01	0.38/0.0	0.34/0.0	0.31/0.0	**1.0/0.0**
		min	**0.99/0.0**	**0.93/0.0**	**0.7/0.0**	0.31/0.0	0.92/0.16

indices (E-coli, Iris, both Wisconsin breast cancer, diagnosis and prognosis), pointing to some similarity between them reflected in their definitions.

On two remaining datasets different indices win. HR index wins over the Zoo dataset with ARI=0.87 (HR also wins on Wine). Close runners-up are EL3 and EL2, in their conventional mode over averaging results post-processing with ARI values 0.85 and 0.80, respectively. The EL2 index wins over the Segmentation dataset (with three close runners-up: EL3, CH and WB, all in their conventional mode). EL2 and EL3 also win on E-coli.

One should notice that the winning modes of the inertia-based indices in Tables 5 and 6 are exactly the same as their winning modes on the synthetic datasets in Tables 2 and 3. This can be considered as an empirical support for our synthetic data generation model.

The winning patterns of the MARE criterion values in Table 6 are almost identical to those obtained for ARI. The

TABLE 8. Mean absolute relative error (MARE) number of clusters recovery results obtained for benchmark datasets (S and unbalance).

		Dataset	S ($K^* = 15$)				Unbalance ($K^* = 8$)
Ind	dist	Agg/var	1.0	2.0	3.0	4.0	
SW	-	-	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	0.28/0.05
CH	Eucl	mean	0.87/0.0	0.87/0.0	0.87/0.0	0.87/0.0	0.75/0.0
		min	0.87/0.0	0.87/0.0	0.87/0.0	0.87/0.0	0.35/0.2
	Conv	mean	0.17/0.04	0.15/0.06	0.04/0.05	0.05/0.04	0.62/0.0
		min	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	0.98/0.79
HR	Eucl	mean	0.57*/0.41	0.95*/0.19	1.0*/0.0	1.0*/0.0	1.82*/0.32
		min	1.0*/0.0	1.0*/0.0	1.0*/0.0	1.0*/0.0	2.27*/0.53
	Conv	mean	0.57*/0.41	1.0*/0.0	1.0*/0.0	1.0*/0.0	2.07*/0.41
		min	1.0*/0.0	1.0*/0.0	1.0*/0.0	1.0*/0.0	2.34*/0.55
WB	Eucl	mean	0.07/0.0	0.73/0.0	0.87/0.0	0.8/0.0	0.62/0.0
		min	0.0/0.0	0.8/0.0	0.87/0.0	0.8/0.0	0.28/0.05
	Conv	mean	0.22/0.03	0.27/0.05	0.06/0.05	0.12/0.09	0.66/0.06
		min	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	1.82/0.32
XU	Eucl	mean	0.87/0.0	0.87/0.0	0.87/0.0	0.87/0.0	0.75/0.0
		min	0.87/0.0	0.87/0.0	0.87/0.0	0.87/0.0	0.28/0.05
	Conv	mean	0.22/0.03	0.26/0.04	0.05/0.05	0.06/0.04	0.66/0.06
		min	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	1.82/0.32
EL1	Eucl	mean	0.15/0.1	0.22/0.16	0.53/0.28	0.47/0.34	0.7/0.36
		min	0.0/0.0	0.0/0.0	0.0/0.0	0.11/0.23	0.27/0.06
	Conv	mean	0.15/0.1	0.67/0.19	0.73/0.0	0.8/0.0	0.74/0.51
		min	0.0/0.0	0.0/0.0	0.0/0.0	0.79/0.05	0.27/0.06
EL2	Eucl	mean	0.12/0.04	0.63/0.23	0.71/0.13	0.73/0.0	0.62/0.0
		min	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	0.32/0.14
	Conv	mean	0.3/0.27	0.73/0.0	0.73/0.0	0.73/0.0	0.67/0.23
		min	0.0/0.0	0.0/0.0	0.64/0.25	0.73/0.0	0.18/0.11
EL3	Eucl	mean	0.08/0.03	0.18/0.1	0.36/0.31	0.67/0.0	0.65/0.05
		min	0.0/0.0	0.0/0.0	0.0/0.0	0.0/0.0	0.46/0.09
	Conv	mean	0.96/0.01	0.38/0.0	0.34/0.0	0.31/0.0	1.0/0.0
		min	0.99/0.0	0.93/0.0	0.7/0.0	0.31/0.0	0.92/0.16

TABLE 9. ARI results, followed by their standard deviations, obtained for the G2 benchmark datasets.

	$V = 8$				$V = 32$			
Index	10	50	90	100	10	50	90	100
SW	1.0/0.0	0.99/0.0	0.75/0.0	0.71/0.0	1.0/0.0	1.0/0.0	1.0/0.0	0.99/0.0
CH	1.0/0.0	0.99/0.0	0.75/0.0	0.71/0.0	1.0/0.0	1.0/0.0	1.0/0.0	0.99/0.0
HR	0.0*/0.0	0.0*/0.0	0.0*/0.0	0.0*/0.0	0.03*/0.11	0.0*/0.0	0.0*/0.0	0.01*/0.06
WB	1.0/0.0	0.99/0.0	0.75/0.0	0.71/0.0	1.0/0.0	1.0/0.0	1.0/0.0	0.99/0.0
XU	1.0/0.0	0.99/0.0	0.75/0.0	0.71/0.0	1.0/0.0	1.0/0.0	1.0/0.0	0.99/0.0
EL1	0.59/0.12	0.6/0.12	0.36/0.01	0.34/0.0	0.53/0.09	0.62/0.12	0.57/0.11	0.58/0.12
EL2	1.0/0.0	0.99/0.0	0.75/0.0	0.71/0.0	1.0/0.0	1.0/0.0	1.0/0.0	0.99/0.0
EL3	1.0/0.0	0.99/0.0	0.75/0.0	0.71/0.0	1.0/0.0	1.0/0.0	1.0/0.0	0.99/0.0

only exception is the winning pattern of the CH index: it does win on three out of four datasets, but it somewhat loses, to its mate, WB, on E-coli.

C. RESULTS FOR THE LITERATURE DATASETS

The ARI cluster recovery results and the MARE number of cluster errors for **S** and **Unbalance** are presented in Tables 7 and 8; the results for **G2** are reported in Table 9.

Let us first focus on the results obtained for the four **S** datasets. The SW index shows almost impeccable performance on all the four datasets, in terms of both ARI and MARE. The MARE results are especially good: no errors at all. Quite a few of the inertia-based indices demonstrate similar performance. We will describe them here: CH, WB, XU – all the three in the conventional mode, using the optimal values (they provide the value of ARI=0.61, which is very close to its maximum of 0.62); and E2 and E3 – both in the Euclidean distance mode.

Regarding the Unbalance datasets, we can see somewhat different winners. Here, CH, WB, and XU are still winners, but at a different style of postprocessing after multiple runs of K-means: the winning conventional mode now is averaging the results rather than picking those optimal. EL2 and EL3 remain winners using the same Euclidean distance mode, also with averaging the results.

It should be noted that the number of clusters here cannot be recovered perfectly with the indices under consideration. SW leads to the average MARE=0.28, which is matched by the WB, XU, and E1 indices. EL2 leads to even a lesser error with MARE=0.18.

For the **G2** datasets, it appears that there are no differences in the results of various approaches of index calculation, so we removed the results for the Distance and Aggregation modes from the table. Moreover, we observed two interesting properties that led us to further decrease in the size of the resulting table: (a) the errors monotonically, with no exception, follow the growth of the within-cluster variance; (b) the errors monotonically, with no exception, decrease when the space dimension grows. Because of Property (a), we report results for four within-cluster variance values (10, 50, 90, 100) only, leaving out all the intermediate variance values. Because of Property (b), we report the results for two space dimensions ($V = 8$ and 32) only, since the ARI values, already quite high for $V = 32$, can only further grow for larger space dimensions. The remainder constitutes the content of Table 9.

The results reported in Table 9 indicate that the indices under consideration work on **G2** datasets in an extremely consistent and unified way, except for the two "failing outliers", HR and EL1. The other indices remain quite steady against the unbalanced cluster sizes.

VI. CONCLUSION

This paper is devoted to a popular approach for choosing the right number of clusters K in K-means clustering. We computationally review the use of cluster validity indices based on the inertia, i.e., a square-error criterion of the conventional K-means algorithm. To make our review more investigative, we bring forth a set of novel uses of these indices. These are: (a) using the Euclidean distance rather than the squared Euclidean distance in criterion (1) with K-means; (b) summarizing a set of inertia values resulting from multiple runs of K-means by the mean, rather than the minimum (or maximum) value. Another novelty is an explicit use of a set of three Elbow indices specified by the step size.

We consider three types of datasets involving cluster intermix. Two of them involve explicit and controllable intermix parameters: the first is a generator of "synthetic" cluster structures at which the intermix is represented by a so-called squeezing parameter, and the second is based on the within-cluster dispersion. The third data type, an ensemble of relevant real-world datasets from the UCI repository has no explicit intermix parameters; moreover, the extent of association between the features and the ground-truth cluster structure is very complex. Therefore, one may hope that our conclusions are valid for a variety of practical situations.

First, one should note that the approach under analysis is not always applicable. Three out of eleven UCI repository datasets have shown no structure to recover – perhaps because features in them are not much related to the ground-truth partitions recovered by any version of K-means. Then, we can safely conclude that the mechanism generating synthetic cluster structures is rich enough to generate both easily recoverable cluster structures and those "die hard", especially with $\alpha = 0.85$. Furthermore, some conclusions, made for synthetic data, appear to hold for the real-world datasets as well.

Our most unexpected observation is that the inertia-based indices appear to work better not with the conventional way of result postprocessing, by using the best try out clustering generated by multiple runs of K-means, but by averaging the results of these multiple runs. This was true for both synthetic data and the real-world UCI repository data.

As to the elbow-based indices EL1, EL2, and EL3, they seem to work equally well with both the Euclidean distance and its squared form. Using the mean postprocessing results is beneficial for each of them too, again with a caveat that at a greater intermix, using the optimal solution is slightly better. Out of these three indices, EL2 is the best overall.

In contrast to our expectations, we cannot indicate a convincing overall winner, although SW usually leads to most balanced solutions. Still, the XU index outperforms SW on synthetic data with greater cluster intermixes, especially for a smaller space dimension ($V = 15$). SW outperforms the other competing indices on four UCI repository datasets, along with WB and CH. These two indices are also good on synthetic data with weaker intermixes; they, however, become hopeless at a greater intermix, $\alpha = 0.85$.

EL2 perhaps is the best match to SW, especially, on the real-world and literature datasets. HR, obviously, is the only candidate to refer to as the overall looser (see its very poor results reported in Tables 7 and 8), as well as the results obtained on synthetic datasets with smaller numbers of clusters. However, we cannot recommend to never use it. HR convincingly wins on Zoo dataset, as well as in a few other cases.

In summary, we believe that our results are instructive enough to the user who is looking for recovering the right number of clusters with multiple running of K-means carried out with different values of K. If in the end, SW and DB (with averaging) suggest the same number of clusters and the user's dataset demonstrates a little intermix, then the results could be accepted as is. Otherwise, the user should execute the program with the XU, HR, and EL2 indices and compare the obtained outcomes. If any two of these indices suggest the same number of clusters, then this solution could be retained as a final one.

Future work should include building more adequate synthetic data generators as well as testing these indices on some additional synthetic and real-world datasets.

Author contribution all authors contributed to the paper conception and design. Material preparation and computations were performed by Andrei Rykov. Analysis of the results by Renato Cordeiro De Amorim, Vladimir Makarenkov and Boris Mirkin. Further conceptualization: Vladimir Makarenkov. The first draft of the manuscript was written by Boris Mirkin and Andrei Rykov, and all co-authors equally participated in reviewing and editing consequent versions of the manuscript. All authors read and approved the final manuscript.

Funding This work was partially supported by Natural Sciences and Engineering Research Council of Canada [249644].

Data Availability The data sets generated during and/or analysed during the current study are available from the corresponding author upon reasonable request.

DECLARATIONS

Conflicts of interest We declare that we have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Ethics approval This article does not contain any studies with human participants or animals performed by any of the authors.

REFERENCES

[1] B. Mirkin, "Choosing the number of clusters," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 1, no. 3, pp. 252–260, May 2011.

[2] T. M. Kodinariya and P. R. Makwana, "Review on determining number of cluster in K-means clustering," *Int. J.*, vol. 1, no. 6, pp. 90–95, 2013.

[3] M. Charrad, N. Ghazzali, V. Boiteau, and A. Niknafs, "NbClust: An R package for determining the relevant number of clusters in a data set," *J. Stat. Softw.*, vol. 61, no. 6, pp. 1–36, 2014.

[4] S. Xu, X. Qiao, L. Zhu, Y. Zhang, C. Xue, and L. Li, "Reviews on determining the number of clusters," *Appl. Math. Inf. Sci.*, vol. 10, no. 4, pp. 1493–1512, Jul. 2016.

[5] H. A. Chowdhury, D. K. Bhattacharyya, and J. K. Kalita, "UIFDBC: Effective density based clustering to find clusters of arbitrary shapes without user input," *Expert Syst. Appl.*, vol. 186, Dec. 2021, Art. no. 115746.

[6] S. O. Mohammadi, A. Kalhor, and H. Bodaghi, "K-splits: Improved K-means clustering algorithm to automatically detect the number of clusters," 2021, *arXiv:2110.04660*.

[7] S. D. Nguyen, V. S. T. Nguyen, and N. T. Pham, "Determination of the optimal number of clusters: A fuzzy-set based method," *IEEE Trans. Fuzzy Syst.*, vol. 30, no. 9, pp. 3514–3526, Sep. 2022.

[8] W. Tong, S. Liu, and X.-Z. Gao, "A density-peak-based clustering algorithm of automatically determining the number of clusters," *Neurocomputing*, vol. 458, pp. 655–666, Oct. 2021.

[9] R. Vangara, K. Rasmussen, G. Chennupati, and B. S. Alexandrov, "Determination of the number of clusters by symmetric non-negative matrix factorization," *Proc. SPIE*, vol. 11730, Apr. 2021, Art. no. 117300G.

[10] Y. Yang, X. Shi, W. Liu, Q. Zhou, M. C. Lau, J. C. T. Lim, L. Sun, C. C. Y. Ng, J. Yeong, and J. Liu, "SC-MEB: Spatial clustering with hidden Markov random field using empirical Bayes," *Briefings Bioinf.*, vol. 23, no. 1, Jan. 2022, Art. no. bbab466.

[11] T. Ullmann, C. Hennig, and A. Boulesteix, "Validation of cluster analysis results on validation data: A systematic framework," *WIREs Data Mining Knowl. Discovery*, vol. 12, no. 3, p. e1444, May 2022.

[12] Y. Zhang, J. Mańdziuk, C. H. Quek, and B. W. Goh, "Curvature-based method for determining the number of clusters," *Inf. Sci.*, vols. 415–416, pp. 414–428, Nov. 2017.

[13] H. Cramer, *Mathematical Methods of Statistics*. Princeton, NJ, USA: Princeton Univ. Press, 1946.

[14] P. Fränti and J. Kivijärvi, "Randomised local search algorithm for the clustering problem," *Pattern Anal. Appl.*, vol. 3, no. 4, pp. 358–369, Dec. 2000.

[15] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *Proc. 18th Annu. ACM-SIAM Symp. Discrete Algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2007, pp. 1027–1035.

[16] B. Mirkin, *Clustering for Data Mining: A Data Recovery Approach*. London, U.K.: Chapman & Hall, 2012.

[17] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.

[18] M. M.-T. Chiang and B. Mirkin, "Intelligent choice of the number of clusters in K-means clustering: An experimental study with different cluster spreads," *J. Classification*, vol. 27, no. 1, pp. 3–40, Mar. 2010.

[19] R. C. de Amorim and B. Mirkin, "Minkowski metric, feature weighting and anomalous cluster initializing in K-means clustering," *Pattern Recognit.*, vol. 45, no. 3, pp. 1061–1075, Mar. 2012.

[20] R. C. de Amorim and V. Makarenkov, "Improving cluster recovery with feature rescaling factors," *Int. J. Speech Technol.*, vol. 51, no. 8, pp. 5759–5774, Aug. 2021.

[21] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Statist.-Simul. Comput.*, vol. 3, no. 1, pp. 1–27, 1974.

[22] J. A. Hartigan, *Clustering Algorithms*. Hoboken, NJ, USA: Wiley, 1975.

[23] L. Xu, "Bayesian Ying–Yang machine, clustering and number of clusters," *Pattern Recognit. Lett.*, vol. 18, nos. 11–13, pp. 1167–1178, 1997.

[24] Q. Zhao and P. Fränti, "WB-index: A sum-of-squares based index for cluster validity," *Data Knowl. Eng.*, vol. 92, pp. 77–89, Jul. 2014.

[25] R. L. Thorndike, "Who belongs in the family?" *Psychometrika*, vol. 18, no. 4, pp. 267–276, 1953.

[26] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *J. Comput. Appl. Math.*, vol. 20, pp. 53–65, Nov. 1987.

[27] G. W. Milligan and M. C. Cooper, "An examination of procedures for determining the number of clusters in a data set," *Psychometrika*, vol. 50, no. 2, pp. 159–179, Jun. 1985.

[28] R. C. de Amorim and C. D. L. Ruiz, "Identifying meaningful clusters in malware data," *Expert Syst. Appl.*, vol. 177, Sep. 2021, Art. no. 114971.

[29] F. Batool and C. Hennig, "Clustering with the average silhouette width," *Comput. Statist. Data Anal.*, vol. 158, Jun. 2021, Art. no. 107190.

[30] E. V. Kovaleva and B. G. Mirkin, "Bisecting K-means and 1D projection divisive clustering: A unified framework and experimental comparison," *J. Classification*, vol. 32, no. 3, pp. 414–442, Oct. 2015.

[31] D. Dua and C. Graff, "UCI machine learning repository," Tech. Rep., 2017.

[32] P. Fränti and S. Sieranoja, "K-means properties on six clustering benchmark datasets," *Int. J. Speech Technol.*, vol. 48, no. 12, pp. 4743–4759, Dec. 2018.

[33] L. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, vol. 2, no. 1, pp. 193–218, Dec. 1985.

[34] N. X. Vinh, J. Epps, and J. Bailey, "Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance," *J. Mach. Learn. Res.*, vol. 11, pp. 2837–2854, Jan. 2010.

ANDREI RYKOV received the bachelor's degree in business informatics from the Higher School of Economics, Moscow, and the M.Sc. degree in data science from the Eindhoven University of Technology (TU/e). His research interests include cluster analysis, deep learning, and the development of the approach combining features of spectral clustering and representation learning with deep learning models.

RENATO CORDEIRO DE AMORIM received the Ph.D. degree in computer science from the Birkbeck, University of London, in 2011. He is currently a Senior Lecturer in computer science and AI with the University of Essex. He has authored a number of articles introducing novel methods following the unsupervised and semi-supervised learning frameworks, with applications in fields, such as security, biosignal processing, and data science in general. He received the Chikio Hayashi Award, in 2017. His research has been funded by Microsoft, the Royal Society, and Innovate U.K. He is also an associate editor of two journals published by Springer and Elsevier.

VLADIMIR MAKARENKOV is currently a Full Professor and the Director of the Graduate Bioinformatics Program, Department of Computer Science, Université du Québec à Montréal. His research interests include bioinformatics, artificial intelligence, and mathematical classification.

BORIS MIRKIN received the Ph.D. degree in computer science (mathematics) and the D.Sc. degree in systems analysis (technology) from Russian Universities. He was one of the leaders in developing clustering and data analysis research in Russia and the USSR. In 1991 and 2011, he travelled through long-term visiting appointments in France, Germany, and USA, and a teaching appointment a Professor in computer science with the Birkbeck, University of London, U.K., from 2000 to 2010. He develops methods for clustering and interpretation of complex data within the "data recovery" perspective. Currently, these approaches are being extended to automation of text analysis problems, including the development and use of hierarchical ontologies. He has published a hundred of refereed articles and a dozen of monographs.