IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 4, 2020, accepted April 23, 2020, date of publication May 8, 2020, date of current version May 22, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2993295

Can the Number of Clusters Be Determined by External Indices?

MOHAMMAD REZAEI[1] **AND PASI FRÄNTI**[1,2]**, (Senior Member, IEEE)**
[1]School of Computing, University of Eastern Finland, 80101 Joensuu, Finland
[2]School of Big Data and Internet, Shenzhen Technology University, Shenzhen 518055, China
Corresponding author: Pasi Fränti (franti@cs.uef.fi)

ABSTRACT External indices have been used in the literature for determining the number of clusters. The idea is to measure the stability of clustering results using an external validity index when adding randomness to the clustering process. The hypothesis is that the clustering results are more stable when the correct number of clusters is used. The goal of this paper is to provide an answer to the research question stated in the paper title. We conduct a systematic study of the main components of the stability-based approach. We will discuss how to add randomness to the process, how to perform the cross-validation, and which external index to use. We will show that the number of clusters can be reliably determined only when the type of clusters is known and all the components of the approach are carefully chosen. Inferior algorithms like k-means, too high or low subsampling rate, null reference for normalization, and ineffective validation indices can all cause the stability-based approach to break. We recommend better design choices for all these components, which leads to better results compared to existing stability-based methods. However, even with the best choices, there are pathological cases where the stability-based method fails.

INDEX TERMS Clustering, cluster validation, stability, number of clusters, external index, resampling.

I. INTRODUCTION

Clustering has two separate sub-problems: determining the number of clusters and finding the clusters. Algorithms like k-means aim at solving the second sub-problem. The number of clusters must be given by the user as input. If the number is unknown or needs a recommendation from user, it must also be estimated. In this paper, we consider centroid-based clustering, which minimizes the well-known *sum of squared error* (SSE) *criterion* for a fixed number of clusters K:

$$SSE = \sum_{j=1}^{k} \sum_{i=1,\, x_i \in C_j}^{N} \|x_i - c_j\|^2 \qquad (1)$$

In other words, for a given dataset with N objects (x_i) in D-dimensional space, the clustering aims at finding the set of k centroids c_j (of clusters C_j) that minimizes the sum of squared distances from their assigned data objects. As an example, the k-means algorithm aims at minimizing SSE. In this paper, we do not question the suitability of SSE as an objective function, but study how well an algorithm like k-means performs in this task. This approach is based on the recommendation in [1] to clearly distinguish between the *clustering method* (objective function) and the *clustering algorithm* (how it is optimized). The above definition also holds for datasets which contain Gaussian clusters as long as the hyperspheres around the clusters are separable.

The number of clusters is usually estimated using *internal validity indices* by comparing the index values of different numbers of clusters [2]. Internal indices are usually based on two measures: compactness and separation. Compactness measures how similar the objects are within the same cluster, and separation measures how dissimilar the clusters are. Several sum-of-squared error indices calculate the ratio of within cluster and between cluster variances [3]. The main characteristic of these indices is that they use no prior information of the data. A number of indices have been compared in [4]–[7] but none has reached a clear state-of-the-art status that would work for a wide range of datasets.

External indices compare the clustering solution with the ground truth data [6], [8]–[10]. They can be used to study the performance of clustering methods with artificial data. External indices are also suitable for comparing two clustering

The associate editor coordinating the review of this manuscript and approving it for publication was Corrado Mencar.

solutions of the same dataset to evaluate the difference of the algorithms [11], [12], and they are utilized in ensemble clustering [12]–[14]. Some authors consider two types of indices: a relative index for comparing two clustering solutions, and an external index for comparing clustering solution with the ground truth [10], [15]. Here we consider both as external indices.

External indices have also been used for determining the number of clusters [9], [11], [16]–[20]. The idea is to generate randomness in the process by resampling the data, clustering the subsamples with a varying number of clusters, and then measuring the stability [18]. The stability is measured by calculating the similarity of the clustered subsamples using an external index. The hypothesis is that the clustering results are more stable (higher similarity) when the correct number of clusters are used.

The majority of the literature [9], [11], [17]–[22] describes the stability-based approach as being good and reliable for determining the number of clusters. However, most of the available papers do not discuss the components of the stability-based approach or its weaknesses. Experimental validation in the papers is usually given only for a very few simple datasets. One paper on this topic by Ben-David *et al.* [23] concluded that the stability-based approach is unsuitable for this problem, and they offer two counter-examples and a mathematical proof (see Section 2) as evidence. However, not even a single experimental result was provided to support the claims.

A wider viewpoint was taken in [24], where several design alternatives of the approach were considered by discussing what implications the theoretical results have in practice. The overall conclusion was that the stability approach has the potential to solve the problem, but it remains an open question as to how the method should be implemented in practice. Specifically, the role of the algorithm, the size of the subsets, and the role of the normalization were all considered as important parameters that should be compared and evaluated in practice.

Despite some promising reports, including [9], [11], [17]–[22], the stability-based approach has not been widely accepted. In fact, recent clustering literature is seriously lacking in related papers; we could find only one critical report [23]. This is therefore an open question: does the stability-based approach work? In this paper, we aim at answering this question. We study all these issues experimentally and by using selected examples. We follow the general framework given in [24], and divide the stability-based approach into the following sub-problems:

1. Adding randomness
2. Cross-validation strategy and normalization
3. Selection of the external index
4. Selection of the clustering algorithm

Randomness is usually added by subsampling. The size and the number of subsamples are the parameters. They are usually straightforward to set, except for when the size of the data is very small. Two alternative approaches are: using a randomized algorithm such as k-means with random initialization [24], and adding noise to the data [25], [26]. However, we will show that k-means itself is unstable and is therefore not reliable for this purpose. Adding noise would require additional noise parameters, which are not trivial to set and which might create unexpected artifacts.

Most external indices are restricted to comparing partitions of exactly the same data. A straightforward approach [18], [20], [27] is to compare the clustering results of a subset to that of the full set by restricting it only to the objects in the subset. Another approach predicts the partition labels of the rest of the objects by nearest neighbor mapping using cluster centroids, or by applying a more complicated classifier process [11], [21], [28], [29]. We will also consider comparing the subsets directly by using centroid index [30], which does not require the partition of the data.

The third sub-problem is selecting an external validity index. Adjusted Rand index [31], information theoretic measures [32], [33], and selected set-matching methods [30], [34], [35] are all suitable for the task. We will show through experiments that the exact choice of the measure is less important than how it is applied, which matters much more. All existing methods simply select the number of clusters that provide maximum stability (global maximum), but we will show using counter-examples and experiments that it is better to choose the last local maximum.

The last sub-problem is the selection of a clustering algorithm. K-means is commonly used but, as shown later through our experiments, it is highly unstable and not suitable for this task. Part of this originates from random initialization [1] and part from its incapability to move centroids between well-separated clusters [36]. These can be partially compensated for by better initialization and repeats, but they cannot remove the fundamental limitations of k-means [37]. A more stable algorithm such as Ward's agglomerative clustering [38], random swap [39], or genetic algorithm for clustering [40] should therefore be chosen.

The objective function (cluster model) that the algorithm should optimize is another question. If we apply the squared error criterion, but the clusters in the data are not spherical, we can get clustering results that do not fit the data. In principle, we should still find the number of clusters that is best for this model. However, the stability assumption does not always hold in this mismatched case.

In this paper, we perform a systematic study on the stability-based methods for solving the number of clusters in sum-of-square type of clustering methods. We first review the stability-based approach. We then study the design choices for every component of the method and show their limitations. We study how the choice of the clustering algorithm affects the result, and we compare the performance of several external indices. We also compare the cross-validation and classification-based approaches. We will show using counter-examples that the maximum stability is not always achieved using the correct number of clusters, and that a more

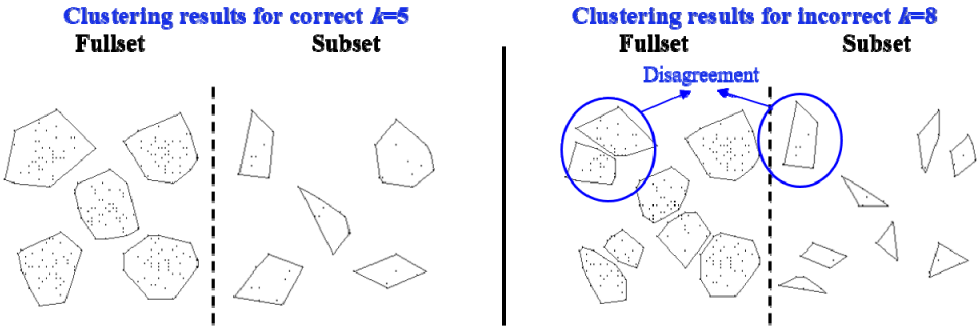

FIGURE 1. Stability-based method for finding number of clusters; stable result is achieved with correct number of clusters (left) and unstable result with incorrect number (right).

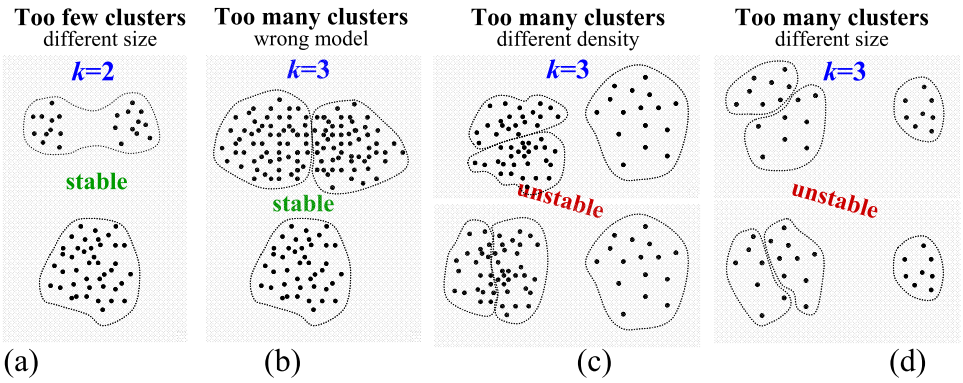

FIGURE 2. (a) Stable result is incorrectly obtained when the number of clusters is too few ($k = 2$) because the two closest clusters are always merged. (b) Stable result for $k = 3$ because of using wrong algorithm and objective function (c) (d) Unstable result is correctly obtained with too many ($k = 3$) clusters. Even though the biggest cluster is always split, the splitting happens in arbitrary direction.

robust criterion called *last local maximum* [41] should be used instead.

To sum up, we will answer whether external stability is applicable for determining the number of clusters, and if so, how it should be done exactly. We limit the study to traditional clustering that minimizes the SSE. Preliminary version of this paper appears in the PhD thesis of Rezaei [2].

II. STABILITY-BASED METHOD

The problem of finding the correct number of clusters is based on the implicit but widely used assumption that we have ground truth knowledge about the number of clusters that we want to determine. If the data do not contain well-separated clusters for a selected objective function, the method then simply provides the number of clusters for which the model best fits the data.

The stability-based approach is defined as follows: find the number of clusters k for which the clustering is stable with respect to the randomization of the process. The definition is the same as in [23], except that we do not make any implicit assumptions about the clustering algorithm employed, the subsampling strategy, or the external index for calculating the distance between the clustering solutions.

A clustering solution is defined as *stable* if it remains the same when applied for several datasets generated with the same process or from the same underlying model [24]. The similarity between every two clustering solutions is measured by an external validity index. It is expected that the most stable result would be achieved when the correct number of clusters is applied [21].

This idea is demonstrated in Fig. 1. Random swap, which is a centroid-based clustering algorithm, is applied to the dataset, which contains five clusters, and to its subset, which has the parameters $k = 5$ and $k = 8$. The clustering results of the full set and the subset are similar when $k = 5$, whereas there are disagreements when $k = 8$. In the full set, the top leftmost cluster is split, whereas in the subset it remains as one; and vice versa, the top rightmost cluster is divided in the subset.

However, stability can also be achieved with a smaller number of clusters when the positioning of clusters is not symmetrical [23]. Fig. 2 (left) shows a dataset with three well-separated clusters. Applying clustering with $k = 2$ gives a stable result because the closest clusters are always merged. Selecting the result with the maximum stability is therefore not sufficient, and a better criterion is needed.

In [23], it was further claimed that stability can also be incorrectly obtained when k is too high because the largest cluster will always be split, see Fig. 2. However, there are two reasons for this to happen that invalidate this claim. First, the split happens simply due to the use of a wrong objective function. The example has one large elliptical cluster, whereas optimizing our objective function would split it into two spherical clusters with a stable result. In other words, it can be modeled by three spherical clusters as well.

Second, it is true that the largest clusters will always be split, but the result depends not only on *which* cluster is split but also *how* the split is done. Consider the other two rightmost examples in Fig. 2 where one of the clusters is bigger either by its *size* or *density*. The way the cluster is split depends arbitrarily on the random subsampling. A proper external index should react to this and provide an unstable result. There is an exception in one-dimensional cases, where density varies but there is no spatial freedom to split the cluster differently [23]. However, it is unlikely that this kind of special case would happen in higher dimensions because there is significantly higher spatial freedom for splitting.

A. ADDING RANDOMNESS

Randomness can be created in one of the following ways:

1. Random subsampling [17], [18], [29]
2. Adding random noise [25], [26]
3. Randomizing the algorithm [24]

The most common approach is to create a number of subsets through *Monte Carlo* subsampling [15], where the size and the number of subsets are parameters. The size of subsets should not be too high (close to 100%) in order to create significant variation between the subsets. Otherwise, the clustering algorithm may always produce the same result, which always indicates stability [24]. Too low of a sampling rate, on the other hand, can break the structure in the data, as shown in Fig. 3. In the literature, choices between 20% and 80% have been considered, but no systematic comparison has been reported. At minimum, the following values have been considered: 50% [28], 80% [17], 33.3% [21], 33%, 60% and 67%, depending on the dataset [18], 50% for 2-fold, and 20% for 5-fold [9], [19].

FIGURE 3. Dataset *spirals* and its subsets with 60% (middle) and 20% (right) subsampling.

Bootstrapping was used in [42] and also mentioned in [24], but has not been studied further. It is essentially the same as random subsampling, but it allows the same object to be chosen multiple times. However, it does not remove the problem of how to select the sampling rate, which is a trade-off between having no effect on the clustering results (too high) and breaking the structure of the data (too low).

The second approach is to add noise to the data by perturbing each individual data object [25], [26]. A noise vector with random orientation is generated, but its magnitude depends on the data and is not trivial to set. In [26], the magnitude of noise is derived based on *k-nearest neighbors*. In [25], a random Gaussian noise with zero mean and fixed standard deviation 0.15 was added to the data. The standard deviation was estimated according to the median standard deviation of the log-ratios for single genes.

The third approach is to randomize the algorithm. Randomness of k-means initialization was studied extensively in [24]. It was observed that the clustering result tends to be unstable when there are too many clusters, and stable with high probability when the correct number of clusters is applied. In the case of too few clusters, both stable and unstable situations were reported, similarly to Fig. 2. However, these analyses were conducted using an algorithm called *idealized k-means,* which uses a better initialization than the standard random initialization of k-means. It was reasoned that an inconsistent clustering algorithm is completely unreliable and should not be used. We fully agree with this and our observations support it; k-means is not suitable for randomization and therefore another more stable algorithm should be used.

B. CROSS-VALIDATION STRATEGY

Depending on the randomization strategy, there are several alternatives for comparing the clustering results. If we use noise addition or randomized algorithm approaches, we can compare the full sets directly using any external index. If the subsampling approach is selected, there are some limitations on what to compare.

Subsampling produces subsets with different sets of objects. Most external indices are based on object-level operations and cannot therefore be applied directly because they require having exactly the same set of objects. It is possible to limit the comparison to the objects that are in both of the subsets. However, the danger is that too small a size of intersection may not reflect the real similarity of the subsets. With an 80% subsampling rate, we have, on average, $0.8 \times 0.8 = 0.64$ shared objects, but with a 20% rate, there are only 0.04.

The second solution is to apply a cluster-level index such as *centroid index* [30], which is independent of the data used to produce the clustering solution. It analyzes how many centroids are differently located in the two solutions. It produces a clear CI=0 value when the clustering structures are identical. It is directly applicable with any model-based clustering, and applies to all the randomization strategies discussed. Other possible alternatives would be the density profile approach [43] for density-based clustering, and the spatially-aware method [44], which uses reproducing *kernel Hilbert Space* (RKHS) to represent each cluster as a single vector. We use the centroid index because of its simplicity.

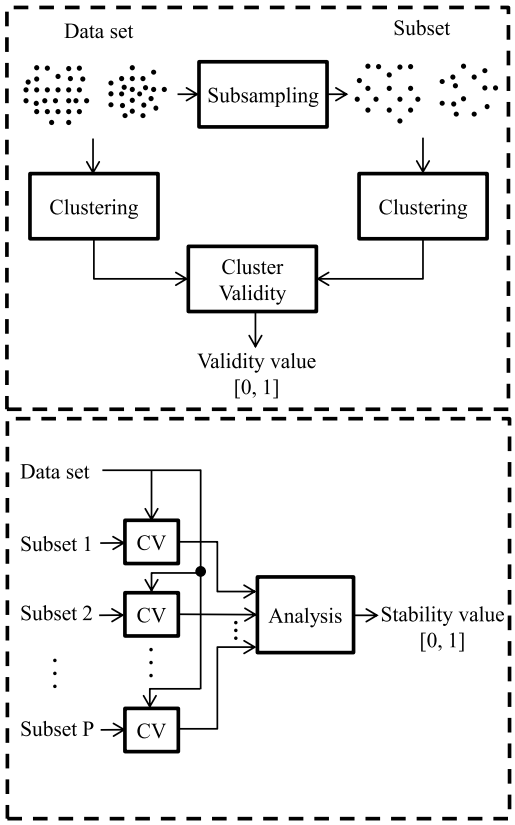

FIGURE 4. Cross-validation technique (CV); clustering solution of a full dataset is compared with the clustering solution of its subset (above). The process is repeated for a number of subsets (below).

The third solution is to predict the missing partition labels in the full set by nearest neighbor mapping based on the cluster centroids [19]. After that, the clustering solution of any subset can be compared to any another. An even simpler variant is to compare the clustering result of a subset to that of the full set by limiting the comparison to the objects that are in the subset [18], [20], [27]. This approach is the most popular in the literature, and we use it as our baseline in the rest of the paper. We refer to these approaches as *cross-validation strategy*. The baseline variant using the subsampling strategy is outlined in Fig. 4.

C. SELECTING THE NUMBER OF CLUSTERS

Most external indices return a similarity value in the range [0, 1]. We study next how we can conclude from these values that two clustering solutions are the same, and that the clustering solution for the given value k is therefore stable. In the following, we consider three approaches (two global, one local) how to select the best k:

1. Maximum stability
2. Normalized maximum stability
3. Last local maximum (proposed)

Almost all of the literature relies on the maximum stability approach, as follows. The cross-validation approach is repeated by applying clustering with all potential number of clusters $k \in [k_{\min}, k_{\max}]$. We denote the mean value of the validity index for k clusters as I_j. The maximum stability approach uses the mean values to select the number of clusters:

$$k = \arg\max_j (I_j) \tag{2}$$

In principle, the goal is to compare the underlying statistical distributions of the index values and to conclude which value of k provides the strongest evidence of stability. Statistical testing of the full distribution could be used instead of only comparing the mean values.

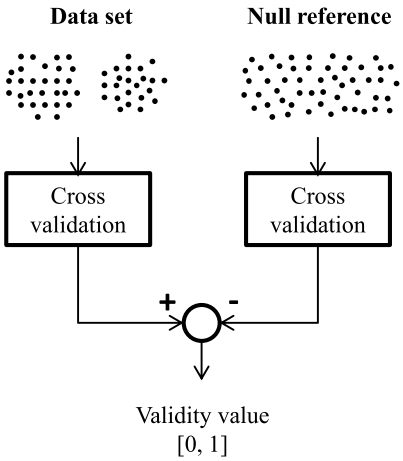

FIGURE 5. Finding the strongest evidence against the null hypothesis by evaluating the difference between the mean index values for a dataset and its null reference.

The normalized maximum stability approach (see Fig. 5) selects the number of clusters as the maximum difference in the mean stability values of the data (I) and the corresponding value (I_0) of the null reference, which is a random dataset drawn from the original data (this is discussed in more detail in section 2.4) [11, 20]:

$$k = \arg\max_j (I_j - I_j^0) \tag{3}$$

This approach is referred to as normalization with regard to the number of clusters [24]. The reason for this is that the stability value depends on k, regardless of the underlying data structure. For example, the stability of clustering results for a random uniform dataset decreases as the number of clusters increases. This bias should be removed, and then the same equation (2) should be used. This approach is also used in the gap statistics index [41].

We consider next a new alternative, which involves selecting the *last local maximum* of the index value. The only existing method which uses this approach was proposed in [22]. The idea arose from the examples in Fig. 2 and the observation made in [24] that stability can also be observed when there are too few clusters, but rarely when there are too many clusters. Thus, instead of finding which value of k provides *highest* stability, the method now tries to determine the

highest value of k that provides a stable clustering solution:

$$k = \arg\max_j (I_j > I_{th} \text{ and } I_{j-1} < I_j > I_{j+1}) \quad (4)$$

For this, a threshold (I_{th}) is needed to decide the point at which an index value is considered stable. Using the centroid index [30], we can interpret CI=0 as stable, and CI>0 as unstable. For all other indices, we set the threshold value at 0.9 throughout this paper. All indices are normalized to [0, 1]; therefore, considering a constant threshold is reasonable. This selection seems robust for the datasets used in this paper, but the downside is that it adds a new control parameter to the process.

D. NULL HYPOTHESIS

We study next the theoretical background of normalization to better understand why it has been considered. Originally, *null hypothesis* H_0 assumes that the data is random and there are no clusters: $k = 1$ [8], [11], [15]. Acceptance or rejection of this hypothesis is based on statistical inference. The alternative hypothesis H_1 assumes a specific structure in the dataset, for example, $k = 3$.

In the stability-based method, H_1 corresponds to X, and H_0 corresponds to a *null reference* X_0, which is a dataset with the same parameters and dimension as X, but its points are randomly sampled from a uniform or normal distribution [11]; see Fig. 6. The null reference based on uniform distribution is generated by randomly sampling objects in the range of the attributes of the original dataset. Sometimes, only the relationships between the data objects are available by a similarity matrix. In this case, the null reference is produced by randomly generating the matrix with the values in the range of minimum and maximum similarity values between the objects in the original dataset [45].

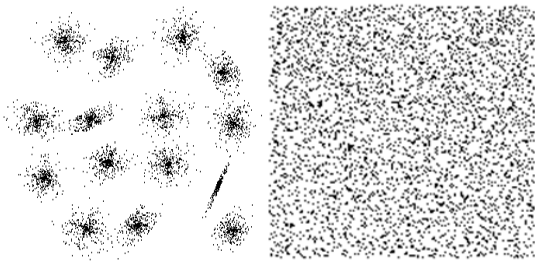

FIGURE 6. Dataset *S1* (left), and the corresponding null reference (right).

Cross-validation is separately performed for X and its null reference X_0 using a large number of repeats. This results in two probability density functions (PDF) of the index I, corresponding to H_1 and H_0. These are considered to be random variables; see Fig. 7 for a theoretical example. The goal is to analyze whether there is statistically significant evidence that the two distributions are different.

A practical example is shown in Fig. 8. We generated a uniform null reference for the dataset in Fig. 1, and produced 100 subsets with the sampling rate of 20%. The histograms

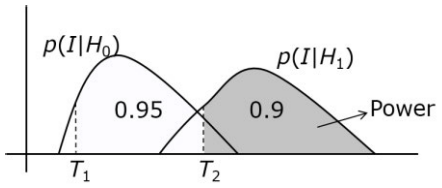

FIGURE 7. Hypothesis testing; PDF of H_0 corresponds to X_0 and PDF of H_1 corresponds to X.

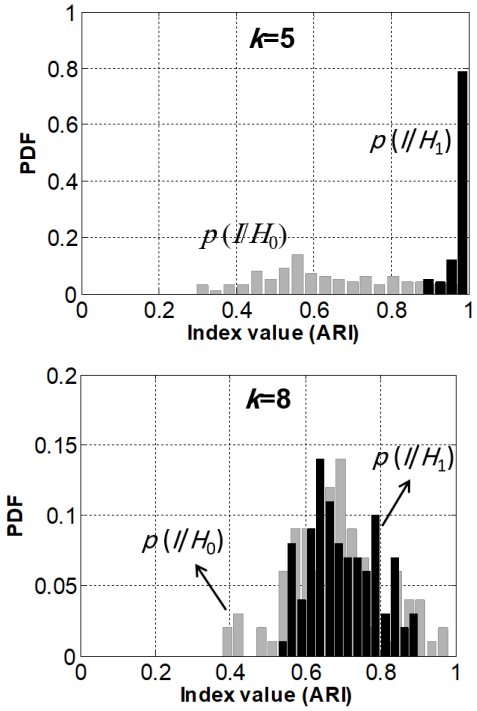

FIGURE 8. Null hypothesis testing for the dataset in Fig. 1 for $k = 5$ and $k = 8$. The PDF of the dataset differs significantly (with respect to ARI) from the null hypothesis only with the correct k value (below).

are the cross-validation values I both for the data (black) and its null reference (gray). In the case of $k = 5$ (left), there is a clear distinction between the two histograms, whereas with $k = 8$ there is no significant difference between the histograms.

Statistical analysis can now be performed to figure out in which k the PDF of H_1 provides the strongest significant evidence against H_0 [11]. The basic approach is to first select a significance level (e.g. 5%), and then find values T_1 and T_2 so that 5% of the points in the distribution have values smaller than T_1, and 5% are larger than T_2, respectively. The number of datasets X for which $I > T_2$ are counted as p_1, where the total number of them is P. H_1 is accepted if p_1/P is larger than a threshold, for example 0.9.

E. CLASSIFICATION-BASED APPROACH

The ideas from supervised learning have also been used to evaluate the stability of clustering results in terms of their reproducibility [11]. The data, in P independent iterations,

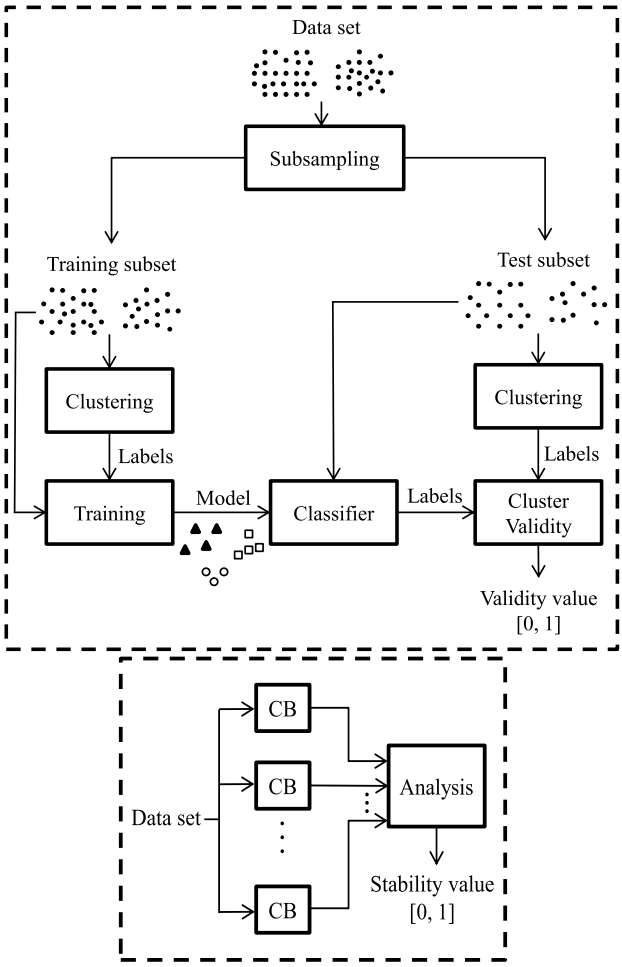

FIGURE 9. One iteration of the classification-based (CB) approach (above). The process is repeated by randomly generating several training and test sets (below).

is randomly divided into two disjoint sets, a training set X_i^{tr} and a test set X_i^{te}, where $i \in [1, P]$. Clustering is applied to both X_i^{tr} and X_i^{te} to produce partitions Y_i^{tr} and Y_i^{te}. Another partition $Y_i^{'te}$ is then predicted for X_i^{te} using a classifier trained on (X_i^{tr}, Y_i^{tr}) [11], [16]. The two partitions for the test set are compared using an external index, see Fig. 9. The P index values corresponding to P test sets are then averaged. The process is repeated for the potential numbers of clusters in the range $k \in [k_{\min}, k_{\max}]$. The hypothesis is that the highest stability (the average similarity between the two partitions of the test set) is achieved for the correct number of clusters.

To derive the labels for the test dataset from the clustering solution of training dataset, a classifier, such as linear discriminant analysis or k-nearest neighbor, is used for training [11], [19]. Selection of a good classifier is a challenging problem. In theory, the classifier is never optimal. A classifier can be derived based on the clustering process that has been used, which leads to a smaller error than that of a general classifier.

For example, the nearest neighbor classifier is suitable for single-link clustering, and the nearest centroid classifier is suitable for centroid-based clustering algorithms such as k-means [19]. In the case of model-based clustering, the labels can be directly determined from the model obtained in the training process without any classifier [16].

The size of training and test sets should be selected carefully. In classification, a larger portion of data is usually considered for training. However, in the current problem, considering more data for training might be problematic. For example, in density-based clustering, different sizes of test and training sets result in different densities, which might lead to different clustering solutions (This will be discussed more in the next section). In this case, training and test sets should have the same size [9].

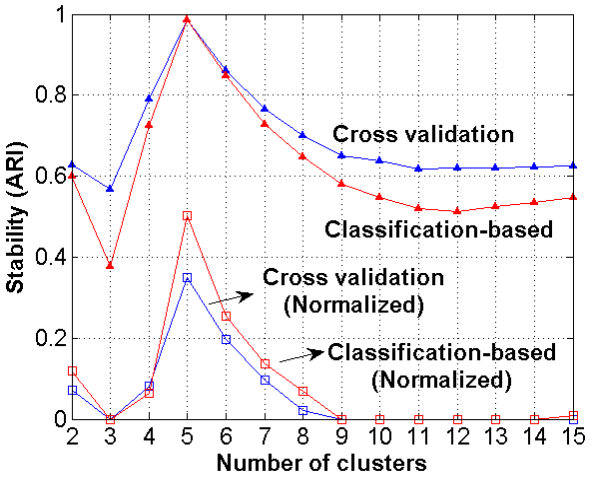

FIGURE 10. Example of the stability-based method for the dataset in Fig. 1. The size of subsets in the cross-validation approach and test sets in the classification-based approach is 20%. Random swap algorithm is used for clustering [39] and adjusted Rand index (ARI) for validation [31].

The normalization based on null reference as in (3) can also be used for the classification-based approach. Fig. 10 shows the results of cross-validation and classification-based approaches with and without normalization for the dataset in Fig. 1. The number of subsets is 100, each 20% of the size of the full dataset in the cross-validation approach. The percentages for the training and test sets in the classification-based approach are 80% and 20%, respectively. Random swap clustering algorithm [39] and *adjusted Rand index* (ARI) [31] are used. The highest stability is found with $k = 5$, the correct number of clusters.

III. CLUSTERING AND VALIDATION

A. CLUSTERING ALGORITHM

A clustering algorithm should have two basic requirements to be suitable for the stability-based method. First, the algorithm itself should be stable so that it provides the same result when applied several times to the same data or to several datasets drawn from the same source [19]. Otherwise, one cannot conclude whether the instability is caused by the artifacts of the clustering algorithm or by the structure of the data.

Second, the objective function that the algorithm uses must be suitable for the dataset so that it is able to find a

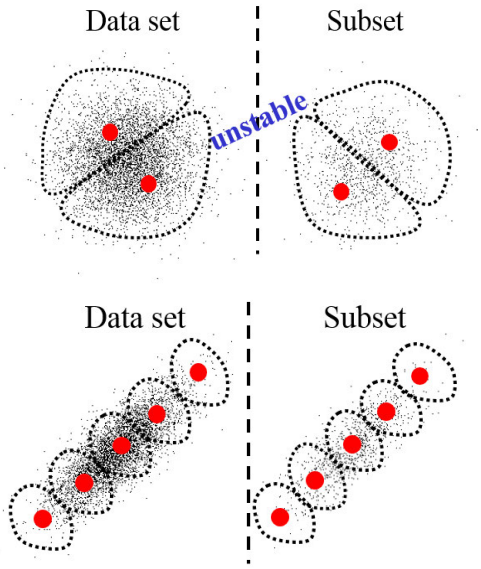

FIGURE 11. Datasets without structure; unstable clustering solution for a spherical 2-D dataset (above), and stable clustering solution for a skewed dataset (below) for $k = 5$.

good solution for the correct number of clusters. Otherwise, it may provide different clusters than what the data contain. For example, Fig. 11 (right) has one Gaussian cluster, but k-means would detect five spherical clusters instead.

Existing clustering methods can be mainly classified into four categories: Centroid-based, Distribution-based, Density-based, and Connectivity-based. Most stability-based approaches use k-means, which is the most well-known centroid-based algorithm aiming at minimizing total squared error (TSE or SSE) [46]. The result of the k-means algorithm strongly depends on the initial choice of the centroids, and it often terminates into a local minimum [37]. K-means is therefore unstable and not suitable for the problem.

The unsuitability of k-means was also recognized in [6], [23], [24], but very little attention was paid to how to resolve the issue. In [23], it is simply assumed that a perfect clustering algorithm is used and that it is merely a question of available computational power. In [24], two alternatives were studied: ideal k-means and actual k-means. The ideal k-means is also based on the same assumption, which is that we have a perfect algorithm. The actual k-means is not the original k-means, but rather an improved variant that uses a better initialization. This modification makes the algorithm less dependent on the initialization and more stable, but the algorithm is still not perfect.

Random swap (RS) [39] is a more stable, centroid-based algorithm that iteratively changes the centroid locations through a trial-and-error manner. Due to its ease of implementation and stable performance, we use it as our baseline throughout the rest of this paper. Another possible algorithm for spherical clusters is agglomerative clustering (AC) with proper merging criterion, such as Ward [38]. Efficient implementation in [48] also minimizes TSE and it usually finds the correct clustering structure, but with minor inaccuracies near the cluster borders. This would provide another suitable compromise of simplicity and stability. The best reported results have been obtained using genetic algorithm (GA). The variant in [40] uses agglomerative clustering as genetic operations and k-means for fine-tuning the results. Existing stability-based methods for spherical clusters have used partitioning around medoids (PAM) [11], repeated k-means [5], [29], competitive learning [5], bisecting k-means [10], and average-link agglomerative clustering [17].

Distribution-based clustering assumes that the objects in each cluster belong to the same probability distribution. The most popular algorithm is *expectation maximization* (EM) [49], which is analogous to k-means. Clusters are modeled by a fixed number of Gaussian mixture models. The algorithm iteratively improves the solution through a two-step process. It was used in the stability-based method in [28]. The problem with EM is that it also depends on the initial configuration. Better algorithms include *split and merge* EM (SMEM) [50], genetic algorithm EM [51], and random swap EM (RSEM) [52], which all aim at overcoming the problem of local optima of EM.

Density-based clustering considers the clusters as more dense areas than the rest of the data. Sparse objects are usually considered as noise. DBSCAN [53] is one of the most popular density-based algorithms. Its basic idea is to create clusters from points whose neighborhood within a given radius (*eps*) contains a minimum number (*minPt*) of other points. The algorithm grows clusters from these points by joining neighboring points within the *eps* distance. The algorithm is simple, but the result strongly depends on its parameters. The number of clusters is automatically derived based on these parameters, and the algorithm performs poorly when there are clusters with different densities in the data. OPTICS [54] generalizes DBSCAN so that the parameter *eps* is derived automatically. There are two main problems with these algorithms. First, they select the number of clusters k indirectly via the input parameters. Second, resampling the subsets will produce different densities than in the original data, and therefore would need different parameters for *eps* and *minPt*. These make it difficult to use the density-based algorithms in the stability-based methods.

Connectivity-based clustering aims at forming a few arbitrary-shaped clusters by connecting nearby objects based on their distance. Agglomerative clustering with single-link and complete-link are two examples of connectivity-based clustering. Single-link would work only if the clusters were well-separated, but gives poor results otherwise. Numerous other algorithms appear in the literature, but it is unclear which one of them would actually work in practice.

In clustering, the main problem is that we usually do not know what kinds of clusters are expected. We can still apply total squared error criterion even if the clusters are not spherical, and find the clustering solution that best fits to the assumed model. It is expected that we will find the number of clusters that best fits the selected model for this data.

Fig. 11 shows an example where applying incorrect cluster models results in a stability with higher number of clusters than what is intuitive. The spherical cluster (left) is stable only if $k = 1$ but becomes unstable if $k = 2$. However, the Gaussian cluster (right) will be stable for $k = 5$, indicating that this is the most natural number of spherical clusters in this data. The same results are provided by most internal indices in [3], according to our tests. In [23], the cause is said to be *asymmetric* structure. We say the cause is that different cluster models are used than what the data would require.

B. EXTERNAL VALIDITY INDEX

External indices are categorized into *pair-counting*, *information theoretic,* and *set-matching* measures [35]. Normalization and correction for chance are desirable properties. Normalization keeps the range of the index either in [-1, 1] or [0, 1], which makes the values comparable across different datasets. Correction for chance adjusts the expected value to zero under normal distribution when random partitioning is applied to the dataset [35].

Pair-counting measures include *Rand index*, *adjusted Rand index*, *Jaccard coefficient, Fowlkes-Mallows index,* and several others [55], [56]. They count the pairs of objects in the dataset on which two different partitions agree or disagree. For instance, if two objects are located in the same cluster, or in different clusters in the two clustering solutions, it is an agreement. Rand index is defined as the number of agreements divided by the total number of pairs of objects. Adjusted Rand index is the corrected form of Rand index [57] for chance [31]:

$$ARI = \frac{RI - E(RI)}{1 - E(RI)} \tag{5}$$

where E(RI) is the expected value of Rand index. Adjusted Rand index is the most popular index in this group.

Information theoretic indices include *entropy*, *mutual information,* and *variation of information* [32], [33], [56]. Mutual information (MI) measures the information that two clustering solutions share by summing up the shared information between every two clusters:

$$MP(P_i, G_j) = \sum_{i=1}^{k}\sum_{j=1}^{k'} p(P_i, G_j) \log \frac{p(P_i, G_j)}{p(P_i)p(G_j)} \tag{6}$$

where $p(P_i)$, $p(G_j)$, and $p(P_i, G_j)$ are estimated as n_i/N, n_j/N, and n_{ij}/N, respectively. N is the size of dataset, P and G are two clustering solutions of the dataset, n_i and n_j are the sizes of clusters P_i and G_j, and n_{ij} is the number of shared objects between two clusters. Variation of information (VI) represents the distance between two clustering solutions, and it is the complement of mutual information. Since there is no upper bound for mutual information and variation of information, normalization is needed [13]. It is shown in [35] that NMI is equal to the similarity variant of adjusted variation of information (AVI_S), the similarity variant of normalized variation of information (NMI_S), and also adjusted mutual information:

$$AVI_s = NVI_s = AMI = NMI \tag{7}$$

Set-matching indices are based on matching the clusters in two clustering solutions, where the similarity between every two clusters is calculated according to a given measure. We classify set-matching indices into two types: point-level, such as *Van Dongen* [34] and *pair set index* (PSI) [35], and cluster-level, such as *centroid index* [30]. Cluster-level indices use only cluster prototypes in contrast to point-level indices, which employ the labels of all objects in the resulting partitions. PSI is defined as follows:

$$PSI = \begin{cases} \dfrac{S - E(S)}{\max(k, k') - E(S)} & S \geq E(S) \\ & \max(k, k') > 1 \\ 0 & S < E(S) \\ 1 & k = k' = 1 \end{cases}$$
$$S = \sum_{i=1}^{\min(k,k')} \frac{n_{ij}}{\max(n_i, n_j)} \tag{8}$$

where i, j are the indices of paired clusters, and $E(S)$ is the expected similarity value when random partitioning is applied.

Centroid index finds for every centroid in clustering solution P its nearest neighbor in clustering solution G. It then calculates the number of times each centroid in G was chosen as nearest, and then sums up the number of centroids that were not chosen at all (*indegree*=0). These are called orphans:

$$CI_1(p \rightarrow G) = \sum_{i=1}^{k'} orphan(G_i)$$
$$CI_2(P, G) = \max\{CI_1(P, G), CI_1(G, P)\} \tag{9}$$

Since the mapping from P to G is not symmetric, CI_2 is defined by calculating the CI_1 measure in both ways.

Existing stability-based methods either define their own index or employ a well-known external index such as Rand [42], Fowlkes and Mallows (FM) [11], [29], or ARI [20] to measure the stability. However, the indices are all similar to the existing external indices. For example, the indices in [9] and [19] are set-matching-based indices, corrected for chance. Optimal pairing for two partitions is derived and then the number of misclassified objects is calculated. In [24], minimal matching distance is used, which is a set-matching-based measure that assumes optimal pairing. The figure of merit [18] is a pair-counting external index, which counts the number of pairs of objects located in the same cluster in both clusterings. Prediction strength [22] is similar to the figure of merit, but the stability is measured only according to the cluster in the test set that has the minimum proportion of the pairs of objects. In [23], a measure called *Hamming distance* is used, which is also a variant of pair counting distance.

TABLE 1. Summary of the datasets from http://cs.uef.fi/sipu/datasets.

Dataset	N	k	d
S1- S4	5000	15	2
Iris	150	3	4
Unbalance	6500	8	2
Bridge	4096	1	16
Birch1	100,000	100	2
Birch2	100,000	100	2
G2-32d	2048	2	32
Unbalance2	6500	8	2
Overlap	1000	6	2
Asymmetric	1000	5	2
Skewed	1000	6	2

IV. EXPERIMENTS

We arrange a set of experiments to answer the following questions:

1. Which external index should be used, and how should k be selected?
2. What is the effect of the sampling rate?
3. Which cross-validation strategy is best?
4. Should null reference be used or not?
5. Which clustering algorithm is best?

For the experiments, we use mostly controlled artificial datasets with known ground truth. In this way, we have no ambiguity about the structure and number of clusters or which clustering method would be suitable. The problem with real datasets is that they usually lack a natural clustering structure and therefore one cannot draw reliable conclusions about the methods for determining the number of clusters.

The programs for generating the subsets and performing the clustering were implemented by C programming, and the external indexes and stability measurements by MATLAB.

A. EXPERIMENTAL SETUP

1) EXTERNAL INDICES

We consider representative indices from the three categories (see Section III.B) of external indices: RI, ARI, NMI, CI, CSI, NVD, and PSI. All the indices are traditional point-level indices normalized in the range [0, 1], except for CI, which is a cluster-level index in the range [0, k]. For visualization purposes, we normalize CI and convert it to a similarity measure in the range [0, 1] using CI$*$ =1-CI/k. NVD is also a distance measure. We consider 1-NVD as a similarity measure in all the experiments. We use PSI as default index in the experiments.

2) CLUSTERING ALGORITHMS

By default, we use the random swap (RS) algorithm [39], [58] unless otherwise noted. We set the number of its iterations to 5000 to make sure that the best possible clustering solution is achieved. To evaluate the impact of the algorithm,

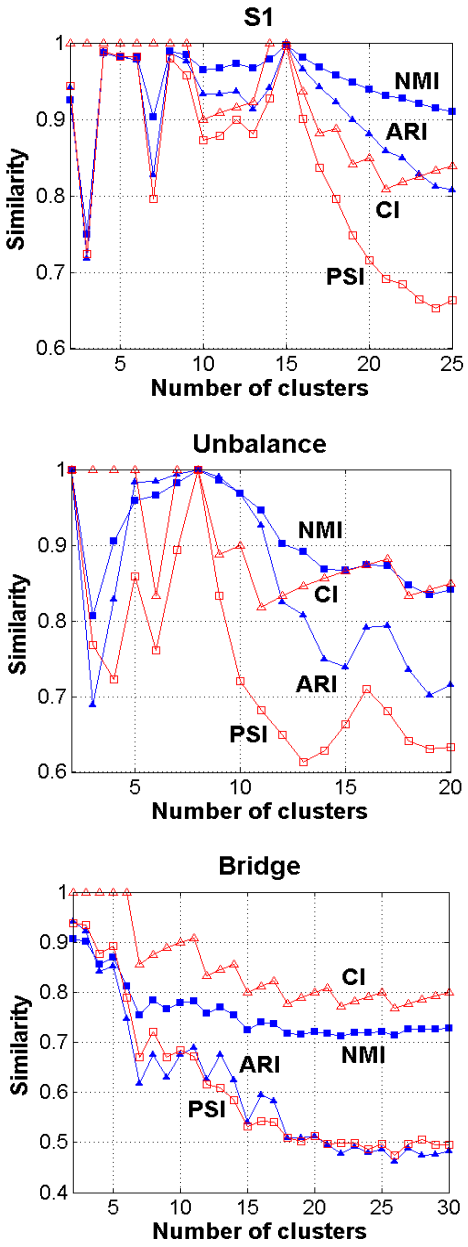

FIGURE 12. Stability results of cross-validation approach using various indices.

we also consider k-means (KM) because of its popularity and genetic algorithm (GA) because of its high clustering accuracy. In GA, we have a population size of 20, which is initialized randomly. Other parameters include the number of generations (10), the number of k-means iterations for fine-tuning (2), no mutations, and using the crossover method proposed in [40]. This setting provides near optimal solutions for all the datasets in our experiments.

3) DATASETS

Table 1 summarizes the 14 datasets used in our tests [36]. They have varying cluster overlap, dimensionality, and structure. *S1*-*S4* are 2-*d* datasets including spherical and Gaussian clusters with varying overlap. The *Overlap* dataset,

TABLE 2. Comparison of external indices when considering the global maximum and the last local maximum. The numbers shown are the resulting number of clusters (k). The emphasis (red color) shows when the index provides a wrong result.

	Datasets						
	Birch1 100	Birch2 100	Unbalance2 8	Overlap 6	Asymmetric 5	Skewed 6	G2-32d 2
	Global maximum						
RI	99..105	100	8	5	2	2	2
ARI	100	100	8	5	2	2	2
NMI	100	100	8	5	2	2	2
PSI	100	100	3	5	2	2	2
NVD	100	100	8	2	2	2	2
CSI	100	100	8	2	2	2	2
CI	100	92, 100	2	2	2	2	2
	Last local maximum						
RI	100	100	19	5	18	15	2
ARI	100	100	8	5	7	5	2
NMI	100	100	8	5	5	4	2
PSI	100	100	8	5	5	5	2
NVD	100	100	8	5	5	6	2
CSI	100	100	8	5	5	6	2
CI	100	100	11	6	13	15	2

	Datasets						
	S1 15	S2 15	S3 15	S4 15	Iris 2	Unbalance 8	Bridge 2
	Global maximum						
RI	15	2, 15	15	15	2	2, 8	2
ARI	15	2	4	2	2	2, 8	2
NMI	15	2	15	15	2	2, 8	2
PSI	15	2	4	2	2	2, 8	2
NVD	15	2	3, 4	2	2	2, 8	2
CSI	15	2	3, 4	2	2	2, 8	2
CI	2..9,14,15	2..15	2..10, 14, 15	2..10, 12..15	2,3,4	2, 3, 4,5,7,8	2..6
	Last local maximum						
RI	15	15	23	19	2	17	27
ARI	15	15	15	15	2	8	2
NMI	15	15	15	15	2	8	2
PSI	15	15	15	15	2	8	2
NVD	15	15	15	15	2	8	5
CSI	15	15	15	15	2	8	5
CI	15	15	15	15	4	8	6

however, contains more overlapping clusters, and it is more challenging. The *Unbalance* dataset has three big clusters, each with a size of 1000 and five small clusters, each with a size of 100. *Unbalance2* is similar to *Unbalance,* but is more challenging, where the five small clusters are close to the big clusters. *Iris* is a small dataset with three clusters. Although *Iris* consists of three classes, the data is distributed so that two spherical clusters would be detected. Thus, $k = 2$ is considered as the correct result. The *Bridge* dataset has 4×4 non-overlapping vectors taken from a 256×256 gray-scale image without visible clusters. Since the selected stability-based methods are used to find $k >= 2$ clusters,

TABLE 3. Comparison of two cross-validation strategies vs. classification-based approach vs. randomized algorithm (R.A.) with different number of iterations

	Datasets						
	Birch1	Birch2	Unbalance2	Overlap	Asymmetric	Skewed	G2-32d
	100	100	8	6	5	6	2
Cross-valid. (FULL)	100	100	8	5	5	5	2
Cross-valid. (SUB)	100	100	8	5	5	3	2
Classification-based	100	100	8	5	5	5	2
R.A. (1)	1	1	10	18	19	19	2
R.A. (10)	1	1	3	17	19	17	2
R.A. (100)	98	1	8	15	17	18	2
R.A. (1000)	100	100	14	16	19	19	2
R.A. (5000)	100	109	19	17	17	19	2

	Datasets						
	S1	S2	S3	S4	Iris	Unbalance	Bridge
	15	15	15	15	2	8	2
Cross-valid. (FULL)	15	15	15	15	2	8	2
Cross-valid. (SUB)	15	15	15	15	2	8	2
Classification-based	15	15	15	15	2	8	2
R.A. (1)	1	4	4	2	3	2	5
R.A. (10)	5	16	10	2	3	4	10
R.A. (100)	16	16	15	19	5	17	9
R.A. (1000)	19	16	22	22	6	17	10
R.A. (5000)	19	16	24	24	6	17	8

the smallest possible result, $k = 2$, is considered as the correct result for the Bridge dataset. *Birch*1 and *Birch*2 include 100 well-separated spherical clusters. The *G*2-32*d* dataset has the dimensionality 32, and contains two clusters (1024 points each). *Asymmetric* includes five clusters, where two groups of two clusters and three clusters provide an asymmetric structure, which can be challenging for the stability-based approach. Dataset *Skewed* has six oblong clusters.

4) SUBSAMPLING

We generate 100 subsets from each dataset by random independent subsampling. The same subsets are used both in the cross-validation and as test sets in the classification-based approach in all experiments. The rest of the data (the complement of each subset) are used as a training set in the classification-based approach. We similarly generate 100 subsets as the uniform null references of datasets. We consider the sampling rates 5%, 10%, 20%, 40%, and 80%. By default, we use 20%.

B. COMPARISON OF EXTERNAL INDICES

We compare the performance of external indices using the subsampling-based cross-validation approach. We consider both the global maximum and the last local maximum approaches (with threshold 0.90).

Fig. 12 shows the average index values for selected datasets, and Table 2 records the number of clusters detected. The first observation is that the choice of index for the last local maximum approach is not very important. Almost all the indices manage to find the correct result for most datasets. The only exception is RI, which fails in 57% of cases.

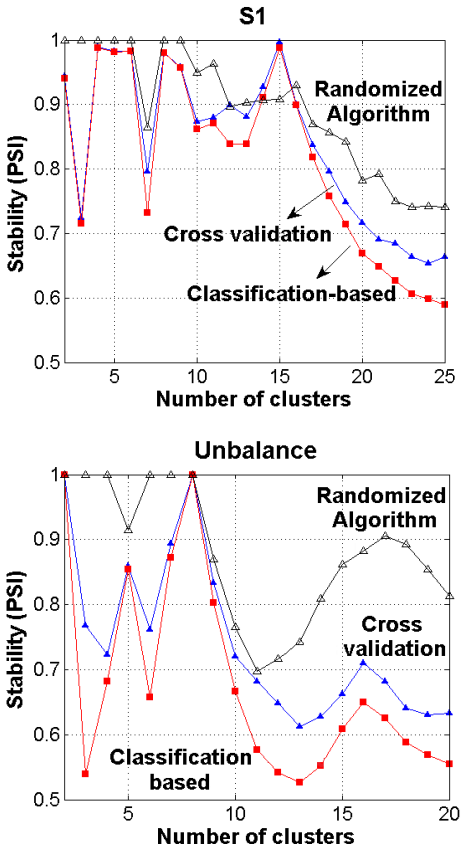

FIGURE 13. Comparison of three approaches: cross-validation, classification-based, and randomized algorithm.

The scale of CI is much rougher than the other indices. It always finds maximum stability for the correct number

TABLE 4. Cross-validation approach without and with null reference: Global = global maximum, Local = last local maximum.

	Datasets						
	Birch1	Birch2	Unbalance2	Overlap	Asymmetric	Skewed	G2-32d
	100	100	8	6	5	6	2
Without	100	100	8	5	5	5	2
Local	100	108	16	1	5	5	2
Global	100	100	2	3	5	3	2

	Datasets						
	S1	S2	S3	S4	Iris	Unbalance	Bridge
	15	15	15	15	2	8	2
Without	15	15	15	15	2	8	2
Local	14	14	7	14	3	8	28
Global	6	7	3	14	3	8	3

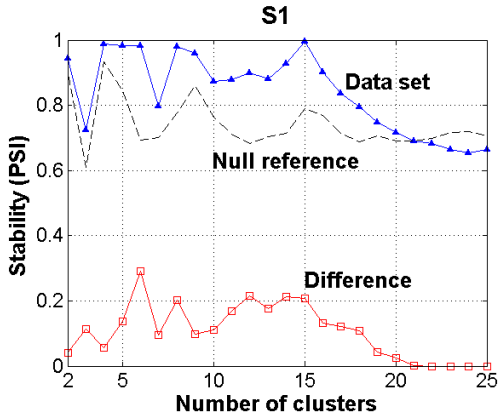

FIGURE 14. Comparing cross-validation approach without and with normalization using null reference.

of clusters, but also several others when there are too few clusters.

The second observation is that the global maximum criterion fails in many cases. It either detects multiple global maxima (especially with CI), or detects a solution with too few clusters. The last local maximum criterion works better in this sense. Most of the indices fail to find the correct number of clusters for the *Overlap* and *Skewed* datasets. The problem is that the employed clustering method cannot find the clusters correctly both for original datasets and their subsets. The reason is that there is too much overlap between clusters for the *Overlap* dataset, and the wrong clustering method is employed for the *Skewed* dataset. The problem with asymmetric datasets (*Asymmetric* and *Unbalance*), as discussed in Section 2, has been solved by employing the last local maximum criteria.

C. CROSS-VALIDATION STRATEGY

We next compare two cross-validation strategies with the classification-based approach. The results for two selected datasets are plotted in Fig. 13. They show the same trend for both cross-validation (subset-fullset) and classification-based approaches with only slight differences.

To compare the clustering results of two subsets, we predict the labels of their full dataset using nearest centroid mapping. We then compare the resulting clustering solutions of the full dataset. Table 3 shows that there is no difference in the performance of the two cross-validation strategies and the classification-based approach when using the last local maximum criterion. The global maximum criterion results in the same errors as in the previous experiment.

We also tested randomization of the algorithm. Instead of k-means, we use random swap, which is a more robust algorithm. We study the level of randomness by setting 1, 10, 100, 1000, and 5000 iterations. The correct clustering solution is found for a few datasets, but with a different number of iterations: 10 (*G2-32d*), 100 (*S3*), or 1000 (*Birch1* and *Birch2*). Fewer iterations would cause more randomness, which potentially allows for detection of the number of clusters via stability. The results for the datasets *S1* and *Unbalance* (100 iterations) in Fig. 13 show the low performance of this approach. Both the global maximum and the last local maximum result in an incorrect number of clusters.

The results of the randomized algorithm in Table 3 show that it rarely works. Correct results are found only for *G2-32d*, *Birch1*, and *Birch2*, but only if the number of iterations is set properly as being slightly less than what would be required to find the optimum solution. Too few iterations cause too much randomness, and stability will not be achieved even with the correct number of clusters. Too many iterations, on the other hand, allow the algorithm to find the same well-optimized clustering solution regardless of the initialization. Even with too many clusters, there is usually a unique global minimum that the algorithm finds. The fundamental problem with this approach is that the randomization caused

TABLE 5. Comparison of clustering algorithms by using PSI

Clustering algorithm	Datasets						
	Birch1	Birch2	Unbalance2	Overlap	Asymmetric	Skewed	G2-32d
	100	100	8	6	5	6	2
RS	100	100	8	5	5	5	2
GA	100	100	8	5	5	5	2
KM	109	1	9	5	2	3	2

Clustering algorithm	Datasets						
	S1	S2	S3	S4	Iris	Unbalance	Bridge
	15	15	15	15	2	8	2
RS	15	15	15	15	2	8	2
GA	15	15	15	15	2	8	2
KM	16	18	15	16	2	4	2

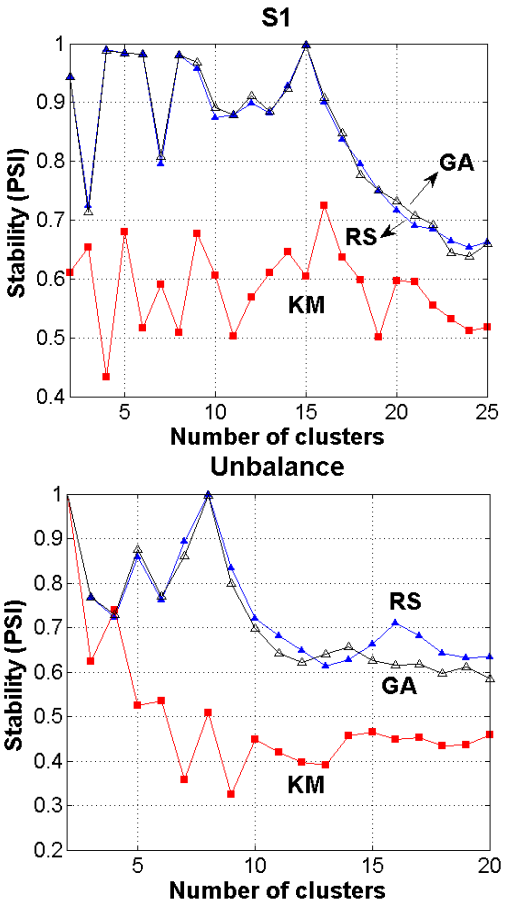

FIGURE 15. Comparison of k-means with two good algorithms: random swap and genetic algorithm.

by an algorithm creates less predictable artifacts than the subsampling approach.

D. NULL REFERENCE

We next test the normalization based on the null reference as used in (3). We report the results both using the last local maximum and the global maximum criteria with the threshold 0.2. The results in Fig. 14 reveal that the stability value, when using the null reference, does not monotonically decrease as expected, but it fluctuates; only the magnitude of the fluctuation decreases with the number of clusters. The overall results (the difference) are negatively affected by the fluctuation and lead to more confusion about the optimal number of clusters for 10 datasets, as shown in Table 4.

E. CLUSTERING ALGORITHM

In this experiment, we compare the performance of three clustering algorithms:

1. Random swap (RS)
2. Genetic algorithm (GA)
3. K-means (KM).

K-means [59] is by far the most popular algorithm used in the literature, and is basically the only one previously considered for the stability-based approach in literature. In addition, we select two algorithms that are proven to be excellent in minimizing SSE. Genetic algorithm [40] has consistently outperformed other algorithms [30] and serves as the state-of-the-art. Random swap serves as a compromise between quality and simplicity: it is almost as easy to implement as k-means, but it provides virtually the same clustering result as GA if iterated long enough.

The results for the datasets $S1$ and *Unbalance* in Fig. 15 show that k-means results in lower stability values, which originates from the instability of the algorithm. This shows that the problem of evaluating the stability related to the structure of the data is mixed with the instability of the clustering algorithm. Therefore, wrong conclusions may be derived due to the choice of a bad algorithm.

To determine the number of clusters, we use the last local maximum criterion with the threshold 0.9. The difference between RS and GA (see Fig. 15) is so small that we expect some other good algorithm, like agglomerative, to be suitable

TABLE 6. Impact of sampling rate on the cross-validation approach. The sampling rate starts from 5%, and is then doubled until it reaches 80%.

Sampling rate	Datasets						
	Birch1	Birch2	Unbalance2	Overlap	Asymmetric	Skewed	G2-32d
	100	100	8	6	5	6	2
5%	100	100	8	1	5	2	2
10%	100	100	8	5	5	3	2
20%	100	100	8	5	5	5	2
40%	100	100	10	5	5	6	2
80%	100	100	13	5	9	15	2

Sampling rate	Datasets						
	S1	S2	S3	S4	Iris	Unbalance	Bridge
	15	15	15	15	2	8	2
5%	15	15	1	1	1	8	2
10%	15	15	15	15	2	8	2
20%	15	15	15	15	2	8	2
40%	15	15	15	15	2	8	5
80%	15	13	15	17	2	8	8

as well. However, k-means is much more problematic. First, it hardly works at all with the same 0.9 threshold. Second, even after tuning the threshold to have more suitable value (0.7), it still fails with 10 datasets out of 14 (see Table 5).

We conclude that the choice of the algorithm is not critical as long as a good algorithm (RS or GA) is chosen. K-means, however, is inferior and should not be used.

F. IMPACT OF SAMPLING RATE

We test the impact of sampling rate on the performance of the cross-validation approach by generating subsets with several sampling rates including 5%, 10%, 20%, 40%, and 80%. A low sampling rate may cause too many changes in the structure of the data, whereas a high sampling rate may result in too few changes. The results in Table 6 show that the subsampling rates 10%, 20%, and 40% provide similarly good results. The low subsampling rate 5% causes an error for S3, S4, and *Iris*, and the high sampling rate 80% causes an error for the $S2$, $S4$, *Unbalance*2, and *Asymmetric* datasets. We recommend a subsample size of 20% merely because it is the safest choice.

G. SUMMARY OF THE RESULTS

Based on the best components found in the previous experiments, we construct the following combination:

- Random swap algorithm
- Cross-validation with 10 subsamples of size 20%
- Adjusted Rand Index (ARI) for cluster validation
- No normalization
- Last local maximum

```
StabilityApproach(X, kmax) → k
    best ← 1
    m    ← 10
    s    ← 0.20
    t    ← 0.90
    FOR i = 1 TO m
        XS[i] ← RandomSubsample(X, s)
    FOR k = 2 TO kmax
        P ← RandomSwap(X, k)
        FOR i = 1 TO m
            Q ← RandomSwap(XS[i], k)
            ari[i] ← AdjustedRand(P, Q)
        ARI ← SUM(ari[i]) / m
        IF ARI > t THEN best ← k
RETURN best

RandomSwap(X, k) ⟶ C, P

    T ← 5000
    C ← Select random centroids(X, k)
    P ← Optimal partition(X, C)
    FOR i = 1 TO T
        (C^new,j) ← Swap(X, C)
        P^new ← Local repartition(X, C^new, P, j, k)
        C^new, P^new ← k-means(X, C^new, P^new, k)
        IF MSE(C^new, P^new) < MSE(C, P) THEN
            (C, P) ← C^new, P^new
RETURN C, P
```

FIGURE 16. Pseudo code of the recommended combination.

The pseudo code of the recommended method is shown in Fig. 16. We compare its performance against selected existing stability-based approaches and few internal indices.

TABLE 7. Selected stability-based methods for the comparison.

Parameter	Method			
	Clest [11]	Prediction Strength [22]	Stability method [19]	Figure of Merit [18]
Randomness	Subsampling	Subsampling	Subsampling	Subsampling
Cross-validation strategy	Classification-based	Classification-based	Classification-based	Cross validation (subset-fullset)
Normalization	Based on Null reference	-	Based on random labeling	-
External index	Fowlkes and Mallows	own index	own index	own index
Clustering algorithm	PAM	*k*-means	*k*-means	Free depending on dataset
Selection	Global	Last maximum	Global	Observation

TABLE 8. Results of the stability-based methods.

	Datasets						
	Birch1 100	Birch2 100	Unbalance2 8	Overlap 6	Asymmetric 5	Skewed 6	G2-32d 2
Clest	N/A	N/A	3	5	2, 5	3	2
Pred. Strength	1	1	2	3	2	2	2
Stab. Method	105	107	4	15	2	3	2
Figure of Merit	100	100	13	16	17	20	2
Recommended	100	100	8	5	5	5	2

	Datasets						
	S1 15	S2 15	S3 15	S4 15	Iris 2	Unbalance 8	Bridge 2
Clest	15	15	15	15	5	9	4
Pred. Strength	1	4	4	2	2	2	2
Stab. Method	17	2	2	17	2	2	2
Figure of Merit	15	16	15	20	2	18	4
Recommended	15	15	15	15	2	8	2

The chosen stability-based methods and their components are summarized in Table 7. We use their original components with two exceptions. For consistency, we use nearest centroid classifier for all classification-based methods. In [18], different clustering algorithms were assumed to be used based on their suitability for the given dataset. We use random swap [39], as it is the recommended choice for centroid-based clustering.

The results are summarized in Tables VIII and IX. The recommended method provides the correct result in 12 out of 14 data sets. The corresponding numbers for Clest [11], Pred.Strength [22], Stab.Method [19], and figure of merit [18] are 7, 3, 3, 6, respectively. Clest works well for G2-32*d*, and S1-S4, but fails with other datasets. It uses the PAM algorithm, which is very slow and cannot therefore be applied to the large datasets (*Birch1* and *Birch2*). The three other methods [18], [19], [22] perform poorly. Even if we changed all other parameters (randomness, algorithm, selection) to what we recommend, they would still fail. These work as well as the recommended method after changing their indices to PSI, which shows that the external indices used in these methods are not suitable for the task.

To compare the stability approach with the method based on internal indices for determining the number of clusters, we apply three well-known internal indices: WB index [3], Silhouette coefficient (SC) [60], and Calinski-Harabasz (CH) [61]. Although CH provides the correct results in more cases, they all perform reasonably well in general. These methods seem to be better choices because they are relatively simple to implement and do not have any tuning parameters whatsoever. The stability-based methods are also more time consuming, as the clustering needs to be repeated; we have applied 100 repeats.

A few examples of failed cases are collected in Fig. 17. In some cases (Clest, Prediction Strength), the problem is the methods find most stable clustering with too few

TABLE 9. Results of the internal indices: CH [61], SC [60], WB [3].

	Datasets						
	Birch1 100	Birch2 100	Unbalance2 8	Overlap 6	Asymmetric 5	Skewed 6	G2-32d 2
CH	100	100	8	6	5	20	2
SC	100	100	8	5	2	2	2
WB	100	100	8	6	5	20	2

	Datasets						
	S1 15	S2 15	S3 15	S4 15	Iris 2	Unbalance 8	Bridge 2
CH	15	15	15	15	3	8	2
SC	15	15	15	15	2	2	2
WB	15	15	15	15	6	8	3

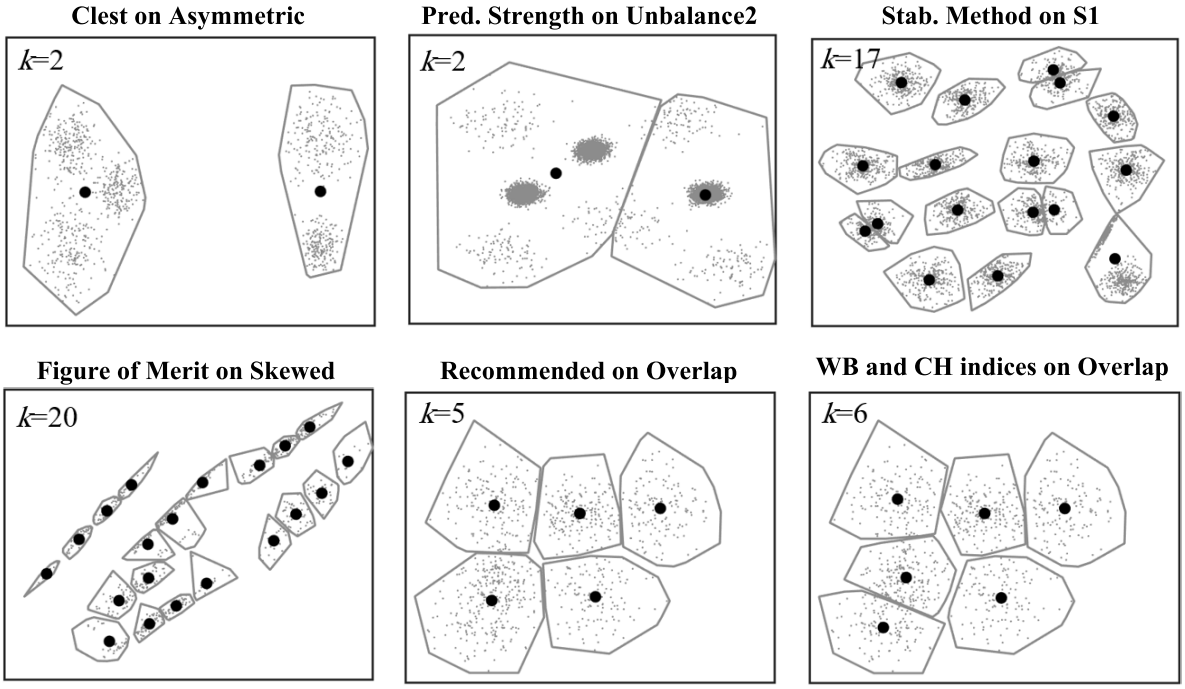

FIGURE 17. Examples of cases when the stability-based methods fail to identify the correct number of clusters. The internal indices WB and CH can find the correct k=6 for Overlap dataset.

clusters ($k = 2$). In these cases, the last local maximum strategy works better. In the case of the stability method, the problem is k-means. It happens to be slightly more stable with $k = 17$.

The other two cases are even more challenging. First, the skewed dataset follows a different model than what SSE tries to capture. Figure of merit finds the most stable result with $k = 20$; the same as what is obtained by the best internal indexes (CH, WB). All other methods also fail with this data. Overlap is another dataset for which all the methods fail. The recommended method and Clest detects k=5, as they only miss the slightly denser cluster, which is hardly visible in the middle left. Among the internal indexes, CH and WB manage to correctly find $k = 6$.

V. CONCLUSIONS

We have performed a systematic study to find out whether the stability-based approach can be used for determining the number of clusters. The simple answer is that, yes, it is possible. We found no fundamental obstacle to using it in cases of spherical clusters, assuming that we know the type of data beforehand and fix the cluster model accordingly. However, even then, we get good results only when using the correct components with proper parameters. By applying an inferior algorithm (k-means), bad sampling rate (5%), normalization (null reference), or ineffective validation index (Rand index), the stability-based approach would not work.

For the individual components, we discuss the choice of the components below in more detail:

1) The exact choice of the cross-validation strategy is not critical. Most indices cannot compare the subsets directly, but comparing a subset to the full set works just fine. Random subsampling is suitable and there is no need to consider other approaches. Our results show that the size of the subset from 10% to 40% is fine. Among the other cross-validation strategies, the classification-based approach adds an extra design question, which is predicting the labels for the missing points. Randomizing the algorithm is not recommended since it gives poor results, even when using a stable algorithm.

2) To select the number of clusters, maximum stability criterion has been mostly used in the literature. However, we have shown that it is not reliable. Instead, we recommend using the last local maximum criterion.

Normalization based on the null reference works worse in most cases. It does not bring enough extra insight into the process, but adds more randomness, which is actually harmful.

3) The choice of external index is not critical. Our results show that any good external index (PSI, ARI, NMI, NVD, CSI) works and that there is no significant difference between them. Only a simple index like Rand index had a negative impact on the result, and should therefore not be used.

4) The choice of the clustering algorithm has a significant effect on the result. We confirm the concern made in [6], [23] that stability-based methods fail when using an unstable algorithm like k-means because it is not stable even with the correct number of clusters. We tested random swap (RS) and genetic algorithm (GA), which both work well. A perfect algorithm as assumed in [23] is not necessary, but something better than the modified k-means variant as in [24] is necessary. We expect that other good algorithms, such as agglomerative clustering (AC), might also work well enough. We leave it for future research to study the stability of the algorithms more extensively.

Using the above guidelines, the stability-based approach can work with reasonable efforts. Despite the positive results, we encountered several challenges that might cause problems when applying the method in different contexts than what we studied here. We briefly discuss them next.

The method has two parameters to set: the sampling rate (20%) and the threshold of the last local maximum criterion (0.90). Too low (5%) or too high (80%) of a sampling rate makes the method fail for some datasets. The suitable range and recommended value of 20% seem a safe choice, but they are still parameters, and it remains an open question as to how well they generalize to other types of data.

Another difficulty is that the clustering result does not depend only on the algorithm, but also on the objective function that it uses. If we know that the clusters are spherical, then using SSE as an objective function works just fine. If we have Gaussian clusters, then we should optimize the Gaussian mixture model. Random swap variant of EM [52] has worked well in our tests in [62]. For more complex data types like clusters with varying densities, arbitrary shapes, or nested clusters, we do not even know which objective function would work well enough. According to the results presented in the literature, algorithms like DBSCAN and single-link work poorly [63]–[65].

A much bigger problem is that clustering is a type of exploratory data analyses and we do not usually know whether the clusters are spherical, Gaussian, arbitrary shaped, or mixed type. We have shown that the stability-based approach works if all its components, including the cluster model for the data, are selected correctly. Slight deviation from the recommendation can make the system fail. It is therefore not expected that the stability-based approach would generalize to unknown data where expected variations in the data are much higher.

REFERENCES

[1] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.
[2] M. Rezaei, "Clustering validation," Ph.D. Dissertation, School Comput., Univ. Eastern Finland, Joensuu, Finland, 2016.
[3] 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.
[4] 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, 1985.
[5] E. Dimitriadou, S. Dolničar, and A. Weingessel, "An examination of indexes for determining the number of clusters in binary data sets," *Psychometrika*, vol. 67, no. 1, pp. 137–159, Mar. 2002.
[6] J. Handl, J. Knowles, and D. B. Kell, "Computational cluster validation in post-genomic data analysis," *Bioinformatics*, vol. 21, no. 15, pp. 3201–3212, Aug. 2005.
[7] Q. Zhao, "Cluster validity in clustering methods," Ph.D. Dissertation, School Comput., Univ. Eastern Finland, Joensuu, Finland, 2012.
[8] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "On clustering validation techniques," *J. Intell. Inf. Syst.*, vol. 17, no. 2, pp. 107–145, Dec. 2001.
[9] V. Roth, T. Lange, M. Braun, and J. Buhmann, "A resampling approach to cluster validation," in *Compstat*. Heidelberg, Germany: Physica, 2002, pp. 123–128.
[10] E. Rendón, I. Abundez, A. Arizmendi, and E. M. Quiroz, "Internal versus external cluster validation indexes," *Int. J. Comput. Commun.*, vol. 5, no. 1, pp. 27–34, 2011.
[11] S. Dudoit and J. Fridlyand, "A prediction-based resampling method for estimating the number of clusters in a dataset," *Genome Biol.*, vol. 3, Jun. 2002, Art. no. research0036.
[12] L. I. Kuncheva and D. P. Vetrov, "Evaluation of stability of k-Means cluster ensembles with respect to random initialization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 11, pp. 1798–1808, Nov. 2006.
[13] A. Strehl and J. Ghosh, "Cluster ensembles—A knowledge reuse framework for combining multiple partitions," *J. Mach. Learn. Res.*, vol. 3, pp. 583–617, Dec. 2002.
[14] S. Zhang, H.-S. Wong, and Y. Shen, "Generalized adjusted rand indices for cluster ensembles," *Pattern Recognit.*, vol. 45, no. 6, pp. 2214–2226, Jun. 2012.
[15] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "Cluster validity methods: Part i," *ACM SIGMOD Rec.*, vol. 31, no. 2, pp. 40–45, Jun. 2002.
[16] J. N. Breckenridge, "Replicating cluster analysis: Method, consistency, and validity," *Multivariate Behav. Res.*, vol. 24, no. 2, pp. 147–161, Apr. 1989.
[17] A. Ben-Hur, A. Elisseeff, and I. Guyon, "A stability based method for discovering structure in clustered data," in *Proc. Pacific Symp. Biocomput.*, 2001, pp. 6–17.
[18] E. Levine and E. Domany, "Resampling method for unsupervised estimation of cluster validity," *Neural Comput.*, vol. 13, no. 11, pp. 2573–2593, Nov. 2001.
[19] T. Lange, V. Roth, M. L. Braun, and J. M. Buhmann, "Stability-based validation of clustering solutions," *Neural Comput.*, vol. 16, no. 6, pp. 1299–1323, Jun. 2004.
[20] Q. Zhao, M. Xu, and P. Franti, "Extending external validity measures for determining the number of clusters," in *Proc. 11th Int. Conf. Intell. Syst. Design Appl.*, Nov. 2011, pp. 931–936.

[21] J. Fridlyand and S. Dudoit, "Applications of resampling methods to estimate the number of clusters and to improve the accuracy of a clustering method," Dept. Statist., UC Berkeley, Berkeley, CA, USA, Tech. Rep. 600, 2001, vol. 600.
[22] R. Tibshirani and G. Walther, "Cluster validation by prediction strength," *J. Comput. Graph. Statist.*, vol. 14, no. 3, pp. 511–528, Sep. 2005.
[23] S. Ben-David, U. Von Luxburg, and D. Pál, "A sober look at clustering stability," in *Proc. COLT*, 2006, pp. 5–19.
[24] U. Von Luxburg, "Clustering stability: An overview," *Found. Trends Mach. Learn.*, vol. 2, no. 3, pp. 235–274, 2010.
[25] M. Bittner *et al.*, "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, vol. 406, no. 6795, pp. 536–540, Aug. 2000.
[26] U. Moller and D. Radke, "A cluster validity approach based on nearest-neighbor resampling," in *Proc. 18th Int. Conf. Pattern Recognit. (ICPR)*, 2006, pp. 892–895.
[27] S. Datta and S. Datta, "Comparisons and validation of statistical clustering techniques for microarray gene expression data," *Bioinformatics*, vol. 19, no. 4, pp. 459–466, Mar. 2003.
[28] P. Smyth, "Clustering using Monte Carlo cross-validation," in *Proc. KDD*, Aug. 1996, pp. 26–133.
[29] O. Abul, A. Lo, R. Alhajj, F. Polat, and K. Barker, "Cluster validity analysis using subsampling," in *Proc. IEEE SMC*, Oct. 2003, pp. 1435–1440.
[30] P. Fränti, M. Rezaei, and Q. Zhao, "Centroid index: Cluster level similarity measure," *Pattern Recognit.*, vol. 47, no. 9, pp. 3034–3045, Sep. 2014.
[31] L. Hubert and P. Arabie, "Comparing partitions," *J. Classification*, vol. 2, no. 1, pp. 193–218, Dec. 1985.
[32] T. O. Kvalseth, "Entropy and correlation: Some comments," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-17, no. 3, pp. 517–519, May. 1987.
[33] M. Meilă, "Comparing clusterings—An information based distance," *J. Multivariate Anal.*, vol. 98, no. 5, pp. 873–895, May 2007.
[34] S. van Dongen, "Performance criteria for graph clustering and Markov cluster experiments," Centrum Voor Wincunde En Infomatica, Amsterdam, The Netherlands, Tech. Rep. INSR0012, 2000.
[35] M. Rezaei and P. Franti, "Set matching measures for external cluster validity," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 8, pp. 2173–2186, Aug. 2016.
[36] P. Fränti and S. Sieranoja, "K-means properties on six clustering benchmark datasets," *Appl. Intell.*, vol. 18, no. 12, pp. 4743–4759., 2018.
[37] P. Fränti and S. Sieranoja, "How much can k-means be improved by using better initialization and repeats?" *Pattern Recognit.*, vol. 93, pp. 95–112, Sep. 2019.
[38] J. H. Ward, "Hierarchical grouping to optimize an objective function," *J. Amer. Stat. Assoc.*, vol. 58, no. 301, pp. 236–244, Mar. 1963.
[39] P. Fränti, "Efficiency of random swap clustering," *J. Big Data*, vol. 5, no. 1, p. 13, Dec. 2018.
[40] P. Fränti, "Genetic algorithm with deterministic crossover for vector quantization," *Pattern Recognit. Lett.*, vol. 21, no. 1, pp. 61–68, Jan. 2000.
[41] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *J. Roy. Stat. Soc., Ser. B, Stat. Methodol.*, vol. 63, no. 2, pp. 411–423, May 2001.
[42] F. Leisch, "Resampling methods for exploring cluster stability," in *Handbook of Cluster Analysis*. Boca Raton, FL, USA: CRC Press, 2015, pp. 658–673.
[43] E. Bae, J. Bailey, and G. Dong, "A clustering comparison measure using density profiles and its application to the discovery of alternate clusterings," *Data Mining Knowl. Discovery*, vol. 21, no. 3, pp. 427–471, Nov. 2010.
[44] P. Raman, J. M. Phillips, and S. Venkatasubramanian, "Spatially-aware comparison and consensus for clusterings," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2011, pp. 307–318.
[45] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, 4th ed. New York, NY, USA: Academic, 2008.
[46] D. MacKay, "An example inference task: Clustering," in *Information Theory, Inference and Learning Algorithms*, vol. 20. Cambridge, U.K.: Cambridge Univ. Press, 2003, pp. 284–292.
[47] J. Wang, "Consistent selection of the number of clusters via crossvalidation," *Biometrika*, vol. 97, no. 4, pp. 893–904, Dec. 2010.
[48] P. Franti, T. Kaukoranta, D.-F. Shen, and K.-S. Chang, "Fast and memory efficient implementation of the exact PNN," *IEEE Trans. Image Process.*, vol. 9, no. 5, pp. 773–777, May 2000.
[49] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Roy. Stat. Soc. Ser. B, Methodol.*, vol. 31, pp. 1–22, Sep. 1977.
[50] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton, "SMEM algorithm for mixture models," in *Proc. Adv. Neural Inf. Process. Syst.*, 1999, pp. 599–605.
[51] F. Pernkopf and D. Bouchaffra, "Genetic-based EM algorithm for learning Gaussian mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 8, pp. 1344–1348, Aug. 2005.
[52] Q. Zhao, V. Hautamäki, I. Kärkkäinen, and P. Fränti, "Random swap EM algorithm for Gaussian mixture models," *Pattern Recognit. Lett.*, vol. 33, no. 16, pp. 2120–2126, Dec. 2012.
[53] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. KDD*, Aug. 1996, pp. 226–231.
[54] M. Ankerst, M. M. Breunig, H.-P. Kriegel, and J. Sander, "OPTICS: Ordering points to identify the clustering structure," *ACM SIGMOD Rec.*, vol. 28, no. 2, pp. 49–60, Jun. 1999.
[55] A. N. Albatineh, M. Niewiadomska-Bugaj, and D. Mihalko, "On similarity indices and correction for chance agreement," *J. Classification*, vol. 23, no. 2, pp. 301–313, Sep. 2006.
[56] 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.
[57] W. M. Rand, "Objective criteria for the evaluation of clustering methods," *J. Amer. Stat. Assoc.*, vol. 66, no. 336, pp. 846–850, Dec. 1971.
[58] 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.
[59] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Statist. Probab.*, Jun. 1967, pp. 281–297.
[60] 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.
[61] T. Caliński and J. Harabasz, "A dendrite method for cluster analysis," *Commun. Statist.-theory Methods*, vol. 3, no. 1, pp. 1–27, 1974.
[62] P. Fränti and M. Rezaei, "Generalizing centroid index to different clustering models," in *Proc. Joint IAPR Int. Workshops Stat. Techn. Pattern Recognit. (SPR) Struct. Syntactic Pattern Recognit. (SSPR) (S+SSPR)*, 2016, pp. 285–296.
[63] H. P. Kriegel, P. Kröger, J. Sander, and A. Zimek, "Density-based clustering," *Wiley Interdiscipl. Rev., Data Mining Knowl. Discovery*, vol. 1, pp. 231–240, 2011.
[64] S. C. Dinger, M. A. Van Wyk, S. Carmona, and D. M. Rubin, "Clustering gene expression data using a diffraction-inspired framework," *Biomed. Eng. OnLine*, vol. 11, no. 1, p. 85, 2012.
[65] Z. Xiong, R. Chen, Y. Zhang, and X. Zhang, "Multi-density dbscan algorithm based on density levels partitioning," *J. Inf. Comput. Sci.*, vol. 9, no. 10, pp. 2739–2749, 2012.

MOHAMMAD REZAEI received the B.Sc. degree in electronic engineering and the M.Sc. degree in biomedical engineering from the Amirkabir University of Technology, Tehran, Iran, in 1996 and 2003, respectively, and the Ph.D. degree in computer science from the University of Eastern Finland. His research interests include data clustering, multimedia processing, classification, and retrieval.

PASI FRÄNTI (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in science from the University of Turku, in 1991 and 1994, respectively.

Since 2000, he has been a Professor of computer science with the University of Eastern Finland. He is currently a Visiting Professor with Shenzhen Technology University, China. He has published 86 journals and 174 peer review conference papers, including 14 IEEE transaction articles. Significant contributions have also been made in image compression, image analysis, vector quantization, and speech technology. His main research interests include machine learning, data mining, and pattern recognition, including clustering algorithms and intelligent location-aware systems.