Generalizing Centroid Index to Different Clustering Models

Pasi Fränti[✉] and Mohammad Rezaei

University of Eastern Finland, Joensuu, Finland
franti@cs.uef.fi

Abstract. Centroid index is the only measure that evaluates cluster level differences between two clustering results. It outputs an integer value of how many clusters are differently allocated. In this paper, we apply this index to other clustering models that do not use centroid as prototype. We apply it to centroid model, Gaussian mixture model, and arbitrary-shape clusters.

Keywords: Clustering · Validity index · External index · Centroid index

1 Introduction

Clustering aims at partitioning a data set of n points into k clusters. *External index* measures how similar a clustering solution is to a given ground truth (if available), or how similar two clustering solutions are. This kind of measure is needed in clustering ensemble, measuring stability and evaluating performance of clustering algorithms.

In clustering algorithms, one of the main challenges is to solve the global allocation of the clusters instead of just tuning the partition borders locally. Despite of this, most external cluster validity indexes calculate only point-level differences without any direct information about how similar the cluster-level structures are.

Rand index (RI) [1] and *Adjusted Rand index* (ARI) [2] count the number of pairs of data points that are partitioned consistently in both clustering solutions (A and B); if a pair of points is allocated in the same cluster in A, they should be allocated into the same cluster also in B. This provides estimation of point-level similarity but does not give much information about the similarity at cluster level.

More sophisticated methods operate at the cluster level. *Mutual information* (MI) [3] and *Normalized mutual information* (NMI) [4] measure the amount of information (conditional entropy) that can be obtained from a cluster in A using the clusters in B. Set-matching based measures include *Normalized van Dongen* (NVD) [5] and *Criterion-H* (CH) [6]. They match the clusters between A and B, and measure the amount of overlap between the clusters, see Fig. 1. However, all of them measure point-level differences. What is missing is a simple structural cluster-level measure.

Figure 2 demonstrates the situation with three clustering results: *k-means* (KM) [7], *random swap* (RS) [8] and *agglomerative clustering* (AC) [9]. The clustering structure of RS and AC is roughly the same, and their differences come mostly from the minor inaccuracies of the centroids in the agglomerative clustering. The result of the k-means

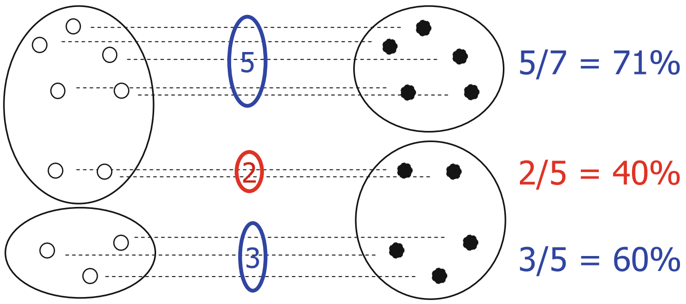
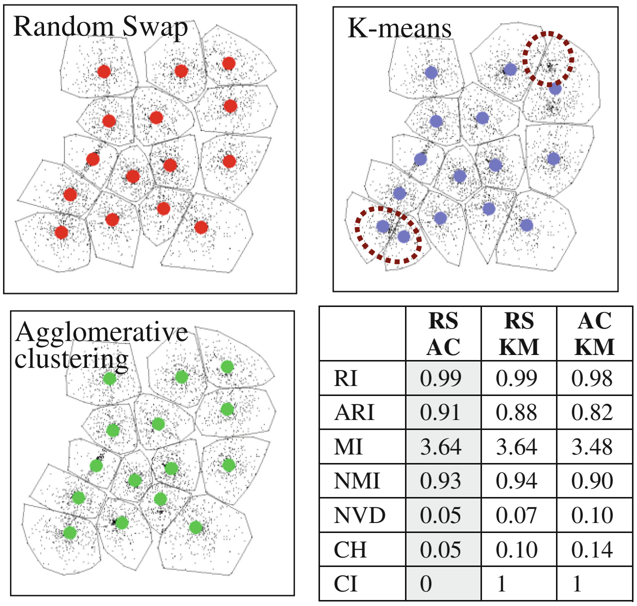


Fig. 1. Principle of set-matching based external validity indexes. The values are the number of overlap between the clusters. Blue indicate clusters that would be selected for matching. Sample index values for this example would be NMI = 0.42, NVD = 0.20, CH = 0.20, CI = 0. (Color figure online)



	RS AC	RS KM	AC KM
RI	0.99	0.99	0.98
ARI	0.91	0.88	0.82
MI	3.64	3.64	3.48
NMI	0.93	0.94	0.90
NVD	0.05	0.07	0.10
CH	0.05	0.10	0.14
CI	0	1	1

Fig. 2. Comparing three clustering results: Random Swap (RS), Agglomerative clustering (AC) and K-means (KM) with eight measures.

has a more significant, structural difference: one centroid is missing at the top, and there are too many centroids at the bottom.

Some indexes (ARI, NVD, CH) indicate that RS-vs-AC in Fig. 2 are more similar (ARI = 0.91; NVD = 0.05; CH = 0.05) than RS-vs-KM (ARI = 0.88; NVD = 0.07; CH = 0.10), or AC-vs-KM (ARI = 0.82; NVD = 0.10; CH = 0.14) but the numbers do not tell that their global structure is the same.

In a recent work [10], we introduced a cluster level index called *centroid index* (CI) to cope with this problem by measuring the number of clusters allocated differently. For each centroid in A, the method finds its nearest centroid in B. Then it calculates the *indegree*-values for each centroid in B; how many times it was selected as the nearest. An orphan centroid (indegree = 0) indicates that a cluster is differently located in the clustering structure. The index value is the count of these orphans so that CI-value indicates the number of cluster-level differences. Value CI = 0 indicates that the clustering structures are identical, CI = 1 that one cluster mismatch, and so on.

The measure is somewhat rough as it ignores the point-level differences. A simple point-level extension called *centroid similarity index* (CSI) was therefore also considered. However, the main idea to measure the cluster level differences is best captured by the raw CI-value, or relative to the number of clusters: CI/k. In Fig. 2, CI-value clearly tells that RS and AC have similar clustering structure (CI = 0), and that k-means has one difference (CI = 1).

A limitation of CI is that we must have the centroids. It is possible to use other clustering models like *k-medoids* [11] and *k-modes* [12] by finding the nearest medoid/mode using the distances in the feature space. However, in more complex clustering models like Gaussian mixture model (GMM), density-based or arbitrary-shape clustering, it is not as simple.

In this paper, we show how to apply centroid index by using partitions instead of the centroids. For every cluster in A, we first find its most similar cluster in B by calculating the amount of overlap by Jaccard coefficient, which is the number of shared points divided by the total number of distinctive data points in the two clusters:

$$J = \frac{|A_i \cap B_j|}{|A_i \cup B_j|} \tag{1}$$

where A_i and B_j are the matching (most similar) clusters.

However, we do not sum up the overlaps but we further analyze the nearest neighbor mapping. The cluster-to-cluster similarity is now determined at point level, but the overall value is still measured at the cluster level. The measure is calculated as the number of orphan clusters (indegree = 0) as in the original definition of centroid index. Thus, the method now generalizes from the centroid-model to other clustering models independent on how the cluster is represented: centroid, medoid, mode, or even no prototype used at all.

For example, in Fig. 1, the topmost clusters are mapped to each other, and the bottom-most clusters to each other. This results in mapping where all indegree = 1 for both clusters, and CI-value becomes 0. Therefore, these two solutions have the same clustering structure and they differ only at the point level.

2 External Cluster Validity

Existing cluster validity indexes can be divided into three categories: *pair-counting*, *information theoretic* and *set-matching based* measures:

Pair-counting measures

- RI = Rand index [1]
- ARI = Adjusted Rand index [2]

Information theoretic measures

- MI = Mutual information [3]
- NMI = Normalized Mutual information [4]

Set-matching based measures

- NVD = Normalized van Dongen [5]
- CH = Criterion H [6]
- CI = Centroid Index [10]
- CSI = Centroid Similarity Index [10]

We next briefly recall the idea of set-matching based methods [13]. The clusters are first either *paired* (NVD) or *matched* (CH and CI). In pairing, best pair of the clusters in A and B are found by minimizing the sum of the similarities of the paired clusters. Hungarian [13], or greedy algorithm [6, 14] has been used to solve it.

In matching, nearest neighbour is searched. This does not always result in bijective mapping where exactly two clusters are paired, but several clusters in A can be mapped to the same cluster in B (or vice versa), see Fig. 3. The mapping is not symmetric, and is usually done in both ways: $A \rightarrow B$ and $B \rightarrow A$.

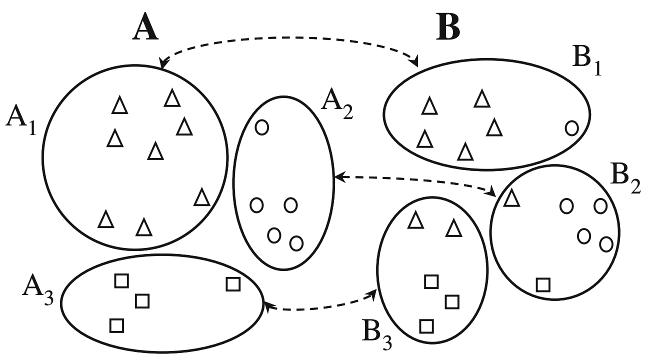


Fig. 3. Matching of clusters using pointwise similarities.

The similarities of the matched/paired clusters are then calculated by summing up the total overlap. In [13], the values are first normalized by the maximum cluster size. In NVD, CH and CSI, the normalization is performed after the summation by dividing the total number of shared points by the total number of points:

$$S = \frac{1}{N} \cdot \sum_{i=1}^{k} |A_i \cap NN(A_i)| \tag{2}$$

where $NN(A_i)$ is the cluster in B that A_i is matched/paired with. Centroid index (CI) finds the nearest centroid without any use of the partitions:

$$NN(A_i) = \underset{1 \le j \le k}{\arg\min} \left\| c[A_i] - c\left[B_j\right] \right\| \tag{3}$$

where $c[A_i]$ and $c[B_j]$ are the centroids of A and B. The other difference is that CI does not use the point-level measure (2). Instead, it calculates the number of mappings (*indegree*) made for each cluster B_j, and then sums up the number of clusters in B that has not been mapped at all (*indegree* = 0). These are called orphans:

$$CI_1(A \to B) = \sum_{j=1}^{k} Orphan\left(B_j\right) \tag{4}$$

where *Oprhan*(B) has value 1 if no cluster in A is mapped to it:

$$Orphan(B) = \begin{cases} 1 & InDegree(A) = 0 \\ 0 & InDegree(A) > 0 \end{cases} \tag{5}$$

In order to have symmetric index, we perform mappings in both ways: $A \to B$ and $B \to A$. The CI-value is then defined as the maximum of these two:

$$CI(A, B) = \max\{CI_1(A \to B), CI_1(B \to A)\} \tag{6}$$

To sum up, the index is easy to calculate, and the result has clear interpretation: how many clusters are differently allocated. An example is shown in Fig. 4.

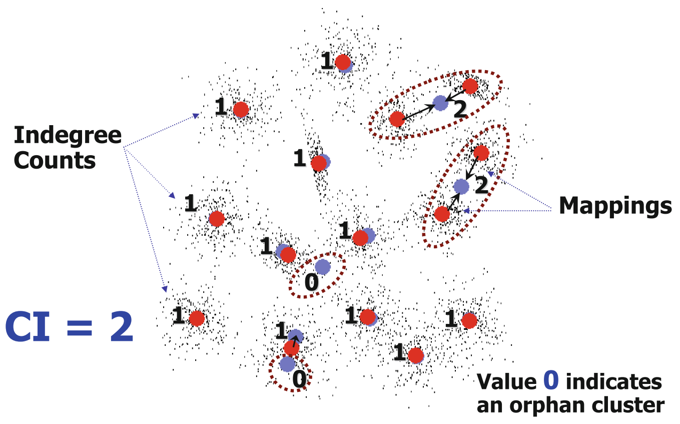


Fig. 4. Example of Centroid index (CI). Only mapping RED→PURPLE is shown. Some purple circles are under the red and not visible. (Color figure online)

3 Centroid Index Using Different Clustering Models

The main limitation of the centroid index is that it requires centroids, which may not exist in other clustering models such as *k-medoids* [11], *k-modes* [12], *Gaussian mixture model* (*GMM*), *density-based* and *arbitrary-shape* clustering. To apply the index to other models, we consider the following alternative approaches:

- Prototype similarity
- Partition similarity
- Model similarity

3.1 Prototype vs. Partition Similarity

K-medoids and k-modes clustering operate also in a feature space (usually Euclidean), so generalization to them is trivial using the prototype similarity approach. We just find the nearest prototype in the other solution using the distance in the feature space. It does not matter whether the prototype is centroid, medoid or mode.

The second approach uses the partitions of the two clustering solutions. The matching of each cluster is done by finding the most similar cluster in the other solution using (1). This applies to any partition-based clustering and it can be calculated from the contingency table in $O(N \cdot k^2)$ time [2]. Contingency table tells how many shared points two clusters A_i and B_j have. See Table 1 for an example.

Table 1. Contingency table for the clusterings presented in Fig. 3. For instance, clusters A_1 shares 5 points with cluster B_1, 1 point with cluster B_2, and 2 points with cluster B_3.

	B_1	B_2	B_3	$\sum$
A_1	5	1	2	8
A_2	1	4	0	5
A_3	0	1	3	4
$\sum$	6	6	5	17

This partition similarity approach applies also to centroid-based clustering but it may be slower to calculate. Finding the most similar prototype takes only $O(dk^2)$, where d is the number of dimensions. This is expected to be faster than the $O(N) + O(k^2)$ of the partition similarity; unless if the dimensionality or the number of clusters is very high. For S1–S4 data (http://cs.uef.fi/sipu/datasets) the estimated numbers are 450 distance calculations for the prototype similarity, and 4352 counter additions for the partition similarity. However, observed values in Table 2 show that the calculation of the squared distance takes so much longer that the speed benefit is practically lost. Thus, the partition-based variant is virtually as fast already when $d = 2$.

Table 2. Processing times for CI using prototype and partition similarity. The third row shows how many times they provided different CI-values. Data sets S1-S4 are from [15].

	S1	S2	S3	S4	Birch2
Prototype	15 ms	15 ms	15 ms	15 ms	120 ms
Partition	14 ms	14 ms	14 ms	14 ms	250 ms
Different	5.8 %	7.1 %	5.2 %	7.9 %	30 %

It is also possible that the two approaches provide different mappings. Even then, the resulting CI-value is expected to be mostly the same. Table 2 also reports the number of times the two approaches give different CI-value.

The third approach is to derive the similarity of clusters directly from the probability density function of the model. Next we study this for Gaussian mixture model.

3.2 Model Similarity

Gaussian mixture model (GMM) represents every cluster by its centroid and covariance matrix. This increases the size of the model from O(1) to O(d^2) per cluster. However, there is often not enough data to estimate the covariances reliably. A simplified variant therefore considers the diagonal of the covariance matrix, thus, reducing the model size to O(d). We also use this simplified variant here.

Expectation maximization (EM) [16] algorithm optimizes GMM analogous to k-means. It iterates *Expectation* and *Maximization* steps in turn to optimize loglikelihood. It also suffers the same problem as k-means: stucks into a local optimum. Better variants include split-and-merge [17], genetic algorithm [18] and random swap [19].

For comparing two clusters (mixtures), we can use any of the three approaches. Prototype similarity approach ignores the covariance and just finds the nearest centroid in the other solution. Partition similarity performs hard partition by assigning each point into the cluster with maximum likelihood, after which the partition similarity can be calculated using the contingency table.

For the model similarity, we use here *Bhattacharyya coefficient*. It measures the similarity between two probability distributions p and q, and is calculated as follows:

$$S_{BC} = \sum \sqrt{p_i \cdot q_j} \tag{7}$$

For two multivariate normal distributions, it can be written as:

$$S_{BC} = \frac{1}{8}\left(c[A_i] - c\left[B_j\right]\right)^T \Sigma^{-1}\left(c[A_i] - c\left[B_j\right]\right) + \frac{1}{2}\ln\left(\frac{|\Sigma|}{\sqrt{|\Sigma_1||\Sigma_2|}}\right) \tag{8}$$

where $c[A_i]$ and $c[B_j]$ are the means (centroids), Σ_1 and Σ_2 are the covariance matrices of the two clusters A_i and B_j. and Σ is the average of Σ_1 and Σ_2. The first term in (8) represents the *Mahalonobis distance*, which is a special case of Bhattacharyya when the covariance matrixes of the two distributions are the same as is the case in GMM.

Figure 5 shows an example of two GMM models by SMEM [17] and RSEM [19] algorithms. We compare the results by calculating the various indexes from the resulting partitions. The result of CI was calculated by all the three approaches; all resulted into the same mapping, and gave exactly the same value CI = 2.

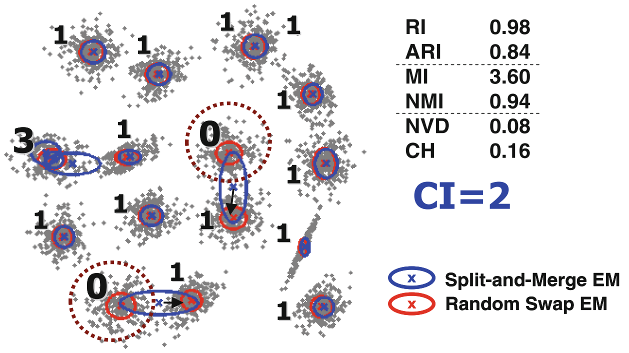


Fig. 5. Centroid index for Gaussian mixture model. Only mapping SMEM→RSEM is shown.

3.3 Arbitrary Partitions

Arbitrary-shape clustering differs from the model-based clustering in that there is no model or prototype for the clusters. This kind of clustering problem is often solved based on connectivity. For example, single link algorithm [20] results in a minimum spanning forest where each cluster is represented by the minimum spanning tree of the points in the cluster.

To compare such clustering results by CI, we use the partition similarity because it is independent on the chosen clustering model. All what is required is that we can access the partitions. However, in on-line clustering where huge amount of data is processed, the original data points might not be stored but deleted immediately when their contribution to the model is calculated. In this case, the partition-based similarity cannot be applied but in most offline applications, we do have access to partitions.

Examples of data with arbitrary-shape clusters is shown in Fig. 6 when clustered by k-means (left) and single link (right). K-means misses the two smaller clusters, and

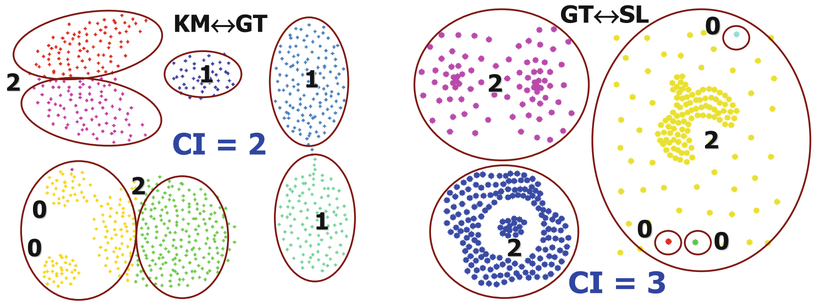


Fig. 6. K-means result for Aggregate (left) and Single Link result for Compound (right) when compared to the ground truth (GT). Mapping of the max value is shown (KM→GT; GT→SL).

divides the bigger ones in the middle and on the top. Single link makes three mistakes in total, by merging three real clusters and by creating three false clusters of size one.

4 Experiments

Here we provide numerical comparison of the centroid index (CI) against the selected existing indexes. We use the following data sets:

- S1–S4 [15]: 5000 points in 15 clusters.
- Unbalanced [13]: 6500 points in 8 clusters
- Birch2 [23]: 100,000 points in 100 clusters
- Aggregate [24]: 788 points in 7 clusters
- Compound [25]: 399 points in 6 clusters

We apply the following algorithms: K-means (KM), K-means++ (KM++) [26], Random Swap (RS) [8], Genetic Algorithm (GA) [27] for the data with spherical clusters; Single Link (SL) [21], DBSCAN [21], Split-and-Merge (SAM) [22] for data with arbitrary-shape clusters. K-means is applied for both.

Same clustering results with the corresponding validity index values are given in Tables 3. Unbalanced is rather easy to cluster by good algorithm but k-means fails because random initialization tends to select all centroids from the bigger clusters to the left, and only one centroid will move to cover the five small clusters, see Fig. 7. Thus, leaving four other clusters empty, which results in CI = 4. Most other indexes react to this but their exact values tell very little about how severe the error is, whereas the CI-value tells that half (4/8) of the clusters are wrongly allocated.

Table 3. Sample clustering results with validity values. CI = 0 indicates correct structure.

	RI	ARI	MI	NMI	NVD	CH	CSI	CI
Birch2								
KM	1.00	0.81	6.26	0.96	0.12	0.24	0.88	18
KM ++	1.00	0.95	6.54	0.99	0.03	0.06	0.97	4
RS	1.00	1.00	6.64	1.00	0.00	0.00	1.00	0
GA	1.00	1.00	6.64	1.00	0.00	0.00	1.00	0
S1								
KM	0.98	0.82	3.57	0.93	0.09	0.17	0.83	2
KM ++	1.00	1.00	3.90	0.98	0.00	0.00	1.00	0
RS	1.00	1.00	3.90	0.98	0.00	0.00	1.00	0
GA	1.00	1.00	3.90	0.98	0.00	0.00	1.00	0
S2								
KM	0.97	0.80	3.46	0.90	0.11	0.18	0.82	2
KM ++	1.00	0.99	3.87	0.99	0.00	0.00	1.00	0
RS	1.00	0.99	3.87	0.99	0.00	0.00	1.00	0
GA	1.00	0.99	3.87	0.99	0.00	0.00	1.00	0

(*continued*)

Table 3. (*continued*)

	RI	ARI	MI	NMI	NVD	CH	CSI	CI
Unbalanced								
KM	0.92	0.79	1.85	0.81	0.14	0.29	0.86	4
KM ++	1.00	1.00	2.03	1.00	0.00	0.00	1.00	0
RS	1.00	1.00	2.03	1.00	0.00	0.00	1.00	0
GA	1.00	1.00	2.03	1.00	0.00	0.00	1.00	0
SL	1.00	0.99	1.91	0.97	0.02	0.05	0.98	3
DBSCAN	1.00	1.00	2.02	0.99	0.00	0.00	1.00	0
SAM	0.93	0.81	1.85	0.82	0.12	0.25	0.88	4
Aggregate								
KM	0.91	0.71	2.16	0.84	0.14	0.24	0.86	2
SL	0.93	0.80	1.96	0.88	0.09	0.18	0.91	2
DBSCAN	0.99	0.98	2.41	0.98	0.01	0.01	0.99	0
SAM	1.00	1.00	2.45	1.00	0.00	0.00	1.00	0
Compound								
KM	0.84	0.54	1.71	0.72	0.25	0.34	0.75	2
SL	0.89	0.74	1.54	0.80	0.13	0.26	0.87	3
DBSCAN	0.95	0.88	1.90	0.87	0.10	0.12	0.90	2
SAM	0.83	0.53	1.78	0.76	0.19	0.34	0.81	2

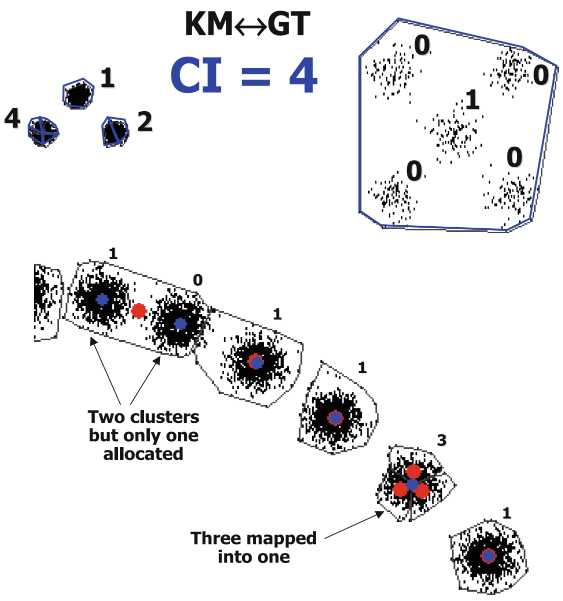


Fig. 7. Datasets unbalance (above) and Birch2 (below) clustered by k-means.

The results for the data with arbitrary-shaped clusters are similar. DBSCAN and SAM work well for the Aggregate providing perfect clustering structure (CI = 0) although DBSCAN leaves out few points as outliers. Compound is more challenging and all the methods make 2 or 3 errors in the clustering structure, usually merging the leftmost clusters and creating too many on the right.

Finally, we study how the indexes react when randomness is added increasingly to artificially created partitions (for details see [13]). Figure 8 shows that the centroid index does not react at all for these point-level changes as long as most points keeps in the original cluster. The values of the set-based measures (NVD, CH, Purity) decrease linearly, which shows that they are most appropriate to measure point-level changes.

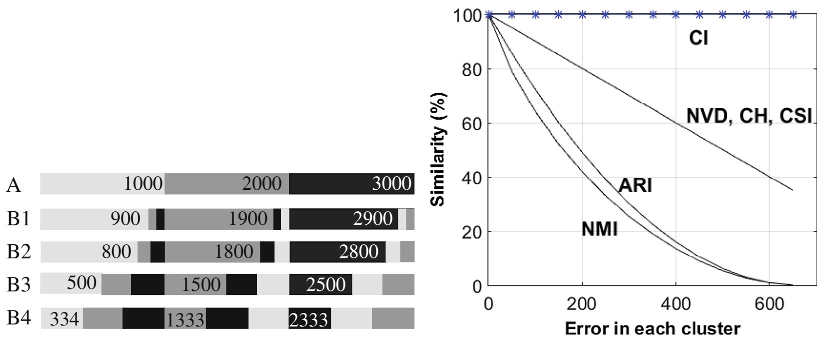


Fig. 8. Effect of increasing randomness in partitions to the clustering result.

5 Conclusions

Centroid Index (CI) is the only validity index that provides cluster level measure. It tells exactly how many clusters are differently allocated, which is more useful information than counting point-level differences. In this paper, we applied it to other clustering models such as Gaussian mixture and data with arbitrary-shaped clusters. Its main advantage is that the significance of the index value can be trivially concluded: value CI > 0 indicate that there is a significant difference in the clustering structure.

References

1. Rand, W.M.: Objective criteria for the evaluation of clustering methods. J. Am. Stat. Assoc. **66**(336), 846–850 (1971)
2. Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**, 193–218 (1985)
3. Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance. J. Mach. Learn. Res. **11**, 2837–2854 (2010)
4. Kvalseth, T.O.: Entropy and correlation: some comments. IEEE Trans. Syst. Man Cybern. **17**(3), 517–519 (1987)

5. Dongen, S.V.: Performance criteria for graph clustering and Markov cluster experiments. Technical report INSR0012, Centrum voor Wiskunde en Informatica (2000)
6. Meila, M., Heckerman, D.: An experimental comparison of model based clustering methods. Mach. Learn. **41**(1–2), 9–29 (2001)
7. MacKay, D.: An example inference task: clustering. In: MacKay, D. (ed.) Information Theory, Inference and Learning Algorithms, pp. 284–292. Cambridge University Press, Cambridge (2003)
8. Fränti, P., Kivijärvi, J.: Randomised local search algorithm for the clustering problem. Pattern Anal. Appl. **3**(4), 358–369 (2000)
9. Fränti, P., Virmajoki, O., Hautamäki, V.: Fast agglomerative clustering using a k-nearest neighbor graph. IEEE TPAMI **28**(11), 1875–1881 (2006)
10. Fränti, P., Rezaei, M., Zhao, Q.: Centroid index: cluster level similarity measure. Pattern Recogn. **47**(9), 3034–3045 (2014)
11. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York (1990)
12. Huang, Z.: A fast clustering algorithm to cluster very large categorical data sets in data mining. Data Min. Knowl. Disc. **2**(3), 283–304 (1998)
13. Rezaei, M., Fränti, P.: Set matching measures for external cluster validity. IEEE Trans. Knowl. Data Eng. **28**(8), 2173–2186 (2016)
14. Zhao, Q., Fränti, P.: Centroid ratio for pairwise random swap clustering algorithm. IEEE Trans. Knowl. Data Eng. **26**(5), 1090–1101 (2014)
15. Fränti, P., Virmajoki, O.: Iterative shrinking method for clustering problems. Pattern Recogn. **39**(5), 761–765 (2006)
16. Dempster, A., Laird, N., Rubin, D.: Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Stat. Soc. B **39**, 1–38 (1977)
17. Udea, N., Nakano, R., Gharhamani, Z., Hinton, G.: SMEM algorithm for mixture models. Neural Comput. **12**, 2109–2128 (2000)
18. Pernkopf, F., Bouchaffra, D.: Genetic-based em algorithm for learning Gaussian mixture models. IEEE Trans. Pattern Anal. Mach. Intell. **27**(8), 1344–1348 (2005)
19. Zhao, Q., Hautamäki, V., Kärkkäinen, I., Fränti, P.: Random swap EM algorithm for Gaussian mixture models. Pattern Recogn. Lett. **33**, 2120–2126 (2012)
20. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice-Hall Inc., Upper Saddle River (1988)
21. Ester, M., Kriegel, H.P., Sander, J., Xu, X.: A density-based algorithm for discovering clusters in large spatial databases with noise. In: KDDM, pp. 226–231 (1996)
22. Zhong, C., Miao, D., Fränti, P.: Minimum spanning tree based split-and-merge: a hierarchical clustering method. Inf. Sci. **181**, 3397–3410 (2011)
23. Zhang, T., et al.: BIRCH: a new data clustering algorithm and its applications. Data Min. Knowl. Disc. **1**(2), 141–182 (1997)
24. Gionis, A., Mannila, H., Tsaparas, P.: Clustering aggregation. ACM Trans. Knowl. Disc. Data (TKDD) **1**(1), 1–30 (2007)
25. Zahn, C.T.: Graph-theoretical methods for detecting and describing gestalt clusters. IEEE Trans. Comput. **100**(1), 68–86 (1971)
26. Arthur, D., Vassilvitskii, S.: K-means ++: the advantages of careful seeding. In: ACM-SIAM Symposium on Discrete Algorithms (SODA 2007), pp. 1027–1035, January 2007
27. Fränti, P.: Genetic algorithm with deterministic crossover for vector quantization. Pattern Recogn. Lett. **21**(1), 61–68 (2000)