Applied Computing and Intelligence

https://www.aimspress.com/journal/aci

Applied Computing and Intelligence
4 (1): 24–44.
DOI: 10.3934/aci.2024003
Received: 06 June 2024
Revised: 13 June 2024
Accepted: 13 June 2024
Published: 17 June 2024

*Research article*

# Clustering accuracy

**Pasi Fränti* and Sami Sieranoja**

Machine Learning, School of Computing, University of Eastern Finland, Finland

* **Correspondence:** Email: pasi.franti@uef.fi.

Academic Editor: Chih-Cheng Hung

**Abstract:** Clustering accuracy (ACC) is one of the most often used measures in literature to evaluate clustering quality. However, the measure is often used without any definition or reference to such a definition. In this paper, we identify the origin of the measure. We give a proper definition for the measure and provide a simple bug fix which allows it to be used also in the case of a mismatch in the number of clusters. We show that the measure belongs to a wider class of set-matching based measures. We compare its properties to centroid index (CI) and normalized mutual information (NMI).

**Keywords:** clustering evaluation; external index; clustering accuracy; normalized mutual information; centroid index

## 1. Introduction

*Clustering accuracy* (ACC) is a frequently used measure in literature to evaluate clustering performance despite it is not well-defined anywhere. The definition is seemingly intuitive and closely related to classification accuracy used in supervised learning. However, clustering is unsupervised learning where ground truth lacks class labels. It is therefore not clear how ACC should be calculated. Many papers do not even define the measure or cite its source. If a citation is given, it is usually just a citation to another paper also using the same measure.

In classification, we have class labels so that a unique matching between the predicted and ground truth classes can be easily made. *Classification accuracy* is then simply the percentage of correctly classified objects. Measures like *precision* and *recall* are also used in binary classification. In clustering, however, the cluster label is just an arbitrary number without any correspondence to the labels used in the ground truth, see Figure 1. The number of clusters can also differ.

In this paper, we review the history of the clustering accuracy measure and discover its origin. No single paper to date has defined this measure precisely. Two early papers used a vaguely described measure which was most likely inspired by other similar measures born around 2000 [1,2]. The clustering accuracy measure started to become popular probably due to its intuitive name and publicly available Matlab implementation[1].

In this paper, we provide a precise definition for the measure and demonstrate the effect of its main design components. In particular, we show that clustering accuracy is a variant of a wider class of *set-matching based measures* [3]. The clustering accuracy measure itself is valid despite a minor issue of not allowing a mismatch or clustering sizes between the detected and ground truth clusters. We revise the measure by fixing this minor problem and provide Python software available in Github[2]. We include two variants: a variant that is backward compatible with the existing measure (ACC-pair), and another simpler variant (ACC-match) that avoids using the Hungarian algorithm.

The rest of the paper is organized as follows. Section 2 reviews the existing approaches for evaluating clustering performance. The history of the clustering accuracy measure is presented in Section 3. A definition for the measure is given in Section 4, and a revised variant in Section 5 with pseudo code and time benchmarking results. Conclusions are then drawn in Section 6.

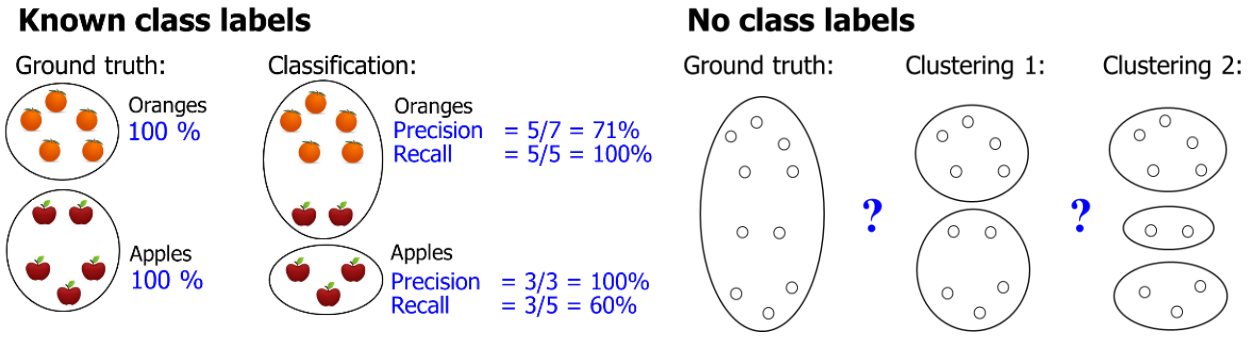


**Figure 1.** Measures such as precision, recall, and classification accuracy can be trivially calculated if we know the class labels. In clustering, this information is missing.

## 2. Evaluating clustering performance

Three common approaches exist for evaluating clustering:

- Internal evaluation
- External evaluation
- Application performance

Internal evaluation measures how well a given objective function is optimized. The most common function for this is *sum-of-squared errors* (SSE) used by k-means and its variants [4–7]. This approach is preferred when the goal is to invent an algorithm with the best possible optimization performance. Internal evaluation is also useful when no ground truth exists.

[1] http://www.cad.zju.edu.cn/home/dengcai/Data/Clustering.html

[2] https://github.com/uef-machine-learning/ClusterAccuracy

*Internal validity indexes* can also be used when the number of clusters ($k$) is unknown and a part of the problem. The main advantage of internal indexes is that they allow comparison when the number of clusters differs. Measures with reasonably good performance include *Silhouette coefficient* [8], *Calinski-Harabasz* [9], *WB-index* [10], and *kCE-index* [11]. For an implementation of these and others, we refer to [12]. In the case of internal evaluation, it would be sensible to use the same function also as the objective function in the algorithm. Note that many complex and highly heuristic algorithms do not use any explicit objective function.

External evaluation can be used when we have ground truth clustering for the data. Such measures are called *external validity indexes*. The most common indexes found in the literature are *adjusted rand index* (ARI) [13], *normalized mutual information* (NMI) [14], and the mysterious measure called *clustering accuracy* (ACC), which belongs to a wider class of set-matching based measures [3]. These measures first make a mapping between the detected and ground truth clusters. The matched clusters are then compared to find out how many same points they share. The overall measure is obtained by summing the total number of shared points. All measures in this class have the same general principle and differ only in the design details.

The third approach is to measure the performance of the application where clustering is used merely as a component in a large and complex pattern recognition application. The evaluation is done simply by comparing how much better the application performs by changing the clustering component. Measures such as classification accuracy and prediction errors can be used here. This approach is useful especially when we do not have ground truth, and when it is not clear which clustering method or objective function would best fit the application.

Measuring application-level performance has two major drawbacks. First, the effect of a single component (clustering) tends to remain marginal. For example, the exact choice of clustering algorithm had only a minor effect on the overall speaker verification performance [15] as long as some reasonably good algorithm was chosen. Even the simple k-means (with repeats) performed well. Only the poorest clustering algorithm showed inferior results and only in the case of the smallest model size ($k = 16$).

Second, this measure often leads to a biased comparison because researchers tend to add application-specific features to their clustering component. This may lead to a flawed interpretation that the improvement was due to better clustering even if the actual improvement was caused by adding the application-level enhancement hidden within the clustering. Examples are *Olivetti faces* and *Letters* datasets, which are thumbnail versions of the face- and character-recognition applications. In image data, even a small spatial shift of the pixels can cause major changes that prevent the object from clustering correctly. It is also easy to improve the performance simply by utilizing spatial correlations via some hidden pre-processing step. Thus, the improvement may be due to incorporating application-specific knowledge into the clustering process, and such results would not generalize to other types of data.

## 3. History of clustering accuracy

We next summarize the use of clustering accuracy in recent pattern recognition papers based on a literature review. We searched on Pattern Recognition[3] (PR), Pattern Recognition Letters[4] (PRL),

[3] https://www.sciencedirect.com/journal/pattern-recognition

[4] https://www.sciencedirect.com/journal/pattern-recognition-letters

and IEEE Xplore[5] (IEEE) archives using "clustering accuracy" as the keyword. We downloaded the papers for further screening and selected those that presented a new clustering method or used clustering in some other application. The search was performed in August 2021, and it covered papers published during 2021 thus far. The aim was to have a reasonable size sample instead of trying to discover all possible clustering-related papers. The final list has 60 papers as summarized in Table 1.

**Table 1.** Summary of papers using normalized mutual information (NMI), clustering accuracy (ACC) and purity. The number of papers citing the source of ACC, or defining it, are also shown.

| | Papers | NMI | ACC | Purity | ACC cited | ACC defined |
|---|---|---|---|---|---|---|
| PR | 26 | 19 | 18 | 7 | 4 | 5 |
| IEEE | 24 | 4 | 5 | 1 | 4 | 3 |
| PRL | 10 | 4 | 5 | 4 | 1 | - |
| Total | 60 | 27 | 26 | 12 | 9 | 8 |

### *3.1. Literature analysis*

The most common themes mentioned in the papers are subspace clustering (10), multi-view, ensemble, fusion, or co-clustering (8), spectral clustering (6), time series (3), and non-negative matrix factorization (3). Among different algorithms, k-means (7), fuzzy c-means (4), deep learning (4), and agglomerative clustering (3) were the most common. Others mentioned include genetic algorithm, DBSCAN, Gaussian mixture model, kernel k-means, kernel FCM, multitype, semi-supervised, feature extraction, motion clustering, gene expression, geo-locations, location points, trajectories, video summarization, watershed segmentation, graphs and graph-based.

Table 1 also summarizes the measures used. The most common are NMI (27), clustering accuracy (26), and purity (12). Clustering error was sometimes named and defined as 1 ACC [16,17]. Other external indexes used are F measure (10), adjusted Rand index (8), precision (5), Rand index (4), and entropy (3). For more detailed documentation of these we refer to [3].

Application-specific measures were also commonly used (17), especially in IEEE journals and conferences which often had more application-oriented themes. Among these, classification and prediction error were the most typical measures used. Internal indexes were occasionally used including Silhouette (4), Calinski-Harabasz, Davies-Bouldin, Elbow method, Gap statistics, Hartigan, I-nice, and Kappa.

Among the 26 papers that used clustering accuracy, only nine gave citations, and only eight made any attempt to define the measure. IEEE papers are the most precise in this regard, whereas only one PRL paper cites any source, and none documented the measure. A typical citation refers to another paper (often by the same authors) that uses the same measure.

Additional citation is sometimes provided to a textbook such as [18], which includes an explanation of the Hungarian algorithm by Kuhn and Munkres [19,20]. This is an optimal method for the pairing (assignment) problem used in set-matching-based measures to pair the clusters, but this does not document the overall measure properly.

[5] https://ieeexplore.ieee.org/search/advanced

We have provided some examples showing how the clustering accuracy has been cited:

- *"Permutation mapping function that maps the predicted cluster label to the equivalent ground truth label"*; [21–23].
- *"Best mapping function that permutes the clustering results to match the ground truth labels"*; [24].
- "*Mapping function map that is optimized overall all possible one-to-one mappings between the clusters and the ground truth labels*"; [25]

These might be understandable for an expert reader, but many times the papers describe this step vaguely as "discover the one-to-one relationship between predicted clusters and real classes" [26], or as "we have to match cluster labels first, then use the matched labels to calculate precision by Kuhn-Munkres" [27]. These leave too much guessing and do not help the reader.

### *3.2. Origin of the accuracy measure*

We followed the citations to trace the source of the accuracy measure. The searches most often came to a dead end for one of the following reasons:

- The search ended with another paper using the accuracy measure; usually containing a brief definition of the measure [28–33].
- The search ended with a paper that uses a completely different measure, such as entropy-based [34], adjusted mutual information, adjusted rand index [35], F-measure [36–38], or a review paper studying other methods but not clustering accuracy.

A likely source for many citations is the web page by Deng Cai[6], which includes simple Matlab code for the measure. The web page refers to two papers, of which one is from 2005 [39]. This paper further cites [33] as its source. The paper [33] is the first (among those we found) that has a clear description of the measure, including a mapping equation (see Table 2). It also cites the Kuhn-Munkres algorithm by a book reference [18], which then became a common citation in other papers.

The paper [33] does not cite any source, but the reference list includes another paper which is the likely source: [40]. This paper defines the mapping problem as perfect matching of a complete weighted bipartite graph and gives a textbook citation to Kuhn-Munkres [18]. However, a greedy algorithm was used in their experiments in [40] instead of the optimal Hungarian algorithm.

Zha et al. [40] cite Slonim and Tishby [41] as the source for the clustering accuracy measure. It is likely the first to describe the measure, since other set-matching based measures were invented around the same time: Criterion-H (CH) in 2001 by Meila and Heckerman [2], and Normalized van Dongen (NVD) by van Dongen in 2000 [1]. The description in [41] is very brief and difficult to understand, but they correctly recognized what many others have ignored: the number of clusters must be the same as in the ground truth. We will return to this limitation a bit later.

The history of the clustering accuracy measure is summarized in Figure 2. After 2005, papers mostly cite [39], or more recent papers that use the same measure.

[6] http://www.cad.zju.edu.cn/home/dengcai/Data/Clustering.html

**Table 2.** Definitions of clustering accuracy from existing papers. Similarity refers to the measurement of a cluster *A* and the mapped ground truth cluster *G*.

| Measure | Year | Ref. | Similarity | Mapping |
|---|---|---|---|---|
| ACC | 2000 | Several authors | $\|A \cap G\|$ | Optimal pairing |
| NVD | 2000 | [1] | $\|A \cap G\|$ | Matching (2-ways) |
| CH | 2001 | [2] | $\|A \cap G\|$ | Greedy pairing |
| Purity | 2011 | [42] | $\|A \cap G\|$ | Matching ($G$®$P$) |
| FM | 2012 | [43] | $\|A\| \times$ SD | Matching ($G$®$P$) |
| CI | 2014 | [44] | 0/1 | Matching (2-ways) |
| CSI | 2014 | [44] | $\|A \cap G\|$ | Matching (2-ways) |
| CR | 2014 | [45] | 0/1 | Greedy pairing |
| GCI | 2016 | [46] | 0/1 | Matching (2-ways) |
| PSI | 2016 | [3] | BB | Optimal pairing |

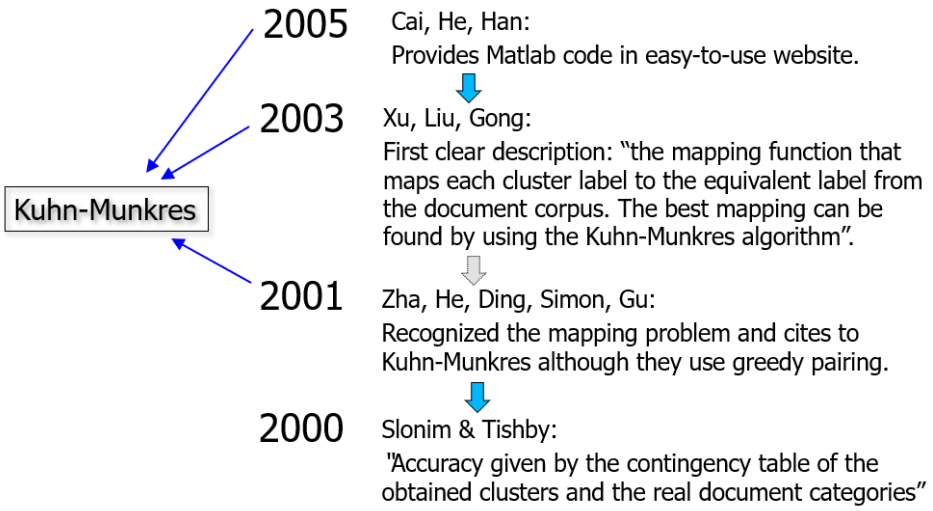


**Figure 2.** History of the clustering accuracy measure. Arrows represent citations. Blue means a direct citation, and gray a citation not directly related to clustering accuracy.

## 4. Definition of clustering accuracy

The goal of clustering is to partition $n$ points $\{x_i\}$ into $k$ clusters. The cluster assignment of a point $x_i$ is denoted as $a_i \in [1, k]$. The ground truth label of the point is denoted as $g_i$. The *clustering accuracy* measure was first defined formally in [33] as:

$$\frac{1}{n} \cdot \sum_{i=1}^{n} \delta\left(g_i, map(a_i)\right), \tag{1}$$

where $d$ is *Kronecker delta* which gives value 1 when the two parameters are equal ($g_i$=map($a_i$)) and 0 otherwise. In other words, we measure the number of points that are correctly clustered with respect to the mapping. This definition itself is valid but insufficient without defining the map function. Another slightly different notation for the same has also been presented as:

$$\frac{1}{n} \cdot \sum_{i=1}^{n} 1 \cdot \{g_i = map(a_i)\}. \quad (2)$$

The main question is how to map between the cluster labels and the ground truth. This is illustrated in Figure 3 (right), where cluster 3 matches perfectly to the ground-truth cluster 2, and the best match for the cluster 1 is the ground-truth cluster 1. However, there are O(*k*!) possible mappings, and it is not trivial which one should be selected.

Clustering accuracy belongs to a wider class of set-matching-based measures [3]. Selecting the mapping is one of the three design questions that must be answered:

- How to map the clusters and the ground truth?
- How to measure similarity between two clusters?
- Do we normalize and how?

The earliest works appeared around 2000, leading to three different measures: *normalized van Dongen* (NVD) [1], *Criterion H* (CH) [2], and *clustering accuracy* (ACC), the focus of this paper. Other measures in this class include Purity, FM, centroid index (CI, GCI), centroid similarity index (CSI), centroid ratio (CR) and pair set index (PSI). Table 2 summarizes the set-matching-based measures.

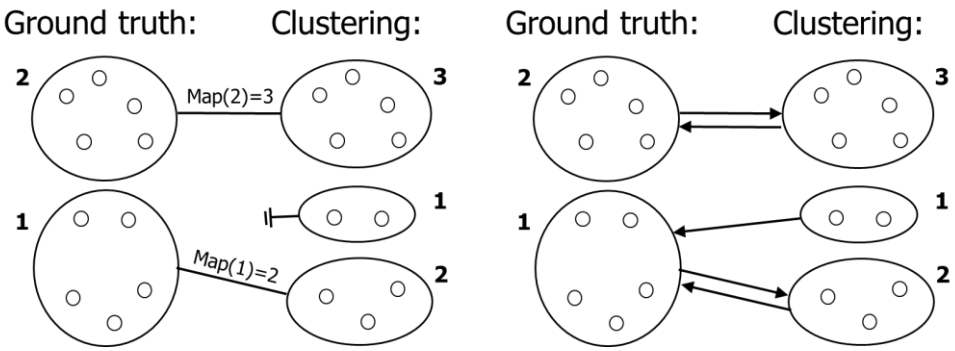


**Figure 3.** Example of pairing (left) and matching (right). In pairing, every cluster can have only one matching cluster in the ground truth, and vice versa. Matching is asymmetric and a cluster can be mapped several times.

### *4.1. Mapping*

Most papers that use clustering accuracy refer to the mapping as an assignment problem and solve it optimally using the Hungarian algorithm, which was originally published by Kuhn [19], and later shown to be solvable in polynomial time by Munkres [20]. In the context of clustering accuracy, it is usually referred to as Kuhn-Munkres, with a citation to a textbook [18].

The methods in Table 2 differ in how they perform the mapping. Only ACC and PSI use the optimal pairing method, whereas CH, CR and ACC variant in [40] use greedy pairing instead. In fact, it is not even necessary to consider the problem as a pairing problem. NVD and most other methods perform nearest-neighbor matching, which allows the same cluster to be matched several times. This avoids the problem of having a mismatch in the number of clusters.

The direction of the matching also matters. It should be performed in both directions, which is the case with NVD, CI, CSI, and GCI. Purity and FM perform matching only from the clustering to

the ground truth. However, this can provide erroneous results when there are more clusters than in the ground truth. For example, purity would provide the maximum score if all clusters were perfect subsets of the ground truth, regardless of how many clusters there are.

### *4.2. Similarity measure and normalization*

Mapping also requires a similarity measure. The classical measures (NVD, CH) and Purity count the total number of shared points in the mapped (or paired) clusters. The literature covering ACC does not document how this is done, but the existing implementation follows the above-mentioned approach. However, this is not the only approach. One could normalize the score by the cluster size, which would make the measure invariant to the cluster sizes.

Normalization by clustering size can be done using any classical-set-matching measure. FM uses Sorensen-Dice but only regarding the size of the ground truth clusters. PSI uses Braun-Banquet. Surprisingly, the only measure using the well-known Jaccard is GCI. The other measures (CI, CSI, and CR) perform the matching by finding the nearest centroid without any point-level calculations. This limits their use to centroid-based clustering, whereas GCI [46] uses cluster similarity in the matching step to overcome this limitation.

A similarity measure is also needed in the calculation of the final score, which is usually in the range [0,1]. This is done by summing up the total number of shared points ($|A \cap G|$) in every cluster pair divided by the total number of points ($n$). FM and PSI sum the normalized similarity scores of each cluster. PSI is the only *chance-corrected* measure which have 0 as the expected value for random clustering [3].

Cluster-level indexes (CI, CR, GCI) perform a cluster-level measure providing a score in the range [0, $k$] so that value 0 indicates that all clusters are correctly located. Any value higher than 0 indicates how many wrongly detected clusters. This result is very intuitive and can indicate whether the correct clustering was found. Normalization by the number of clusters can be done by dividing as CI/$k$ if wanted [3].

### *4.3. Summary*

Most measures work reasonably well in practice. It is difficult to find a research paper where the clustering result differs significantly with different measures, such as ACC or NMI. The relative performance of the clustering methods remains consistent regardless of the measure used.

The most challenging condition is reported in [47]. It used a stability-based approach to solve the number of clusters. The approach works if all parameters (e.g., sampling rate) are properly set, the assumption of having spherical clusters holds, and an accurate clustering algorithm such as random swap [5] or genetic algorithm [4] is used (k-means did not work). The choice of clustering evaluation measure was shown not to be critical. All point-level measures worked except Rand index.

The result in [47] indicates that the exact choice of the clustering-evaluation measure is not critical as long as Rand index is not used. Otherwise, the choice is practical: does the measure generalize to all types of data, is there a software package available, and is the measure fast enough?

Centroid index is a cluster-level measure that gives an intuitive and easy-to-interpret result by telling exactly how many of the clusters are incorrect. In specific, CI = 0 explicitly tells that all

clusters are roughly at their correct location with respect to the ground truth. Due to this intuitive interpretation, we recommend using CI as a cluster-level measure.

ACC provides a point-level measure with more fine-tuned result where every point contributes. However, the result is just a number in scale [0,1] without any clear interpretation. For example, ACC = 0.84 is better than ACC = 0.79 but how much better? And what does it mean in practice?

The precision of ACC becomes important when the measure is used by an algorithm instead of given to a human to interpret the result. For instance, the stability-based measure [47] requires a very precise measure to detect even small variations between the two clustering results compared. To sum up, we recommend CI as cluster-level and ACC as a point-level measure. Although ACC does not apply any normalization, there is nothing fundamentally wrong with it because in a point-level measure it is expected that all points contribute equally.

## 5. Clustering accuracy revisited

There is no fundamental flaw in the existing clustering accuracy measure. It has only a minor issue in that the number of clusters must match the ground truth because of pairing. However, a simple patch for this is to add empty clusters to the one (clustering or ground truth) with fewer clusters. Figure 4 demonstrates the idea.

Nearest-neighbor matching does not have this problem. It is also faster than the optimal pairing which requires O($k^3$) time, which can be impractical for a large number of clusters. The Hungarian algorithm is also more complex to implement, but this is not an issue for are using an existing package. So, which method should we choose?

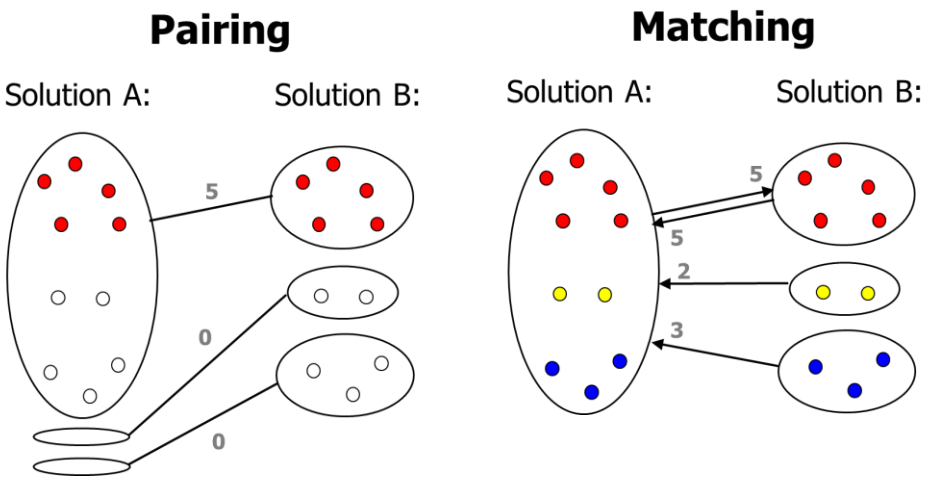


**Figure 4.** Example of optimal pairing (left) and nearest-neighbor matching (right). The result of pairing is 5/10 = 50%, and the result of matching (5+10)/20 = 75%. Matching tends to provide a higher score since it can find a perfect match when mapping from solution B to solution A.

### *5.1. Pairing versus matching?*

To answer the question, we compare the two variants (ACC-pair, ACC-match) with artificially generated data with ten clusters each consisting of 500 points. There is no actual data but only the ground truth class labels, see Figure 5. The total size of the dataset is $n$ = 5000. Clustering is then

created by making random errors to the cluster labels as detailed in the following. The process is adopted from [3].

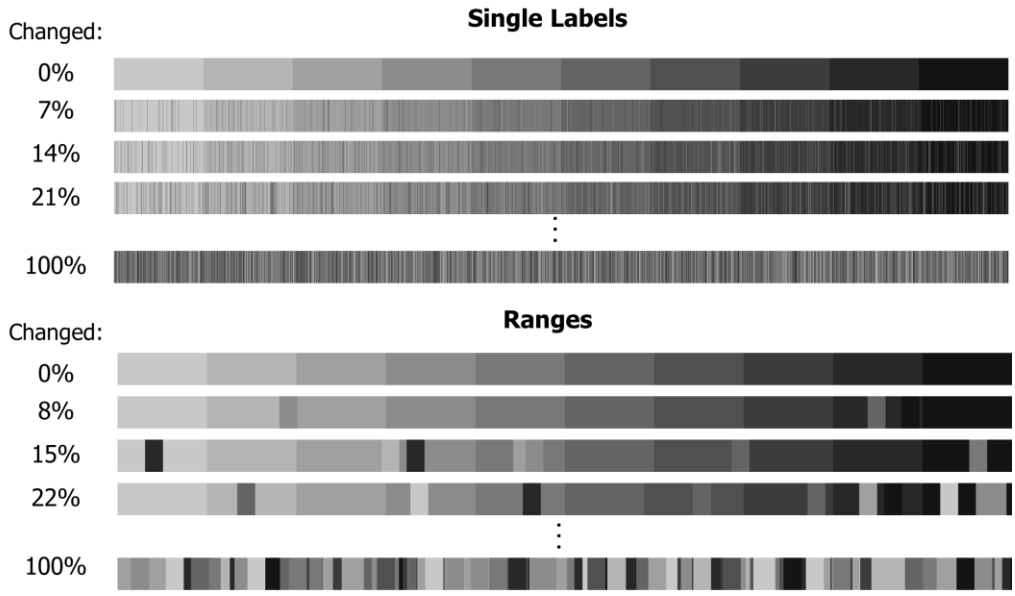


**Figure 5.** Artificially generated $n$ = 5000 data (http://cs.uef.fi/sipu/datasets/) divided into $k$=10 clusters organized from left to right. The top row (0%) corresponds to the ground truth. The other rows represent the clustering results generated by swapping random points into a wrong cluster. The percentage is the proportion of swapped points. In the above (single labels), the points are swapped independently. In the below (ranges), the points are swapped in groups of 100.

In the first experiment, errors are created by changing the cluster label of a randomly chosen datapoint to another (randomly chosen) cluster label. The number of errors is a parameter which we vary from 0% to 100%. The effect is demonstrated on GCI, NMI and two variants of ACC, one using pairing and the other using matching. The results are reported in Figure 6 (left).

The most important observation is that both ACC variants perform identically. It does not matter which variant is used, both NMI and ACC decrease consistently with the increasing error rate. The behavior of ACC is linear but does not reach zero. The behavior of NMI is slightly curved but reaches zero due to normalization. GCI reacts only for error rates above 90%. This is because the location of the clusters remains correct even with high rates of point-level errors.

The second experiment tries to simulate the type of non-uniform errors that clustering algorithms typically commit. We randomly select 100 consecutive points and change all their cluster labels to another randomly chosen cluster. Results are reported in Figure 6 (right).

The results of the second experiment do not differ much from the first. The two variants of ACC (pairing and matching) give nearly identical values. There is a slight difference with the higher number of errors (randomized %), where matching provides slightly higher scores. The results of NMI are less curvy but do not reach zero. GCI reacts earlier due to the errors being grouped and some clusters can no longer be correctly detected.

Figure 7 demonstrates cases when ACC with matching provides a higher score than ACC with pairing. In the case of 3-vs-3 clustering, pairing always gives a 50% score, whereas matching

gives 75% because it allows multiple clusters to map to the same cluster. The same tendency shows in the cases of mismatching number of clusters (3-vs-4 and 3-vs-2). Here the pairing variant gives a 75% score and the matching variant 87%.

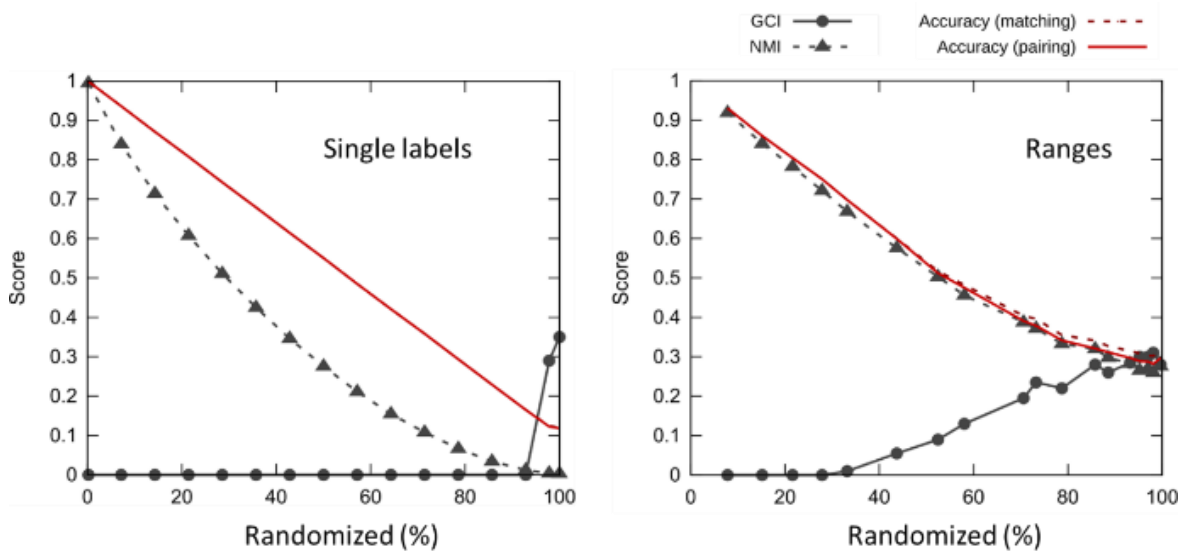


**Figure 6.** Effect of random errors created for each point independently (single labels), and jointly for groups of 100 points (ranges). The number of errors (Randomized %) is the parameter to control the error rate. Results are averaged over 20 runs.

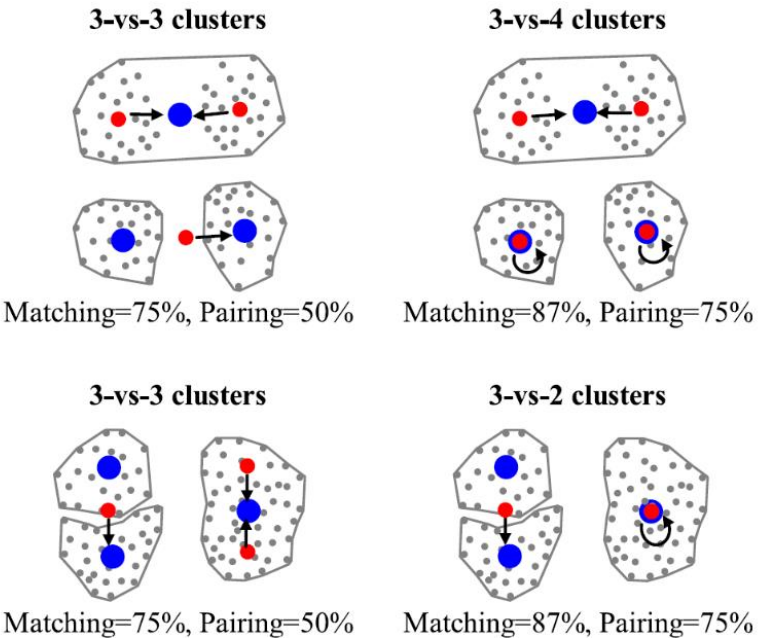


**Figure 7.** Pathological cases when the matching variant provides a higher score than the pairing variant. Here two clustering results (red and blue) are compared. Cluster borders are shown for the blue. Arrows are shown only for the mapping from red to blue.

*5.2. Algorithm*

The pseudo code of clustering accuracy is given below with two variants: accuracy with pairing (ACC-pair) and accuracy with matching (ACC-match). The first variant is backward-compatible so that the result is directly comparable to those reported in literature. The only difference is the minor fix so that the number of clusters can be different in the two solutions ($kA \neq kB$). This is done by adding empty rows or columns to the contingency matrix, whichever is smaller.

The new variant is recommended for two reasons. First, it avoids the use of the Hungarian algorithm completely. Second, it is computationally more efficient. The matching can also be performed in cases of different cluster sizes without any modifications. The key idea in the implementation is to apply the matching twice, once for each direction. This is implemented by switching the order of the partition label parameters (*La* and *Lb*) before calling the function second time. The final score is the minimum of two matching results.

The time complexity of the matching variant is O($N$). A straightforward implementation of the contingency table requires O(max{$k^2$, $N$}) where $k$ = max{$kA$, $kB$}. However, when $k^2 > N$, most elements in the contingency table will become zero as it can have at most $N$ non-zero elements. For this reason, we use a hash table implementation for the sparse matrix, which speeds up the ARGMAX step in the pseudocode from O($k$) to O($N$/$k$) and maintains the total time complexity at O($N$).

The time complexity of the pairing variant is O($k^3$). A straightforward implementation of the Kuhn-Munkres algorithm would yield to time complexity of O(max{$k^3$, $N$}). We use the Jonker-Volgenant variant [48] of the Kuhn-Munkres algorithm, which has the same worst-case time complexity, but it works faster for sparse matrices. This helps when $k^3 > N$. We will show in the next section that the O($k^3$) worst case does not necessarily occur, but the actual observed processing times closely follow the simulated curve of O($k^2$).

```
ACC-pair(La,Lb,mappingMethod):
kA = Number of unique labels in La
kB = Number of unique labels in Lb
N = length of La and Lb
contg = ContingencyMatrix(La,Lb,N,kA,kB,"pairing")
mapAtoB = hungarian(-contg)
return CountMatches(La,Lb,mapAtoB)

ACC-match(La,Lb):
acc1 = MatchOneWay(La,Lb)
acc2 = MatchOneWay(Lb,La)
return min(acc1,acc2)
```

```
MatchOneWay(La,Lb):
contg = ContingencyMatrix(La,Lb,N,kA,kB,"matching")
FOR i=1:kA
  mapAtoB[i] = ARGMAX{contg[i][j]: j=1,...,kB}
return CountMatches(La,Lb,mapAtoB)

CountMatches(La,Lb,N,mapAtoB):
correct=0
FOR i=1..N
  IF Lb[i] == mapAtoB[La[i]]
    correct+=1
return correct/N

ContingencyMatrix(La,Lb,N,kA,kB,mappingMethod)
IF mappingMethod == "pairing"
  contg = matrixOfZeros(MAX{kA,kB},MAX{kA,kB})
ELSE
  # Sparse matrix implemented using hash tables
  contg = matrixOfZeros(kA,kB)
FOR i=1..N
  # Number of points belonging to both clusters
  contg[La[i]][Lb[i]] += 1
```

### *5.3. Time benchmark*

We conducted a short benchmark to test the time complexity performance of the two Python implementations. We generated random labels for datasets of varying size and number of clusters. The dataset size varied from $N$ = 10,000 to $N$ = 50,000; and the number of clusters from $k$ = 100 to $k$ = 50,000 (restricting to $k < N$). We then measured the processing time to calculate ACC.

The results in Figure 8 show that the processing time for ACC-match increases linearly with respect to both the number of clusters, $k$, and the size of the dataset, $N$. As the number of clusters is always smaller than the size of the data, $k < N$, the implementation has O($N$) time complexity.

The ACC-pair variant runs much faster than the theoretical O($k^3$) worst-case time complexity. It seems to be very close to O($k^2$) due to the sparsity of the input matrix. For example, if $k$ = 1,000 and $N$ = 10,000, at most 1% of the cells contain non-zero values. We can see also that the size of the dataset does not significantly affect the processing time, and that the number of clusters is the limiting factor.

We managed to run the pairing algorithm up to $N$ = 50,000 and $k$ = 46,300. For larger $k$ the implementation ran out of memory. However, clustering with values of $k$ this high is rarely performed, so we conclude that the ACC-pair is useful for all relevant clustering problems.

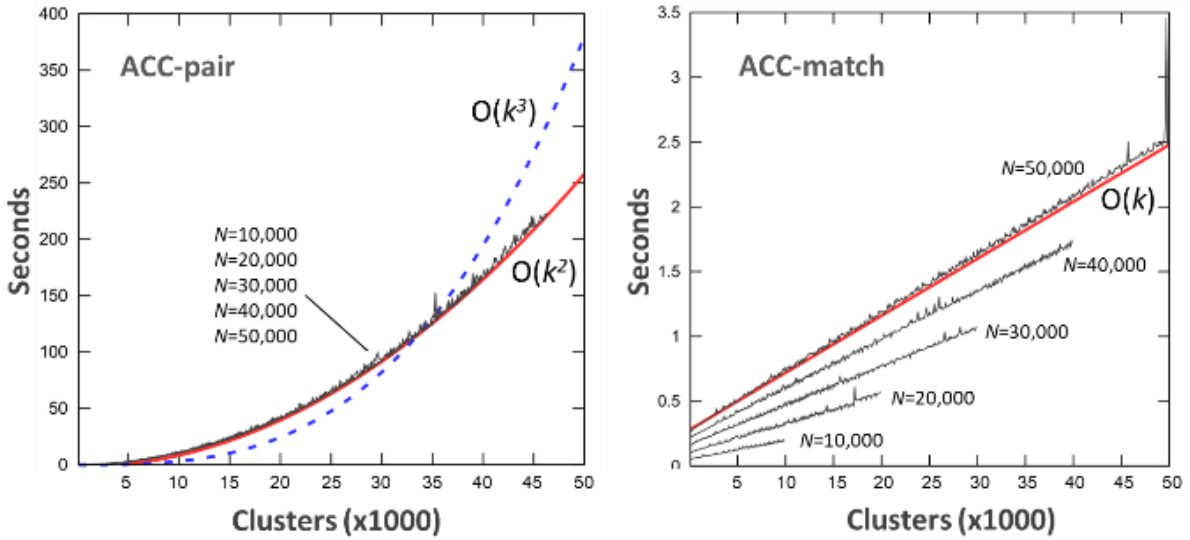


**Figure 8.** Run-time benchmarking with varying number of clusters ($k$) of the ACC-pair (left) and ACC-match (right) implementations. The theoretical time complexities are shown with red and blue lines.

## 6. Conclusions

We have documented the widely used clustering accuracy measure (ACC) and its development history in a literature review. It is a valid measure and belongs to a wider class of set-matching-based measures. Compared to normalized mutual information (NMI), ACC decreases linearly with the amount of clustering errors but does not reach zero because it lacks normalization. Otherwise, there is no significant difference in their performance.

The popularity of the clustering accuracy is addressed mostly to its appealing name and a publicly available Matlab package. We have modified the measure by correcting the inability to deal with mismatched numbers of clusters. In addition to the optimal pairing of the Hungarian algorithm, we also implemented the nearest-neighbor matching variant. Both variants provide identical values or with minor differences in the case of high error rates in certain pathological cases. In our view, both variants are useful. The matching variant has lower time complexity and has become the preferred choice as it can also work for a very high number of clusters.

## Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools for this article.

## Conflict of interest

All authors declare no conflict of interest regarding the publication of this paper.

## References

1. S. van Dongen, *Performance criteria for graph clustering and Markov cluster experiments*, Amsterdam: Centrum voor Wiskunde en Informatica, 2000.

2. M. Meila, D. Heckerman, An experimental comparison of model based clustering methods, *Mach. Learn.*, **41** (2001), 9–29. https://doi.org/10.1023/A:1007648401407
3. M. Rezaei, P. Fränti, Set matching measures for external cluster validity, *IEEE Trans. Knowl. Data Eng.*, **28** (2016), 2173–2186. https://doi.org/10.1109/TKDE.2016.2551240
4. P. Fränti, Genetic algorithm with deterministic crossover for vector quantization, *Pattern Recogn. Lett.*, **21** (2000), 61–68. https://doi.org/10.1016/S0167-8655(99)00133-6
5. P. Fränti, Efficiency of random swap clustering, *J. Big Data*, **5** (2018), 13. https://doi.org/10.1186/s40537-018-0122-y
6. B. Fritzke, Breathing k-means, arXiv: 2006.15666.
7. C. Baldassi, Recombinator-k-means: an evolutionary algorithm that exploits k-means++ for recombination, *IEEE Trans. Evolut. Comput.*, **20** (2022), 991–1003. https://doi.org/10.1109/TEVC.2022.3144134
8. P. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, *J. Comput. Appl. Math.*, **20** (1987), 53–65. https://doi.org/10.1016/0377-0427(87)90125-7
9. T. Calinski, J. Harabasz, A dendrite method for cluster analysis, *Commun. Stat.*, **3** (1974), 1–27. https://doi.org/10.1080/03610927408827101
10. Q. Zhao, P. Fränti, WB-index: a sum-of-squares based index for cluster validity, *Data Knowl. Eng.*, **92** (2014), 77–89. https://doi.org/10.1016/j.datak.2014.07.008
11. J. Hämäläinen, S. Jauhiainen, T. Kärkkäinen, Comparison of internal clustering validation indices for prototype-based clustering, *Algorithms*, **10** (2017), 105. https://doi.org/10.3390/a10030105
12. M. Niemelä, S. Äyrämö, T. Kärkkäinen, Toolbox for distance estimation and cluster validation on data with missing values, *IEEE Access*, **10** (2021), 352–367. https://doi.org/10.1109/ACCESS.2021.3136435
13. L. Hubert, P. Arabie, Comparing partitions, *J. Classif.*, **2** (1985), 193–218. https://doi.org/10.1007/BF01908075
14. T. Kvålseth, Entropy and correlation: some comments, *IEEE Trans. Syst. Man Cybern.*, **17** (1987), 517–519. https://doi.org/10.1109/TSMC.1987.4309069
15. T. Kinnunen, I. Sidoroff, M. Tuononen, P. Fränti, Comparison of clustering methods: a case study of text-independent speaker modeling, *Pattern Recogn. Lett.*, **32** (2011), 1604–1617. https://doi.org/10.1016/j.patrec.2011.06.023
16. C. Peng, Z. Kang, Q. Cheng, Integrating feature and graph learning with low-rank representation, *Neurocomputing*, **249** (2017), 106–116. https://doi.org/10.1016/j.neucom.2017.03.071
17. M. Wu, B. Schölkopf, A local learning approach for clustering, In: *Advances in neural information processing systems 19*, Cambridge: MIT Press, 2007, 1529–1536.
18. L. Lovasz, M. D. Plummer, *Matching theory*, North Holland: Akademiai Kiado, 1986.
19. H. W. Kuhn, The Hungarian method for the assignment problem, *Naval Research Logistics Quarterly*, **2** (1955), 83–97. https://doi.org/10.1002/nav.3800020109
20. J. Munkres, Algorithms for the assignment and transportation problems, *J. Soc. Ind. Appl. Math.*, **5** (1957), 32–38. https://doi.org/10.1137/0105003
21. X. Liu, M. Li, C. Tang, J. Xia, J. Xiong, L. Liu, et al., Efficient and effective regularized incomplete multi-view clustering, *IEEE Trans. Pattern Anal.*, **43** (2021), 2634–2646. https://doi.org/10.1109/TPAMI.2020.2974828

22. M. Alshammari, J. Stavrakakis, M. Takatsuka, Refining a k-nearest neighbor graph for a computationally efficient spectral clustering, *Pattern Recogn.*, **114** (2021), 107869. https://doi.org/10.1016/j.patcog.2021.107869
23. J. Ma, Y. Zhang, L. Zhang, Discriminative subspace matrix factorization for multiview data clustering, *Pattern Recogn.*, **111** (2021), 107676. https://doi.org/10.1016/j.patcog.2020.107676
24. J. Liu, S. Teng, L. Fei, W. Zhang, X. Fang, Z. Zhang, et al., A novel consensus learning approach to incomplete multi-view clustering, *Pattern Recogn.*, **115** (2021), 107890. https://doi.org/10.1016/j.patcog.2021.107890
25. X. Yang, J. Shi, C. Wang, Y. Zhou, Z. Zhou, T. Chen, et al., Binary clustering for deep network trained by feature growth, *IEEE Access*, **9** (2021), 8354–8366. https://doi.org/10.1109/ACCESS.2020.3047467
26. Y. Xu, S. Chen, J. Li, L. Luo, J. Yang, Learnable low-rank latent dictionary for subspace clustering, *Pattern Recogn.*, **120** (2021), 108142. https://doi.org/10.1016/j.patcog.2021.108142
27. Y. Chen, L. Zhou, N. Bouguila, C. Wang, Y. Chen, J. Du, BLOCK-DBSCAN: fast clustering for large scale data, *Pattern Recogn.*, **109** (2021), 107624. https://doi.org/10.1016/j.patcog.2020.107624
28. R. Wang, F. Nie, Z. Wang, H. Hu, X. Li, Parameter-free weighted multi-view projected clustering with structured graph learning, *IEEE Trans. Knowl. Data Eng.*, **32** (2020), 2014–2025. https://doi.org/10.1109/TKDE.2019.2913377
29. C. Peng , Z. Kang , S. Cai , Q. Cheng, Integrate and conquer: double-sided two-dimensional k-means via integrating of projection and manifold construction, *ACM Tran. Intel. Syst. Tech.*, **9** (2018), 57. https://doi.org/10.1145/3200488
30. C. Peng, Z. Kang, Y. Hu, J. Cheng, Q. Cheng, Robust graph regularized nonnegative matrix factorization for clustering, *ACM Trans. Knowl. Discov. Data*, **11** (2017), 33. https://doi.org/10.1145/3003730
31. D. Cai, X. He, X. Wu, J. Han, Non-negative matrix factorization on manifold, *Proceedings of IEEE International Conference on Data Mining*, 2008, 63–72. https://doi.org/10.1109/ICDM.2008.57
32. D. Cai, X. He, J. Han, Locally consistent concept factorization for document clustering, *IEEE Trans. Knowl. Data Eng.*, **23** (2011), 902–913. https://doi.org/10.1109/TKDE.2010.165
33. W. Xu, X. Liu, Y. Gong, Document clustering based on non-negative matrix factorization, *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, 2003, 267–273. https://doi.org/10.1145/860435.860485
34. M. Yin, J. Gao, Z. Lin, Laplacian regularized low-rank representation and its applications, *IEEE Trans. Pattern Anal.*, **38** (2016), 504–517. https://doi.org/10.1109/TPAMI.2015.2462360
35. R. Liu, H. Wang, X. Yu, Shared-nearest-neighbor-based clustering by fast search and find of density peaks, *Inform. Sciences*, **450** (2018), 200–226. https://doi.org/10.1016/j.ins.2018.03.031
36. Y. Chen, S. Tang, L. Zhou, C. Wang, J. Du, T. Wang, et al., Decentralized clustering by finding loose and distributed density cores, *Inform. Sciences*, **433–434** (2018), 510–526. https://doi.org/10.1016/j.ins.2016.08.009
37. Y. Chen, S. Tang, S. Pei, C. Wang, J. Du, N. Xiong, DHeat: a density heat-based algorithm for clustering with effective radius, *IEEE Trans. Syst. Man Cybern.*, **48** (2018), 649–660. https://doi.org/10.1109/TSMC.2017.2745493

38. E. Müller, S. Günnemann, I. Assent, T. Seidl, Evaluating clustering in subspace projections of high dimensional data, *Proc. VLDB Endow.*, **2** (2009), 1270–1281. https://doi.org/10.14778/1687627.1687770
39. D. Cai, X. He, J. Han, Document clustering using locality preserving indexing, *IEEE Trans. Knowl. Data Eng.*, **17** (2005), 1624–1637. https://doi.org/10.1109/TKDE.2005.198
40. H. Zha, X. He, C. Ding, M. Gu, H. Simon, Spectral relaxation for k-means clustering, In: *Advances in neural information processing systems 14*, Cambridge: MIT Press, 2001, 1057–1064.
41. N. Slonim, N. Tishby, Document clustering using word clusters via the information bottleneck method, *Proceedings of the 23rd annual international ACM SIGIR conference on Research and development in information retrieval*, 2000, 208–215. https://doi.org/10.1145/345508.345578
42. E. Rendón, I. Abundez, A. Arizmendi, E. M. Quiroz, Internal versus external cluster validation indexes, *International Journal of Computer Networks and Communications*, **5** (2011), 27–34.
43. M. C. P. de Souto, A. L. V. Coelho, K. Faceli, T. C. Sakata, V. Bonadia, I. G. Costa, A comparison of external clustering evaluation indices in the context of imbalanced data sets, *Proceedings of Brazilian Symposium on Neural Networks*, 2012, 49–54. https://doi.org/10.1109/SBRN.2012.25
44. P. Fränti, M. Rezaei, Q. Zhao, Centroid index: cluster level similarity measure, *Pattern Recogn.*, **47** (2014), 3034–3045. https://doi.org/10.1016/j.patcog.2014.03.017
45. Q. Zhao, P. Fränti, Centroid ratio for pairwise random swap clustering algorithm, *IEEE Trans. Knowl. Data Eng.*, **26** (2014), 1090–1101. https://doi.org/10.1109/TKDE.2013.113
46. P. Fränti, M. Rezaei, Generalized centroid index to different clustering models, *Proceedings of Structural, Syntactic, and Statistical Pattern Recognition*, 2016, 285–296. https://doi.org/10.1007/978-3-319-49055-7_26
47. M. Rezaei, P. Fränti, Can the number of clusters be determined by external indices? *IEEE Access*, **8** (2020), 89239–89257. https://doi.org/10.1109/ACCESS.2020.2993295
48. R. Jonker, A. Volgenant, A shortest augmenting path algorithm for dense and sparse linear assignment problems, *Computing*, **38** (1987), 325–340. https://doi.org/10.1007/BF02278710

## Appendix. Papers used for the literature review

*Appendix A. Pattern Recognition*

1. J. Hou, A. Zhang, N. Qi, Density peak clustering based on relative density relationship, *Pattern Recogn.*, **108** (2020), 107554. https://doi.org/10.1016/j.patcog.2020.107554
2. J. Liu, S. Teng, L. Fei, W. Zhang, X. Fang, Z. Zhang, et al., A novel consensus learning approach to incomplete multi-view clustering, *Pattern Recogn.*, **115** (2021), 107890. https://doi.org/10.1016/j.patcog.2021.107890
3. M. Alshammari, J. Stavrakakis, M. Takatsuka, Refining a k-nearest neighbor graph for a computationally efficient spectral clustering, *Pattern Recogn.*, **114** (2021), 107869. https://doi.org/10.1016/j.patcog.2021.107869
4. J. Tan, Z. Yang, Y. Cheng, J. Ye, B. Wang, Q. Dai, SRAGL-AWCL: a two-step multi-view clustering via sparse representation and adaptive weighted cooperative learning, *Pattern Recogn.*, **117** (2021), 107987. https://doi.org/10.1016/j.patcog.2021.107987

5. J. Zhou, W. Pedrycz, X. Yue, C. Gao, Z. Lai, J. Wan, Projected fuzzy C-means clustering with locality preservation, *Pattern Recogn.*, **113** (2021), 107748. https://doi.org/10.1016/j.patcog.2020.107748
6. H. Peng, H. Wang, Y. Hu, W. Zhou, H. Cai, Multi-dimensional clustering through fusion of high-order similarities, *Pattern Recogn.*, **121** (2022), 108108. https://doi.org/10.1016/j.patcog.2021.108108
7. S. Peng, W. Ser, B. Chen, Z. Lin, Robust semi-supervised nonnegative matrix factorization for image clustering, *Pattern Recogn.*, **111** (2021), 107683. https://doi.org/10.1016/j.patcog.2020.107683
8. J. Xia, J. Zhang, Y. Wang, L. Han, H. Yan, WC-KNNG-PC: watershed clustering based on K-nearest-neighbor graph and Pauta criterion, *Pattern Recogn.*, **121** (2022), 108177. https://doi.org/10.1016/j.patcog.2021.108177
9. X. Si, Q. Yin, X. Zhao, L. Yao, Consistent and diverse multi-view subspace clustering with structure constraint, *Pattern Recogn.*, **121** (2022), 108196. https://doi.org/10.1016/j.patcog.2021.108196
10. H. Li, Z. Liu, Multivariate time series clustering based on complex network, *Pattern Recogn.*, **115** (2021), 107919. https://doi.org/10.1016/j.patcog.2021.107919
11. Z. Fu, Y. Zhao, D. Chang, Y. Wang, A hierarchical weighted low-rank representation for image clustering and classification, *Pattern Recogn.*, **112** (2021), 107736. https://doi.org/10.1016/j.patcog.2020.107736
12. J. Saha, J. Mukherjee, CNAK: cluster number assisted K-means, *Pattern Recogn.*, **110** (2021), 107625. https://doi.org/10.1016/j.patcog.2020.107625
13. K. Song, X. Yao, F. Nie, X. Li, M. Xu, Weighted bilateral K-means algorithm for fast co-clustering and fast spectral clustering, *Pattern Recogn.*, **109** (2021), 107560. https://doi.org/10.1016/j.patcog.2020.107560
14. Z. Kang, C. Peng, Q. Cheng, X. Liu, X. Peng, Z. Xu, et al., Structured graph learning for clustering and semi-supervised classification, *Pattern Recogn.*, **110** (2021), 107627. https://doi.org/10.1016/j.patcog.2020.107627
15. Y. Ge, P. Peng, H. Lu, Mixed-order spectral clustering for complex networks, *Pattern Recogn.*, **117** (2021), 107964. https://doi.org/10.1016/j.patcog.2021.107964
16. H. Zhang, H. Li, N. Chen, S. Chen, J. Liu, Novel fuzzy clustering algorithm with variable multi-pixel fitting spatial information for image segmentation, *Pattern Recogn.*, **121** (2022), 108201. https://doi.org/10.1016/j.patcog.2021.108201
17. C. Peng, Q. Zhang, Z. Kang, C. Chen, Q. Cheng, Kernel two-dimensional ridge regression for subspace clustering, *Pattern Recogn.*, **113** (2021), 107749. https://doi.org/10.1016/j.patcog.2020.107749
18. J. Ma, Y. Zhang, L. Zhang, Discriminative subspace matrix factorization for multiview data clustering, *Pattern Recogn.*, **111** (2021), 107676. https://doi.org/10.1016/j.patcog.2020.107676
19. J. Hua, J. Yu, M. Yang, Star-based learning correlation clustering, *Pattern Recogn.*, **116** (2021), 107966. https://doi.org/10.1016/j.patcog.2021.107966
20. M. Yang, K. Sinaga, Collaborative feature-weighted multi-view fuzzy c-means clustering, *Pattern Recogn.*, **119** (2021), 108064. https://doi.org/10.1016/j.patcog.2021.108064

21. S. Huang, Z. Kang, Z. Xu, Q. Liu, Robust deep k-means: an effective and simple method for data clustering, *Pattern Recogn.*, **117** (2021), 107996. https://doi.org/10.1016/j.patcog.2021.107996
22. J. Zheng, P. Yang, G. Shen, S. Chen, W. Zhang, Enhanced low-rank constraint for temporal subspace clustering and its acceleration scheme, *Pattern Recogn.*, **111** (2021), 107678. https://doi.org/10.1016/j.patcog.2020.107678
23. Y. Xu, S. Chen, J. Li, L. Luo, J. Yang, Learnable low-rank latent dictionary for subspace clustering, *Pattern Recogn.*, **120** (2021), 108142. https://doi.org/10.1016/j.patcog.2021.108142
24. S. Baek, G. Yoon, J. Song, S. Yoon, Deep self-representative subspace clustering network, *Pattern Recogn.*, **118** (2021), 108041. https://doi.org/10.1016/j.patcog.2021.108041
25. Y. Chen, L. Zhou, N. Bouguila, C. Wang, Y. Chen, J. Du, BLOCK-DBSCAN: fast clustering for large scale data, *Pattern Recogn.*, **109** (2021), 107624. https://doi.org/10.1016/j.patcog.2020.107624
26. T. Qiu, Y. Li, Enhancing in-tree-based clustering via distance ensemble and kernelization, *Pattern Recogn.*, **112** (2021), 107731. https://doi.org/10.1016/j.patcog.2020.107731

*Appendix B. Pattern Recognition Letters*

1. W. Wang, F. Chen, Y. Ge, S. Huang, X. Zhang, D. Yang, Discriminative deep semi-nonnegative matrix factorization network with similarity maximization for unsupervised feature learning, *Pattern Recogn. Lett.*, **149** (2021), 157–163. https://doi.org/10.1016/j.patrec.2021.06.013
2. H. Q. Ung, C. T. Nguyen, K. M. Phan, V. T. Minh Khuong, M. Nakagawa, Clustering online handwritten mathematical expressions, *Pattern Recogn. Lett.*, **146** (2021), 267–275. https://doi.org/10.1016/j.patrec.2021.03.027
3. E. Rica, S. Álvarez, F. Serratosa, On-line learning the graph edit distance costs, *Pattern Recogn. Lett.*, **146** (2021), 55–62. https://doi.org/10.1016/j.patrec.2021.02.019
4. M. Kutbi, Y. Chang, P. Mordohai, Inlier clustering based on the residuals of random hypotheses, *Pattern Recogn. Lett.*, **150** (2021), 101–107. https://doi.org/10.1016/j.patrec.2021.07.007
5. E. Riddle-Workman, M. Evangelou, N. M. Adams, Multi-type relational clustering for enterprise cyber-security networks, *Pattern Recogn. Lett.*, **149** (2021), 172–178. https://doi.org/10.1016/j.patrec.2021.05.021
6. D. Paul, S. Saha, A. Kumar, J. Mathew, Evolutionary multi-objective optimization based overlapping subspace clustering, *Pattern Recogn. Lett.*, **145** (2021), 208–215. https://doi.org/10.1016/j.patrec.2021.02.012
7. H. Zhang, T. Zhan, I. Davidson, A self-supervised deep learning framework for unsupervised few-shot learning and clustering, *Pattern Recogn. Lett.*, **148** (2021), 75–81. https://doi.org/10.1016/j.patrec.2021.05.004
8. A. Sahu, A. S. Chowdhury, First person video summarization using different graph representations, *Pattern Recogn. Lett.*, **146** (2021), 185–192. https://doi.org/10.1016/j.patrec.2021.03.013
9. Z. Wu, C. Su, M. Yin, Z. Ren, S. Xie, Subspace clustering via stacked independent subspace analysis networks with sparse prior information, *Pattern Recogn. Lett.*, **146** (2021), 165–171. https://doi.org/10.1016/j.patrec.2021.03.026
10. S. Bianco, L. Celona, P. Napoletano, Disentangling image distortions in deep feature space, *Pattern Recogn. Lett.*, **148** (2021), 128–135. https://doi.org/10.1016/j.patrec.2021.05.008

*Appendix C. IEEE Journals and Conferences*


1. M. A. Masud, J. Z. Huang, M. Zhong, X. Fu, Generate pairwise constraints from unlabeled data for semi-supervised clustering, *Data Knowl. Eng.*, **123** (2019), 101715. https://doi.org/10.1016/j.datak.2019.101715
2. A. S. Shirkhorshidi, T. Y. Wah, S. M. R. Shirkhorshidi, S. Aghabozorgi, Evolving fuzzy clustering approach: an epoch clustering that enables heuristic postpruning, *IEEE Trans. Fuzzy Syst.*, **29** (2021), 560–568. https://doi.org/10.1109/TFUZZ.2019.2956900
3. X. Zhao, Z. Wang, L. Gao, Y. Li, S. Wang, Incremental face clustering with optimal summary learning via graph convolutional network, *Tsinghua Sci. Technol.*, **26** (2021), 536–547. https://doi.org/10.26599/TST.2020.9010024
4. Y. Cheng, W. Jia, R. Chi, A. Li, A clustering analysis method with high reliability based on Wilcoxon-Mann-Whitney testing, *IEEE Access*, **9** (2021), 19776–19787. https://doi.org/10.1109/ACCESS.2021.3053244
5. X. Hong, H. Li, P. Miller, J. Zhou, L. Li, D. Crookes, et al., Component-based feature saliency for clustering, *IEEE Trans. Knowl. Data Eng.*, **33** (2021), 882–896. https://doi.org/10.1109/TKDE.2019.2936847
6. X. Liu, M. Li, C. Tang, J. Xia, J. Xiong, L. Liu, et al., Efficient and effective regularized incomplete multi-view clustering, *IEEE Trans. Pattern Anal.*, **43** (2021), 2634–2646. https://doi.org/10.1109/TPAMI.2020.2974828
7. Z. Zhang, C. Wang, X. Peng, H. Qin, H. Lv, J. Fu, et al., Solar radiation intensity probabilistic forecasting based on k-means time series clustering and Gaussian process regression, *IEEE Access*, **9** (2021), 89079–89092. https://doi.org/10.1109/ACCESS.2021.3077475
8. L. Huang, L. Gan, B. Ling, A unified optimization model of feature extraction and clustering for spike sorting, *IEEE Trans. Neur. Syst. Reh.*, **29** (2021), 750–759. https://doi.org/10.1109/TNSRE.2021.3074162
9. M. Li, W. Liang, X. Liu, Multi-view clustering with learned bipartite graph, *IEEE Access*, **9** (2021), 87952–87961. https://doi.org/10.1109/ACCESS.2021.3060135
10. H. H. Zhao, X. C. Luo, R. Ma, X. Lu, An extended regularized K-means clustering approach for high-dimensional customer segmentation with correlated variables, *IEEE Access*, **9** (2021), 48405–48412. https://doi.org/10.1109/ACCESS.2021.3067499
11. W. Xiong, Y. Qiao, L. Bai, M. Ghahramani, N. Wu, P. Hsieh, et al., Wafer reflectance prediction for complex etching process based on K-means clustering and neural network, *IEEE Trans. Semiconduct. M.*, **34** (2021), 207–216. https://doi.org/10.1109/TSM.2021.3068974
12. K. Li, Y. Qin, Q. Ling, Y. Wang, Z. Lin, W. An, Self-supervised deep subspace clustering for hyperspectral images with adaptive self-expressive coefficient matrix initialization, *IEEE J.-STARS*, **14** (2021), 3215–3227. https://doi.org/10.1109/JSTARS.2021.3063335
13. X. Yang, J. Shi, C. Wang, Y. Zhou, Z. Zhou, T. Chen, et al., Binary clustering for deep network trained by feature growth, *IEEE Access*, **9** (2021), 8354–8366. https://doi.org/10.1109/ACCESS.2020.3047467
14. S. Lee, D. E. Lim, Y. Kang, H. J. Kim, Clustered multi-task sequence-to-sequence learning for autonomous vehicle repositioning, *IEEE Access*, **9** (2021), 14504–14515. https://doi.org/10.1109/ACCESS.2021.3051763

15. Y. Chen, L. Zhou, S. Pei, Z. Yu, Y. Chen, X. Liu, et al., KNN-BLOCK DBSCAN: fast clustering for large-scale data, *IEEE Trans. Syst. Man Cybern.*, **51** (2021), 3939–3953. https://doi.org/10.1109/TSMC.2019.2956527
16. I. Czarnowski, J. Jędrzejowicz, P. Jędrzejowicz, Designing RBFNs structure using similarity-based and kernel-based fuzzy C-means clustering algorithms, *IEEE Access*, **9** (2020), 4411–4422. https://doi.org/10.1109/ACCESS.2020.3048104
17. Y. Guo, S. Tierney, J. Gao, Robust functional manifold clustering, *IEEE Trans. Neur. Net. Lear.*, **32** (2021), 777–787. https://doi.org/10.1109/TNNLS.2020.2979444
18. S. Dong, Y. Quan, W. Feng, G. Dauphin, L. Gao, M. Xing, A pixel cluster CNN and spectral-spatial fusion algorithm for hyperspectral image classification with small-size training samples, *IEEE J.-STARS*, **14** (2021), 4101–4114. https://doi.org/10.1109/JSTARS.2021.3068864
19. J. Yang, Y. Zhang, G. Zhu, S. Kwong, A clustering-based framework for improving the performance of JPEG quantization step estimation, *IEEE Trans. Circ. Syst. Vid.*, **31** (2021), 1661–1672. https://doi.org/10.1109/TCSVT.2020.3003653
20. S. Ma, X. Li, Z. Zhang, Y. Xu, X. Pang, Positions-only clustering on passive location points of communication transmitters, *IEEE Trans. Instrum. Meas.*, **70** (2021), 3000911. https://doi.org/10.1109/TIM.2021.3055285
21. J. Song, H. Xing, H. Zhang, Y. Xu, Y. Meng, An adaptive network-constrained clustering (ANCC) model for fine-scale urban functional zones, *IEEE Access*, **9** (2021), 53013–53029. https://doi.org/10.1109/ACCESS.2021.3070345
22. S. Cao, L. Zhou, Z. Zhang, Prediction of dissolved oxygen content in aquaculture based on clustering and improved ELM, *IEEE Access*, **9** (2021), 40372–40387. https://doi.org/10.1109/ACCESS.2021.3064029
23. L. Wang, S. Niu, X. Gao, K. Liu, F. Lu, Q. Diao, et al., Fast high-order sparse subspace clustering with cumulative MRF for hyperspectral images, *IEEE Geosci. Remote S.*, **18** (2021), 152–156. https://doi.org/10.1109/LGRS.2020.2968350
24. J. Xu, M. Yu, L. Shao, W. Zuo, D. Meng, L. Zhang, et al., Scaled simplex representation for subspace clustering, *IEEE Trans. Cybern.*, **51** (2021), 1493–1505. https://doi.org/10.1109/TCYB.2019.2943691


AIMS Press