Volume 47, Issue 9 September 2014 ISSN 0031-3203

ELSEVIER

PATTERN RECOGNITION

Available online at www.sciencedirect.com

ScienceDirect

Pattern Recognition 47 (2014) 3034–3045

Contents lists available at ScienceDirect

Pattern Recognition

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

Centroid index: Cluster level similarity measure

Pasi Fränti [a,*], Mohammad Rezaei [a], Qinpei Zhao [b]

[a] *Speech & Image Processing Unit, Department of Computer Science, University of Eastern Finland, P.O. Box 111, FIN-80101 Joensuu, Finland*
[b] *School of Software Engineering, Tongji Unversity, Shanghai, China*

ARTICLE INFO

Article history:
Received 8 May 2013
Received in revised form
25 November 2013
Accepted 18 March 2014
Available online 29 March 2014

Keywords:
Clustering
k-Means
External validity
Similarity measure

ABSTRACT

In clustering algorithm, one of the main challenges is to solve the global allocation of the clusters instead of just local tuning of the partition borders. Despite this, all external cluster validity indexes calculate only point-level differences of two partitions without any direct information about how similar their cluster-level structures are. In this paper, we introduce a cluster level index called centroid index. The measure is intuitive, simple to implement, fast to compute and applicable in case of model mismatch as well. To a certain extent, we expect it to generalize other clustering models beyond the centroid-based *k*-means as well.

1. Introduction

Quality of centroid-based clustering is usually evaluated by internal validity indexes, most commonly by measuring intra-cluster variance. Internal validity indexes use information intrinsic to the data to assess the quality of a clustering. These include measures such as *Dunn's index* [1], *Davies–Bouldin index* [2] and *Silhouette coefficient* [3]. For a recent survey, see [4].

External indexes can be applied to compare the clustering against another solution or ground truth (if available). The ground truth can be a small representative training set given by an expert of the application domain. However, synthetic data is often used to test different aspects of the clustering methods, where their ground truth is easier to obtain. The indexes can also be applied in clustering ensemble [5,6] and used in genetic algorithms [7] to measure genetic diversity in a population. In [8], external indexes have been used for comparing the results of multiple runs to study the stability of the *k*-means algorithm. In [9], a framework for evaluating popular internal validity indexes was introduced by using external indexes on ground-truth labels. To sum up, in all these cases the main goal is to measure the similarity of two given clusterings.

Most external indexes are based on counting how many pairs of data points are co-located into the same (or different) cluster in both solutions. Examples of these are *Rand index* [10], *adjusted Rand index* [11], *Fowlkes and Mallows index* [12] and *Jaccard coefficient* [13]. A popular application-dependent approach is to measure classification error, which is quite often employed in supervised learning. Another type of external validity indexes is based on finding matches between the clusters in two solutions. Normalized *van Dongen criterion* [14,15] has a simple computation form and it can measure data with imbalanced class distributions. Other indexes utilize the entropy in different manners to compare two solutions. *Mutual information* [16] is derived from conditional entropy and *variation of information* [17] is a complement of the mutual information. Studies of external indexes can be found in [15,18].

For comparing clusterings, external indexes have been widely used by counting how many pairs of data points are partitioned consistently in the two clustering solutions. In order to be consistent, a pair of points must be allocated in both solutions either in the same cluster, or in a different cluster. This provides estimation of point-level similarity but does not give any direct information about the similarity at cluster level. For example in Fig. 1, both examples have large point-level mismatches (marked by yellow) but only the second example has cluster level mismatches.

In this paper, we propose a cluster level measure to estimate the similarity of two clustering solutions. First, nearest neighbor mapping is performed between the two sets of cluster prototypes (centroids), and the number of zero-mappings is then calculated. Each zero count means that there is no matching cluster in the other solution. The total number of zero-mappings gives direct information of how many different cluster locations are there in the two clustering solutions in total. In case of a perfect match, the index provides zero value indicating that the solutions have the same cluster-level structure. We denote the measure as *centroid index* (*CI*).

[*] Corresponding author.
E-mail address: franti@cs.joensuu.fi (P. Fränti).

http://dx.doi.org/10.1016/j.patcog.2014.03.017
0031-3203/© 2014 Elsevier Ltd. All rights reserved.

Most similar to our method are set-based measures [14,17]. They perform matching of the clusters and then measure the proportion of overlap across the matching clusters. Heuristic matching by a greedy algorithm is often done [14,31] because the optimal matching by Hungarian algorithm, for example, is not trivial to implement and takes O(N^3) time. Matching problem assumes that the number of clusters is equal. If this is not the case, some clusters must be left out and dealt with another manner. The set-based methods are also restricted to measure point-level differences.

Fig. 2 demonstrates the difference between a local point-level index (Adjusted Rand index) and the new centroid index (*CI*). The results of agglomerative clustering [19,20] and random swap algorithms [21,22] have only point level differences but have the same cluster level structure. The corresponding *CI*-value is 0. The result of the *k*-means, however, has one differently allocated centroid and the corresponding *CI*-values are 1. Adjusted Rand index reflects only to point level differences (values of 0.82, 0.88 and 0.91), which have less clear interpretation in practice. The proposed index is therefore more informative.

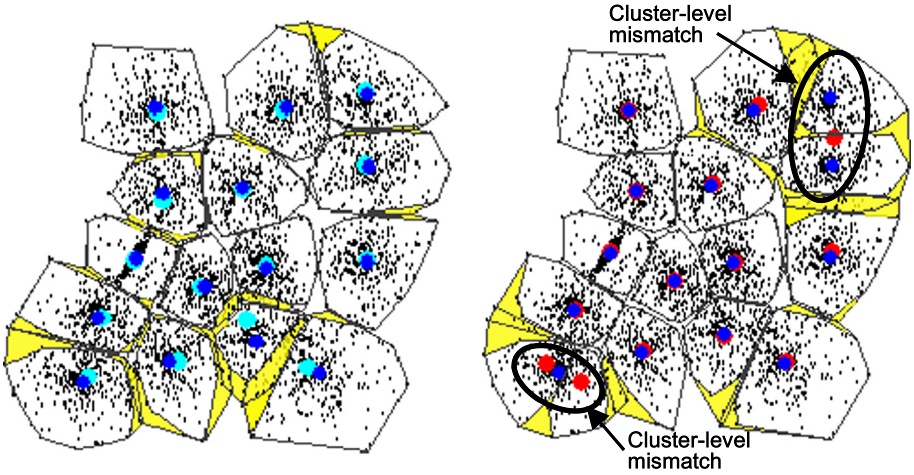


Fig. 1. Two different point-level clustering comparisons. Differences in the partitions are emphasized by yellow coloring. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

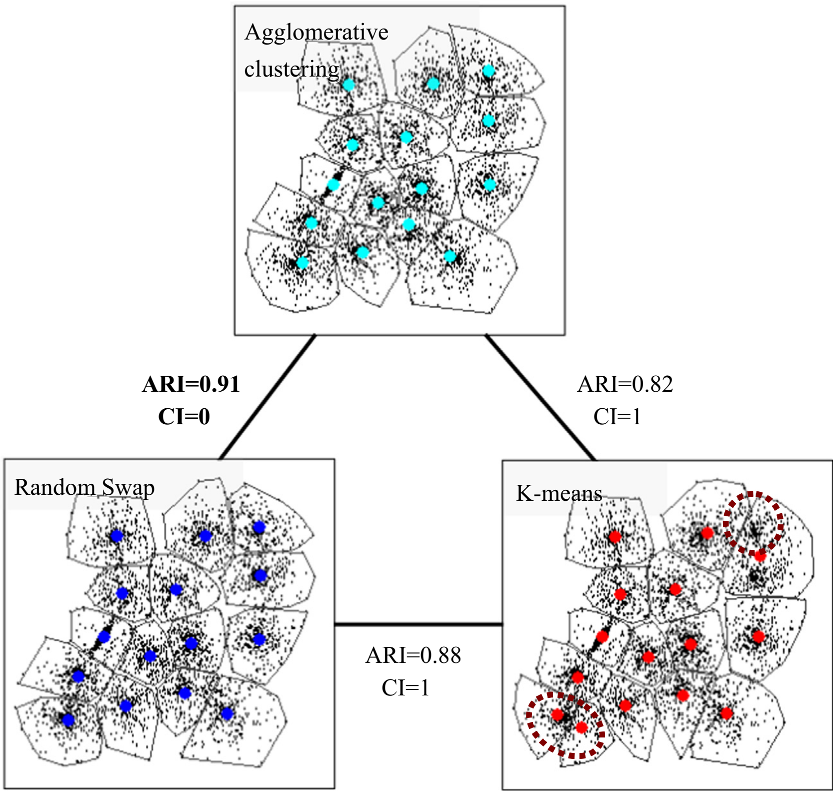


Fig. 2. Three clustering solutions and the corresponding values of Adjusted Rand index and the proposed centroid index (*CI*). The *k*-means solution has one incorrectly allocated cluster at the bottom left corner and one cluster missing at the top right corner. Otherwise the three solutions have only point level differences.

The main advantage of the centroid index is its clear intuitive interpretation. Each zero-count indicates exactly one missing cluster in the solution, either caused by different global allocation or by different number of clusters. The other benefits are that the centroid index is simple to implement and fast to compute. We expect that the main idea can be generalized to other clustering models beyond the centroid-based model (*k*-means).

The rest of the paper is organized as follows. We first define the centroid index in Section 2. We also give extension to measure point-level differences and discuss generalization to other type clustering problems. In Section 3, the index is compared against the existing indexes using artificial and real data. Furthermore, we apply the index for studying highly optimized clustering solutions and find out that it can recognize structural differences even between near-optimal clusterings that have seemingly similar partition. Another application of the index is to measure the stability of clustering algorithms. Conclusions are then drawn in Section 4.

2. Cluster level similarity

K-means clustering problem is defined as follows. Given a set of *N* data points *x* in *D*-dimensional space, partition the points into *K* clusters so that intra cluster variance (mean square error) is minimized. Centroids c_k represents the prototypes in *k*-means. The cost function is defined as

$$f = \frac{1}{N}\sum_{i=1}^{N}\sum_{x_i \in c_k} ||x_i - c_k||^2 \tag{1}$$

2.1. Duality of centroids and partition

Partition and the set of centroids are defined as

$$p_i \leftarrow \arg\min_{1 \le j \le M} ||x_i - c_j||^2 \quad \forall i \in [1, N] \tag{2}$$

$$c_j \leftarrow \sum_{p_i = j} x_i \Big/ \sum_{p_i = j} 1 \quad \forall j \in [1, K] \tag{3}$$

For a given partition $\{p_i\}$, the optimal prototype of a cluster is its centroid (arithmetic mean). And vice versa, for a given prototypes, optimal partition can be solved by assigning each point to the cluster whose prototype c_j is nearest. Thus, partition and centroids can be considered as *dual* structures (see also Appendix A): if one of them is given, the other one can always be uniquely determined using (2) and (3).

The duality is utilized in the *k-means algorithm* [23], which finds the nearest local minimum for a given initial solution by repeating these two properties in turn. The steps are called *partition step* and *centroid step*. However, *k*-means is limited to make local point-level changes only. More advanced algorithms, on the other hand, focus on solving the cluster location globally by operating with the prototypes, and solve the partition trivially by Eq. (2). Most common approach is to use *k*-means for the point-level fine-tuning, integrated either directly within the algorithm, or applying it as a separate post processing step.

Incremental algorithms add new clusters step by step by splitting an existing cluster [24,25], or by adding a new prototype [26], which attracts points from neighbor clusters. The opposite approach is to assign every data point into its own cluster, and then stepwise merge two clusters [27] or remove an existing one [28]. Fine-tuning can be done by *k*-means either after each operation, or after the entire process. Most successful iterative algorithms swap the prototypes randomly [21,22] or by deterministic manner [29], whereas genetic algorithms combine two entire clustering solutions by a crossover [30]. The success of all these algorithms is based on making cluster level changes. It is therefore reasonable that the similarity of solutions is measured at cluster level also.

2.2. Centroid index

Centroid Index (*CI*) measures cluster-level differences of two solutions. Since most essential cluster-level information is captured by the prototypes (cluster centroids), the calculations are based on them. Given two sets of prototypes $C=\{c_1, c_2, c_3,\ldots, c_{K1}\}$ and $C'=\{c'_1, c'_2, c'_3,\ldots, c'_{K2}\}$, we construct nearest neighbor mappings ($C \rightarrow C'$) as follows:

$$q_i \leftarrow \arg\min_{1 \le j \le K2} \| c_i - c'_j \|^2 \quad \forall i \in [1, K1] \tag{4}$$

For each target prototype c'_j, we analyze how many prototypes c_i consider it as the nearest ($q_i = j$). In specific, we are interested in the ones which no prototype is mapped to

$$orphan(c'_j) = \begin{cases} 1 & q_i \ne j \ \forall i \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

The dissimilarity of *C* in respect to *C′* is the number of orphan prototypes

$$CI_1(C, C') = \sum_{j=1}^{K2} orphan(c'_j) \tag{6}$$

We define that two clusterings (with same number of clusters $K1 = K2$) have the same cluster-level structure if every prototype is mapped exactly once ($CI_1 = 0$). Otherwise, every orphan indicates that there is a cluster in *C′* that is missing in *C*. For example, in Fig. 3 there are two sets of prototypes. Two prototypes are orphans, which is interpreted that there are two differently allocated prototypes with respect to the reference solution.

Note that the mapping is not symmetric ($C \rightarrow C' \ne C' \rightarrow C$). Symmetric version of the index is obtained by making the mapping in both ways

$$CI_2(C, C') = \max\{CI_1(C, C'), CI_1(C', C)\} \tag{7}$$

The index has clear intuitive interpretation: it measures how many clusters are differently located in the two solutions. In specific, if there are no orphans (each prototype has been mapped exactly once in both ways), the two clustering structures are equal. This kind of bijective 1:1 mapping happens only if the solutions have the same number of clusters, and the prototypes have the same global allocation. From algorithm point of view, the value of the index indicates how many prototype need to be swapped in order to transform one of the clustering solution to the other.

2.3. Generalizations

2.3.1. Different number of clusters

With the symmetric variant (CI_2), the number of clusters does not matter because the index is not limited by the pairing as other set-based measures. Instead, it gives a value that equals to the difference in the number of clusters ($K2 - K1$), or higher if other cluster-level mismatches are also detected. Intuitive interpretation of the value is the same as in Section 2.2. If the one-way variant (CI_1) is used, it should be calculated by mapping from the solution with fewer clusters to the solution with more clusters. Sample values are shown in Table 1, where three clusters found by *k*-means are compared to the ground truth (GT) with two clusters.

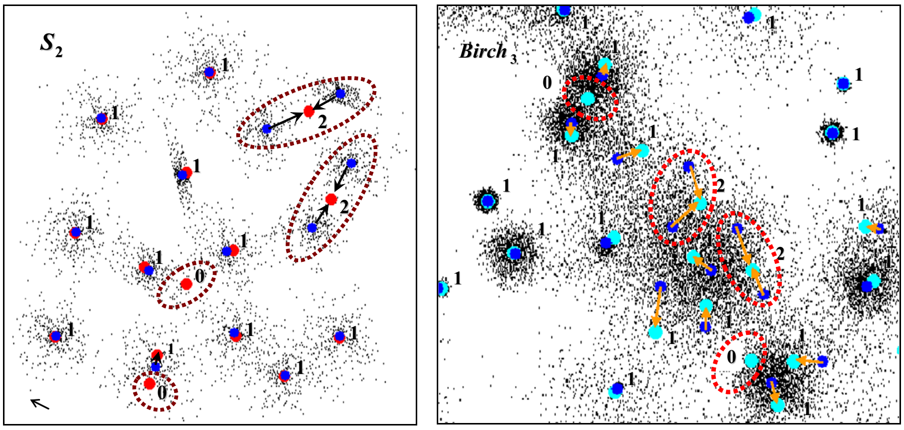


Fig. 3. Two sets of prototypes and their mappings are shown for S_2 (left) and for $Birch_3$ (right). In both examples, there are two orphans resulting to index value of $CI=2$.

Table 1
CI, CSI, Normalized van Dongen index (NVD) and Criterion-H (CH) values between the four different *k*-means clustering (3 clusters) and ground truth (GT). Perfect match are indicated by the following values: $CI=0$, $NVD=0$, $CH=0$, $CSI=1$.

NVD / CI	1	2	3	4	GT
1	-	0.23	0.22	0.22	0.11
2	1	-	0.13	0.13	0.12
3	1	0	-	0.22	0.11
4	1	0	1	-	0.11
GT	1	1	1	1	-

CH / CSI	1	2	3	4	GT
1	-	0.47	0.44	0.44	0.22
2	0.53	-	0.13	0.12	0.25
3	0.56	0.87	-	0.25	0.22
4	0.56	0.87	0.65	-	0.22
GT	0.78	0.75	0.78	0.78	-

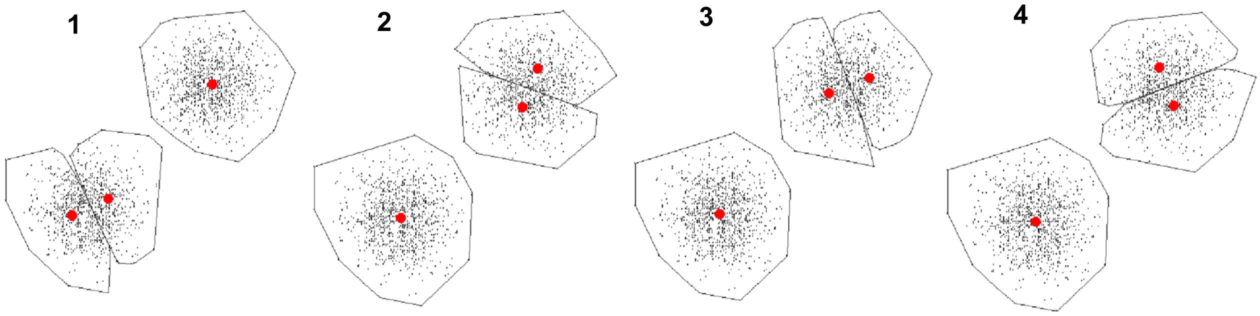


Fig. 4. Four different *k*-means solutions. Solution 1 has clearly different allocation than the others, whereas solutions 2–4 have mainly local differences.

2.3.2. *Point-level differences*

One limitation of the index is that it provides only very coarse (integer) values. This is suitable to measure cluster-level differences but not to measure more accurate point-level differences. Sample calculations are shown in Table 1 using the four sample data sets of Fig. 4. Here *CI* detects that Clustering 1 has different global allocation than 2–3–4. Among these three, the result is 0 (2–3, 2–4) or 1 (3–4) depending on the amount of variation of the topmost two clusters.

The centroid index, however, easily extends to measure point-level differences by combining it with a set-matching index [15,31] such as *criterion-H* [32] or *van Dongen index* [14]. In set-matching measures, the clusters are first paired by maximum weighted matching or by a greedy algorithm. The paired clusters are analyzed how many points they share relative to the cluster size. Our approach is simpler than that. We search for the nearest match without the pairing constraint, and allow 1:*N* type of matches. This is useful especially when the solutions have different number of clusters. Point-level centroid similarity index (CSI) can then be calculated as

$$\mathrm{CSI}=\frac{S_{12}+S_{21}}{2}\quad \text{where } S_{12}=\frac{\sum_{i=1}^{K_1} C_i \cap C_j}{N},\ S_{21}=\frac{\sum_{j=1}^{K_2} C_j \cap C_i}{N}$$

The results of CSI as well as the two set-based measures are shown in Table 1. We conclude that the point-level indexes provide more accurate measurements than *CI* but lack the intuitive interpretation of how many clusters are differently allocated. For more thorough study of the point-level measurement and their normalizations we refer to a follow-up paper [33], which is currently under process.

For better understanding the capability and limitations of the measure, on-line visualization on 2-D data sets is available for interactive testing here: http://cs.uef.fi/sipu/clustering/animator/.

2.3.3. *Other clustering models*

So far we have focused on *k*-means clustering assuming that the data is in (Euclidean) vector space. This restriction, however, is not really necessary. The only requirement for the index is that we can calculate similarity between any two clusters, and in this way, find the nearest neighbor clusters in the other solution. In *k*-means, the clusters are assumed to be spherical (e.g. Gaussian) and have uniform variance, in which case the nearest neighbor is trivially found by calculating the centroid distances.

In Gaussian mixture model (GMM), each cluster (called component) is represented by the centroid and covariance matrix (often just its diagonal) in order to model elliptical clusters. In this case, it is possible to solve the nearest neighbor by finding the most similar Gaussian component as in [34]. After this, the number of orphan models can be calculated in the same way to measure the similarity of two GMMs. Potential utilization of this could be done in a swap-based EM algorithm [35].

Extension to density-based clustering is less straightforward but possible. In [36], clustering is represented by their density profiles along each attribute. Our index can be generalized using this or any other definition of the similarity between two clusters, and then performing the nearest neighbor mapping.

3. Experiments

We compare the centroid index against popular point-level external validity indexes such as adjusted Rand index (ARI) [5], normalized van Dongen (NVD) [14] and normalized mutual information (NMI) [42].

Denote the two clustering partitions by $P=\{p_1, p_2,\ldots, p_{K1}\}$ and $S=\{s_1, s_2,\ldots, s_{K2}\}$ whose similarity we want to measure. For every pair of data points (x_i, x_j), the following counts are calculated:

a=the number of point pairs in the same cluster in P and in S.
b=the number of point pairs in the same cluster in P but in different in S.
c=the number of point pairs in the same cluster in S but in different in P.
d=the number of point pairs in different clusters in P and in S.

A contingency table of P and S is a matrix with n_{ij}, which is the number of objects that are both in clusters P_i and S_j, i.e., $n_{ij}=|P_i \cap G_j|$. The pair counting index ARI is based on counting the pairs of points on which the two clusterings agree or disagree. The indexes are defined based on the contingency table as follows:

$$\mathrm{ARI}=\frac{a-(a+c)(a+b)/d}{(a+c)+(a+b)/2-(a+c)(a+b)/d} \tag{8}$$

$$\mathrm{NVD}=\frac{\left(2N-\sum_{i=1}^{K}\max_{j=1}^{K\prime} n_{ij}-\sum_{j=1}^{K\prime}\max_{i=1}^{K} n_{ij}\right)}{2N} \tag{9}$$

$$\mathrm{NMI}=\frac{MI(P,G)}{(H(P)+H(G))/2} \tag{10}$$

where $H(P)$ is the entropy of clustering P. The value indicating complete match is 0 for *NVD*, and 1 for ARI and NMI.

3.1. *Data sets*

We consider the data sets summarized in Table 2 consisting of four generated data sets (Fig. 5), three image data sets (Fig. 6), and *Birch* data sets [37] (Fig. 7). The points in the first set (*Bridge*) are 4 × 4 non-overlapping vectors taken from a gray-scale image, and in the second set (*Miss America*) 4 × 4 difference blocks of two subsequent frames in video sequence. The third data set (*House*) consists of color values of the *RGB* image. *Europe* consists of differential coordinates from a large vector map. The number of clusters in these is fixed to $M=256$. The data sets S_1–S_4 are two-dimensional artificially generated data sets with varying complexity in terms of spatial data distributions with $M=15$ predefined clusters.

3.2. *Clustering algorithms*

For generating clustering, we consider the following algorithms: *k-means* (KM), *repeated k-means* (RKM), *k-means++* [38], *X-means* [25], *agglomerative clustering* (AC) [39], *global k-means* [26], *random swap* [21], and *genetic algorithm* [30]. For more comprehensive quality comparison of different clustering algorithms, we refer to [28].

K-means++ selects the prototypes randomly one by one so that, at each selection, the data points are weighted according to their distance to the nearest existing prototype. This simple initialization strategy distributes the prototypes more evenly among the data points. Both *k*-means++ and RKM are repeated 100 times.

X-means is a heuristic hierarchical method that tentatively splits every cluster and applies local *k*-means. Splits that provide improvement according to Bayesian information criterion are accepted. Kd-tree structure is used to speed-up *k*-means.

Agglomerative clustering (AC) and Global *k*-means (GKM) are both locally optimal hierarchical methods. AC generates the clustering using a sequence of merge operations (bottom-up approach) so that at each step, the pair of clusters is merged that increases objective function value least.

Global *k*-means (GKM) uses the opposite top-down approach. At each step, it considers every data point as a potential location for a new cluster, applies *k*-means iterations (here 10 iterations) and then selects the candidate solution that decreases the objective function value most. The complexity of the method is very high and it is not able to process the largest data sets in reasonable time.

Random swap (RS) finds the solution by a sequence of *prototype swaps* and by fine-tuning their exact location by *k*-means. The prototype and its new location are selected randomly, and the new trial solution is accepted only if it improves the previous one. This iterative approach is simple to implement and it finds the correct solution if iterated long enough.

Genetic algorithm (GA) maintains a set of solutions. It generates new candidate solutions by AC-based crossover, and fine-tuned by two iterations of *k*-means. We use population of 50 candidate solutions, and generate 50 generations. In total, there are 2500 high quality candidate solutions, and the best clustering result is produced, which is also visually verified to be the global optimum (S_1–S_4, *Birch*$_1$, *Birch*$_2$).

3.3. *Experiments with artificial data*

We made visual comparison of the results of all algorithms against the known ground truth with all 2-D data sets. Figs. 8 and 9 show selected cross-comparison samples for S_1–S_4, *Birch*$_1$ and *Birch*$_2$. For S_1–S_4, all algorithms provide correct cluster allocation except *k*-means, *X*-means for S_2, and AC for S_4. For *Birch*$_1$ and *Birch*$_2$, AC, RS and GA all provide correct results, with $CI=0$. In all cases, it was visually confirmed that *CI* equals to the number of

Table 2
Summary of the data sets.

Data set	Type of data set	Number of data points (N)	Number of clusters (M)	Dimension of data (D)
Bridge	Gray-scale image blocks	4096	256	16
House[a]	*RGB* image	34,112	256	3
Miss America	Residual image blocks	6480	256	16
Europe	Differential coordinates	169,673	256	2
$Birch_1$–$Brich_3$	Synthetically generated	100,000	100	2
S_1–S_4	Synthetically generated	5000	15	2

[a] Duplicate data points are combined and their frequency information is stored instead.

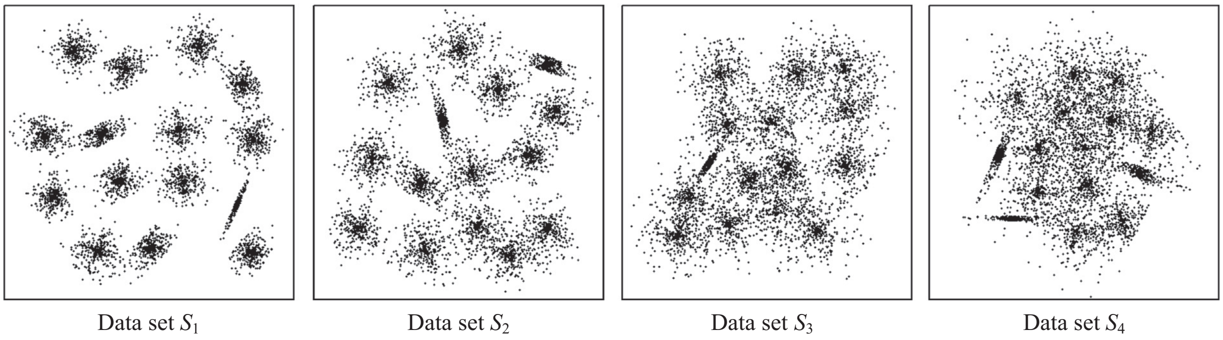


Fig. 5. Generated data sets with varying degrees of spatial overlap.

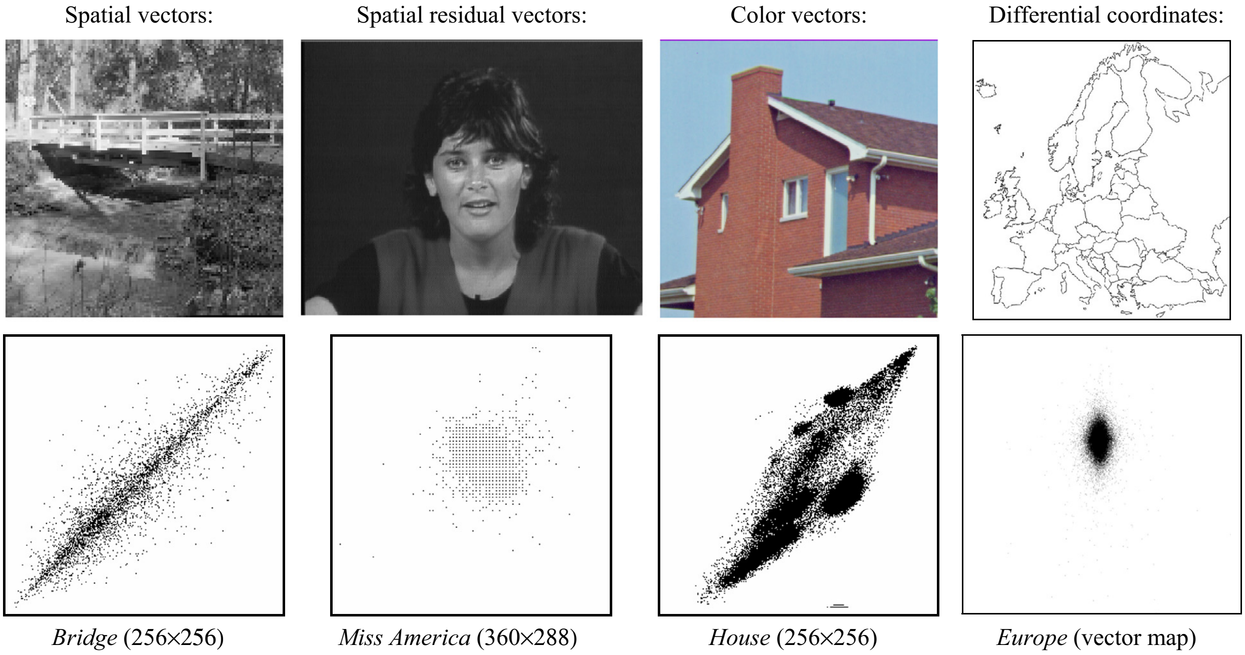


Fig. 6. Image data sets and their two-dimensional plots.

incorrectly located prototypes. Fig. 9 demonstrates the kind of clustering mistakes that typically appear.

For $Birch_3$, ground truth is not known. A visual comparison between RS and GA results is therefore provided in Fig. 10 as these algorithms provide the most similar results. Two clusters are differently located, and the other clusters have only minor point-level differences. At the lower part there are few point-level differences that demonstrate how large differences are tolerated by the *CI*-measure to be recognized as having the same cluster level structure.

3.4. *Comparison of clustering algorithms*

We next study the numerical results of the centroid index and the four point-level indexes. First, we report MSE values in Table 3 to give rough understanding about the clustering quality of the generated solutions. *K*-means provide clearly weaker results in all cases but it is difficult to make further conclusions about how good or bad the results are exactly. With *Bridge* we get 179.76 (KM), 173.64 (KM++), 168.92 (AC), 167.61 (RS) and 161.47 (GA) whereas the best reported value is 160.73 in [22]. With $Birch_1$, we get 5.47

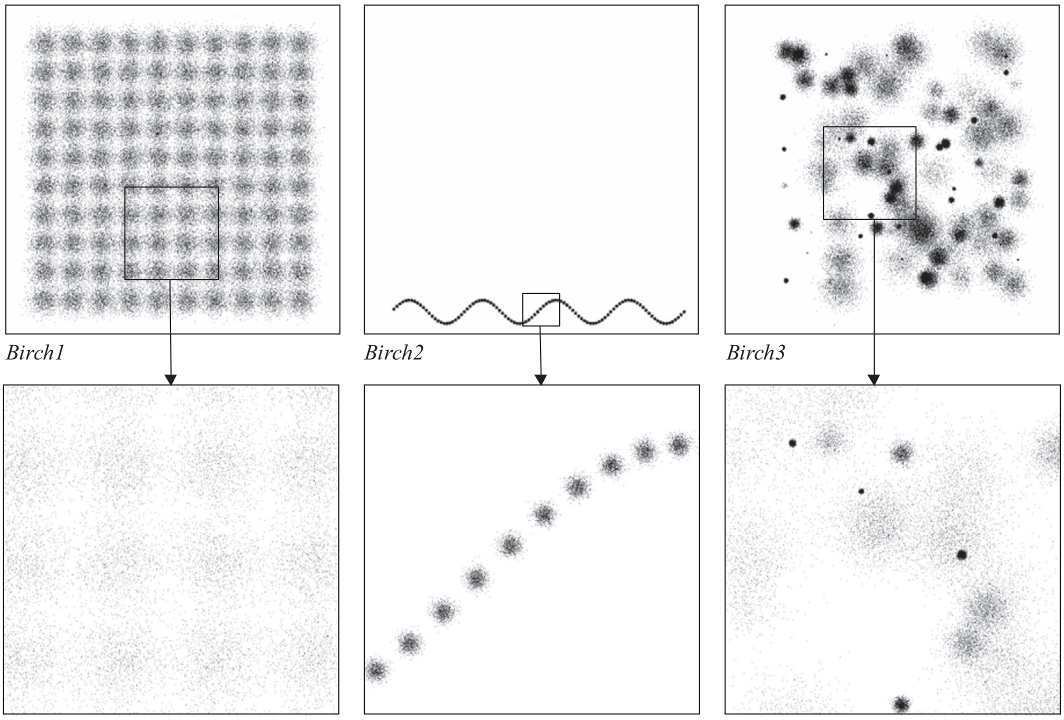
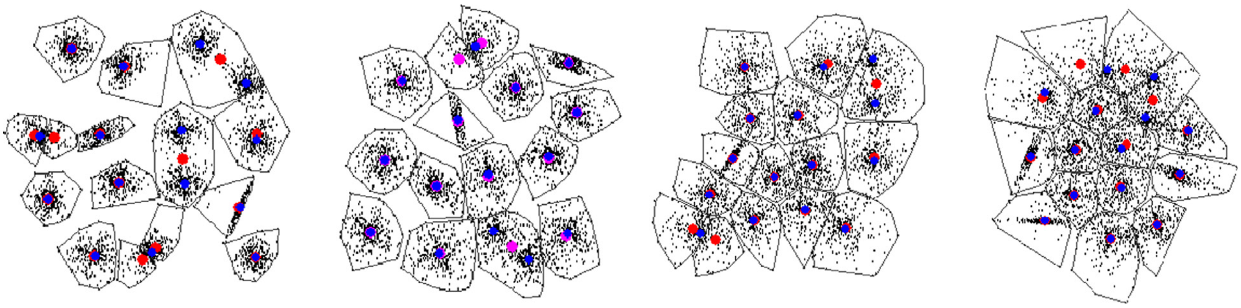


Fig. 7. *Birch* data sets.



$\boldsymbol{S}_1$: ARI=0.83, NVD=0.09, NMI=0.93 CI$_2$=2

$\boldsymbol{S}_2$: ARI=0.89, NVD=0.08, NMI=0.90, CI$_2$=1

$\boldsymbol{S}_3$: ARI=0.86, NVD=0.06, NMI=0.94, CI$_2$=1

$\boldsymbol{S}_4$: ARI=0.82, NVD=0.10, NMI=0.90, CI$_2$=1

Fig. 8. Values of three indexes when comparing random swap (blue) against *k*-means (red) for S_1, S_3, S_4, and versus *X*-means (purple) for S_2. The partition borders are drawn for the *k*-means and X-means algorithms. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

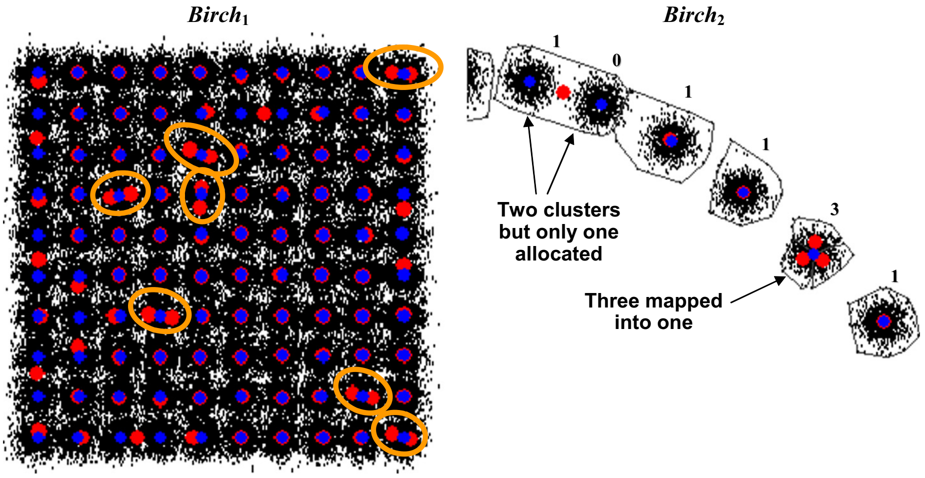


Fig. 9. *K*-means clustering (red points) versus reference solution (blue) – which is random swap clustering (left), and genetic algorithm (right). The values are $CI_2=7$ for *Birch*$_1$ and $CI_2=18$ for *Birch*$_2$ (only small fragment of the data shown here). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

(KM), 4.88 (KM++), 4.73 (AC), 4.64 (RS) without any clear evidence whether the AC and KM++ results can be considered essentially similar to that of RS.

Table 4 provides the corresponding values for all the point-level indexes. Known ground truth is used as the reference solution when available (S_1–S_4, *Birch*$_1$, *Birch*$_2$) and for the remaining data sets the result of GA is used as reference.

Adjusted Rand index provides higher values for all the correct clustering results with S_1–$S_{4,}$ than for any of the incorrect ones. However, the scale is highly data dependent, and there is no way to distinct between correct and incorrect clustering based on the value. The correct clustering results are measured by values 1.00 (S_1) 0.98–0.99 (S_2), 0.92–0.96 (S_3) but 0.93–0.94 (S_4). *Europe* data set is even more problematic as the measure makes almost no distinction among the clustering methods.

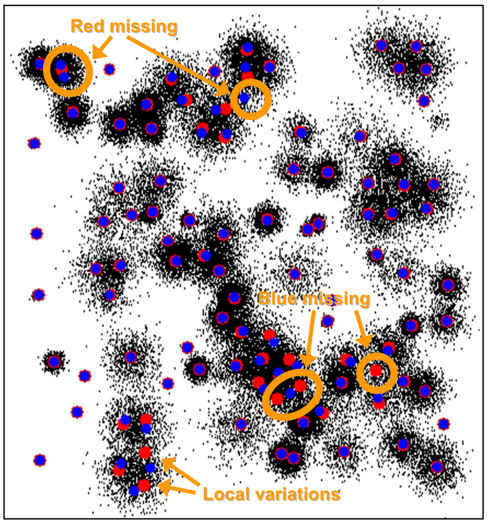


Fig. 10. Random swap (blue) versus genetic algorithm (red) with $CI_2=2$. There are two places (marked by yellow) where the results have different allocation of prototypes. In few places there are local variations of the prototypes that do not reflect to *CI*. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

The other two indexes perform similarly to ARI. The values of NVD are rather consistent whereas NMI provides higher variation and have the same problems with *Europe* and the S_1–S_4 sets. The point-level variant of the proposed index (CSI) provides 0.98–1.00 values when the clustering is correct. It somewhat suffers from the same problem as the other point-level indexes (Birch$_1$ for XM providing value 0.98 despite clustering is not correct) but overall it is much more consistent than ARI, NMI and NVD.

The *CI*-values are collected in Table 5. The results of S_1–S_4, *Birch*$_1$ and *Birch*$_2$ are consistent with the visual observations: the values indicate exactly how many clusters are incorrectly allocated. In specific, the index recognizes the failures of *X*-means (S_2) and AC (S_4).

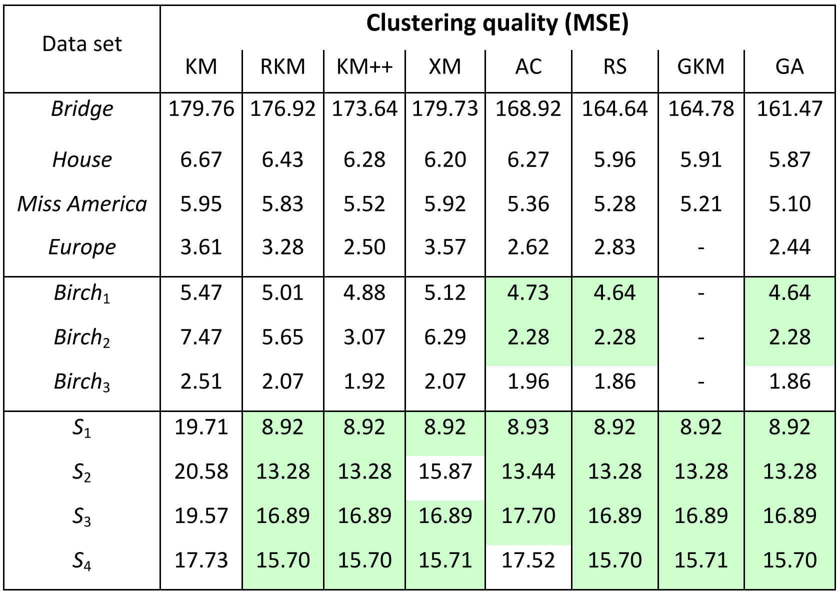
With higher dimensional image sets the results cannot be visually confirmed, and since the data is not expected to have clear clusters, the interpretation is less intuitive. Nevertheless, *CI* provides good estimation of the clustering quality and is useful for comparing the algorithms. For example, we can see that agglomerative clustering (AC), random swap (RS) and Global *k*-means (GKM) provide *CI*-values varying from 18 to 42, in comparison to the values 43–75 of *k*-means. This gives more intuitive understanding how much each solution differs to that of the reference solution.

Among the algorithms, only RS, GKM and GA are capable for finding the correct cluster allocation ($CI=0$) for the data sets for which ground truth is known. Agglomerative clustering has one incorrect allocation with S_4. The improved *k*-means variants (RKM, KM++ and XM) fail to find the optimal cluster allocation for *Birch* sets, whereas the plain *k*-means fails in all cases.

3.5. Comparison of highly optimized solutions

The results in Table 5 indicate that although the best algorithms provide quite similar results in terms of minimizing the cost function (MSE), the clusters have different global allocation. For example, the results of GA (161.47) and GKM (164.78) have 33 clusters (13%) allocated differently. We therefore study whether this is an inevitable phenomenon when clustering non-trivial multi-dimensional image data.

Table 3
Clustering quality measured by internal index (variance).



Data set	Clustering quality (MSE)							
	KM	RKM	KM++	XM	AC	RS	GKM	GA
Bridge	179.76	176.92	173.64	179.73	168.92	164.64	164.78	161.47
House	6.67	6.43	6.28	6.20	6.27	5.96	5.91	5.87
Miss America	5.95	5.83	5.52	5.92	5.36	5.28	5.21	5.10
Europe	3.61	3.28	2.50	3.57	2.62	2.83	-	2.44
Birch$_1$	5.47	5.01	4.88	5.12	4.73	4.64	-	4.64
Birch$_2$	7.47	5.65	3.07	6.29	2.28	2.28	-	2.28
Birch$_3$	2.51	2.07	1.92	2.07	1.96	1.86	-	1.86
S_1	19.71	8.92	8.92	8.92	8.93	8.92	8.92	8.92
S_2	20.58	13.28	13.28	15.87	13.44	13.28	13.28	13.28
S_3	19.57	16.89	16.89	16.89	17.70	16.89	16.89	16.89
S_4	17.73	15.70	15.70	15.71	17.52	15.70	15.71	15.70

Table 4
Clustering quality measured by the point-level indexes. The cases when the clustering was visually confirmed to be correct are emphasized by shading, and the six incorrect clusterings with S_1–S_4 are emphasized by **boldface**.

Data set	Adjusted Rand Index (ARI)							
	KM	RKM	KM++	XM	AC	RS	GKM	GA
Bridge	0.38	0.40	0.39	0.37	0.43	0.52	0.50	1
House	0.40	0.40	0.44	0.47	0.43	0.53	0.53	1
Miss America	0.19	0.19	0.18	0.20	0.20	0.20	0.23	1
Europe	0.46	0.49	0.52	0.46	0.49	0.49	-	1
Birch $_1$	0.85	0.93	0.98	0.91	0.96	1.00	-	1
Birch $_2$	0.81	0.86	0.95	0.86	1	1	-	1
Birch $_3$	0.74	0.82	0.87	0.82	0.86	0.91	-	1
S_1	**0.83**	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S_2	**0.80**	0.99	0.99	**0.89**	0.98	0.99	0.99	0.99
S_3	**0.86**	0.96	0.96	0.96	0.92	0.96	0.96	0.96
S_4	**0.82**	0.93	0.93	0.94	**0.77**	0.93	0.93	0.93

Data set	Normalized Mutual Information (NMI)							
	KM	RKM	KM++	XM	AC	RS	GKM	GA
Bridge	0.77	0.78	0.78	0.77	0.80	0.83	0.82	1.00
House	0.80	0.80	0.81	0.82	0.81	0.83	0.84	1.00
Miss America	0.64	0.64	0.63	0.64	0.64	0.66	0.66	1.00
Europe	0.81	0.81	0.82	0.81	0.81	0.82	-	1.00
Birch $_1$	0.95	0.97	0.99	0.96	0.98	1.00	-	1.00
Birch $_2$	0.96	0.97	0.99	0.97	1.00	1.00	-	1.00
Birch $_3$	0.90	0.94	0.94	0.93	0.93	0.96	-	1.00
S_1	**0.93**	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S_2	**0.90**	0.99	0.99	**0.95**	0.99	0.93	0.99	0.99
S_3	**0.92**	0.97	0.97	0.97	0.94	0.97	0.97	0.97
S_4	**0.88**	0.94	0.94	0.95	**0.85**	0.94	0.94	0.94

Data set	Normalized Van Dongen (NVD)							
	KM	RKM	KM++	XM	AC	RS	GKM	GA
Bridge	0.45	0.42	0.43	0.46	0.38	0.32	0.33	0.00
House	0.44	0.43	0.40	0.37	0.40	0.33	0.31	0.00
Miss America	0.60	0.60	0.61	0.59	0.57	0.55	0.53	0.00
Europe	0.40	0.37	0.34	0.39	0.39	0.34	-	0.00
Birch $_1$	0.09	0.04	0.01	0.06	0.02	0.00	-	0.00
Birch $_2$	0.12	0.08	0.03	0.09	0.00	0.00	-	0.00
Birch $_3$	0.19	0.12	0.10	0.13	0.13	0.06	-	0.00
S_1	**0.09**	0.00	0.00	0.00	0.00	0.00	0.00	0.00
S_2	**0.11**	0.00	0.00	**0.06**	0.01	0.04	0.00	0.00
S_3	**0.08**	0.02	0.02	0.02	0.05	0.00	0.00	0.02
S_4	**0.11**	0.04	0.04	0.03	**0.13**	0.04	0.04	0.04

Data set	Centroid Similarity Index (CSI)							
	KM	RKM	KM++	XM	AC	RS	GKM	GA
Bridge	0.47	0.51	0.49	0.45	0.57	0.62	0.63	1.00
House	0.49	0.50	0.54	0.57	0.55	0.63	0.66	1.00
Miss America	0.32	0.32	0.32	0.33	0.38	0.40	0.42	1.00
Europe	0.54	0.57	0.63	0.54	0.57	0.62	---	1.00
Birch $_1$	0.87	0.94	0.98	0.93	0.99	1.00	---	1.00
Birch $_2$	0.76	0.84	0.94	0.83	1.00	1.00	---	1.00
Birch $_3$	0.71	0.82	0.87	0.81	0.86	0.93	---	1.00
S_1	**0.83**	1.00	1.00	1.00	1.00	1.00	1.00	1.00
S_2	**0.82**	1.00	1.00	**0.91**	1.00	1.00	1.00	1.00
S_3	**0.89**	0.99	0.99	0.99	0.98	0.99	0.99	0.99
S_4	**0.87**	0.98	0.98	0.99	**0.85**	0.98	0.98	0.98

Next we consider only highly optimized (near-optimal) clustering results produced by three different optimization processes:

- GAIS: Genetic Algorithm with Iterative Shrinking (long variant) [28].
- RS: Random Swap [21].
- PRS: Perturbation Random Swap (experimental algorithm).

GAIS is a variant of the genetic algorithm (GA) that uses random initial solutions and iterative shrinking as the crossover method. The best known algorithms are all based on this variant one way or another. Random Swap is another powerful optimization technique that always finds the global minimum or very close to it – if iterated long. We consider here 1.000.000 (1 M) and 8.000.000 (8 M) iterations, and an experimental alternative (PRS) that perturbs the attributes of every centroid by 2-5% after every 10 iterations.

We use three different starting points for the optimization, see Table 6. First one is a random clustering optimized by RS (RS_{8M}). The other two are different runs produced by GAIS labeled by the year when ran (GAIS-2002 and GAIS-2012). These two are further optimized by various combinations of RS and PRS aiming at the lowest possible MSE-value.

In Table 7, we compare all these high quality solutions against each other. Although their MSE-values are very close to each other, the results indicate that they all have different global allocation. In specific, the RS-optimized results have 22–25 difference cluster allocations compared to the GAIS results. However, when we compare the results within the 'GAIS-2002 family', they have exactly the same global allocation ($CI=0$). This indicates that RS is capable for optimizing the MSE further (from 160.72 to 160.43) but only via local fine-tuning while keeping the global allocation unchanged.

The same observation applies to the results of the 'GAIS 2012 family': fine-tuning by MSE is observed (from 160.68 to 160.39) but only minor (one cluster) difference in the global allocations, at most. Despite similar behavior when optimizing MSE, the two GAIS families have systematic differences in the global allocation: 13–18 differently allocated clusters, in total.

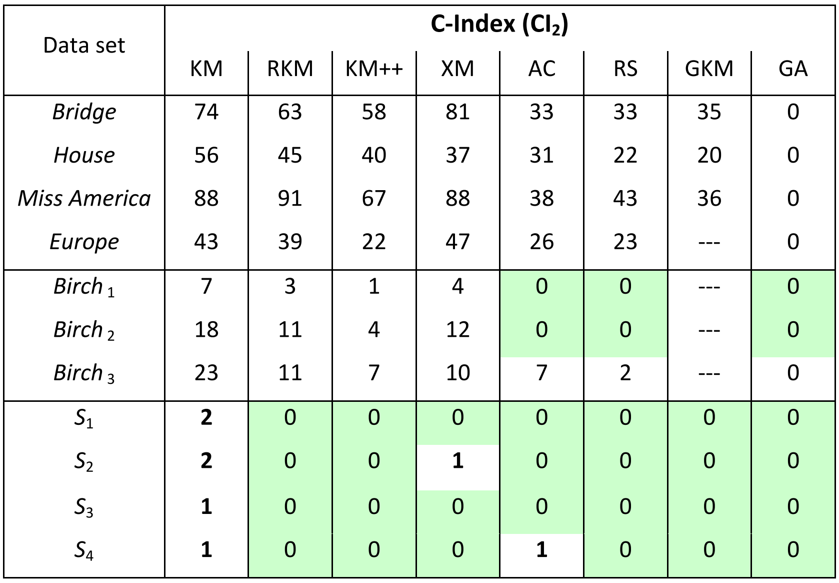
From the results we conclude that, in case of multi-dimensional image data, the index reveals existence of multiple clustering structures providing the same level of MSE-values but with different global cluster allocation. This indicates the existence of multiple global optima and that the proposed index can detect this. The point-level indexes can reveal the differences as well (into a certain extent) but without knowing the source of the differences originating from different global structure.

3.6. Stability of clustering

We next apply the index for measuring stability of clustering [40]. For this purpose, we generate from each data set 10 subsets by random sub-sampling, each of size 20% (overlap allowed). Each subset is then clustered by all algorithms. We measure the similarity of the results across the subsets within the same algorithm. In case of stable clustering, we expect the global structure to be the same expect minor changes due to the randomness in the sampling.

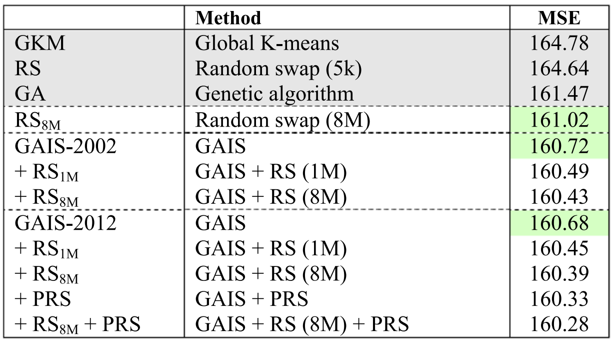
The results (Table 8) show that no variation is observed (0%) when applying a good algorithm (RS, GKM and GA) for the data sets S_1–S_4, *Birch*$_1$ and *Birch*$_2$. These all correspond to the case when the algorithm was successful with the full data as well (see Table 5). Results for NVD can also recognize stability for S_1 and *Birch*$_1$ only but not for S_2–S_4 and *Birch*$_2$. In general, instability can originate from several different reasons: applying inferior

Table 5
Clustering quality measured by the proposed centroid index (CI_2).



Data set	C-Index (CI$_2$)							
	KM	RKM	KM++	XM	AC	RS	GKM	GA
Bridge	74	63	58	81	33	33	35	0
House	56	45	40	37	31	22	20	0
Miss America	88	91	67	88	38	43	36	0
Europe	43	39	22	47	26	23	---	0
Birch $_1$	7	3	1	4	0	0	---	0
Birch $_2$	18	11	4	12	0	0	---	0
Birch $_3$	23	11	7	10	7	2	---	0
S_1	**2**	0	0	0	0	0	0	0
S_2	**2**	0	0	**1**	0	0	0	0
S_3	**1**	0	0	0	0	0	0	0
S_4	**1**	0	0	0	**1**	0	0	0

Table 6
Highly optimized clustering results for *Bridge*. First three rows are reference results from previous experiments. The numbers in the parentheses refer to the number of random swap iterations applied.



	Method	MSE
GKM	Global K-means	164.78
RS	Random swap (5k)	164.64
GA	Genetic algorithm	161.47
RS$_{8M}$	Random swap (8M)	161.02
GAIS-2002	GAIS	160.72
+ RS$_{1M}$	GAIS + RS (1M)	160.49
+ RS$_{8M}$	GAIS + RS (8M)	160.43
GAIS-2012	GAIS	160.68
+ RS$_{1M}$	GAIS + RS (1M)	160.45
+ RS$_{8M}$	GAIS + RS (8M)	160.39
+ PRS	GAIS + PRS	160.33
+ RS$_{8M}$ + PRS	GAIS + RS (8M) + PRS	160.28

Table 7
CI$_1$-values between the highly optimized algorithms for *Bridge*.

Centroid index (CI$_1$)									
Main algorithm:	RS$_{8M}$	GAIS 2002			GAIS 2012				
+Tuning 1 +Tuning 2	× ×	× ×	RS$_{1M}$ ×	RS$_{8M}$ ×	× ×	RS$_{1M}$ ×	RS$_{8M}$ ×	PRS ×	RS$_{8M}$ PRS
RS$_{8M}$	–	19	19	19	23	24	24	23	22
GAIS (2002)	23	–	0	0	14	15	15	14	16
+RS$_{1M}$	23	0	–	0	14	15	15	14	13
+RS$_{8M}$	23	0	0	–	14	15	15	14	13
GAIS (2012)	25	17	18	18	–	1	1	1	1
+RS$_{1M}$	25	17	18	18	1	–	0	0	1
+RS$_{8M}$	25	17	18	18	1	0	–	0	1
+PRS	25	17	18	18	1	0	0	–	1
+RS$_{8M}$+PRS	24	17	18	18	1	1	1	1	–

algorithm (k-means variants), using too small sub-sample size relative to the number of clusters, or using wrong number of clusters ($K=14$ or $K=16$ for S_1–S_4), or using inferior validity measure.

An open question is whether the stability could be used for detecting the number of clusters. Further tests would be needed as clustering tend to be stable also when only few ($K=3$) clusters are used. Thus, an external validity index such as *CI* alone is not sufficient for this task. This is left as future studies.

4. Conclusions

We have introduced a cluster level similarity measure called centroid index (*CI*), which has clear intuitive interpretation by corresponding to the number of differently allocated clusters. Value $CI=0$ indicates that the two clustering have the same global structure, and only local point-level differences may appear. Values $CI>0$ are indications of how many clusters are differently allocated. In swap-based clustering, this equals to the number of swaps needed, and an attempt has been made in [41] for recognizing the potential swaps.

The centroid index is trivial to implement and can be computed fast in O(K^2) time based on the cluster centroids only. Point-level extension (CSI) was also introduced by calculating the (proportional) number of same points between the matched clusters. This provides more accurate result at the cost of losing the intuitive interpretation of the value.

The index was demonstrated to be able to recognize structural similarity of highly optimized clustering of 16-dimensional image data. General belief is that nearest neighbor search (and clustering itself) would become meaningless when dimension increases, yet the index found out similarity of the clustering structures that was not previously known. We also used the index to measure stability of clustering under random sub-sampling. The results are promising in such extent that we expect the index to be applicable for solving the number of clusters even though not in trivial manner as such. This is a point of further studies.

The centroid index is also expected to generalize to other clustering models such as Gaussian mixture models and density-based clustering. All what would be needed is to define similarity of two clusters in order to perform the nearest neighbor mapping.

Table 8
Stability of the clustering. The values are the proportion of differently allocated centroids (calculated as *Index*/*K*), across all 10 subsets, on average. Zero value implies stability in respect to random sub-sampling.

Data set	Relative CI-values (%)							
	KM	RKM	KM++	XM	AC	RS	GKM	GA
	K=100							
Birch$_1$	11	8	9	3	0	0	---	0
Birch$_2$	23	19	11	4	0	0	---	0
Birch$_3$	19	15	14	9	7	4	---	5
	K=16							
S_1	19	9	13	8	5	5	5	5
S_2	16	7	14	6	5	5	5	5
S_3	15	7	11	11	5	5	5	5
S_4	15	5	12	9	5	4	4	4
	K=15							
S_1	10	5	11	0	0	0	0	0
S_2	19	5	10	5	0	0	0	0
S_3	16	6	13	5	0	0	0	0
S_4	11	4	10	11	2	0	0	0
	K=14							
S_1	17	10	15	7	4	4	4	4
S_2	16	6	13	5	2	1	1	1
S_3	16	7	9	7	2	5	5	5
S_4	10	2	8	9	5	2	2	2
	K=3							
S_1	2	5	4	4	10	8	5	5
S_2	0	0	3	4	0	0	0	0
S_3	16	0	5	17	5	0	0	0
S_4	4	0	3	1	6	0	0	0

Data set	Relative NVD-values (%)							
	KM	RKM	KM++	XM	AC	RS	GKM	GA
	K=100							
Birch$_1$	15	11	12	5	3	1	---	1
Birch$_2$	16	14	9	3	0	0	---	0
Birch$_3$	17	16	15	10	11	9	---	9
	K=16							
S_1	9	6	4	3	1	2	2	2
S_2	12	5	5	3	3	3	3	3
S_3	15	7	12	8	8	5	5	5
S_4	14	8	11	13	11	8	8	8
	K=15							
S_1	11	6	6	4	0	0	0	0
S_2	11	4	11	3	2	1	1	1
S_3	17	8	12	6	7	3	3	3
S_4	19	9	15	11	10	5	5	5
	K=14							
S_1	12	8	8	5	5	5	5	4
S_2	14	6	11	9	4	1	1	2
S_3	14	10	15	7	11	8	9	9
S_4	16	7	16	14	14	8	7	6
	K=3							
S_1	22	16	21	22	12	16	16	13
S_2	15	13	17	13	8	11	11	13
S_3	11	2	8	12	11	2	2	2
S_4	13	8	16	8	23	8	8	8

Conflict of interest statement

None declared.

Appendix A. Duality property

An important property of centroid-based clustering is that the distortion difference originates from the movement of the centroid to any other point depends on the size of the cluster and the distance between the centroid and the point.

Lemma 2.1. *Given a subset S of points in R^d with size n, let c be the centroid of S. Then for any $z \in R^d$, there is*

$$\sum_{x_i \in S} ||x_i - z||^2 - \sum_{x_i \in S} ||x_i - c||^2 = n||c - z||^2 \tag{A1}$$

Proof. By expanding the left side, we have

$$\begin{aligned}
&\sum_{x_i \in S} ||x_i - z||^2 - \sum_{x_i \in S} ||x_i - c||^2 \\
&\quad = \sum_{x_i \in S} (||x_i||^2 - 2x_i z + ||z||^2) - \sum_{x_i \in S} (||x_i||^2 - 2x_i c + ||c||^2) \\
&\quad = \sum_{x_i \in S} 2x_i c - 2x_i z + ||z||^2 - ||c||^2 \\
&\quad = 2\sum_{x_i \in S} x_i(c - z) + \sum_{x_i \in S} ||z||^2 - \sum_{x_i \in S} ||c||^2 \\
&\quad = 2nc(c - z) + nz^2 - nc^2 = n||c - z||^2
\end{aligned}$$

The fourth equality follows from the fact that $c = 1/n\Sigma_{x_i \in S} x_i$.

For a given partition, the optimal set of prototypes is the centroid (arithmetic mean) of the clusters. And vice versa, for a given set of prototypes, optimal partition can always be obtained by assigning each point to its nearest centroid. Thus, partition and centroids are dual structures.

Lemma 2.2. *For each iteration $t \ge 0$ in k-means, we have that*

$$f\left(\left\{p_i^{(t)}\right\}_{i=1}^{N}\right) \ge f\left(\left\{p_i^{(t+1)}\right\}_{i=1}^{N}\right) \tag{A2}$$

Proof. Define $S_j^{(t+1)} = \{x \in \{x_i\}_{p_i = j}, 1 \le j \le M\}$, x satisfies that $||x - c_j^{(t)}||^2 < ||x - c_h^{(t)}||^2$, where $1 \le h \le K, j \ne h$.

According to the definition in Eq. (1),

$$\begin{aligned}
f\left(\left\{p_i^{(t)}\right\}_{i=1}^{N}\right) &= \sum_{j=1}^{K} \left(\sum_{x \in S_j} ||x - c_j^{(t)}||^2 \right) = \sum_{j=1}^{K} \left(\sum_{h=1}^{K} \sum_{x \in S_j^{(t)} \cap S_h^{(t+1)}} ||x - c_j^{(t)}||^2 \right) \\
&\ge \sum_{j=1}^{K} \left(\sum_{h=1}^{K} \sum_{x \in S_j^{(t)} \cap S_h^{(t+1)}} ||x - c_h^{(t)}||^2 \right) = \sum_{h=1}^{K} \left(\sum_{j=1}^{K} \sum_{x \in S_h^{(t+1)} \cap S_j^{(t)}} ||x - c_h^{(t)}||^2 \right) \\
&= \sum_{h=1}^{K} \left(\sum_{x \in S_h^{(t+1)}} ||x - c_h^{(t)}||^2 \right) \ge \sum_{h=1}^{K} \left(\sum_{x \in S_h^{(t+1)}} ||x - c_h^{(t+1)}||^2 \right) \\
&= f(\{p_i^{(t+1)}\}_{i=1}^{N})
\end{aligned}$$

The second inequality follows the Lemma 2.1. Intuitively, Lemma 2.2 indicates the duality between the centroids and partitions.

References

[1] J.C. Dunn, Well separated clusters and optimal fuzzy partitions, J. Cybern. 4 (1974) 95–104.

[2] D.L. Davies, D.W. Bouldin, Cluster separation measure, IEEE Trans. Pattern Anal. Mach. Intell. 1 (2) (1979) 95–104.

[3] L. Kaufman, P.J. Rousseeuw, Finding Groups in Data. An Introduction to Cluster Analysis, Wiley, New York, 1990.

[4] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J.M. Pérez, I. Perona, An extensive comparative study of cluster validity indices, Pattern. Recognit. 46 (1) (2013) 243–256.

[5] H. Wong, S. Zhang, Y. Chen, Generalized adjusted rand indices for cluster ensemble, Pattern Recognit. 45 (6) (2012) 2214–2226.

[6] A. Lourenco, A.L. Fred, A.K. Jain, On the scalability of evidence accumulation clustering, in: Proceedings of the 20th International Conference on Pattern Recognition (ICPR'10), 2010, pp. 782–785.

[7] P. Fränti, J. Kivijärvi, T. Kaukoranta, O. Nevalainen, Genetic algorithms for large scale clustering problems, Comput. J. 40 (9) (1997) 547–554.

[8] L.I. Kuncheva, D.P. Vetrov, Evaluation of stability of *k*-means cluster ensembles with respect to random initialization, IEEE Trans. Pattern Anal. Mach. Intell. 28 (11) (2006) 1798–1808.

[9] I. Gurrutxaga, J. Muguerza, O. Arbelaitz, J.M. Pérez, J. Martín, Towards a standard methodology to evaluate internal cluster validity indices, Pattern Recognit. Lett. 32 (2011) 505–515.

[10] W.M. Rand, Objective criteria for the evaluation of clustering methods, J. Am. Stat. Assoc. 66 (1971) 846–850.

[11] L. Hubert, P. Arabie, Comparing partitions, J. Classif. 2 (1) (1985) 193–218.

[12] E.B. Fowlkes, C.L. Mallows, A method for comparing two hierarchical clusterings, J. Am. Stat. Assoc. 78 (383) (1983) 553–569.

[13] R.R. Sokal, P.H.A. Sneath, Principles of Numeric Taxonom, W.H. Freeman, San Francisco, 1963.

[14] S. van Dongen, Performance Criteria for Graph Clustering and Markov Cluster Experiments, Technical Report INSR0012, Centrum voor Wiskunde en Informatica, 2000.

[15] J. Wu, H. Xiong, J. Chen, Adapting the right measures for *k*-means clustering, ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD'09), 2009, pp. 877–886.

[16] A. Strehl, J. Ghosh, C. Cardie, Cluster ensembles – a knowledge reuse framework for combining multiple partitions, J. Mach. Learn. Res. 3 (2002) 583–617.

[17] M. Meila, Comparing clusterings – an information based distance, J. Multivar. Anal. 98 (2007) 873–895.

[18] J. Epps, N.X. Vinh, J. Bailey, Information theoretic measures for clusterings comparison: variants, properties, normalization and correction for chance, J. Mach. Learn. Res. 11 (2010) 2837–2854.

[19] H. Frigui, R. Krishnapuram, Clustering by competitive agglomeration, Pattern Recognit. 30 (7) (1997) 1109–1119.

[20] P. Fränti, O. Virmajoki, V. Hautamäki, Fast agglomerative clustering using a *k*-nearest neighbor graph, IEEE Trans. Pattern Anal. Mach. Intell. 28 (11) (2006) 1875–1881.

[21] P. Fränti, J. Kivijärvi, Randomised local search algorithm for the clustering problem, Pattern Anal. Appl. 3 (4) (2000) 358–369.

[22] T. Kanungo, D.M. Mount, N. Netanyahu, C. Piatko, R. Silverman, A.Y. Wu, A local search approximation algorithm for *k*-means clustering, Comput. Geom. 28 (1) (2004) 89–112.

[23] A.K. Jain, Data clustering: 50 years beyond K-means, Pattern Recognit. Lett. 31 (2010) 651–666.

[24] P. Fränti, T. Kaukoranta, O. Nevalainen, On the splitting method for vector quantization codebook generation, Opt. Eng. 36 (11) (1997) 3043–3051.

[25] D. Pelleg, A. Moore, X-means: extending *k*-means with efficient estimation of the number of clusters, in: Proceedings of the Seventeenth International Conference on Machine Learning (ICML'00), Stanford, CA, USA, 2000.

[26] A. Likas, N. Vlassis, J.J. Verbeek, The global *k*-means clustering algorithm, Pattern Recognit. 36 (2003) 451–461.

[27] T. Kaukoranta, P. Fränti, O. Nevalainen, Iterative split-and-merge algorithm for VQ codebook generation, Opt. Eng. 37 (10) (1998) 2726–2732.

[28] P. Fränti, O. Virmajoki, Iterative shrinking method for clustering problems, Pattern Recognit. 39 (5) (2006) 761–765.

[29] B. Fritzke, The LBG-U method for vector quantization – an improvement over LBG inspired from neural networks, Neural Process. Lett. 5 (1) (1997) 35–45.

[30] P. Fränti, Genetic algorithm with deterministic crossover for vector quantization, Pattern Recognit. Lett. 21 (1) (2000) 61–68.

[31] H. Xiong, J. Wu, J. Chen, *K*-means clustering versus validation measures: a data-distribution perspective, IEEE Trans. Syst. Man Cybern. Part B 39 (2) (2009) 318–331.

[32] M. Meila, D. Heckerman, An experimental comparison of model based clustering methods, Mach. Learn. 41 (1/2) (2001) 9–29.

[33] M. Rezaei, Q. Zhao, P. Fränti, 2014. Set matching based external cluster validity indexes (in preparation).

[34] J.E. Harmse, Reduction of Gaussian mixture models by maximum similarity, J. Nonparametric Stat. 22 (6) (2010) 703–709.

[35] Q. Zhao, V. Hautamäki, I. Kärkkäinen, P. Fränti, Random swap EM algorithm for Gaussian mixture models, Pattern Recognit. Lett. 33 (2012) 2120–2126.

[36] E. Bae, J. Bailey, G. Dong, A clustering comparison measure using density profiles and its application to the discovery of alternate clusterings, Data Min. Knowl. Discov. 21 (2010) 427–471.

[37] T. Zhang, R. Ramakrishnan, M. Livny, BIRCH: a new data clustering algorithm and its applications, Data Min. Knowl. Discov. 1 (2) (1997) 141–182.

[38] D. Arthur, S. Vassilvitskii, *K*-means++: the advantages of careful seeding, in: ACM-SIAM Symposium on Discrete Algorithms (SODA'07), New Orleans, LA, 2007, pp. 1027–1035.

[39] P. Fränti, T. Kaukoranta, D.-F. Shen, K.-S. Chang, Fast and memory efficient implementation of the exact PNN, IEEE Trans. Image Process. 9 (5) (2000) 773–777.

[40] S. Ben-david, D. Pál, H.U. Simon, Stability of *k*-means clustering, Conference on Computational Learning Theory (COLT), LNCS 4539, Budapest, Hungary, 2007, pp. 20–34.

[41] Q. Zhao, P. Fränti, Centroid ratio for pairwise random swap clustering algorithm, IEEE Trans. Knowl. Data Eng. (2014) (in press) http://doi.ieeecomputersociety.org/10.1109/TKDE.2013.113.

[42] T.O. Kvalseth, Entropy and correlation: some comments, IEEE Trans. Syst. Man Cybern. 17 (3) (1987) 517–519.

Pasi Fränti received his M.Sc. and Ph.D. degrees from the University of Turku, 1991 and 1994 in Science. Since 2000, he has been a Professor of Computer Science at the University of Eastern Finland. He has published 63 journals and 140 peer review conference papers, including 12 IEEE transaction papers. His current research interests include clustering algorithms and location-based recommendation systems. He has supervised 19 Ph.D.s and is currently the head of the East Finland doctoral program in Computer Science & Engineering (ECSE).

Mohammad Rezaei received his B.Sc. degree in Electronic engineering in 1996 and his M.Sc. degree in biomedical engineering in 2003 both from Amirkabir university of Technology, Tehran, Iran. Currently he is a Ph.D. student in university of Eastern Finland. His research interests include data clustering, multimedia processing, classification and retrieval.

Qinpei Zhao received the M.Sc. degrees in pattern recognition and image processing from the Shanghai Jiaotong University, China in 2007, and Ph.D. degree in computer science from the University of Eastern Finland, 2012. Her research interests include clustering, data mining, location-based applications and pattern recognition. Since 2013 she is with the Tongji University, Shanghai, China.