Centroid Ratio for a Pairwise Random Swap Clustering Algorithm

Qinpei Zhao and Pasi Fränti, *Senior Member, IEEE*

Abstract—Clustering algorithm and cluster validity are two highly correlated parts in cluster analysis. In this paper, a novel idea for cluster validity and a clustering algorithm based on the validity index are introduced. A *Centroid Ratio* is firstly introduced to compare two clustering results. This centroid ratio is then used in prototype-based clustering by introducing a *Pairwise Random Swap* clustering algorithm to avoid the local optimum problem of k-means. The swap strategy in the algorithm alternates between simple perturbation to the solution and convergence toward the nearest optimum by k-means. The centroid ratio is shown to be highly correlated to the mean square error (MSE) and other external indices. Moreover, it is fast and simple to calculate. An empirical study of several different datasets indicates that the proposed algorithm works more efficiently than *Random Swap*, *Deterministic Random Swap*, *Repeated k-means* or *k-means++*. The algorithm is successfully applied to document clustering and color image quantization as well.

Index Terms—Data clustering, random /deterministic swap, clustering evaluation, k-means

1 Introduction

Prototype-based clustering is a typical clustering method for finding a sequence of prototypes that best fit data with unknown structure. For example, a single prototype (centroid) is used to represent a cluster in *k-means* [1], which has been widely applied for data grouping in real applications not only because it has low computational and memory space requirements but it also achieves good results in most cases. However, it is known to be sensitive to its initialization.

A common way to address the initialization problem is to run k-means multiple times with a different set of randomly chosen initial parameters [2] and to choose the best solution as a result. We call this variant *repeated k-means* (RKM). For different data sets, the proper number of repetitions for RKM is an empirical choice. *Swap-based clustering algorithm* [3] is a local search heuristic to find optimal centroids based on the convergence property of k-means. In each iteration, a swap strategy is employed to look for a pair of centroids, of which one is to be removed, and the other inserted, to arrive at an improved solution. If better prototypes are found, the swap is made. This procedure is repeatedly performed after a fine-tuning step by k-means. This swap-based clustering is simple to implement and obtains good quality results independently of its initialization. This swap strategy could be either random or deterministic.

Several other methods have been developed, which are based on stochastic global optimization such as *simulated annealing* [4] and *genetic algorithms* [5]. These methods have not gained wide acceptance because of their great time complexity. A *global k-means* algorithm (GKM) [6] is an incremental approach that dynamically adds one cluster center at a time through a deterministic global search procedure. The search procedure consists of N (data size) executions of the k-means algorithm from suitable initial positions. The *k-means++* algorithm [7] chooses initial values (seeds) for k-means, and improves both the speed and accuracy of k-means. It is $\Theta(\log M)$-competitive with the optimal clustering [7], i.e., $E[\phi] \leq 8(\log M + 2)\phi_{OPT}$ where ϕ indicates the cost function and M represents the number of clusters.

People have identified some data characteristics that can greatly affect the k-means clustering analysis. These data characteristics include: high-dimensionality, the size of the data, the sparseness of the data, noise, outliers, types of attributes and data sets, and scales of attributes [8]. The conventional k-means uses the Euclidean distance to calculate the distance between data points, which puts restrictions on high-dimensional data. In a high dimensional space, the data becomes sparse, and traditional indexing and algorithmic techniques fail to be efficient and/or effective [9]. Therefore, many clustering algorithms based on conventional k-means do not work well for high-dimensional data. The k-means algorithm with cosine distance, which is known as *spherical k-means* [10], is a popular method for clustering high-dimensional data, for example in document clustering.

The clustering algorithm and the validity of the clustering are two essential parts of cluster analysis. In general, cluster validity can be categorized into two classes: external [11] and internal validity [12]. Partitions at the point level are often used for evaluating clusterings by

- *Q. Zhao is with the School of Software Engineering, Tongji University, Shanghai 200092, China. E-mail: qinpeizhao@gmail.com.*
- *P. Fränti is with the School of Computing, University of Eastern Finland, Joensuu 80110, Finland. E-mail: pasi.franti@uef.fi.*

Manuscript received 7 Mar. 2012; revised 13 Apr. 2013; accepted 20 June 2013. Date of publication 1 July 2013; date of current version 7 May 2014. Recommended for acceptance by M. Ester.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier 10.1109/TKDE.2013.113

external indices [11] such as the *Rand index* and the *Jaccard coefficient*. Since these evaluation measures are at the point level, they provide high accuracy but their time complexity is related to both $O(M)$ and $O(N)$, typically $O(MN)$, where N is the data size. Centroids are representatives for clusters in prototype-based clusterings. However, there has been very little work done on clustering evaluation based only on centroids. In [33], a cluster-level measure to estimate the similarity of two clustering solutions is firstly proposed. Centroids represent a global structure of prototypes, and utilizing only centroids in the evaluation reduces the time complexity to $O(M^2)$.

In this paper, we propose a cluster validity index called the *centroid ratio*, which can be used to compare two clusterings and find unstable and incorrectly located centroids in them. As the centroid ratio can find incorrectly located centroids in two clusterings, we use this property and propose a novel clustering algorithm called the *Pairwise Random Swap* (PRS) clustering algorithm. The incorrectly located centroids detected by the centroid ratio are selected as the clusters to be swapped in PRS. Meanwhile, the similarity value for comparing two clusterings from the centroid ratio can be used as a stopping criterion in the algorithm.

In Section 4, we demonstrate that the proposed centroid ratio has a high correlation with other evaluation measures. The proposed algorithm is then compared to other algorithms such as random swap clustering (RS), deterministic random swap clustering (DRS), repeated k-means (RKM), and k-means++ (KM++), on a variety of data sets. The experimental results indicate that the proposed algorithm requires 26% to 96% less processing time than the second fastest algorithm (RS) and avoids the local optimality problem better than the other swap strategies. To investigate the feasibility of the centroid ratio and PRS in a high-dimensional space, we modify the distance definition in the centroid ratio and PRS from the Euclidean distance to the cosine distance, and study them in document clustering.

2 Related Work

2.1 *k*-means

Given $X = \{x_1, x_2, \ldots, x_N\}$, a set of N points in a d-dimensional Euclidean space to be clustered, we define C and P as a specific partition of these points into M clusters, where $C = \{c_1, c_2, \ldots, c_M\}$ presents the centroids and $P = \{p_1, p_2, \ldots, p_N\}$ the point level partitions. A cost function is used to evaluate the quality of the clustering. There is no universal function for all clustering problems, and the choice of the function depends on the application. We consider the clustering as an optimization problem, and the *mean squared error* (MSE) is the most common cost function, calculated as

$$f = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} \left\| x_i - c_j \right\|^2 I(x_i \textit{ is closest to } c_j). \qquad (1)$$

where I is an indicator function. The k-means (Algorithm 1) is the most famous clustering algorithm, which aims at partitioning N objects into M clusters so that each object belongs to the cluster with the minimum Euclidean distance to the cluster centroid.

Algorithm 1: *k-means* algorithm

```
Input: $X$, $M$
Output: $C$, $P$, $MSE$
$c_j = x_i | i = random(1, N)$, $0 \le j \le M$ ;
while ! convergence do
    $p_i \leftarrow \underset{1 \le j \le M}{\operatorname{argmin}} \|x_i - c_j\|^2$, $\forall i \in [1, N]$ ;
    $c_j \leftarrow (\sum_{p_i = j} x_i) / (\sum_{p_i = j} 1)$ ;
    $MSE = f$ (see Eq. 1) ;
end
return $C$, $P$, $MSE$ ;
```

It is known that k-means has issue with initialization. With different initial solutions, k-means converges to different local minima, which makes the final result unstable. Previous work on improving the clustering results based on standard k-means has employed different strategies [2]–[7], [13], of which the swap-based approach is simple but effective.

2.2 Swap-based clustering

In swap-based clustering, the centroids are perturbed by a certain strategy in order to not get stuck in a local minima. A swap is accepted if it improves the clustering quality. This trial-and-error approach is simple to implement and very effective in practice.

The *Random Swap* algorithm (RS), originally called *Randomized Local Search* [3], is based on randomization: a randomly selected centroid is swapped to another randomly selected location. After that, a local repartition is performed and the clustering is fine-tuned by two k-means iterations. Pseudocode of the random swap algorithm is described in Algorithm 2.

To ensure a good clustering quality, the number of iterations for random swap should be set large enough to find successful swaps. For a more accurate analysis, the algorithm has a linear dependency on the number of data vectors, quadratic on the number of clusters, and an inverse dependency on the dimensionality [14].

Deterministic swap aims at finding good swaps by a systematic analysis rather than by trial-and-error. In

Algorithm 2: Pseudocode of *Random Swap*

```
Input: $X$, $M$
Output: $C$, $P$, $MSE$
$C \leftarrow InitializeCentroids(X)$ ;
$P \leftarrow OptimalPartition(X, C)$ ;
for $T$ times do
    $C^{new} \leftarrow RandomSwap(C)$;
    $P^{new} \leftarrow LocalRepartition(P, C^{new})$ ;
    $KmeansIteration(P^{new}, C^{new})$ ;
    if $f(P^{new}, C^{new}) < f(P, C)$ then
        $(P, C) \leftarrow P^{new}, C^{new}$ ;
    end
end
$MSE = f$ (see Eq. 1) ;
return $C$, $P$, $MSE$ ;
```

general, the clustering can be found in a few swaps only if the algorithm knows the centroid that should be swapped and the location where it should be relocated.

Several heuristic criteria have been considered for the selection of the centroids to be swapped, but simple criteria such as selecting the clusters with the smallest size or variance do not work very well in practice. Other approaches remove one cluster [15], or merge two existing clusters as in agglomerative clustering [16]. Deterministic removal takes N distance calculations for each of the M clusters. Thus, the overall time complexity of the deterministic removal step becomes $O(MN)$.

The replacement location of the swapped centroid can be chosen by considering the locations of all possible data points: this, however, would be very inefficient. In order to find the correct location, the task can be divided into two parts: select an existing cluster and select a location within this cluster. One heuristic selection is to choose the cluster that has the largest distortion (Eq. 1). The exact location within the cluster can be chosen considering the following heuristics: 1) current centroid of the cluster with small movement; 2) furthest data point; 3) middle point of the current centroid and furthest data point; 4) random.

With the random and deterministic swap strategies, an analysis combining the deterministic heuristic with random swap was conducted in [17].

3 Methodology

3.1 Centroid Ratio

The design of the internal indices is based on three elements: the data set, the point level partitions, and centroids. Mean square error (MSE) is a conventional criterion for evaluating clustering, which is calculated by these three elements. External indices [11], however, use only partitions by comparing the given clustering against the ground truth. The ground truth is usually built by using human assessors or the output of another clustering algorithm. External indices count the pairs of points of agreement or disagreement of the two partitions. These evaluation measures have been well studied in the literature [11], [12], [18].

A criterion such as MSE uses quantities and features inherent in the dataset, which gives a global level of evaluation. Since it relates to points and clusters, its time complexity is at least $O(MN)$. The partition-based criteria are based on pointwise evaluation of two partitions, which usually gives a time complexity of $O(N^2)$. The time complexity of point-pair measures can be reduced to $O(N + M^2)$ [19] by a contingency matrix.

There has been little research on cluster level evaluation measure based on centroids only. As an important structure of the clustering, the centroid reveals the allocation of the clusters. Two clusterings $\{X, P_1, C_1\}$ and $\{X, P_2, C_2\}$ from k-means are shown in Fig. 1, where the centroids and the partitions are highly correlated with each other. The partition shows little difference (left) at the border of the clusters, while the centroids also display little difference on the location. For incorrectly located centroids (right), the partitions differ greatly. The evaluation of the clustering can be performed on either the partition P or the centroids C. Motivated by this, we introduce a cluster level criterion in this section.

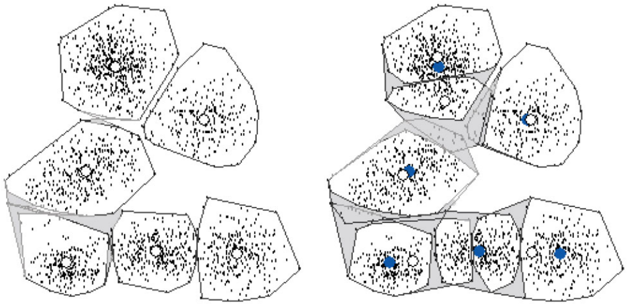


Fig. 1. Clusterings from k-means, showing the connection between centroids and partitions.

Let $C_1 = \{c_{11}, c_{12}, \ldots, c_{1M}\}$ and $C_2 = \{c_{21}, c_{22}, \ldots, c_{2M}\}$ be the centroids of two clusterings C_1 and C_2 respectively and $|C_1| = |C_2|$.

A pairing problem between two sets of centroids can be represented by a bipartite graph in which the vertex classes are the centroids in C_1 and C_2 separately, and centroids in C_1 are joined by edges to centroids in C_2.

Definition 1. *The Nearest Pairing of two sets of centroids (C_1 and C_2) is exactly the same as the minimum matching of a given bipartite graph in graph theory, where the nodes correspond to the centroids, the edges connect the centroids from different clusterings, and the edge cost stands for the centroid distance.*

The minimum matching in the nearest pairing (see Fig. 2) is solved here by a greedy algorithm. For each i,j, where $1 \le i \le M$, $1 \le j \le M$, we consider them to be paired if c_{2j} is the closest centroid to c_{1i} out of $\{c_{21}, c_{22}, \ldots, c_{2M}\}$. We thus iterate M times the operations

$$\begin{aligned} \{i, j\} &= \underset{c_{1i}\in C_1, c_{2j}\in C_2}{\operatorname{argmin}} \left\| c_{1i} - c_{2j} \right\|^2 \\ C_1 &\leftarrow C_1 \backslash \{c_{1i}\} \\ C_2 &\leftarrow C_2 \backslash \{c_{2j}\}. \end{aligned} \tag{2}$$

For paired centroids $c_{1i} \in C_1$ and $c_{2j} \in C_2$, we define the distances

$$\begin{aligned} D_1(i) &= \min_{c_{1s}\in C_1} \| c_{1i} - c_{1s} \|^2 \\ D_2(i) &= \min_{c_{2s}\in C_2} \left\| c_{2j} - c_{2s} \right\|^2 \\ D_{12}(i) &= \left\| c_{1i} - c_{2j} \right\|^2. \end{aligned} \tag{3}$$

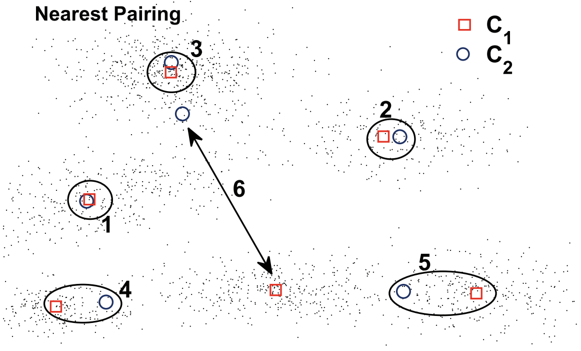


Fig. 2. Nearest pairing of two clusterings C_1 and C_2.

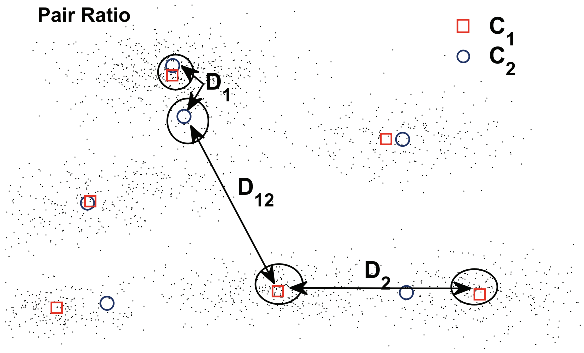


Fig. 3. Calculate the *Pair Ratio* for one pair of centroids.

The value of D_{12} is the distance of the matched centroids in two clustering results C_1 and C_2. Here, the "distance" need not be the Euclidean distance. D_1 is the nearest distance of two centroids in the same set of centroids C_1, and similarly, D_2 is the nearest distance in C_2. The centroids in two clustering sets are strictly matched when $D_{12} = 0$. We say the centroid i is *stable*, or *correctly located*, when $D_{12} \leq D_1$ and $D_{12} \leq D_2$.

Definition 2. *The Pair Ratio for centroid i, denoted by PR(i), is the degree of matching between centroid i from C_1 and C_2 after nearest pairing.*

We define the pair ratio for a centroid i of the clustering C_1 with respect to C_2 (see Fig. 3) by

$$PR(i) = \frac{D_{12}(i)}{D_1(i)} \times \frac{D_{12}(i)}{D_2(i)}. \tag{4}$$

A centroid i is said to be *stable*, or *correctly located*, when $PR(i) \leq 1$. For unstable and incorrectly located centroids, $PR(i) > 1$.

Definition 3. *The similarity S between two clusterings C_1 and C_2 is*

$$S(C_1, C_2) = 1 - \sum_{i=1}^{M} \gamma_i / M, \tag{5}$$

$$\textit{where } \gamma_i = \begin{cases} 1 & \textit{if } PR(i) > 1 \\ 0 & \textit{otherwise.} \end{cases}$$

Here, the value of S is in $[0, 1]$*, where* 1 *indicates a complete match of the two clusterings and* 0 *indicates a complete mismatch.* S_{id} *is the set of incorrectly located centroids in a pair of clusterings and* $S_{id} = \{i | PR(i) > 1\}$*.*

Definition 4. *Given T sets of clustering results, the degree of stability of centroid i is defined as*

$$stability(i) = \frac{\sum_{t=1}^{T} \sum_{s=1}^{T} (1 - \gamma_i)_{\{C_t, C_s\}}}{T^2}. \tag{6}$$

If the stability is 1, the centroid i is completely stable and 0 is completely unstable.

Definition 5. *The* Centroid Ratio *is defined as the union of the pair ratio (PR) and the similarity S, where PR finds incorrectly located centroids and the S value indicates the similarity of two clusterings.*

There are papers [20]–[22] that have addressed the question of finding the distance metric for clustering. One of the choices is to measure the L_q norm distance for some $q \geq 1$ between two optima from the clustering. For instance, the distance for two k-means clustering solutions can be taken to be the absolute value (L_1 norm) of their MSE values. In our paper, this provides another choice for measuring the similarity between the centroid sets of two clusterings. We show in the following that the two choices in k-means clustering are highly correlated with each other.

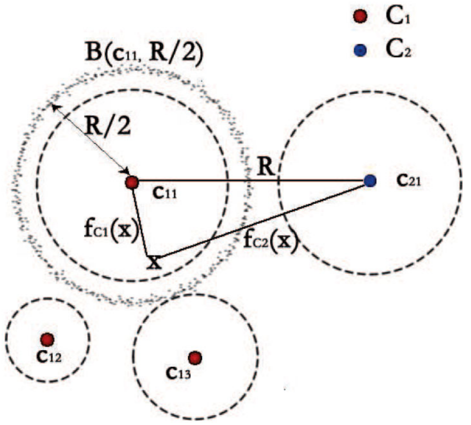


Fig. 4. Point x is located in a ball of radius $R/2$ centers at c_{11}. Centroids c_{11} and c_{21} are two paired centroids with distance R in clustering C_1 and C_2 respectively.

Let (X, C, P_r) be a probability space with a probability measure P_r over X. We assume that the data X lies in an Euclidean ball in $\mathbb{R}^d$, i.e. $X \subset \mathbb{R}^d$ and X is closed. Considering two sets of k-means clustering solutions (C_1, C_2), for any $x \in X$, where X follows the probability distribution Pr, we define:

$$f_{C_1}(x) = \sum_{i=1}^{M} \|x - c_{1i}\|^2 \, I(c_{1i} \textit{ is closest to } x)$$
$$f_{C_2}(x) = \sum_{j=1}^{M} \left\|x - c_{2j}\right\|^2 I(c_{2j} \textit{ is closest to } x). \tag{7}$$

and the total error for all of the points X is:

$$f_{C_1} = \int f_{C_1}(x) dP_r(x)$$
$$f_{C_2} = \int f_{C_2}(x) dP_r(x). \tag{8}$$

The MSE values for two solutions are f_{C_1}/N and f_{C_2}/N respectively.

Assume that $B(c_{11}, R/2)$ is a ball of radius $R/2$ centered at c_{11} and that the distance of $\|c_{11} - c_{21}\|^2 = R$ (see Fig. 4). We have $f_{C_1}(x) \leq f_{C_2}(x)$ for $x \in B(c_{11}, R/2)$. Note that $c_{2j} \notin B(c_{11}, R/2)$, $j \in 2, \ldots, M$. The L_1 norm distance for the two clustering solutions is

$$\begin{aligned} \|f_{C_1} - f_{C_2}\|_{L_1(P_r)} &= \int \left|f_{C_1}(x) - f_{C_2}(x)\right| dP_r(x) \\ \text{for } x \in B(c_{11}, R/2), \\ &\geq \int_{B(c_{11}, R/2)} \left|f_{C_1}(x) - f_{C_2}(x)\right| dP_r(x) \\ &= \int_{B(c_{11}, R/2)} \left(f_{C_2}(x) - f_{C_1}(x)\right) dP_r(x) \end{aligned}$$

$$
\begin{aligned}
&= \int_{B(c_{11},R/2)} \left(\sum_{j=1}^{1} \left\|x - c_{2j}\right\|^2 I - \sum_{i=1}^{M} \|x - c_{1i}\|^2 I \right) dP_r(x) \\
&\text{as } \|x - c_{11}\|^2 \geq f_{C_1}(x) = \sum_{i=1}^{M} \|x - c_{1i}\|^2 I, \\
&\geq \int_{B(c_{11},R/2)} \left(\|x - c_{21}\|^2 - \|x - c_{11}\|^2 \right) dP_r(x) \\
&\text{since } \|x - c_{21}\|^2 \geq \|c_{11} - c_{21}\|^2 - \|x - c_{11}\|^2 \\
&\qquad\qquad \geq R - R/2 = R/2, \\
&\geq \int_{B(c_{11},R/2)} \left(R/2 - \|x - c_{11}\|^2 \right) dP_r(x) \\
&\geq a_1 \int_0^{R/2} \left(R/2 - r^2 \right) r^{d-1} dr \\
&= a_1 \left(\frac{R}{2} \right)^{d+2} \left[\frac{1}{d} - \frac{1}{d+2} \right] \\
&= aR^{d+2} = a \, \|c_{11} - c_{21}\|^{2(d+2)}, \qquad (9)
\end{aligned}
$$

where a is a constant and d is the dimension of the data. From Eq. 9, we show that the L_1 norm distance of two k-means optima is highly correlated with the proposed similarity measure on two centroid sets. This can also be observed from Table. 3 in the experiment.

The value of the MSE reflects a global view, but there is no way to track the detailed information of each point through it. External indices such as the Rand index can compare two clusterings pointwise, but they do not directly provide information on clusters. The proposed centroid ratio can reveal information at a cluster level, which is able to give a global evaluation and detect unstable or incorrectly located centroids.

3.2 The Pairwise Random Swap algorithm

We introduce a pairwise random swap algorithm employing the centroid ratio, which efficiently ameliorates the local optimum problem of k-means. The pairwise random swap algorithm (PRS) takes a given data set X and the number of clusters M as inputs. It starts by generating two sets of centroids (C_1, C_2) and MSE values (MSE_1, MSE_2) from conventional k-means as described in Algorithm 3.

Then, we calculate the pair ratio value $PR(i)$ for each set of paired centroids in (C_1, C_2) to get a set of incorrectly located centroids S_{id} and the similarity value $S(C_1, C_2)$ using Eq. 5.

We perform the *Swap* function (Algorithm 4) to get improved solutions for both of the clusterings, in which we randomly swap the detected centroids c_{1j} and c_{2j} in C_1 and C_2 ($j \in S_{id}$) to a random location and fine-tune the result by k-means.

The algorithm stops when the similarity of the two centroid sets S is 1, which indicates that the centroids of the two clusterings are completely matched. The final solution of the PRS algorithm is the centroid set that has the lower MSE value, i.e., $\min(MSE_1, MSE_2)$.

On occasion, the initial centroid sets C_1 and C_2 are completely matched but the partition is locally optimal, i.e., $S(C_1, C_2) = 1$ and $S_{id} \in \emptyset$ at the beginning, in which case the PRS algorithm performs a random swap on the centroids.

Algorithm 3: *Pairwise Random Swap* clustering algorithm

```
Input: $X$, $M$
Output: $C$, $MSE$
Two initializations: $I_1, I_2$;
$(C_1, MSE_1)$ = k-means($X$, $I_1$, $M$);
$(C_2, MSE_2)$ = k-means ($X$, $I_2$, $M$);
Calculate $S_{id} = i|PR(i) > 1$ and $S(C_1, C_2)$;
while $S \neq 1$ do
    $(C_1^{'}, C_2^{'}, MSE_1^{'}, MSE_2^{'})$ = Swap($X$, $M$, $C_1$, $C_2$, $MSE_1$, $MSE_2$, $S_{id}$);
    $MSE_1 = MSE_1^{'}$; $MSE_2 = MSE_2^{'}$;
    $C_1 = C_1^{'}$; $C_2 = C_2^{'}$;
    Calculate $S_{id} = \{i|PR(i) > 1\}$ and $S(C_1, C_2)$;
end
return $\min(MSE_1, MSE_2)$ and corresponding $C_1$ or $C_2$;
```

The proposed algorithm is a type of deterministic swap clustering (DR) since the centroids to be swapped are chosen by the centroid ratio and the allocated position is random. The time complexity of the removal step is $O(M^2)$, and $O(1)$ for the addition step. Although the swap heuristic is capable of moving out of a local minimum, it may take a long time to move to near a local minimum. Thus, it is profitable to use k-means for fine-tuning after the swap heuristic [23]. A note for the PRS algorithm is that other prototype-based clustering algorithms can be used instead of k-means.

3.3 Efficiency Analysis

The efficiency of a swap-based clustering algorithm depends on two issues: how many iterations (swaps) are needed and how much time each iteration consumes. Swap-based clusterings can be categorized into four types in terms of the swap strategy: *RR*, *RD*, *DR* and *DD* [17].

In RR, the swap step is completely random so it needs a large number of iterations to provide a good

Algorithm 4: Function of *Swap*

```
Input: $X$, $m$, $C_1$, $C_2$, $MSE_1$, $MSE_2$, $S_{id}$
Output: $C_{r1}^{'}, C_{r2}^{'}$ and $MSE_{r1}^{'}, MSE_{r2}^{'}$
$MSE_{r1}^{'}$ = $MSE_1$ + 1;
while $MSE_{r1}^{'} > MSE_1$ do
    $C_{r1} \leftarrow$ random swap $S_{id}$ on $C_1$;
    $(C_{r1}^{'}, MSE_{r1}^{'})$ = k-means($X$, $C_{r1}$, $m$);
end
$MSE_{r2}^{'}$ = $MSE_2$ + 1;
while $MSE_{r2}^{'} > MSE_2$ do
    $C_{r2} \leftarrow$ random swap $S_{id}$ on $C_2$;
    $(C_{r2}^{'}, MSE_{r2}^{'})$ = k-means($X$, $C_{r2}$, $m$);
end
return $C_{r1}^{'}, C_{r2}^{'}$ and $MSE_{r1}^{'}, MSE_{r2}^{'}$;
```

TABLE 1
Summary of Time Complexities on One Iteration of Deterministic Swap

	RD	DR	DD	PRS
Removal	$O(1)$	$O(MN)$	$O(MN)$	$O(M^2)$
Addition	$O(N)$	$O(1)$	$O(N)$	$O(1)$
fine-tuning	$O(sN)$	$O(sN)$	$O(sN)$	$O(sN)$
Total	$O(sN+N)$	$O(sN+MN)$	$O(sN+MN)$	$O(sN+M^2)$

RD represents random removal and deterministic addition; DR, deterministic removal and random addition and DD, deterministic removal and deterministic addition. PRS is for the proposed PRS algorithm.

quality of result. It takes $O(sN)$ (s is the number of neighboring clusters on average) at least for each iteration with a fast variant of k-means for fine-tuning [24]. The main bottleneck of random swap is that the number of iterations T has depends quadratically on the number of clusters M [14], which increases the overall time complexity.

The selection criterion for swapping in *DR* and *RD* is to find clusters that involve the least increase in the cost function (MSE) when they are swapped. In *DD*, the centroid to be removed is chosen by calculating the removal cost, and the addition is made within the cluster of the highest distortion. In this case, the number of iterations is limited because the algorithm will stop whenever there is no improvement. However, the time required for each iteration is high. It takes $O(MN)$ to find the minimum removal cost, $O(N)$ for the addition cost, and $O(sN)$ for the local partition and fine-tuning, so the total time complexity of one iteration in *DD* is $O(sN + MN)$. The time complexities for the variants of deterministic swap are summarized in Table 1. As shown in the table, the time complexities of the existing deterministic strategies are either related to $O(N)$ or $O(MN)$. The time complexity of the swap strategies is the only difference in the total time complexity of the variants.

In the proposed method, the algorithm needs $O(M^2)$ to find incorrectly located centroids and $O(1)$ for the addition. The main computation is in the repartitioning and fine-tuning by the k-means iterations, which takes $O(sN)$. The total time complexity is $O(k_2(k_1sN + M^2))$, where k_1 is the number of iterations of k-means and k_2 is the repeated times of the centroid ratio step. It is shown by experiment that the selection of k_1 affects the final result very little, and the algorithm always stops after a relatively small number of iterations.

TABLE 2
Attributes of the Data Sets Used in Our Experiments

Name	Dimensionality	Data Size	#Clusters
Synthetic data sets			
S1-S4 [15]	2	5000	15
Aggregation [25]	2	788	7
R15 [26]	2	600	15
BIRCH1-BIRCH2 [27]	2	100000	100
Real data sets			
CM [28]	9	68040	NA(20)
CT [28]	16	68040	NA(20)
Documents			
re0 [29]	2886	1504	13
re1 [29]	3758	1657	25
tr31 [29]	10128	927	7
wap [29]	8460	1560	20

For the data sets where the number of clusters is unknown, the model sizes used in the experiments are shown in parenthesis.

To sum up, random swap needs a large number of iterations to provide a good quality of clustering. The deterministic swap needs fewer iterations, but takes more time for each iteration. For the variants of deterministic swap, the main computational burden comes from the local partitioning, and is the same for the different variants. However, the time complexities of the deterministic strategies differ, and the number of iterations depends on the swap strategy.

The time complexity for global k-means is $O(TM^2N^2)$ with incrementally adding one cluster at a time through a deterministic global search, where T is an average k-means iterations. The k-means++ algorithm has an additional procedure for choosing initial cluster centers, which adds $O(MN)$ to the time complexity of the standard k-means.

4 Experiments

We tested the algorithms using synthetic, real, and documental data sets from various sources as summarized in Table 2.

The synthetic data sets are two dimensional and contain a known number of clusters, which makes things easy from the visualization point of view. The ground truth labels are known for S1 to S4[1], which have gradually more overlapping clusters: i.e., in S1 the overlap is the smallest, whereas in S4 the overlap is the greatest. BIRCH sets [27] are large data sets with 100 clusters among 100,000 data points. BIRCH1 contains clusters in regular grid structures, BIRCH2 has clusters at a Sine curve. R15 [26] is generated as 15 similar 2-D Gaussian distributions that are positioned in rings. Data Aggregation (A7) [25] consists of seven perceptually distinct groups of points, where there are non-Gaussian clusters. The distributions of the two dimensional data sets are shown in Fig. 5.

The real data sets are the color moments (CM) and co-occurrence texture (CT) data sets from [30]. It is unknown whether the data is clustered. We selected the number of components of CM and CT to be 20 in the experiment, because the number of clusters of these two data sets are unknown and the problem of determining the number of clusters is outside the scope of this paper.

Documents [29] re0 and re1 are from the Reuters-21578 text categorization test collection, Distribution 1.0, and tr31 is from TREC. The data set wap is from the WebACE project (WAP), where each document corresponds to a web page listed in the subject hierarchy of Yahoo!. The ground-truth partitions are available for all of these data sets.

[1] http://cs.joensuu.fi/sipu/datasets/

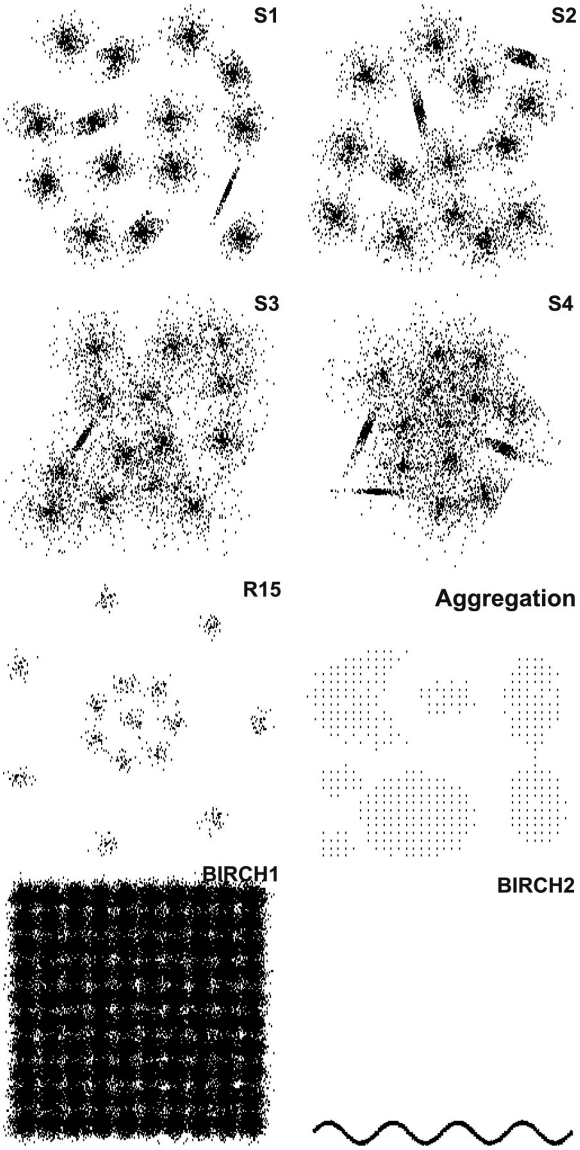


Fig. 5. Visualization of two-dimensional data sets.

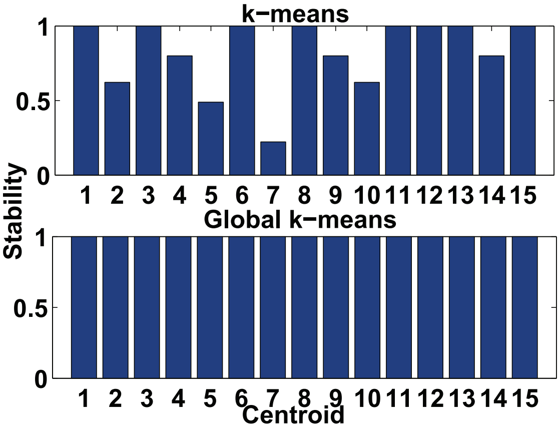


Fig. 6. Stability of each centroid, and finding unstable centroids by centroid ratio.

4.1 Centroid Ratio Validity

We study the validity of the proposed centroid ratio in this section. To compare with other clustering evaluation measures, we define consistency in terms of the similarity between their rankings on a number of clustering results. The compared measures include the Rand index (RI), the Adjusted Rand index (ARI), the Jaccard coefficient (Jac), the Fowlkes and Mallows index (FM), and ΔMSE. The similarity is based on Spearman's rank correlation, which is a nonparametric measure of the statistical dependence between two variables. The clustering results are obtained from 50 runs of standard *k*-means clustering on data set S2 with 15 clusters until convergence. The ground-truth labels are known for the data set S2. The 50 clustering results are compared with the ground-truth labels by the evaluation measures. The Rand correlation is then calculated pairwise between the two measures with 50 values each.

TABLE 3
Spearman's Rank Correlation for Different Clustering Validity Measures

	RI	ARI	Jac	FM	-ΔMSE	CR
RI	1	1	1	1	0.90	0.96
ARI	1	1	1	1	0.90	0.96
Jac	1	1	1	1	0.90	0.96
FM	1	1	1	1	0.90	0.96
-ΔMSE	0.90	0.90	0.90	0.90	1	0.94
CR	0.96	0.96	0.96	0.96	0.94	1

The values in Table 3 indicate that the external measures are very highly correlated with each other. The proposed centroid ratio has a higher correlation with the external measures than with ΔMSE. From the high correlation of the centroid ratio with the other measures, we conclude that the centroid ratio is valid for clustering evaluation.

Using the definition for the stability degree of centroids in Section 3.1, we tested the stability of the centroids in the standard *k*-means and Global *k*-means (GKM) [6] separately. We carried out $T = 10$ runs and calculated pairwise the degree of stability from ten clusterings using Eq. 6. The degree of stability for each centroid in *k*-means and GKM is shown in Fig. 6, centroids $2, 4, 5, 7, 9, 10, 14$ are not stable from *k*-means, while all centroids are stable in GKM. The degree of the stability is reflected in each centroid, for example, centroid 7 is the most unstable centroid in *k*-means.

4.2 Validity of the Pairwise Random Swap algorithm

We compare the PRS with other variants of *k*-means, including repeated *k*-means (RKM) and *k*-means++ (KM++) [7]. We also compare it with the Random Swap algorithm (RS) and the Deterministic Random Swap algorithm (DRS). The clustering algorithms[2] are implemented in C and tested under the same environment.

Swapping iterations are needed in RS and DRS and repetitions are needed for RKM and KM++ to guarantee good performance. We summarize the parameter settings for the experiments in Table 4. The number of swapping iterations in RS comes from [14]. For RKM and KM++, the number of repetitions was selected experimentally. All algorithms employ *k*-means, the number of iterations of *k*-means in RS and DRS is set to two, but runs until convergence in RKM and KM++.

[2] http://cs.joensuu.fi/sipu/soft/

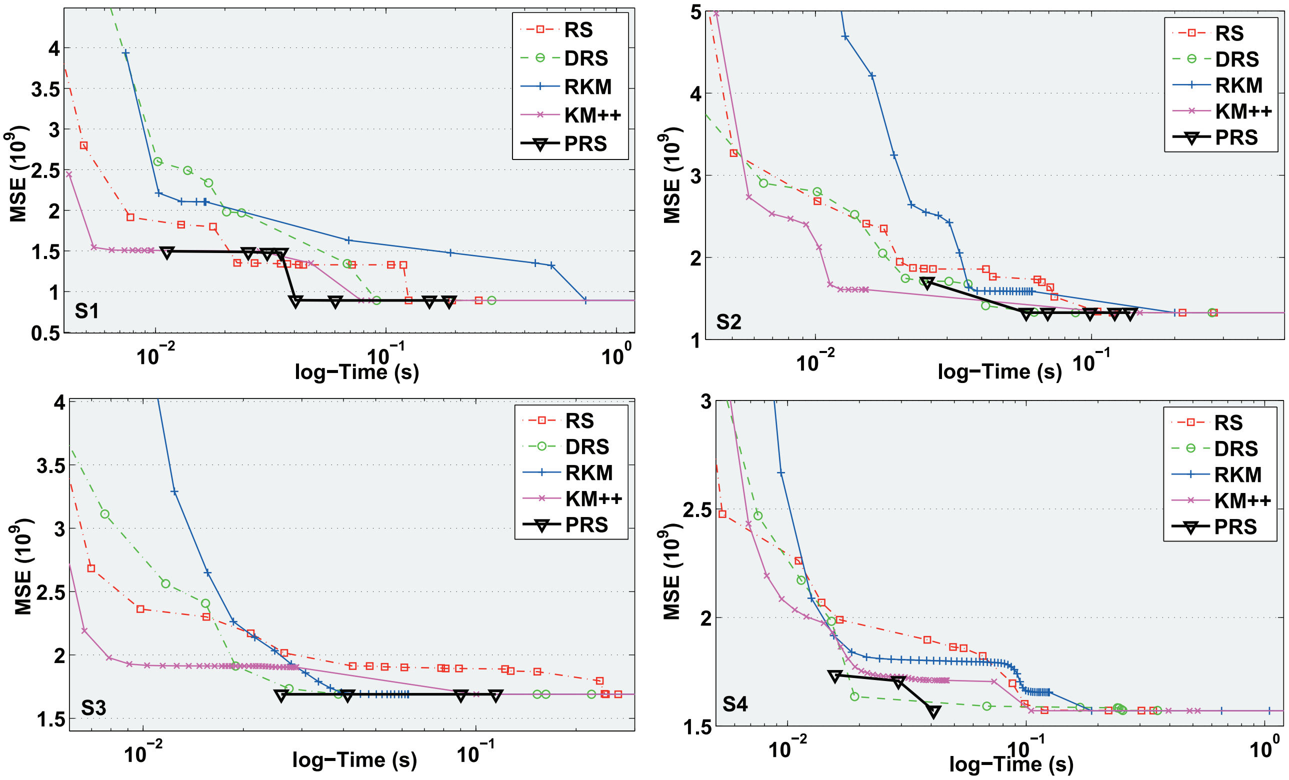


Fig. 7. MSE values with increasing time from clustering algorithms on S-sets.

We study the relationship between the number of k-means iterations, clustering result (MSE) and processing time (seconds) in Fig. 8. We repeated PRS 50 times on each number of iterations for k-means. The differences of MSE values among the runs with different numbers of k-means iterations are less than 0.0007%, which is negligible. The processing time has variance on each run with different numbers of iterations. However, a larger number of k-means iterations does not necessarily lead to a better result and higher processing time according to Fig. 8. Thus, we set k-means iterations in PRS to ten.

TABLE 4
Parameter Settings for RS, DRS, RKM and KM++

Data	RS/DRS	RKM/KM++
S1-S4	130	130
R15	130	130
A7	60	60
BIRCH1	1400	300
BIRCH2	10000	300
CM	2000	300
CT	2000	300

The numbers represent the number of iterations for swaps and repetitions in the experiments.

From the experiments on several synthetic data sets, we observe that the number of PRS iterations remains always $k_2 \leq M$. We verify this observation in Fig. 9, where PRS was run 100 times on data sets S1–S4, Aggregation, and R15.

One way to compare the performance of the methods is to plot the *MSE* values with increasing time. With enough processing time, the time-distortion figure can be used to check the estimated quality at the time axis.

We performed 50 runs of each algorithm on each data set to study their average performance (see Fig. 11). Box plots

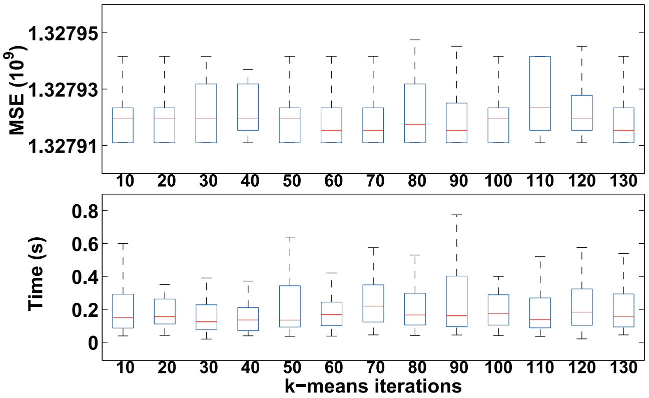


Fig. 8. MSE and time vs. the number of iterations of k-means in PRS for data set S2.

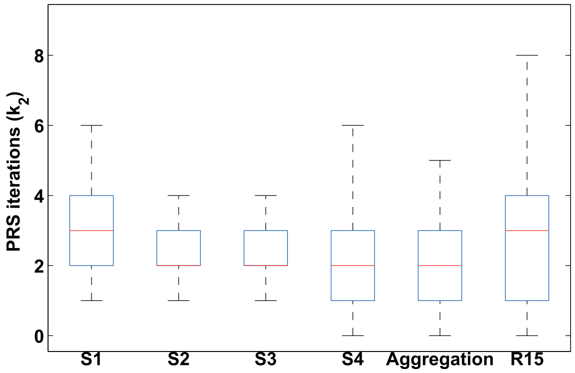


Fig. 9. Boxplot of the required number of PRS iterations. The probability is 100% for $k_2 \leq M$, where $M = 15$ for S1–S4 and R15, and $M = 7$ for Aggregation.

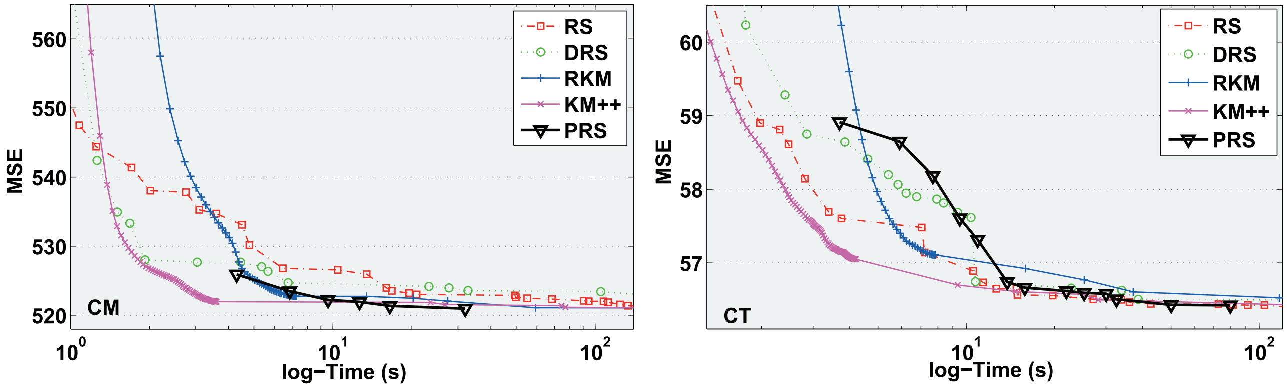


Fig. 10. MSE values with increasing time from clustering algorithms on CM and CT.

of the *MSE* values and processing times reflect the performance of the algorithms on average. Each box includes the minimum, the median (the central red line), the 25^{th} and 75^{th} percentiles (the edges of the box) and the maximum values.

The time-distortion plots on the S-sets (S1–S4) are compared in Fig. 7. Among the clustering algorithms (RS, DRS, PRS, RKM, KM++), PRS performs best in terms of the MSE and the processing time. Because of the stopping criterion, PRS stops while the other algorithms are still running. DRS reaches the local minimum faster than RS because DRS stops whenever there is no improvement and RS stops when the number of iterations has been reached. Deterministic selection converges faster than random selection. RKM is the most inefficient algorithm, since not every repetition helps the final result and a waste of computation exists in RKM. For example, too many repetitions fail to improve the result for S4 (see Fig. 7). KM++ reaches a local minimum as fast as RS, DRS and PRS, and the setting of the repetitions for KM++ is over-set in the experiment. This raises the question as to how many iterations are proper for RKM and KM++ in order to obtain a good performance in an efficient way.

As shown in the box plot for the S-sets (Fig. 11), enough running time guarantees good performance of RKM and KM++. The degree of overlapping of the S-sets increases the running time needed by RKM and KM++ and has a minor effect on the swap-based clustering algorithms. The running time of RS is stable. The swapping candidates in the deterministic swap depend on the selection criterion. Thus, both DRS and PRS have high variance in their processing times. PRS is a good choice according to its MSE values and processing time.

CM and CT contain multi-dimensional data. When the data has high dimensionality, the feature space is usually sparse [31]. The standard *k*-means algorithm for cluster analysis often does not work well in high dimensional spaces. Thus, the algorithms employing *k*-means are

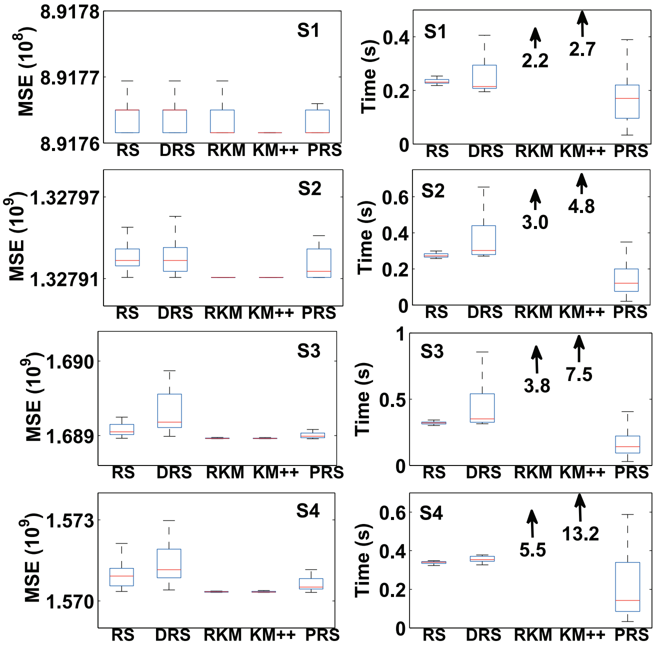


Fig. 11. Box plots of S-sets including minimum, 25th percentile, median, 75th percentile, and maximum. The central red line represents the median value, the edges of the box are the 25th and 75th percentiles.

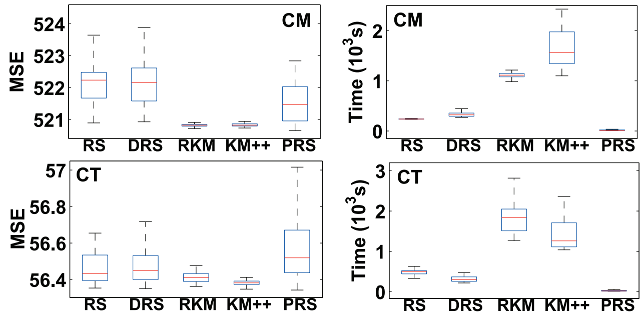


Fig. 12. Box plots of data sets CM and CT.

TABLE 5
Summary of the Median Processing Times (in Seconds)

	RS	DRS	RKM	KM++	PRS
S1	0.23	0.21	2.22	0.50	**0.2**
S2	0.27	0.30	2.97	4.59	**0.1**
S3	0.32	0.35	3.80	6.78	**0.1**
S4	0.34	0.35	5.47	12.93	**0.1**
A7	**<0.1**	**<0.1**	**<0.1**	**<0.1**	**<0.1**
R15	**<0.1**	**<0.1**	0.121	0.117	**<0.1**
BIRCH1	262	173	2413	1787	**134**
BIRCH2	315	413	535	539	**126**
CM	237	321	1112	1562	**14**
CT	497	306	1845	1261	**19**

TABLE 6
Comparison of Spherical *k*-means (SKM) and PRS for High-Dimensional Document Clustering by External Indices and Centroid Ratio

	RI		ARI		Jac		FM		CR	
	SKM	PRS	SKM	PRS	SKM	PRS	SKM	PRS	SKM	PRS
re0	0.68	**0.75**	0.03	**0.14**	0.14	**0.16**	0.25	**0.29**	0.08	0.08
re1	0.82	**0.86**	0.04	**0.10**	0.07	**0.09**	0.14	**0.17**	0.08	0.08
tr31	0.70	**0.75**	0.14	**0.30**	0.19	**0.30**	0.33	**0.46**	0	**0.14**
wap	0.82	**0.88**	0.18	**0.34**	0.16	**0.26**	0.29	**0.41**	0.10	**0.15**

The range of the values is [0,1], where one is a complete match and zero is a complete mismatch.

TABLE 7
MSE, PNSR (dB) and Processing Time (Seconds) of Different Clusterings on Subsampled Images at Quantization Level 32

image		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
MSE	RS	173	118	80	256	68	492	100	**38**	**101**	174	162	**82**	**76**	229	12
	FCM	191	151	159	267	69	417	97	40	114	201	194	90	85	238	17
	GA	177	118	**79**	260	67	399	102	**38**	103	171	158	82	77	233	**11**
	KM++	**134**	**100**	80	**188**	**53**	236	69	41	123	**110**	144	93	84	**208**	16
	PRS	140	101	107	189	58	**202**	**67**	43	137	114	**129**	93	90	212	16
PSNR	RS	25.7	27.4	29.1	24.0	29.8	21.2	28.1	**32.4**	**28.1**	25.7	26.0	**29.0**	**29.3**	24.5	**37.5**
	FCM	25.3	26.4	26.1	23.9	29.7	21.9	28.3	32.1	27.6	25.1	25.3	28.6	28.8	24.4	35.7
	GA	25.7	27.4	**29.2**	24.0	29.8	22.1	28.1	**32.4**	28.0	25.8	26.2	29.0	**29.3**	24.5	**37.5**
	KM++	**26.9**	28.1	29.1	**25.4**	**30.9**	24.4	29.7	32.0	27.2	**27.7**	26.6	28.4	29.0	**24.9**	36.1
	PRS	26.7	**28.8**	27.9	**25.4**	30.5	**25.1**	**29.9**	31.8	26.8	27.6	**27.0**	28.4	28.6	**24.9**	36.1
time	RS	152	133	78	150	125	139	129	203	112	117	105	89	131	162	42
	FCM	73	47	41	66	80	52	62	74	73	58	51	108	58	58	24
	GA	642	712	407	833	889	789	785	4353	530	623	577	685	666	756	259
	KM++	226	174	96	207	184	174	201	298	158	158	144	111	183	222	47
	PRS	**17**	**46**	**13**	**4**	**5**	**34**	**13**	**17**	**42**	**16**	**17**	**3**	**19**	**19**	**34**

restricted by the performance of *k*-means. RKM obtains a result a little bit better than KM++ on the dataset CM, while KM++ works better than RKM on CT. In terms of MSE values, RKM and KM++ work better than swap-based algorithms on both CM and CT (Fig. 10). However, the running times of RKM and KM++ are higher than those of swap-based algorithms. For a highly separated data space, the probability of getting a good swap is relatively low, which explains the high variance of the MSE values for RS, DRS and PRS. PRS performs better than RS and DRS on CM; however, PRS is not stable on CT (Fig. 12). In terms of the processing time, PRS is still the most efficient of the tested algorithms.

A summary table of the processing times in Table 5 presents the numerical results of the different algorithms. PRS requires 26% to 96% less processing time than the others on different data sets.

4.3 Extension for High-dimensional Document Clustering

The definition of the centroid ratio is not restricted to the Euclidean distance. The conventional *k*-means can be extended to spherical *k*-means by changing the Euclidean distance to the cosine distance [29]. We therefore extend the centroid ratio and PRS by modifying the distance function for high-dimensional data. Documents usually involve a feature space with thousands of dimensions. We tested the spherical *k*-means (SKM) and PRS algorithms with cosine distance on the documents listed in Table 2 using the ground-truth partitions for each document. The larger is the index value, the closer is the result to the ground-truth partition. We also use the centroid ratio to compare the clustering results. The partitions are converted to centroids by taking the mean value for each cluster. The results in Table 6 show that the PRS is effective in document clustering as well.

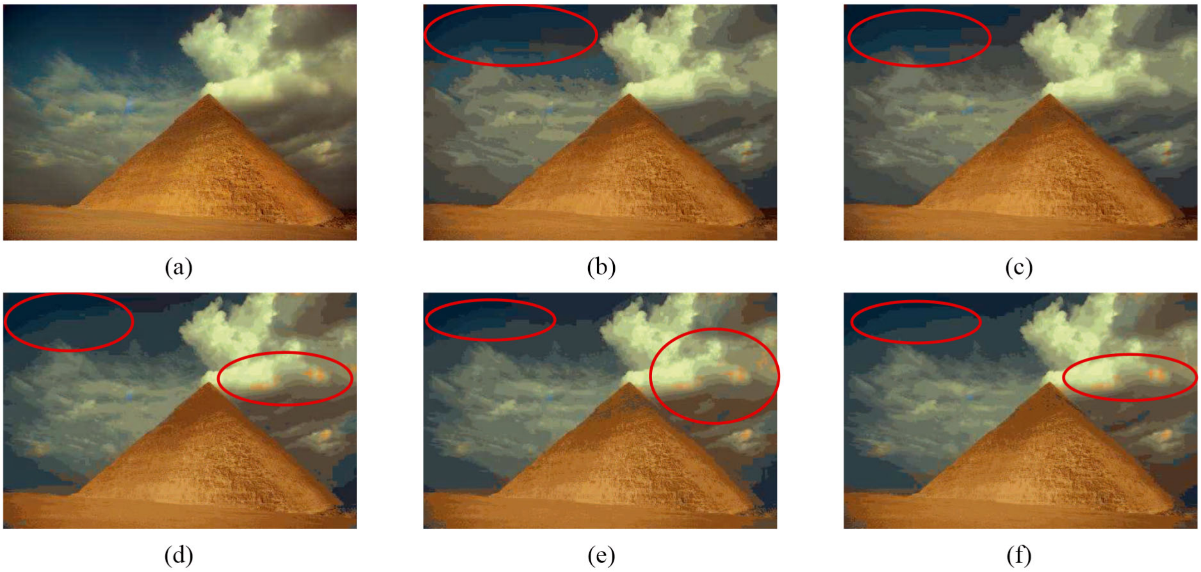
4.4 An Application to Image Color Quantization

The most straightforward application of clustering algorithms in image processing is color quantization. When the input data set is the color space of the images, the clustering points in three-dimensional space are treated as standard color quantizations. After the clusters have been located, typically the points in each cluster are averaged to obtain the representative color to which all the colors in that cluster are mapped.

We compare the proposed clustering algorithm with other popular clusterings on the images[3] for color quantization. The images are in RGB color space with a size of 481×321 pixels. In order to speed up the running time for all of the clustering algorithms, we reduced the amount of image data by a subsampling method [32]. The subsampling method can reduce the size of the image from 14% to 42% while the running time is thus reduced by from 55% to 94%. The difference of the MSE values for the original images and the subsampled images is from -23% to 19%. Based on the numbers, we conclude that the subsampling method is applicable to color quantization.

The proposed method is compared to the algorithms including random local search (RS), Fuzzy c-means (FCM) and Genetic algorithm (GA), and *k*-means++ (KM++). The evaluations of the clusterings by mean square error (MSE) and peak signal-to-noise ratio (PNSR) are

[3] http://cs.joensuu.fi/~zhao/DATA/



(a) (b) (c)

(d) (e) (f)

Fig. 13. Sample quantization results for image 11 at quantization level 32. The main difference is shown in the red circles.

listed in Table 7. Comparing the MSE and PNSR values, there is no clustering algorithm that works for all images. The performance is equally distributed among the algorithms and images. The proposed algorithm has the best performance in its running time. A visualization of the quantization results from the different algorithms is shown in Fig. 13.

5 Conclusion

We proposed a novel evaluation criterion called the centroid ratio, based on the centroids in prototype-based clustering, which compares two clusterings and detects unstable centroids and incorrectly located centroids. The centroid ratio is highly correlated with external indices and MSE values. Since the centroid ratio can detect incorrectly located clusters, it is employed as a swap criterion in the Pairwise Random Swap algorithm. Meanwhile, the similarity value obtained from the centroid ratio is employed as a stopping criterion in the algorithm. The algorithm has been compared with other algorithms, such as Random Swap, Deterministic Random Swap, Repeated *k*-means, and *k*-means++. It is the most efficient method among these algorithms according to the experimental results. The applications of the proposed algorithm to document clustering and color image quantization indicate that the algorithm is useful and is not restricted by the distance function of *k*-means. For high-dimensional data, the running time of the proposed algorithm has high variance, which can be improved in future work. The centroid ratio as a cluster validity index will also be studied in our future work.

Acknowledgments

The authors thank the Editors and the anonymous reviewers for their constructive comments. They also thank Dr. Z. Wen for his support on mathematical derivations. The work was partially supported by the National Natural Science Foundation of China (No. 61202382).

References

[1] A. K. Jain, “Data clustering: 50 years beyond K-means,” *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, 2010.

[2] A. Jain, M. Murty, and P. Flynn, “Data clustering: A review,” *ACM CSUR*, vol. 31, no. 3, pp. 264–323, 1999.

[3] P. Fränti and J. Kivijärvi, “Randomized local search algorithm for the clustering problem,” *Pattern Anal. Applicat.*, vol. 3, no. 4, pp. 358–369, 2000.

[4] G. Babu and M. Murty, “Simulated annealing for selecting optimal initial seeds in the k-means algorithm,” *Indian J. Pure Appl. Math.*, vol. 25, no. 1–2, pp. 85–94, 1994.

[5] K. Krishna and M. Murty, “Genetic k-means algorithm,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 29, no. 3, pp. 433–439, Jun. 1999.

[6] A. Likas, N. Vlassis, and J. Verbeek, “The global *k*-means clustering algorithm,” *Pattern Recognit.*, vol. 36, no. 2, pp. 451–461, 2003.

[7] D. Arthur and S. Vassilvitskii, “K-means++: The advantages of careful seeding,” in *Proc. 18th Annu. ACM-SIAM SODA*, Philadelphia, PA, USA, 2007, pp. 1027–1035.

[8] H. Xiong, J. Wu, and J. Chen, “K-means clustering versus validation measures: A data-distribution perspective,” *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 39, no. 2, pp. 318–331, Apr. 2009.

[9] C. Aggarwal, A. Hinneburg, and D. Keim, “On the surprising behavior of distance metrics in high dimensional space,” in *Proc. 8th ICDT*, vol. 1973. London, U.K., 2001, pp. 420–434.

[10] I. Dhillon, Y. Guan, and J. Kogan, “Iterative clustering of high dimensional text data augmented by local search,” in *Proc. IEEE ICDM*, Washington, DC, USA, 2002, pp. 131–138.

[11] J. Wu, H. Xiong, and J. Chen, “Adapting the right measures for k-means clustering,” in *Proc. 15th ACM SIGKDD Int. Conf. KDD*, Paris, France, 2009, pp. 877–886.

[12] Y. Liu, Z. Li, H. Xiong, X. Gao, and J. Wu, “Understanding of internal clustering validation measures,” in *Proc. 10th ICDM*, Sydney, NSW, Australia, 2010, pp. 911–916.

[13] S. Khan and A. Ahmad, “Cluster center initialization algorithm for k-means clustering,” *Pattern Recognit. Lett.*, vol. 25, no. 11, pp. 1293–1302, 2004.

[14] P. Fränti, O. Virmajoki, and V. Hautamäki, “Probabilistic clustering by random swap algorithm,” in *Proc. 19th ICPR*, Tampa, FL, USA, 2008, pp. 1–4.

[15] P. Fränti and O. Virmajoki, “Iterative shrinking method for clustering problems,” *Pattern Recognit.*, vol. 39, no. 5, pp. 761–775, 2006.

[16] H. Frigui and R. Krishnapuram, “Clustering by competitive agglomeration,” *Pattern Recognit.*, vol. 30, no. 7, pp. 1109–1119, 1997.

[17] P. Fränti and O. Virmajoki, "On the efficiency of swap-based clustering," in *Proc. 9th ICANNGA*, Kuopio, Finland, 2009, pp. 303–312.

[18] E. Dimitriadou, S. Dolnicar, and A. Weingassel, "An examination of indexes for determining the number of clusters in binary data sets," *Psychometrika*, vol. 67, no. 1, pp. 137–160, 2002.

[19] C. Michele and M. Antonio, "A fuzzy extension of some classical concordance measures and an efficient algorithm for their computation," in *Proc. 12th Int. Conf. KES*, Zagreb, Croatia, 2008, pp. 755–763.

[20] M. Meila, "Comparing clusterings—An information based distance," *J. Multivar. Anal.*, vol. 98, no. 5, pp. 873–895, 2007.

[21] S. Wagner and D. Wagner, "Comparing clusterings—An overview," Faculty of Informatics, Univ. Karlsruhe, Karlsruhe, Germany, Tech. Rep. 2006-4, 2006.

[22] A. Rakhlin and A. Caponnetto, "Stability of *k*-means clustering," in *Advances in Neural Information Processing System*, vol. 19. Cambridge, MA, USA: MIT Press, 2007.

[23] T. Kanungo *et al.*, "A local search approximation algorithm for *k*-means clustering," *Comput. Geom.*, vol. 28, no. 2–3, pp. 89–112, 2004.

[24] T. Kaukoranta, P. Fränti, and O. Nevalainen, "A fast exact GLA based on code vector activity detection," *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1337–1342, Aug. 2000.

[25] A. Gionis, H. Mannila, and P. Tsaparas, "Clustering aggregation," *ACM Trans. Knowl. Discov. Data*, vol. 1 no. 1, pp. 1–30, 2007.

[26] C. Veenman, M. Reinders, and E. Backer, "A maximum variance cluster algorithm," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 9, pp. 1273–1280, Sept. 2002.

[27] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: A new data clustering algorithm and its applications," *Data Min. Knowl. Discov.*, vol. 1, no. 2, pp. 141–182, 1997.

[28] A. Asuncion and D. Newman. (2007). *UCI Machine Learning Repository* [Online]. Available: http://archive.ics.uci.edu/ml/

[29] M. Steinbach, G. Karypis, and V. Kumar, "A comparison of document clustering techniques," in *Proc. Knowl. Discov. Databases Workshop Text Mining*, 2000.

[30] M. Ortega *et al.*, "Supporting ranked Boolean similarity queries in MARS," *IEEE Trans. Knowl. Data Eng.*, vol. 10, no. 6, pp. 905–925, Nov./Dec. 1998.

[31] S. Dasgupta and L. Schulman, "A probabilistic analysis of EM for mixture of separated, spherical Gaussians," *J. Mach. Learn. Res.*, vol. 8, pp. 203–226, Feb. 2007.

[32] T. Hasan, Y. Lei, A. Chandrasekaran, and J. Hansen, "A novel feature sub-sampling method for efficient universal background model training in speaker verification," in *Proc. IEEE ICASSP*, Dallas, TX, USA, 2010, pp. 4494–4497.

[33] P. Franti, M. Rezaei, and Q. Zhao, Centroid index: Cluster level similarity measure, Pattern Recognition.

Qinpei Zhao received the B.Sc. degree in automation technology from Xi'dian University, Xi'an, China in 2004. She received the M.Sc. degree in pattern recognition and image processing from Shanghai Jiaotong University, Shanghai, China in 2007. She received the Ph.D. degree in computer science from University of Eastern Finland, Joensuu, Finland in 2012. Her current research interests include clustering algorithm and multimedia processing.

Pasi Fränti received the M.Sc. and the Ph.D. degrees in computer science in 1991 and 1994, respectively, from the University of Turku, Finland. From 1996 to 1999, he was a Post-Doctoral Researcher with the University of Joensuu, funded by the Academy of Finland. Since 2000, he has been a Professor with the same department. He has published 61 journals and more than 140 peer-review conference papers, including 10 IEEE Transactions' papers. He is currently the Head of the East Finland Doctoral Program in Computer Science and Engineering. His current research interests include clustering algorithms, vector quantization, lossless image compression, voice biometrics, and location-based systems. He is a senior member of the IEEE.