Journal of Computers Vol. 33 No. 5, October 2022, pp. 1-17
doi: 10.53106/199115992022103305001

DP-Kmeans and Beyond: Optimal Clustering with a New Clustering Validity Index

Zhu-Juan Ma[1*], Zi-Han Wang[2], Xiang-Hua Chen[3], Feng Liu[2]

[1] School of Big Data and Artificial Intelligence, Anhui Xinhua University, Hefei 230088, China
zjmsjtu16@163.com
[2] School of Computer Science and Technology, Anhui University, Hefei 230601, China
w64074511@163.com; fengliu@ahu.edu.cn
[3] Computer Applied Research Institute of Traditional Chinese Medicine,
Anhui University of Traditional Chinese Medicine, Hefei 230038, China
xhchen62@163.com

Received 27 September 2021; Revised 2 February 2022; Accepted 2 March 2022

Abstract. The K-means clustering algorithm is widely used in many areas for its high efficiency. However, the performance of the traditional K-means algorithm is very sensitive to the selection of initial clustering centers. Furthermore, except the convex distributed datasets, the traditional K-means algorithm still cannot optimally process many non-convex distributed datasets and datasets with outliers. To this end, this paper proposes the DP-Kmeans, an improved K-means algorithm based on the Density Parameter and center replacement, which can be more accurate than the traditional K-means by dropping the random selection of the initial clustering centers and continuous updating of the new centers. Due to the unsupervised learning feature, the number of clusters and the quality of data partitions generated by the clustering algorithm cannot be guaranteed. In order to evaluate the results of the DP-Kmeans algorithm, this paper proposes the SII, a new clustering validity index based on the Sum of the Inner-cluster compactness and the Inter-cluster separateness. Based on the DP-Kmeans algorithm and the SII index, a new method is proposed to determine the optimal clustering numbers for different datasets. Experimental results on ten datasets with different distributions demonstrate that the proposed clustering method is more effective the existing ones.

Keywords: K-means, clustering validity, optimal clustering number, data mining

1 Introduction

As a classical data mining method, clustering is widely used in fields of pattern recognition, artificial intelligence and so on, which can find the underlying structures of the datasets without prior information. Clustering aims at dividing the dataset into several clusters where data points in the same cluster are as similar as possible [1]. Tons of clustering algorithms have been proposed in many areas. Among them, the K-means is one of the most popular algorithms because of its simplicity, effectiveness, and scalability [2-3]. The K-means algorithm is consisted of two stages, initialization, and iteration. Initialization computes the initial clustering centers randomly, while iteration normally takes two sub-steps: The first sub-step, assignment, assigns all the data points to their nearest clusters, and then the second sub-step, updating, re-computes the clustering centers with the assigned data points.

The performance of the K-means highly depends on the initialization stage [4], whose, random selection property may seriously degrade the clustering accuracy. Several re-initializations may be needed before achieving the acceptable clustering quality. Even, an improper initialization will bring exponential running time overhead [5]. In addition, the updating sub-step in the iteration of the K-means may also seriously affect the performance of the K-means. Here, the average of all data points in each cluster is calculated to get its new center. However, oftentimes, the new center can be disturbed by the outliers because improperly processing outliers will cause the big deviation between the computed center and the real one.

In summary, the performance of the K-means can be affected by the initialization and update of the clustering centers, i. e., the random selection of initial clustering centers and incorrect update due to the outliers. To overcome these two problems, this paper proposes an improved K-means, DP-Kmeans. In this algorithm, methods of density parameter and center replacement are introduced to select the initial clustering centers and accurately compute the new clustering centers respectively.

* Corresponding Author

Clustering algorithms provide the effective ways to partition different datasets. However, it is difficult to determine how many clusters are feasible to a given dataset because of their unsupervised learning property. Different partitions can be obtained with different clustering algorithms under different parameters [6]. Meanwhile, the quality of data partitions generated by the clustering algorithm cannot be assured. The clustering validity index (CVI) is widely adopted to evaluate the results of the clustering algorithms, which determines the optimal clustering number (K_{opt}) of a given dataset. CVI takes the target dataset and clustering number K as parameters and computes the optimal (usually the biggest value or the smallest value) index value, K_{opt}, by repeatedly executing the CVI function with different values of K.

Many CVIs have been proposed. However, there is no CVI can optimally process all types of datasets [7]. For example, the commonly used CH [8] is very effective for the convex datasets. However, it cannot work well with non-convex and unbalance datasets. The SIL [9] can be applied to many types of datasets. However, its performance is poor when it is used to handle the overlapping datasets. The COP [10] is good at the convex datasets and the partially overlapping datasets. But it cannot well cope with the non-convex datasets. The SMV [11] is able to process datasets with noisy data points but has the difficulty in dealing with the non-convex and overlapping datasets. To stably evaluate the clustering results for varieties of datasets, this paper proposes the SII, a novel clustering validity index based on the linear combination of the newly defined inner-cluster compactness and the inter-cluster separability. Generally speaking, the contributions of this paper are as follows:

(1) *The new clustering algorithm, DP-Kmeans*. In the DP-Kmeans, the density parameter of each data point of the target dataset is computed. By taking the evenly distributed and large density parameter points as the initial clustering centers, the problem of performance instability of the K-means caused by the random selection initial clustering centers is resolved. To eliminate the influence caused by the outliers, in each iteration, the real points closest to the fake centers generated by the traditional K-means are taken as the clustering centers of the DP-Kmeans. These two improvements ensure that the DP-Kmeans is more stable and accurate than the existing ones.

(2) *The novel clustering validity index, SII*. Like many existing CVIs, the SII index is defined by the linear combination of the inner-cluster compactness and inter-cluster separability. However, there is generally a large difference between the values the inner-cluster compactness and inter-cluster separability. In order to gap the two values, the weighted value is introduced to balance the influences of the two values on the SII index. By this improvement, the new SII index can stably evaluate the clustering results for more datasets than many existing CVIs.

(3) *The new K_{opt} determination method*. By the combination of the DP-Kmeans and minimum value of the SII index, a new method is proposed to determine the K_{opt} for many types of datasets, such as the convex/non-convex datasets, the balance/unbalance datasets, the arc datasets, and datasets with outliers.

This manuscript is the continuation work of the previous conference paper "Effective Clustering Analysis based on New Designed CVI and Improved Clustering Algorithms" [12]. Here, the original density parameter-based K-means algorithm is extended by the center replacement method to accurately process more kinds of datasets. A novel SII based on the linear combination of the newly defined inner-cluster compactness and inter-cluster separability is defined to effectively evaluate the results of the clustering algorithms. Meanwhile, more datasets and existing clustering methods (including clustering algorithms and CVIs) are taken to test the performance of the proposed clustering method.

The remainder of this paper is organized as follows: Section 2 gives a brief overview of the related works. The DP-Kmeans, the SII, and the K_{opt} determination method are discussed in Section 3, 4 and 5 separately. Section 6 discusses the experimental results with the whole paper concluded in Section 7.

2 Related Work

The clustering analysis research of the K-means can be divided into two types: clustering algorithm and CVIs. The former focuses on partitioning the target datasets into several clusters, while the latter aims to evaluate the partitions of datasets computed by clustering algorithms.

2.1 Improvements on K-means Algorithm

Due to huge impact on the performance, many works are proposed to improve the initialization stage of the K-means. Redmond [4] used the kd-trees to improve the initialization of the K-means where the kd-tree is used to estimate the density of data points at various locations with the highest estimated densities taken as the near

optimal clustering centers. This estimation method is high efficiency in determining the initial clustering centers. However, it cannot guarantee all the estimated centers are the correct. The GK-means [13] combined the grid structure and the spatial index with the K-means algorithm so that the initial clustering centers can be effectively specified. However, its performance highly depends on the partition of the grids. Yoder [14] proposed the semi-supervised K-means++, which finds the clustering centers incrementally. In the K-means++, the first center is randomly selected while the other centers are specified as far as possible from the first one. This algorithm works well based on the suitable selection of labeled data points. Hussain proposed kCC [15], a unified framework of co-clustering on the K-means, to improve the initialization of the K-means (and K-means++) as well as the ability of handing high dimensional datasets. However, its complexity is much higher than many other improvements of the K-means. Huang et al. [16] proposed W-K-means, which assigns different weights to features and selects the central point by feature weighting. This algorithm comprehensively considers the influence of datasets with different dimensions on the clustering results. However, it does not explain the relationship between the feature weight and the scale of feature values. Inspired by the Canopy clustering algorithm, Zhang et al. [17] proposed the DC-K-means algorithm, which uses the sample density to find the initial points. This algorithm works well on the low-density datasets. However, it cannot properly cluster datasets with many outliers. In order to accurate find initial clustering centers, the improved K-means proposed by Wang et al. [18] integrates the individual sample density, the dimension-weighted Euclidean distance, and the local-global sample distributions. This algorithm eliminates the impact of outliers on performance. However, it is time-consuming on clustering large scale datasets.

There are also many works on improving or balancing the performance, accuracy, and stability of the K-means. To avoid the local optima and sensition to noise in the traditional K-means, Islam [19] proposed GenClust, which intervenes fast hill-climbing cycles of the K-means and thus obtain the high quality of the clustering results quickly. However, this algorithm cannot process large scale datasets due to the high computational complexity. Fadaei [20] proposed the re-clustering method to reduce the processing cost of the K-means in dynamic networks. In this method, the number of checked nodes and the total consumed time during the iterations are reduced. However, this method need carefully tradeoff between the clustering accuracy and data processing speed. The Grid-K-means [21] is proposed to overcome the drawbacks (low efficiency, poor clustering accuracy and more sensitivity to noise points) of the traditional K-means. It combines the idea of meshing in grid clustering with K-means and dynamically changes grid operations to substitute data point operations. Meanwhile, grid step size and threshold value dynamically change with the varying K. Based on the Gaussian-weighted distance, Zhang et al. [17] proposed a modified clustering method for coarse K-means. However, this method is not widely applicable which only effective for those real datasets without clear boundaries. Jones et al. [22] proposed the FilterK algorithm, an improvement of K-means by reducing the effect of outliers. Based on the grid density of data objects, Fan et al. [23] improved the K-means by extracting data objects from all dimensions. In this improvement, the initial clustering centers are selected by calculating the average value of all dimensions.

The above researches mainly target at easing the first problem of the K-means, i.e. the initialization of the clustering centers [24]. However, the influence of the iteration stage on the performance of the K-means is not well studied. Due to the existence of outliers, the positions of some clustering centers specified by the initial stage may deviate from the actual ones. However, the DP-Kmeans improves the K-means in both stages. It uses the density parameter to incrementally select the initial clustering centers and uses the center replacement method to update the deviated clustering centers.

2.2 The Clustering Validity Indexes

Existing CVIs can be broadly divided into three categories according to their components: The CVIs based on the geometric structures of datasets, the CVIs based on the membership of data points and the CVIs based on the combination of geometric structures and membership degree of datasets.

The DI [25] presented by Dunn in 1974 is the representative CVI of the first category. This index is calculated by the ratio of the minimum distance between clusters to the maximum distance within clusters. Due to sensitive to the outliers and noise data points, except for the convex datasets, the DI index cannot properly process datasets with irregular spatial distributions. By standardizing the compactness within clusters, Hubert presented the CI [26]. This index is simple and easy to calculate by only considering the compactness within clusters. However, this index is not stable in processing many kinds of datasets. The I [27] proposed by Maulik is composed of three components: $1/K$, E_1/E_K and $\max_{i,j \in [1,K]} d(v_i,v_j)$. Where, K is the clustering number; $d(v_i,v_j)$ is the Euclidean distance between clustering centers v_i and v_j; E_K is defined as $\sum_{k \in [1,K]} \sum_{j \in [1,n]} u_{kj}\, d(v_j,v_k)$. The three components reactive with each other to constitute the I index. However, this index is unstable due to the uncertainty of the param-

eters. The CH [8] is determined by the ratio of inner-cluster compactness and the inter-cluster separability. The CH is superior to most of CVIs in evaluating the results of the clustering algorithms. The DBI [28] is defined by the ratio of the inner-cluster compactness and the inter-cluster separability. This index is suitable for processing the non-convex datasets. However, it cannot properly deal with the overlapping datasets. The SIL [9] can process the datasets with different spatial distributions. However, it is difficult to process the overlapping datasets. Meanwhile, the computational complexity of the SIL index is higher than many other indexes. The COP [10] is also based on the ratio of the inner-cluster compactness and the inter-cluster separability. The inner-cluster compactness is calculated by the average distance between all the data points to the cluster center. For all the clusters of the target dataset, the inter-cluster separability is measured by farthest distances among different centers. This index works well for datasets with the characteristic of "within-cluster compactness, between-cluster separation". However, the greater the overlap of the target dataset is, the worse the performance of COP index has. The SMV [11] is the newly developed index which uses the new measurement, also called the dual center, to represent the separation among clusters. The SMV index has high accuracy but narrow range of applications.

The CVIs based on the degree of membership of datasets are mainly used to evaluating partitions of the fuzzy clustering. The PC (partition coefficient) [29] and PE (partition entropy) [30] are the classical CVIs of this category. Both show the monotonic trend with the changing number of clusters. The two indexes are efficient for the fuzzy clustering. However, the two indexes exhibit poor performances on the large-scale datasets. Zalik proposes the CO_r [31] for fuzzy clustering based on the compactness and overlap measures. Here, only the data points with sufficient membership are considered to compute the compactness and only the data points with small degree of membership between two clusters are considered to estimate the overlapping degree. This index is stable and effective when evaluating partitions with clusters that widely differ in size or density. However, it is difficult to find the K_{opt} for the non-convex datasets. Kim presented the OS [32] for fuzzy clustering. This index is constructed based on the ratio of the overlapping degree to the separability. The P [33] is set up as a fuzzy parameter, this index is not stable as the other fuzzy indexes because of the uncertainty of the fuzzy parameter.

Tang proposed the VT [34] to overcome the monotone decreasing trend when the clustering number tends to the sample number and the strong interaction between the clustering number and the fuzziness. Based on the geometric structure and the membership degree of datasets, this index avoids the numerical instability of validation index when fuzzy weighting exponent increases. The PBM [35] is the product of the three factors competing to achieve the optimal value of the index. The maximum value of PBM indicates the appropriate partitioning of the target dataset. The PCAES [36] is composed of two items to measure the compactness: The first item calculates the sum of the ratio of squared fuzzy membership and its minimum value and the second item (the exponential index) calculates the relative value of the distance between the center points of the two nearest clusters. This index is built with a complex structure and thus more computations are needed than those which only consider the geometric structure information or membership degree of datasets.

It can be seen that there is no CVI can optimally process all types of datasets. Many of the existing CVIs have good clustering performance for the datasets of "within-cluster compactness, between-cluster separation". However, most of them cannot properly deal with datasets with non-convex distributions and datasets with a large degree of overlapping. The SII index proposed in this paper attempts to optimal process more types of datasets, thanks to the redefined inner-cluster compactness and inter-cluster separability.

3 DP-Kmeans: An Improved K-means Algorithm

The K-means has been widely applied to many data division problems. However, it suffers from low accuracy and cannot optimally process non-convex datasets and datasets with lots of outliers. The DP-Kmeans which is based on the density parameter and the center replacement is introduced to overcome these shortcomings.

3.1 Finding the Initial Clustering Centers

We first show how to robustly select the initial clustering centers, which is based on the following assumptions: "*In the Euclidean space R^m, a m-dimensional dataset $D=\{x_1, x_2, ..., x_n\}$ contains n data points. In this dataset, each points $x_i=\{x_{i1}, x_{i2}, ..., x_{im}\}$ has m feature attributes. Meanwhile, the dataset D should to be divided into K clusters.*"

In the dataset D, the Euclidean distance between the two different points x_i and x_j (i, j=1, 2, …, n) can be computed as:

$$d(x_i, x_j) = | x_i - x_j | = \sqrt{(x_{i1} - x_{j1})^2 + ... + (x_{in} - x_{jn})^2} \ . \quad (1)$$

Consequently, the largest (marked as *LaDist*) and smallest (marked as *SmDist*) distances of all the data point pairs are defined as follows:

$$LaDist = \sum_{i=1}^{n-1} max_{1 \le i < j \le n} d(x_i, x_j)^2 \ . \quad (2)$$

$$SmDist = \sum_{i=1}^{n-1} min_{1 \le i < j \le n} d(x_i, x_j)^2 \ . \quad (3)$$

The dataset D is supposed to be divided into K clusters, however, for different clustering algorithms, the sizes of generated clusters may be different. When the sizes of clusters change, the distances between each sample point pair is also changed. To this end, the dynamic average distance (marked as *DyAveDist*) that evaluates the average values of the largest and smallest distances for all data points is defined as follows:

$$DyAveDist = \frac{LaDist + SmDist}{2 \times K} \ . \quad (4)$$

Based on the dynamic average distance, the density parameter can be defined as Definition 1.

Definition 1 (Density parameter, ρ). In the dataset D, the number of data points in the circular region with x_i (i = 1, 2, …, n) as the center and the *DyAveDist* as the radius is called the density parameter of the data point x_i. Assume $u(x)$ (if $x \geq 0$, $u(x)=1$, otherwise, $u(x)=0$) to be the jump function. Specifically, the density parameter of x_i can be calculated as:

$$\rho(x_i, DyAveDist) = \sum_{i=1, j \neq i}^{n} u(DyAveDist - d(x_i, x_j)) \ . \quad (5)$$

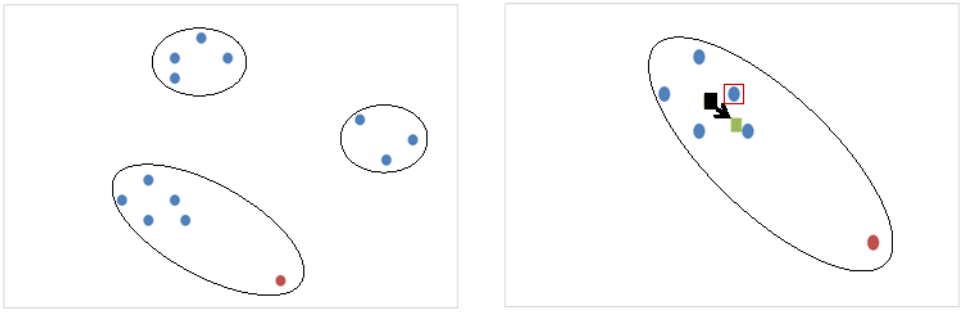
The initial clustering centers can be specified with the density parameter of each data point. Actually, the data point with the highest density parameter is taken as the first initial clustering center when finding initial clustering centers. Meanwhile, data points in the circular region corresponding to the first initial clustering center are all removed from the original dataset D. Then, the second initial clustering center is the data point with the highest density parameter in the remainder points of dataset D. This process is continued until all the K initial clustering centers are specified.

3.2 Replacing the Clustering Centers

The initial clustering centers are not randomly selected with the density parameter. Therefore, it is more stable than the traditional K-means. However, it is still sensitive to the outliers. Actually, some of the initial clustering centers generated by the traditional K-means may not be the real points (these points are called *fake center* in this paper) in the target datasets. Furthermore, due to the influence of the outliers, the estimated clustering centers will deviate from the actual ones. This problem will seriously degrade the precision of the traditional K-means.

Interestingly, the centers generated by the K-medoids clustering algorithm are always the real data points of the target dataset. Inspired by the K-medoids, this paper proposes the center replacement method to update the *fake center* generated by the traditional K-means. Specifically, the *fake center* is replaced by its nearest neighbor point within this cluster. Meanwhile, this neighbor point should as far away as possible from the outliers. Our method only updates the *fake centers*, which is taken by the K-medoids algorithm. The real initial clustering centers (generated by the K-means) are still taken as the clustering centers of the final result of the clustering. In the process of the clustering, the *fake centers* are updated one by one until all the real clustering centers are specified.

As an example, Fig. 1 shows the *fake center* replacement for the cluster that contains an outlier point. Here, the blue dots represent the normal data points while the red ones represent the outliers. Fig. 1(a) shows the dataset consisted of three clusters that are generated by MATLAB randomly, where the cluster in the lower part contains the outlier point. This cluster is shown in Fig. 1(b) separately for a clear view.



(a) A dataset contains the outlier point (b) Clustering center replacement

Fig. 1. An example of the outlier and its replacement

In Fig. 1(b), the black rectangle is taken as the center by the traditional K-means algorithm if there is no interference from outlier point. As can be seen in this figure, this is the *fake center* of this cluster. However, the center will deviate from the "actual" (specified by the traditional K-means algorithm) one if the outlier point is taken into the consideration. As in Fig. 1(b), the center is moved from the black rectangle to the green rectangle along the arrow. This deviation will result in bad clustering performance. Actually, a large number of data points that do not belong to this cluster will be included in the next iteration of the clustering algorithm with this deviation. In Fig. 1(b), the blue dot in the red rectangle is taken as the final clustering center by our improved method. This center is the actual point in the dataset which is the nearest actual data point to the black rectangle. Meanwhile, it is far from the outlier point.

3.3 Description of the DP-Kmeans Algorithm

The DP-Kmeans incorporates the density parameter and the center replacement to the traditional K-means algorithm (Fig. 2). The DP-Kmeans can not only stably find the clustering centers but also deal with the outliers. In this algorithm, Step (1) computes the dynamic average distances of all the data point pair (x_i, x_j) in the dataset D. Step (2) computes the density parameter for all of the data points. Based on the density parameter, Step (3) finds the K initial clustering centers of the dataset D and puts them into the set V. Step (4) - Step (8) iteratively form the final division of the dataset D. Specifically, Step (5) initializes each cluster; Step (6) puts the data points into the corresponding clusters; Step (7) updates the clustering centers by the center replacement method.

Suppose the target dataset $D=\{x_1, x_2, \ldots, x_n\}$ has n data points. Meanwhile, each data point $x_i=(x_{i1}, x_{i2}, \ldots, x_{im})$ has m attributes. The DP-Kmeans clustering algorithm divides dataset D into K clusters $\{C_1, C_2, \ldots, C_K\}$ by p iterations. The number of center replacements is c. The computational complexity of the DP-Kmeans algorithm is calculated as follows:

(1) The computational cost of obtaining the dynamic average distances of all the data point pairs (x_i, x_j) can be:

$$T_1(n) = m \times n^2 + n^2/2 + n^2/2 + 1. \quad \textbf{(6)}$$

(2) The computational cost of finding the K maximum density points and taking them as the initial clustering centers can be:

$$T_2(n) = n^2 + n + K \times n. \quad \textbf{(7)}$$

(3) The computational cost of putting the remainder data points into the corresponding nearest clusters can be:

$$T_3(n) = K \times m \times n. \quad \textbf{(8)}$$

(4) The computational cost of getting the new clustering centers by the center replacement method can be:

$$T_4(n) = K \times m \times n + c \times K \times m \times n. \quad \textbf{(9)}$$

(5) Since the DP-Kmeans algorithm repeatedly executes Step (5) – Step (7) by p times, the whole computational cost of the algorithm can be:

$$T(n) = T_1(n) + T_2(n) + p\times(T_3(n)+T_4(n)+K\times|C_i|^2). \quad \textbf{(10)}$$

In the general cases, the values of K, m, c and p are far less than the value of n (K, m, c, $p << n$). They can be taken as constants. So, the computational complexity of the DP-Kmeans is $O(n^2+n^2+p\times K\times m\times n\times|C_i|^2) = O(n^2)$.

Input: Dataset $D=\{x_1, x_2, \ldots, x_n\}$; The value of the clustering number K.
Output: Dataset D is divided into K clusters $D=\{C_1, C_2, \ldots, C_K\}$.

(1) Calculate the dynamic average distances (*DyAveDist*) of all the data points pair (x_i, x_j) in the dataset D;
(2) for i =1, 2, ..., n do
  Calculate the density parameter $\rho(x_i, DyAveDist)$ of data point x_i ;
(3) for j =1, 2, ..., K do //Get K initial clustering centers and put them into the set V.
  Select the data point x_j with the j^{th} largest density parameter ($\rho(x_i, DyAveDist)$) from D;
  Set x_j as the j^{th} initial clustering center v_j;
  $V \leftarrow v_j$; //Put v_j into the initial clustering center set V.
(4) Repeat
(5) Let $C_i=\emptyset$ ($1\le i\le K$); // C_i is i^{th} cluster of the dataset D.
(6) for j=1, 2,..., n do //Put the data points into the corresponding clusters.
  Calculate the distance between data point x_j to all the specified initial clustering centers in V;
  According to the nearest distance principle, put data point x_j into the corresponding cluster;
(7) for j=1, 2, ..., K do //Update the initial clustering centers
  Calculate the distance between v_j and the other data points in cluster C_j;
  Find the nearest neighbor (v_j') of v_j, meanwhile, this neighbor is as far as possible away from the outliers of cluster C_j;
  if $v_j \neq v_j$' $v_j \leftarrow v_j$' // v_j' is updated as the new clustering center of cluster C_j;
(8) Until the criterion function ( ) converges to a constant. //vi is the center of Ci.

Fig. 2. The main steps of the DP-Kmeans

4 SII: A New Clustering Validity Index

It is well known that different clustering algorithms or even the same clustering algorithm with different configurations may produce different clustering partitions. In order to evaluate the results generated by the clustering algorithms, this section presents the SII, a new defined clustering validity index. The SII is based on the sum of the new defined inner-cluster compactness and the inter-cluster separateness.

4.1 Definition of the SII

The definitions in this part are based on the following assumptions: "*In the Euclidean space R^m, a m-dimensional dataset $D=\{x_1,x_2,\ldots,x_n\}$ contains n sample points. In this dataset, each data points $x_i=\{x_{i1},x_{i2},\ldots,x_{im}\}$ has m feature attributes. For the DP-Kmeans clustering algorithm, dataset D is divided into K clusters $C=\{C_1,C_2,\ldots,C_K\}$. The corresponding clustering centers are $V=\{v_1,v_2,\ldots,v_K\}$. For each cluster C_k ($k=1,2,\ldots,K$) in C, $|C_k|$ is the number of data points in this cluster. The Euclidean distance between data point x_i and x_j can be calculated by Equation (1)*".

Definition 2 (Inner-cluster distance of the cluster C_k, T_k). For a given cluster C_k in C (k = 1, 2, …, K), the weighted mean squared Euclidean distance (abbreviated by Inner-cluster distance) of point pairs in this cluster is defined as:

$$T_k = \frac{2K}{|C_k|(|C_k|-1)}\sum_{i,j=1}^{|C_k|}\frac{|C_k|}{n}(d(x_i,x_j))^2 = \frac{2K}{n(|C_k|-1)}\sum_{i,j=1}^{|C_k|}(d(x_i,x_j))^2 . \quad \textbf{(11)}$$

In Equation (11), $|C_k|$ is the number of the data points in the cluster C_k; $|C_k|$/n is the weight of the cluster C_k in the dataset D. The *Inner-cluster distance* is the main component for the construction of the measure of the inner-cluster compactness. This component will endow the SII with the ability of processing the arc and the overlapping datasets.

Definition 3 (Inner-cluster compactness, *T*). The inner-cluster compactness of the dataset *D* is defined as:

$$T = \sum_k^K T_k = \frac{2K}{n} \sum_k^K \frac{1}{(|C_k| - 1)} \sum_{i,j=1}^{|C_k|} (d(x_i, x_j))^2 \ . \tag{12}$$

Many existing CVIs, such as the DBI [28], the COP [10] and the CH [8], define the inner-cluster compactness with the average values of the compactness of all clusters. However, as can be seen in the Equation (12), the inner-cluster compactness of the SII is defined by the sum of the compactness of all clusters of the dataset *D*. So, the influence of each single cluster on the SII is enlarged. By this method, the overlapping datasets can be properly processed by the SII

Definition 4 (Global center, V_0). The global center of the dataset *D* is defined as:

$$V_0 = (\frac{1}{n} \sum_{i=1}^n x_{i1}, \frac{1}{n} \sum_{i=1}^n x_{i1}, ..., \frac{1}{n} \sum_{i=1}^n x_{im}) \ . \tag{13}$$

Definition 5 (Inter-cluster distance, S_0). The weighted mean squared Euclidean distance between all data points v_k (k = 1, 2, …, K) in the center set V and the global center V_0 (abbreviated by inter-cluster distance) is defined as:

$$S_0 = (\frac{1}{K} \sum_{k=1}^K \frac{|C_k|}{n} (d(v_k, V_0))^2 = \frac{1}{nK} \sum_{k=1}^K |C_k| (d(v_k, V_0))^2 \ , \tag{14}$$

where, $|C_k|$ is the number of the data points in the cluster C_k; $|C_k|$/n is the weight of the cluster C_k in the entire dataset *D*. The inter-cluster distance is taken to measure the influence of the corresponding cluster on the separateness among all the clusters. Based on this component, the SII can process the datasets with a large number of differences among clusters (also called the unbalanced datasets).

Definition 6 (Inter-cluster separateness, *S*). The separateness among clusters in the dataset *D* can be calculated as:

$$S = 2 \times K \times S_0 = \frac{2}{n} \sum_{k=1}^K |C_k| (d(v_k, V_0))^2 \ . \tag{15}$$

As can be seen in the Equation (12), the inner-cluster compactness *T* of the dataset *D* is defined by the sum of the compactness of all clusters other than the average value of them. So, the value of *T* is the big number when it is compared with the value of S_0. In order to balance the impacts of *T* and *S* on the formation of the SII index, S_0 is multiplied by the constant of 2*K*. The two components, *T* and *S*, play roughly the same roles in the formation of the SII index according to this method.

Definition 7 (*SII*). The SII index is defined by the sum of the inner-cluster compactness *T* and the inter-cluster separateness *S*:

$$SII(K) = T + S = \frac{2K}{n} \sum_k^K \frac{1}{(|C_k| - 1)} \sum_{i,j=1}^{|C_k|} (d(x_i, x_j))^2 + \frac{2}{n} \sum_{k=1}^K |C_k| (d(v_k, V_0))^2 \ . \tag{16}$$

As the Equation (17), the optimal clustering number K_{opt} of the dataset *D* is obtained by finding the minimum value of SII(K), K=1, 2, …, $\sqrt{n}$.

$$K_{opt} = \{K \mid min_{2 \le K \le \sqrt{n}} SII(K)\} \ . \tag{17}$$

As can be seem in Equation (16), the calculation of the time complexity (marked as $T(n)$) of the $SII(K)$ is composed of two parts, the time complexity on computing the inner-cluster compactness (marked as $T_1(n)$) and the

computation of the inter-cluster separateness (marked as $T_2(n)$). From Equation (11) and Equation (12), the time complexity of $T_1(n)$ can be:

$$T_1(n)= K\times m\times(C_{max})^2, \tag{18}$$

where, K is the number of the clusters the target dataset D to be divided; m is the number of attributes of the dataset D; C_{max} is the number of data points in the biggest cluster of the dataset D. Since different clusters may have different number of data points, the biggest cluster is used to estimate the time complexity of the inner-cluster compactness, that is $C_{max}\leftarrow max\{|C_1|,|C_2|,\ldots, |C_K|\}$.

From Equation (13) and (15), the time complexity of $T_2(n)$ can be:

$$T_2(n)= m\times n + K\times m, \tag{19}$$

where, $m\times n$ is the time complexity on the calculation of the global center V_0.

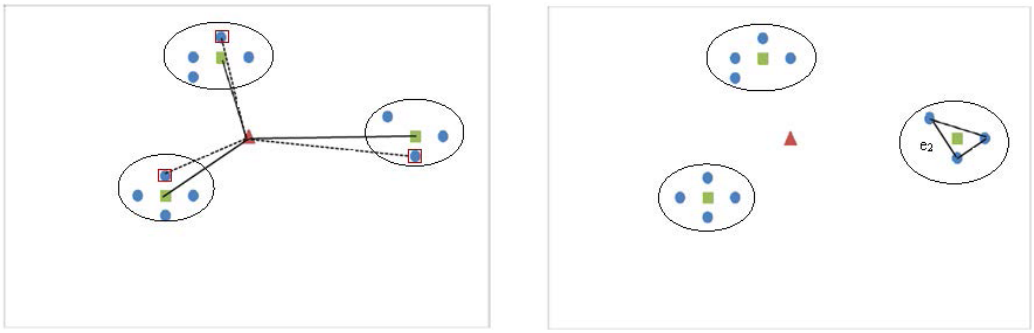
Based on Equation (18) and (19), the time complexity of the SII(K) can be:

$$T(n)= T_1(n)+ T_2(n)= K\times m\times(C_{max})^2 + m\times n + K\times m = m\times(K\times(C_{max})^2 + n + K). \tag{20}$$

In the general cases, the values of K and m are far less than the value of n, they can be taken as constants. So, the time complexity of the $T(n)$ can be roughly expressed as $\max\{O(n), O((C_{max})^2)\}$.

4.2 Rationality Analysis of SII

S (inter-cluster separateness) in Equation (15) evaluates the degree of dissimilarity among different clusters of the target dataset D. As an example, Fig. 3 shows a dataset that contains three clusters, which is randomly generated by the MATLAB. The data points (blue dots) are divided into three clusters. The green squares are the corresponding clustering centers (they are not the real data points of the target dataset) specified by the traditional K-means. The blue dots in the red boxes (they are the real data points of the target dataset) are the new clustering centers updated by the DP-Kmeans. The global center of this dataset (red triangle) is specified by the Equation (13). It can be seen that the global center and the three clustering centers are connected by the lines: the original clustering centers (green squares) are connected by solid lines and the replaced centers (blue dots in the red boxes) are connected by the dotted ones. In each dotted line, the distance between the corresponding clustering center and the global center is assigned. the inter-cluster distance $S_0 = (|C_1|e_1^2 + |C_2|e_2^2 + |C_3|e_3^2)/n$, according to the Equation (14). Consequently, the time complexity on the calculation of the inter-cluster separateness can be reduced to $K\times m$.



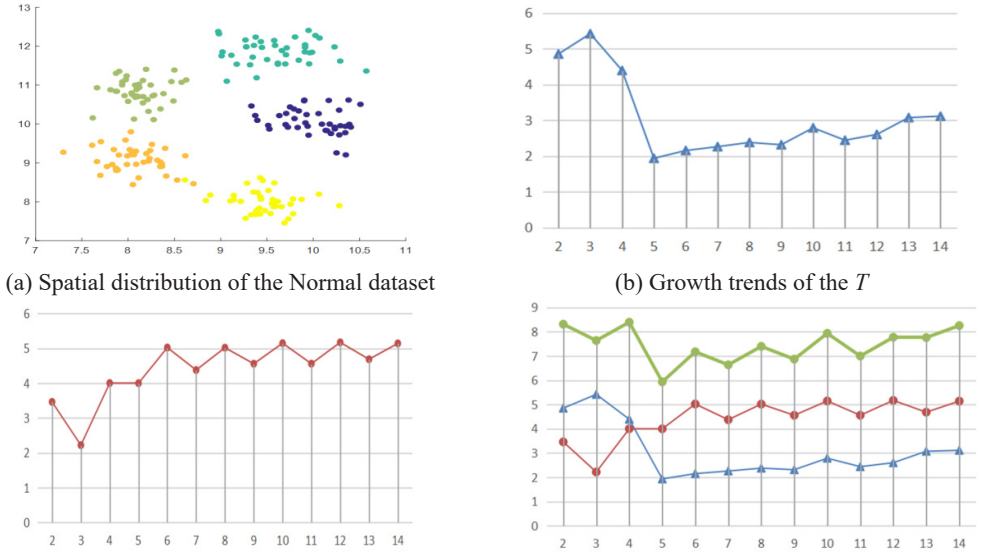
(a) Inter-cluster separateness (b) Inner-cluster compactness

Fig. 3. An example of computing the inter-cluster separability and the inner-cluster compactness of the SCVI index

The inner-cluster compactness T evaluates the degree of similarity of data point in each single cluster. By the Equation (11), the inner-cluster distance of each cluster is calculated. In Fig. 3(b), the inner-cluster distance of the cluster C_2 can be computed by $T_2 = 2K(e_1^2 + e_2^2 + e_3^2)/n(|C_2|-1)$ according to Equation (11). This method avoids the direct connections of different clustering centers. Therefore, the designed SII index is able to process the arc and overlapping datasets.

Fig. 4(a) is the two-dimensional spatial distribution of the Normal simulated dataset been processed by the DP-Kmeans. The Normal dataset is composed of 5 clusters. Fig. 4(b), Fig. 4(c) and Fig. 4(d) show the growth trends of inner-cluster compactness (T), the inter-cluster separateness (S) and the SII(K) respectively. In the 3 sub-graphs, the abscissas and the ordinates represent the clustering number K and the corresponding index values respectively.

It can be seen from Fig. 4(b) that, except K=2, the value of T is generally decreasing when K reaches 5. After that, the value of T is slightly increasing with the growth of K. It is worth noting that the values of T decreases sharply from K=4 to K=5. So, K=5 is the inflection point of T. So, Equation (12) can compute the inner-cluster compactness for each cluster. As can be seen in Fig. 4(c) that the value of S fluctuates greatly when K<5. However, the fluctuation of S flattens out gradually when K>5. This means the values of S get the stable state when K≥5. In other words, the values of S changes slightly with the growth of K. Therefore, Equation (15) can compute inter-cluster separateness among different clusters. The green polygonal line in Fig. 4(d) shows the growth trend of the values of the SII(K). In this sub-graph, the blue and red polygonal lines are the growth trends of T and S shown in Fig. 4(b) and Fig. 4(c) respectively. As can be seen in Fig. 4(d), when K reaches the value of five, T gets the minimum value; the values of S is into a stable state; the minimum value of the SII index is acquired at the inflection point. So, the corresponding K is the optimal clustering number of the Normal dataset.



(a) Spatial distribution of the Normal dataset

(b) Growth trends of the T

(c) Growth trends of the S

(d) Growth trends of the SII(K)

Fig. 4. The spatial distribution of the Normal dataset and the growth trends of the index values

Input: Dataset $D=\{x_1, x_2, \ldots, x_n\}$
Output: The optimal clustering number (K_{opt}) and corresponding values of SII(K).

(1) Determine the search range [K_{min}, K_{max}] of clustering number;
(2) For K =K_{min} to K_{max} do //Conventionally, the values of K is in the interval of [2,$\sqrt{n}$ $\sqrt{n}$].
(3) Use the DP-Kmeans algorithm to cluster dataset D;
(4) Use the Equation (12) to calculate the inner-cluster compactness for each cluster;
(5) Use the Equation (15) to calculate the inter-cluster separateness among different clusters;
(6) let *min*←SII(2); // Use the Equation (16) to calculate the value of SII(K);
for K=3,4,…, $\sqrt{n}$ $\sqrt{n}$ do // Use the Equation (17) to get the optimal clustering number K_{opt};
if *min*>SII(K)
then *min*←SII(K);
K_{opt}←K;
else Keep *min* unchanged.

Fig. 5. The new method for determining the K_{opt}

5 A New Method for Determining the K_{opt}

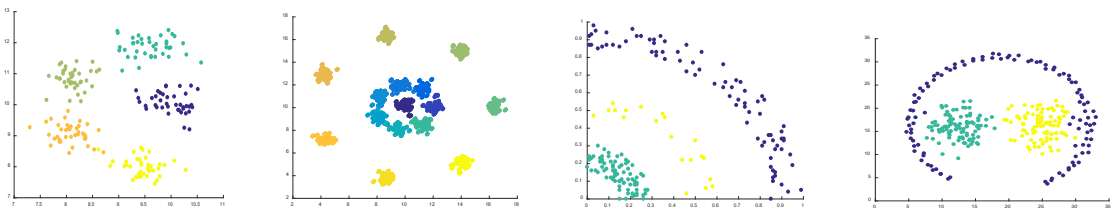
A new method for determining the K_{opt} can be obtained based on the DP-Kmeans and the SII. This method is able to optimally process different types of datasets, such as the arc datasets, the convex datasets, the non-convex datasets, the overlapping datasets and the unbalance datasets. Fig. 5 gives the main steps of this method. Step (1) - Step (3) divide the dataset D into K different clusters. According to the empirical rule, the value of K is in the interval of $[2, \sqrt{n}\ ]$. Step (4) computes the inner-cluster compactness for each generated cluster. Step (5) computes the inter-cluster separateness among all the generated clusters. Step (6) uses the Equation (16) and Equation (17) to get the K_{opt} for the dataset D.

6 Experimental Results

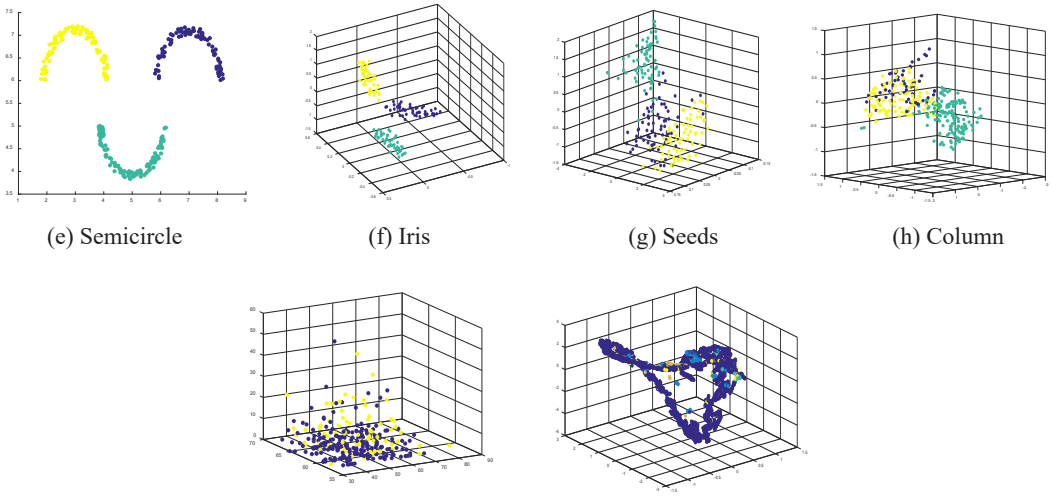
This section presents the experimental results on the performances of the DP-Kmeans and the SII. As listed in Table 1, the tested datasets in this section are composed of the five simulated datasets (http://cs.joensuu.fi/sipu/datasets/) and the five UCI real machine learning datasets (https://archive.ics.uci.edu/ ml/datasets.php). In the experiments, the empirical rule $K \leq \sqrt{n}$ is firstly used to get the range of K for different tested datasets. Then, for different datasets, the clustering accuracy of the DP-Kmeans algorithm is tested. Meanwhile, the accuracy of the DP-Kmeans is compared with the ones of the K-medoids, the K-means++, CCIA [37] and DC-K-means [17]. Lastly, the SII is used to evaluate the clustering results generated by the DP-Kmeans. The results of the SII are also compared with the ones of the eight existing CVIs (CH^+ [8], I^+ [27], STR^+ [38], DBI^- [28], COP^- [10], SMV^- [11], $BCVI^-$ [21] and $DCVI^-$ [12]). The CH, I and STR get the optimal clustering numbers at the biggest index values and thus they are marked as CH^+, I^+ and STR^+ respectively. On the contrary, the DBI, COP, SMV, BCVI and DCVI indexes get the optimal clustering numbers at the smallest index values and thus they are marked as DBI^-, COP^-, SMV^-, $BCVI^-$ and $DCVI^-$ respectively. The SII gets the optimal clustering number at the smallest index value according to Equation (17) and thus is marked as SII^-.

Table 1. Characteristics of the ten tested datasets

Datasets	Points	Clusters	Dimensions	Range of K
Simulated datasets				
Normal	200	5	2	2<=K<=14
R15	600	15	2	2<=K<=24
Curve	180	3	2	2<=K<=13
Pathbased	300	3	2	2<=K<=17
Semicircle	300	3	2	2<=K<=17
UCI real machine learning datasets				
Iris	150	3	4	2<=*K*<=12
Seeds	210	3	7	2<=*K*<=14
Column	310	3	6	2<=*K*<=17
Haberman	306	2	3	2<=*K*<=17
PageBlocks	5473	5	10	2<=*K*<=73



(a) Normal (b) R15 (c) Curve (d) Pathbased



(e) Semicircle (f) Iris (g) Seeds (h) Column

(i) Haberman (j) PageBlocks

Fig. 6. Spatial distributions of the ten tested datasets

6.1 Performance Evaluation of the DP-Kmeans

The spatial distributions of the ten tested datasets after the clustering partition by the DP-Kmeans are given in Fig. 6. As shown in Fig. 6(a), the data points in the Normal dataset are divided into five clusters. This dataset is generally spherical distributed. However, there are also some outliers in these clusters. As shown in Fig. 6(b), the data points of the R15 datasets are divided into 15 clusters. The Curve, Pathbased and Semicircle datasets shown in Fig. 6(c) to Fig. 6(e) are the non-spherical distributed datasets. In Fig. 6(c), the three clusters of the Curve dataset are formed into three concentric arcs. Fig. 6(d) shows the spatial distribution of the Pathbased dataset. In this dataset, the inner two spherical clusters are surrounded by a semi-circle cluster. As shown in Fig. 6(e), the three clusters of the Semicircle dataset are formed into three semi-circles.

Fig. 6(f) to Fig. 6(j) shows the spatial distributions of the five UCI real machine learning datasets which are processed by the DP-Kmeans. As listed in Table 1, most of these datasets are high dimensional. It is needed to reduce the dimensions before displaying them in the low dimensional space [39]. In this paper, the widely used non-linear dimensionality reduction tool T-SNE [40] is used to preprocess all the high dimensional datasets.

As shown in Fig. 6(f), the 150 points in the Iris are divided into three clusters. The data points in the three clusters are distributed as “inner-cluster compactness and inter-cluster separateness”. As shown in Fig. 6(g) and Fig. 6(h), both the Seeds and Column datasets are divided into three clusters. Meanwhile, there are many overlapping points among these clusters. The Haberman dataset is composed of two clusters. As shown in Fig. 6(i), the two clusters of it are almost overlapped completely. As shown in Fig. 6(j), the PageBlocks dataset is more complex than the other datasets. The dataset is also containing much more data points than the others. The 5473 data points of the PageBlocks datasets is divided into five clusters. Meanwhile, the PageBlocks dataset is also having much overlapping data points among different clusters.

Table 2. Spatial distribution characteristics of the ten datasets

Datasets	Points	Compositions	Overlap	Arc	Convex	Balance	Outlier
Normal	200	5*40	×	×	√	√	√
R15	600	15*40	×	×	√	√	√
Curve	180	20+80+80	×	√	×	×	×
Pathbased	300	93+97+110	×	√	×	×	×
Semicircle	300	90+99+111	×	√	×	×	×
Iris	150	3*50	√	×	√	√	×
Seeds	210	3*70	√	×	√	√	√
Column	310	60+100+150	√	×	×	×	√
Haberman	306	82+225	√	×	×	×	√
PageBlocks	5473	4913+329+28+88+115	√	×	×	×	√

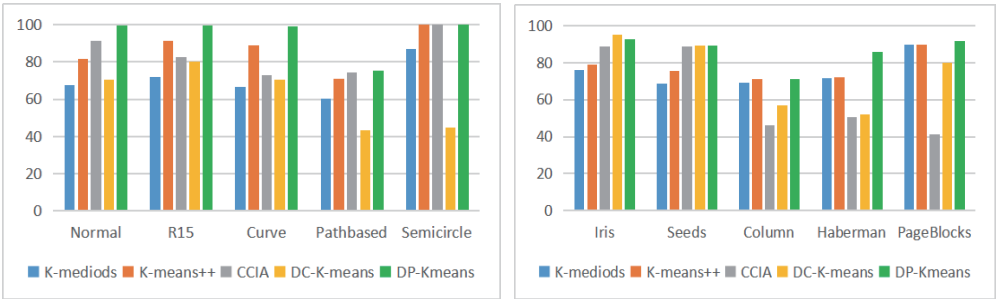
Table 2 collects all the spatial distribution characteristics of the ten datasets that are discussed in this section. Table 2 is divided into eight columns by the dataset names, data point's numbers, compositions, and structures of datasets. For a given dataset, the symbol of "√" specifies it has the corresponding characteristic with, the "×" symbol represents null. For example, in the fourth row of this table, the three clusters of the Curve dataset have 20, 80 and 80 sample points respectively. It is the arc dataset. The spatial distributions in Fig. 6 have shown that the DP-Kmeans can optimally process many kinds of datasets.

Table 3. The accuracy comparisons among different algorithms (%)

Datasets	Algorithms				
	K-medoids	K-means++	CCIA	DC-K-means	DP-Kmeans
Normal	67.50	81.55	91.50	70.5	99.50
R15	72.20	91.22	82.83	80.17	99.67
Curve	66.48	88.89	72.78	70.56	99.22
Pathbased	60.40	71.17	74.25	43.48	75.25
Semicircle	87.03	100	100.00	44.67	100.00
Iris	76.07	78.87	88.67	95.34	92.67
Seeds	68.62	75.53	89.05	89.52	89.52
Column	69.39	71.23	46.13	56.77	71.29
Haberman	71.90	72.29	50.65	52.29	85.95
PageBlocks	90.01	89.97	41.26	79.95	91.78

The sixth column of the Table 3 lists the accuracy of the DP-Kmeans for the ten datasets listed in Table 1. For the better of comparisons, the experimental results of another four clustering algorithms, the K-medoids, the K-means++, the CCIA and the DC-K-means are also included in Table 3. In this table, the experimental results are shown with the average values of ten repeatedly experiments. In order to display the accuracy of each algorithm more intuitively, Fig. 7 shows the accuracy histograms of the five algorithms on the five synthetic datasets (a) and the five real datasets (b).

As can be seen in the Table 3 and Fig. 7, due to the randomly selection of the initial clustering centers, the accuracy of the K-medoids is worse than the K-means++. In the K-means++, except the first initial clustering center, the other clustering centers are no longer randomly selected. Therefore, the accuracy of the K-means++ is higher than the K-medoids. The CCIA algorithm initializes the centers by the iterative clustering. In this algorithm, the density based multi-scale data aggregation method is used to merge similar clusters. As results listed in the fourth column of Table 3, the accuracy of this algorithm is better than these of the K-medoids and K-means++. However, CCIA algorithm is not a good choice for large-scale datasets and datasets with many overlapping points among different clusters. The DC-K-means finds the initial clustering centers by the product of the sample density, the cluster distance, and the maximum weighted value. Due to cannot properly deal with the non-convex datasets and the overlapping datasets, the performance of this algorithm is the worst on average among the five algorithms. In the DP-Kmeans, the density parameter and the center replacement methods are introduced to select the initial clustering centers. Due to the two improvements, the accuracy of the DP-Kmeans is the best among the ones of the five clustering algorithms.



(a) Accuracy on five synthetic datasets (b) Accuracy on five real datasets

Fig. 7. The accuracy of different clustering algorithms on the ten datasets

6.2 Performance Evaluation of the SII Index

In this section, the performance of the SII^- is firstly evaluated with ten tested datasets. Then, the performance of the SII^- is compared with the eight existing CVIs (CH^+, I^+, STR^+, DBI^-, COP^-, SMV^-, $BCVI^-$ and $DCVI^-$). For fairness, all the nine CVIs are used to evaluate the clustering results that are generated by the DP-Kmeans.

Table 4. The index of the SII^- on the ten tested datasets

K	Datasets									
	Normal	R15	Curve	Pathbased	Semicircle	Iris	Seeds	Column	Haberman	PageBlocks
2	7.706	71.788	0.551	296.377	14.034	11.158	35.725	8964	**559.488**	79145808
3	7.600	75.927	**0.402**	**277.532**	**9.734**	**8.573**	**30.565**	**8649**	609.887	67842680
4	8.409	88.303	0.536	354.893	12.720	11.458	39.778	11307	665.974	85415552
5	**5.839**	78.697	0.457	340.109	10.878	9.979	37.656	9424	673.770	**67395928**
6	7.062	84.770	0.513	371.280	12.843	12.390	45.933	11156	761.896	78670032
7	6.492	75.834	0.454	346.348	12.130	11.806	41.575	11477	754.686	72720576
8	7.339	76.091	0.473	364.556	13.369	12.692	43.832	11847	834.138	80033016
9	6.750	67.738	0.502	353.227	10.676	11.369	40.969	12042	795.408	76201808
10	7.714	72.735	0.549	379.988	11.814	14.370	50.636	13163	896.901	77091800
11	7.010	51.248	0.536	361.503	10.240	13.150	43.134	14756	819.847	75975976
12	7.647	53.348	0.543	397.172	10.918	14.348	50.684	16152	865.571	82999048
13	7.708	48.373	0.569	393.946	10.082	--	44.230	14514	963.312	80579744
14	8.198	49.638	--	360.570	10.820	--	47.075	15408	998.450	86503864
15	--	**44.971**	--	351.862	10.250	--	--	15836	913.581	84464936
16	--	48.027	--	351.194	10.630	--	--	16497	1008.586	92610352
17	--	45.747	--	337.427	10.234	--	--	17430	979.979	84908464
18	--	48.520	--	--	--	--	--	--	--	92475552
19	--	46.372	--	--	--	--	--	--	--	92072912
20	--	48.953	--	--	--	--	--	--	--	96898816
21	--	46.967	--	--	--	--	--	--	--	95548784
22	--	49.117	--	--	--	--	--	--	--	99750712
23	--	47.220	--	--	--	--	--	--	--	86759768
24	--	49.211	--	--	--	--	--	--	--	101636712
25	--	--	--	--	--	--	--	--	--	100918320
26	--	--	--	--	--	--	--	--	--	80638400
27	--	--	--	--	--	--	--	--	--	78092016
28	--	--	--	--	--	--	--	--	--	82433072
29	--	--	--	--	--	--	--	--	--	81010624
30	--	--	--	--	--	--	--	--	--	83138976
.....	--	--	--	--	--	--	--	--	--	.....
73	--	--	--	--	--	--	--	--	--	90426528
.....	--	--	--	--	--	--	--	--	--	--
137	--	--	--	--	--	--	--	--	--	--

Table 4 collects the values of the SII^- on the ten tested datasets. In this table, the ranges of *K* of different datasets are limited by the empirical rule $2 \le K \le \sqrt{n}$. For example, since there are 200 data points in the Normal dataset, as listed in Table 4, we only have to calculate the values of SII(*K*) when the values of *K* fall into the interval of [2, 14]. The values of *K* corresponding to the bold underline numbers are the optimal clustering numbers acquired by SII^- index for different datasets. Since there are 5473 data points in the PageBlocks datasets, the ranges of *K* are limited into the intervals of [2, 73]. For sparing the spaces, as listed in Table 4, only part of CVI values of the PageBlocks dataset is displayed. But in the remainder results of these datasets, all the optimal clustering numbers of SII^- index are obtained.

The third column of the Table 1 gives the real numbers of clusters of the ten tested datasets. Compared the experimental results collected in Table 5 with the third column of the Table 1, we can find that the SII^- can obtain the optimal clustering numbers for all the ten tested datasets.

Table 5. Clustering results computed by different CVIs for the ten datasets

Dataset	CH^+	I^+	STR^+	DBI^-	COP^-	SMV^-	$BCVI^-$	$DCVI^-$	SII^-
Normal	√(5,630.84)	×(2,0.410)	√(5,6.110)	√(5,0.436)	√(5,0.210)	√(5,0.401)	√(5,0.844)	√(5,0.960)	√(5,5.539)
R15	√(15,4871.94)	×(2,0.933)	√(15,24.530)	√(15,0.315)	√(15,0.156)	√(15,0.253)	×(5,7.003)	×(5,8.189)	√(15,44.971)
Curve	×(8,476.35)	○(4,0.084)	×(8,4.152)	√(3,0.538)	×(8,0.291)	√(3,0.531)	○(4,0.049)	√(3,0.059)	√(3,0.402)
Pathbased	×(17,407.76)	○(2,3.912)	×(17,0.776)	√(3,0.686)	○(4,0.316)	×(14,0.594)	×(5,31.789)	√(3,37.546)	√(3,277.532)
Semicircle	×(16,2312.15)	○(2,0.734)	×(13,2.727)	×(13,0.507)	×(13,0.244)	×(11,0.516)	○(4,0.854)	○(4,1.087)	√(3,9.734)
Iris	√(3,560.30)	√(3,0.806)	○(2,2.272)	○(2,0.405)	○(2,0.205)	○(2,2.484)	√(3,1.088)	√(3,3.123)	√(3,8.573)
Seeds	√(3,375.81)	○(2,1.722)	×(13,1.033)	○(2,0.691)	√(3,0.311)	√(3,0.619)	√(3,3.305)	√(3,4.186)	√(3,30.565)
Column	×(5,224.38)	○(2,203.77)	√(3,4.351)	○(2,0.099)	○(2,0.088)	○(2,0.131)	√(3,1170.50)	√(3,1282.97)	√(3,8649.06)
Haberman	×(4,256.30)	√(2,5.196)	×(4,0.359)	×(4,0.847)	√(2,0.255)	×(13,0.702)	×(4,62.003)	×(4,73.877)	√(2,599.488)
PageBlocks	×(47,17915.6)	×(3,2650.91)	×(42,15.278)	×(2,0.365)	×(2,0.027)	×(68,0.361)	×(3,12394817)	×(3,16343546)	√(5,67395928)

Table 5 collects the experimental results of the nine CVIs on the ten datasets. In this table, the column of "K_{opt}" specifies the actual number of clusters of the corresponding datasets. The symbols of "√" specify the CVIs listed in the first row of this table can get the optimal clustering numbers for the corresponding datasets listed in the first column of this table; the symbols of "○" specify the CVIs can only get the near optimal clustering numbers for the corresponding datasets; the symbols of "×" specify the CVIs cannot get the correct optimal clustering numbers for the corresponding datasets. The number pair in the round bracket means the "optimal clustering number" and the "index value" obtained by the corresponding CVI. For example, the number pair (5, 630.84) at the row #2 and column #3 means the CH^+ index gets the biggest value (630.84) when the value of K is 5 on the Normal dataset.

Table 5 shows that the performance of the SII^- is the best because it can get the optimal clustering numbers for all the ten tested datasets. The other eight CVIs cannot get the optimal clustering numbers for all the tested datasets. Specifically, the CH^+ can get the optimal clustering numbers for the Normal, R15, Iris and Seeds datasets. The I^+ can get the optimal clustering numbers for the Iris and Harberman datasets and the near optimal clustering numbers for the Curve, Pathbased, Semicircle, Seeds and Column datasets. The STR^+ can get the optimal clustering numbers for the Normal, R15 and Column datasets and the near optimal clustering numbers for the Iris datasets. The DBI^- is able to get the optimal clustering numbers for the Normal, R15, Curve and Pathbased datasets and the near optimal clustering numbers for the Iris, Seeds and Column datasets. The COP^- gets the optimal clustering numbers for the Normal, R15, Seeds and Haberman datasets and the near optimal clustering numbers for the Pathbased, Iris and Column datasets. The SMV^- obtains the optimal clustering numbers for the Normal, R15, Curve and Seeds datasets and the near optimal clustering numbers for the Iris and Column datasets. The $BCVI^-$ gets the optimal clustering numbers for the Normal, Iris, seeds and Column datasets and the near optimal clustering numbers for the Curve and Semicircle datasets. The $DCVI^-$ gets the optimal clustering numbers for the Normal, Curve, Pathbased, Iris, seeds and Column datasets and the near optimal clustering numbers for the Semicircle datasets.

Table 6. Time costs of the nine CVIs on the ten datasets (*ms*)

Dataset	CH^+	I^+	STR^+	DBI^-	COP^-	SMV^-	$BCVI^-$	$DCVI^-$	SII^-
Normal	0.306178	0.371982	0.419426	0.1156	12.51277	0.336827	0.1222	0.085333	0.312772
R15	0.237672	0.188996	0.904568	3.3243	19.81088	0.317596	0.2028	0.152038	0.481845
Curve	0.118986	0.087436	0.290434	0.1409	5.93458	0.374086	0.3374	0.142122	0.711835
Pathbased	0.148132	0.122892	0.482703	3.1128	8.38431	0.447399	0.2737	0.15985	0.529547
Semicircle	0.221363	0.150836	0.533002	0.2223	9.172443	0.279136	0.1129	0.077221	0.69813
Iris	0.128000	0.115080	0.360204	0.1556	4.314747	0.231061	0.0970	0.063399	0.526463
Seeds	0.325408	0.231361	0.894019	0.3933	8.074826	0.436883	0.3308	0.120788	0.927331
Column	0.377090	0.311287	1.518696	0.6132	15.076676	0.359662	0.2310	0.056488	4.324143
Haberman	0.563081	0.182686	0.610886	0.1886	7.952835	0.282442	0.1340	0.097652	2.716366
PageBlocks	4.914185	0.408939	23.363724	1.7718	221.54844	1.792903	4.3634	0.252695	199.2679

Table 6 lists the time cost of the nine CVIs on evaluating the clustering results of the ten datasets. The clustering results of these datasets are all generated by the DP-Kmeans. Due to the $O(n^2)$ time complexity, the COP^- index consumes the largest time cost among the nine CVIs. The other eight CVIs are all the linear time complexity. Due to similar data distributions of the Normal, R15, Pathbased, Semicircle, Iris and Seeds datasets, the time cost of SII^- is roughly equal to the ones of the I^+, STR^+, DBI^-, COP^-, SMV^-, $BCVI^-$ and $DCVI^-$. However, on evaluat-

ing the results of the uneven-distributed datasets (i.e., the Curve, Column, Haberman and Pageblocks), the time cost of SII$^-$ is higher than the ones of the seven linear time complexity CVIs. As discussed in the last part of the Section 4.1, on processing the uneven-distributed datasets, the time cost of the SII$^-$ is mainly determined by the largest cluster of the target datasets. Overall, the SII$^-$ can optimally evaluate the clustering results without consuming much time cost.

7 Conclusion and Future Works

In this paper, the DP-Kmeans is firstly proposed to resolve the drawbacks of the traditional K-means algorithm: In the initial stage, the DP-Kmeans uses the density parameter to avoid the problem of randomly selection initial clustering centers; and, in the iteration stage, the DP-Kmeans uses the center replacement method to update clustering centers when they are distorted by the outliers. The data points used to replace the distorted centers are the ones with the smallest distances from the distorted centers and the largest distances from the outliers. After the target datasets being partitioned by the DP-Kmeans, the new index, SII, is defined to evaluate the quality of the clustering results. The SII is designed by the sum of the new defined inner-cluster compactness and the inter-cluster separability. Finally, a new algorithm, OCNS, based on the DP-Kmeans and the SII is designed to determine the optimal clustering number for different datasets. Experimental results on testing many types of datasets have demonstrated that the proposed DP-Kmeans is effective and widely applicable. In the iterative process of the DP-Kmeans, the adoption of the center point replacement method effectively abandons the influence of outliers. For this reason, the DP-Kmeans has obvious advantages over the four existing algorithms (K-medoids, the K-means++, CCIA and DC-K-means) in dealing with datasets with many data points. In future work, the DP-Kmeans will be further studied and used to deal with many types of large-scale datasets. The experimental results have also demonstrated that the proposed SII index has higher performance in evaluating the clustering results than the other existing CVIs. However, the SII index has relatively higher time cost than the other linear time complexity CVIs when the uneven-distributed datasets are processed. As discussed in the Section 4.1, the time complexity of the SII index can be expressed as $max\{O(n), O((C_{max})^2)\}$. Where, n and C_{max} are the numbers of data points of the target dataset and the largest cluster respectively. Therefore, the time cost of the SII is mainly determined by the largest cluster of the target datasets. In the future, further study will be expected to overcome this shortcoming.

Acknowledgement

This work was supported by the University Natural Science Research Projects of Anhui Province, China (Grant No. KJ2020A1175, KJ2021A0041) and the Natural Science Foundation of Anhui Province, China (Grant No. 2008085MF188).

References

[1] A. Jain, Data clustering: 50 years beyond K-means, Pattern Recognition Letters 31(2010) 651-666.
[2] R. Xu, D. Wunsch II, Survey of clustering algorithm, IEEE Transactions on Neural Networks 16(3)(2005) 645-678.
[3] L. Bai, J.-Y. Liang, C. Sui, C.-Y. Dang, Fast global K-means clustering based on local geometrical information, Information Sciences 245(2013) 168-180.
[4] S. Redmond, C. Heneghan, A method for initializing the K-means clustering algorithm using kd-trees, Pattern Recognition Letters 28(8)(2007) 965-973.
[5] A. Vattani, K-means requires exponentially many iterations even in the plane, in: Proc. of the 25th annual symposium on Computational geometry, 2009.
[6] S.-B. Zhou, Z.-Y. Xu, A novel internal validity index based on the cluster centre and the nearest neighbour cluster, Applied Soft Computing 71(2018) 78-88.
[7] E.-Z. Zhu, R.-H. Ma, An effective partitional clustering algorithm based on new clustering validity index, Applied Soft Computing 71(2018) 608-621.
[8] T. Calinski, J. Harabasz, A dendrite method for cluster analysis, Communications in Statistics 3(1)(1974) 1-27.
[9] P. Rousseeuw, Silhouettes: A graphical aid to the interpretation and validation of cluster analysis, Journal of Computational and Applied Mathematics 22(1987) 53-65.
[10]I. Gurrutxaga, I. Albisua, O. Arbelaitz, J.I. Martín, J. Muguerza, J.M. Pérez, I. Perona, SEP/COP: An efficient method to find the best partition in hierarchical clustering based on a new cluster validity index, Pattern Recognition 43(2010) 3364-3373.
[11]S.-H. Yue, J.-P. Wang, J. Wang, X.-J. Bao, A new validity index for evaluating the clustering results by partitional cluster-

ing algorithm, Soft Computing 20(3)(2016) 1127-1138.
[12]E.-Z. Zhu, B.-B. Zhu, P. Wen, F. Liu, X.-J. Li, F.-T. Wang, Effective Clustering Analysis based on New Designed CVI and Improved Clustering Algorithms, in: Proc. of the 16th IEEE International Symposium on Parallel and Distributed Processing with Applications, 2018.
[13]X.-Y. Chen, Y.-L. Su, Y. Chen, G.-H. Liu, GK-means: an Efficient K-means Clustering Algorithm Based on Grid, in: Proc. of the 1st International Symposium on Computer Network and Multimedia Technology, 2009.
[14]J. Yoder, C. Priebe, Semi-supervised k-means ++, Journal of Statistical Computation and Simulation 87(13)(2017) 2597-2608.
[15]S. Hussain, M. Haris, A k-means based co-clustering (kCC) algorithm for sparse, high dimensional data, Expert Systems with Applications 118(2019) 20-34.
[16]J.-Z. Huang, M.K. Ng, H.-Q. Rong, Z.-C. Li, Automated variable weighting in k-means type clustering, IEEE transactions on pattern analysis and machine intelligence 27(5)(2005) 657-668.
[17]G. Zhang, C.-X. Zhang, H.-Y. Zhang, Improved K-means algorithm based on density canopy, Knowledge Based Systems 145(29)(2018) 289-297.
[18]Z.-L. Wang, J. Li, Y.-F. Song, Improved K-means algorithm based on distance and weight, Computer Engineering and Applications 56(3)(2020) 87-94.
[19]M.Z. Islam, V.E. Castro, M.A. Rahman, T. Bossomaier, Combining K-means and a genetic algorithm through a novel arrangement of genetic operators for high quality clustering, Expert Systems with Applications 91(2018) 402-417.
[20]A.H. Fadaei, S.H. Khasteh, Enhanced K-means re-clustering over dynamic networks, Expert Systems with Applications 132(2019) 126-140.
[21]E.-Z. Zhu, Y.-X. Zhang, P. Wen, F. Liu, Fast and Stable Clustering Analysis based on Grid-mapping K-means Algorithm and New Clustering Validity Index, Neurocomputing 363(2019) 149-170.
[22]J.J. Petra, K.J. Matthew, J.D. Melanie, K. Khunti, M. Catt, T. Yates, A.V. Rowlands, E.M. Mirkes, FilterK: A new outlier detection method for k-means clustering of physical activity, Journal of biomedical informatics 104(2020) 103397.
[23]Z.-X. Fan, Y. Sun, H. Luo, Clustering of college students based on improved K-means algorithm, Journal of Computers 28(6)(2017) 676-679.
[24]M. Erisoglu, N. Calis, S. Sakallioglu, A new algorithm for initial cluster centers in K-means algorithm, Pattern Recognition Letters 32(14)(2011) 1701-1705.
[25]J.C. Dunn, A fuzzy relative of the ISODATA process and its use in detecting compact well-separated clusters, Journal of Cybernetics 3(1974) 32-57.
[26]L. Hubert, J. Schultz, Quadratic assignment as a general data analysis strategy, British Journal of Mathematical and Statistical Psychology 29(2)(1976) 190-241.
[27]U. Maulik, S. Bandyopadhyay, Performance evaluation of some clustering algorithms and validity indices, IEEE Transactions on Pattern Analysis & Machine Intelligence 24(12)(2002) 1650-1654.
[28]D.L. Davies, D.W. Bouldin, A cluster separation measure, IEEE Transactions on Pattern Analysis and Machine Intelligence 1(2)(1979) 224-227.
[29]J.C. Bezdek, Numerical taxonomy with fuzzy sets, Journal of Mathematical Biology 7(1)(1974) 57-71.
[30]J.C. Bezdek, Cluster validity with fuzzy sets, Journal of Cybernetics 3(3)(1974) 58-74.
[31]K.R. Zalik, Cluster validity index for estimation of fuzzy clusters of different sizes and densities, Pattern Recognition 43(10)(2010) 3374-3390.
[32]D.-W. Kim, K. H. Lee, D. Lee, On cluster validity index for estimation of the optimal number of fuzzy clusters, Pattern Recognition 37(10)(2004) 2009-2025.
[33]M.-Y. Chen, D.A. Linkens, Rule-base self-generation and simplification for data-driven fuzzy models, Fuzzy Sets and Systems 142(1)(2004) 243-265.
[34]Y.-G. Tang, F.-C. Sun, Z.-Q. Sun, Improved validation index for fuzzy clustering, in: Proc. of the 2005 American Control Conference, 2005.
[35]M.K. Pakhira, S. Bandyopadhyay, U. Maulik, Validity index for crisp and fuzzy clusters, Pattern Recognition 37(3)(2004) 487-501.
[36]K.-L. Wu, M.-S. Yang, A cluster validity index for fuzzy clustering, Pattern Recognition Letters 26(9)(2005) 1275-1291.
[37]S.S. Khan, A. Ahmad, Cluster center initialization algorithm for K-means clustering, Pattern Recognition Letters 25(11)(2004) 1293-1302.
[38]A. Starczewski, A new validity index for crisp clusters, Pattern Analysis and Applications 20(2017) 687-700.
[39]R.-Y. Li, L.-F. Zhang, B. Du, A robust dimensionality reduction and matrix factorization framework for data clustering, Pattern Recognition Letters 128(2019) 440-446.
[40]L.V.D. Maaten, G. Hinton, Visualizing data using t-SNE, Journal of Machine Learning Research 9(2605)(2017) 2579-2605.