Egyptian Informatics Journal 23 (2022) 133–144

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

WeDIV – An improved *k*-means clustering algorithm with a weighted distance and a novel internal validation index

Zilan Ning [a,b], Jin Chen [a], Jianjun Huang [a], Umar Jlbrilla Sabo [a], Zheming Yuan [a,*], Zhijun Dai [a,*]

[a] Hunan Engineering & Technology Research Centre for Agricultural Big Data Analysis & Decision-making, Hunan Agricultural University, Changsha 410128, China
[b] College of Information and Intelligence, Hunan Agricultural University, Changsha 410128, China

ARTICLE INFO

Article history:
Received 16 December 2021
Revised 21 September 2022
Accepted 21 September 2022
Available online 17 October 2022

Keywords:
Clustering
EP_dis
Weighted distance
RCH
Internal validation index

ABSTRACT

Designing appropriate similarity metrics (distance) and estimating the optimal number of clusters have been two important issues in cluster analysis. This study proposed an improved *k*-means clustering algorithm involving a Weighted Distance and a novel Internal Validation index (WeDIV). The weighted distance, *EP_dis*, was designed by considering the relative contribution between Euclidean and Pearson distances with a weighted strategy. This strategy can effectively capture information reflecting the globally spatial correlation and locally variable trend simultaneously in high-dimensional space. The new internal validation index,*RCH*, inspired by the Calinski-Harabasz (*CH*) index and the analysis of variance, was developed to automatically estimate the optimal number of clusters. The *EP_dis* was proved reliable in mathematics and was validated on two simulated datasets. Four simulated datasets representing different properties were used to validate the effectiveness of *RCH*. Furthermore, We compared the clustering performance of WeDIV with 12 prevailing clustering algorithms on 16 UCI datasets. The results demonstrated that WeDIV outperforms the others regardless of specifying the number of clusters or not.

1. Introduction

Clustering is an unsupervised learning method in pattern recognition and machine learning. It is also a branch of multivariate statistical analysis, aiming to discover the natural groups of a set of patterns, points, or objects [1–3]. Clustering has been used in several fields, including image processing [4–7], text mining [8,9], bioinformatics [10–12], and medical studies [13–16]. From the statistics viewpoint, clustering can be categorized into a probability model-based and non-parametric approach [17]. The former assumes that the dataset follows a mixture probability distribution and uses a mixed likelihood method for clustering [18], where a representative is an expectation and maximization (EM) algorithm [19,20]. The non-parametric clustering can be further divided into hierarchical clustering and partitional methods based on the objective function of similarity metrics (distance) [3]. Hierarchical clustering methods [21,22] group data according to the distance matrix and association criteria defined on a dataset to form a dendrogram. Conversely, partitional methods assume that finite cluster prototypes can represent the dataset's structure based on their objective functions [23,24]. An appropriate similarity metric (distance) is essential in partitional methods. The *k*-means algorithm is one of the most famous partitional algorithms [25]. Because of its simplicity and efficiency, it has been widely studied and formed a series of improved algorithms derived, such as partitioning around medoids (PAM) [26], fuzzy c-means (FCM) [27,28], and X-means [29]. These extended algorithms typically use Euclidean distance as the similarity metric (distance). Generally, the similarity metric can be categorized into two groups [30]. The distance-based metrics can characterize the global correlation in high-dimensional space between objects, including Euclidean, Minkowski, and Mahalanobis distances. The correlation-based metrics can capture the locally variable trend between objects, including the Pearson correlation coefficient and Cosine similarity. A clustering algorithm using either of the two groups of similarity metrics (distance) may lose necessary information that can be captured by the other, especially for a dataset with complex structures. Designing a new

* Corresponding author.
E-mail address: daizhijun@hunau.edu.cn (Z. Dai).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.eij.2022.09.002

similarity metric that can simultaneously consider the two groups of distance to capture more intrinsic relationships between data points would help advance clustering performance. The number of clusters (k) in most clustering scenarios is generally unidentified. Nevertheless, an internal validation index could be used to estimate the number of clusters [31,32], such as Daves-Bouldin (*DB*) index [33], Silhouette (*Sil*) index [34], Calinski-Harabasz (*CH*) index [35], and *Gap* Statistic [36]. Milligenet et al. compared 30 internal validation indices comprehensively and found that *CH* performs the best in estimating the number of clusters [37,38]. However, the drawback of *CH* can be easily revealed by analyzing its equation: $\frac{SS_B}{k-1}/\frac{SS_W}{N-k}$, where SS_B represents the sum of the square between clusters, SS_W represents the sum of the square within clusters, and N is the number of data points. From the statistics viewpoint, *CH*'s essence is to analyze variance (ANOVA) among different clusters under a specified k. *CH* values calculated at different k are incomparable as the degrees of freedom vary. Suppose we have a dataset comprising 105 data points distributed in 6 clusters. Now there are two *CH* values calculated, $CH_1 = 2.40 < F_{0.05(4,100)} = 2.46$, when $k = 5$; and $CH_2 = 2.38 > F_{0.05(5,99)} = 2.30$, when $k = 6$, where $F_{0.05(k-1,N-k)}$ is the corresponding F-test threshold at the significance level of 0.05, N is the number of data points. A simple comparison between the *CH* value and its corresponding threshold shows that the differences between and within clusters are statistically significant when $k = 6$. Therefore, the optimal number of clusters should be 6. However, a direct comparison between the two *CH* values without considering the threshold would provide an incorrect conclusion that the optimal number of clusters is 5. This study proposed an improved k-means clustering algorithm involving a weighted distance and a novel internal validation index (WeDIV), which improved the k-means algorithm in two aspects. First, we designed a new distance, *EP_dis*, by combining the Euclidean distance and Pearson distance through a weighted strategy, which can capture global spatial correlation and locally variable trends between data points. Then, a new internal validation index named relative *CH* (*RCH*) was developed to estimate the number of clusters. We applied the new algorithm to 16 UCI datasets and compared it with 12 prevailing clustering methods. Experimental results showed that WeDIV outperformed the others regardless of specifying the cluster numbers or not. Our work was organized as follows: in Section 2, we reviewed the related work; in Section 3, we proposed the WeDIV algorithm in detail; in Section 4, we presented the results of different clustering algorithms and discussed; in Section 5, the conclusion and future work were stated.

2. Related work

This section reviews several works closely related to k-means clustering and internal validation index. Supposing a dataset $X = \{x_i\}_{i=1}^{N}$ comprising N data points with M features. k-means clustering aims to assign every data point to k clusters $\{V_j\}_{j=1}^{k}$, where $X = \bigcup_{j=1}^{k} V_j$, and $V_i \cap V_j = \varnothing$ $(1 \leqslant i \neq j \leqslant k)$.. The clustering process of k-means can be summarized as follows: 1) select k initial centroids randomly; 2) assign each data point to a centroid with the minimal distance; 3) update the centroids by recalculating the means (centroids) for data points that assigned to each cluster; 4) repeat steps 2 and 3 until the objective function converged [25,39]. Generally, the k-means algorithm adopts the minimization of the sum of squares error (*SSE*) as an objective function [40], which is defined as below:

$$SSE = \sum_{j=1}^{k} \sum_{x_i \in V_j} \|x_i - c_j\|^2, \tag{1}$$

where $c_j = \sum_{x_i \in V_j} x_i / n_j$ is the center of cluster V_j, n_j is the number of data points in cluster V_j, and $\|\cdot\|^2$ denotes the Euclidean norm reflecting the similarity between data points. Three parameters, including the scheme of cluster initialization, the similarity metric (distance), and the number of clusters k, must be specified before clustering with the k-means algorithm [41]. Most extensions of k-means made attempts concerning the three aspects.

2.1. Cluster initialization

The performance of k-means is highly dependent on initial centroids. Random centroid [42,43] are the most popular initialization method, along with k-means++ [44,45]. In addition, there are many initialization techniques, such as random partition [46], density-based [47], Maxmin [48], and Kaufman's variant of Maxmin [1]. As a classical version of k-means to deal with the initialization problem, k-means++ reduces the impact of random sampling on the result by selecting data points with specific probability as initial centroids [49]. In PAM [26], cluster centroids are represented using the median of data points instead of the mean, reducing the impact of noise and making the clustering result more stable. Hatamlous et al. [50] and Kumar et al. [51] proposed several binary search techniques to address the initialization problem of k-means. Feng et al. [52] highlighted that the initialization of clustering should consider the effect of outliers. Kumar and Reddy [53] proposed an efficient method for selecting the initial seed and improved the performance of k-means by locating seed points in dense dataset areas. Mittal et al. [54] proposed a method that selects the farthest data points as the initial seed and uses the adaptive neighborhood distance to estimate the number of clusters. Furthermore, Fränti and Sieranoja thoroughly compared various initialization techniques of cluster centroids. They demonstrated that initialization using a simple furthest point heuristic (Maxmin) method achieves better performance, whose clustering errors of k-means reduce from 15% to 6% on average [55]. The 'Maxmin' strategy randomly selects a data point as the first cluster centroid and then determines other centroids sequentially, where a data point with the maximal distance to the predefined centroids is determined as the next centroid. The 'Maxmin' technique is employed as the cluster initialization method in this study.

2.2. Similarity metrics

The choice of similarity metric (distance) plays a crucial role in clustering. Euclidean distance is the default similarity metric in most clustering algorithms. Some non-Euclidean distances are also used in clustering. Wu and Yang [56] designed a new similarity metric more robust than the Euclidean norm in c-means clustering and developed two new clustering algorithms, AHCM and AFCM. The km-M+ [57] algorithm gears clusters toward roughly elliptical shapes and uses the Mahalanobis distance for clustering. The *i*-MWK-means [58] algorithm employing a weighted Minkowski distance could overcome the drawbacks of the k-means that lack defending against noisy features. Nevertheless, optimization of the parameter β in *i*MWK-means remains a challenge. The *S*-k-means [59] algorithm derives a new distance metric *S*-distance with the *S*-divergence and is used in k-means. However, the algorithm must provide the number of clusters as a parameter. Meng et al. [60] defined a new similarity metric in k-means to capture the difference in trend characteristics between data points by considering the importance of derivative information.

2.3. Internal validation index

Several clustering algorithms, such as the k-means, must provide the cluster numbers as a parameter. An internal validation

index is a simple approach to estimating the optimal cluster numbers. The *DB* [33] index is a function of the ratio of the sum of intra-cluster scatter to inter-cluster separation. A smaller *DB* indicates less inter-cluster similarity. The *Sil* [34] index is a measure of how similar an object is to its cluster (cohesion) compared with other clusters (separation). The range of *Sil* is $[-1, 1]$. A larger *Sil* indicates less intra-cluster scatter and a more inter-cluster separation. The *CH* [35] index is a function of the ratio of inter-cluster dispersion to intra-cluster dispersion. A larger *CH* indicates a more similar intra-cluster and a more separate inter-cluster. The *Gap* [36] index is based on a statistical hypothesis test. It works by comparing the change in intra-cluster dispersion with what would be expected under an appropriate reference null distribution for each value of k. The k corresponding to the maximum *Gap* is the optimal number of clusters. Furthermore, several new internal validation indices have been proposed in recent years. Arima et al. [61] proposed a modified *Gap* index in fuzzy k-means clustering and applied it to gene expression datasets. Mur et al. [62] designed the *GS* index by combining the *Sil* index and the concept of local scaling to estimate the number of clusters in spectral clustering. Zhang et al. [63] proposed a curvature-based method for determining the number of clusters. Gupta et al. [64] proposed two techniques (LL and LML) to estimate the number of clusters in k-means by observing the minimum distance between cluster centroids.

2.4. Clustering algorithms determine the cluster numbers automatically

Except for utilizing the internal validation index, some clustering algorithms can automatically determine the cluster numbers. X-means [29] is an extension of k-means by making local decisions for cluster centroids in each iteration of k-means and using Bayes Information Criterion [65] or Akaike Information Criterion [66] to split itself to obtain a better cluster. Nonetheless, the X-means is still affected by the initialization problem. R-EM [20] is a robust EM clustering algorithm based on the Gaussian mixture model. This algorithm solves the problem that EM is sensitive to initial values. C-FS [47] assumed cluster centroids are characterized by a higher density than their neighbors and by a relatively large distance from points with higher densities. This algorithm determines the cluster centroids by finding the density peaks using a heuristic approach of a decision graph. Nevertheless, the performance of C-FS is highly dependent on two parameters: the local density ρ and cutoff distance β. A $-$ Ward$_{p\beta}$ [22] uses a feature weighted of the Minkowski distance in a hierarchical clustering algorithm, making it able to detect clusters with shapes other than spherical, but how to determine an appropriate parameter p and β should be considered. RL-FCM [17] introduces an entropy penalty term to adjust the bias-free of fuzziness index and creates a robust learning-based model to find the optimal number of clusters. However, this algorithm does not allow the provision of the number of clusters.

3. WeDIV algorithm

This section introduced the WeDIV algorithm in detail and theoretically verified the properties of the new distance metric, *EP_dis*, and the novel internal validation index, relative *CH* (*RCH*).

3.1. A new distance metric: EP_dis

As one of the most commonly used distances, Euclidean distance (E) (Eq. 2) is defined to represent the globally spatial correlation between data points x_i and x_j in M-dimensional space. The range of Euclidean distance is $[0, \infty]$. A greater Euclidean distance score means less similarity between two data points.

$$E_{x_i x_j} = \sqrt{\sum_{t=1}^{M} (x_{it} - x_{jt})^2} \tag{2}$$

The Pearson correlation coefficient (R) (Eq. 3) can be interpreted as a normalized covariance, reflecting the local variation trend between two data points.

$$R_{x_i x_j} = \frac{cov(x_i, x_j)}{\delta_{x_i} \delta_{x_j}}, \tag{3}$$

where $cov(x_i, x_j)$ is the covariance between x_i and x_j, δ_{x_i} and δ_{x_j} are the standard deviations of x_i and x_j, respectively. A larger R represents a stronger linear correlation between two data points. In order to determine the relative contribution between the Euclidean distance (E) and the Pearson distance $P = (1 - R)$ [67] in characterizing the similarity of two data points, we designed a new distance *EP_dis* by combining the two distances through a weighted process. The *EP_dis* is defined as below:

$$EP\_dis = wE + (1 - w)P, \tag{4}$$

w is the weight to be optimized with the range of [0, 1]. A smaller *EP_dis* represents a stronger similarity between the two data points. *EP_dis* would be Euclidean distance when $w = 1$ and Pearson distance when $w = 0$. Since the ranges of Euclidean distance (E) and Pearson distance (P) are different, before calculating the *EP_dis*, the distance matrix of the dataset must be normalized.

Property 1. The measure *EP_dis* in Eq. (4) is a distance metric. As is well known, a measure $d(x, y)$ being a distance metric should satisfy three properties as follow:

(1) Nonnegativity: $\forall x \neq y, d(x, y) > 0, d(x, x) = 0$
(2) Symmetry: $d(x, y) = d(y, x)$
(3) Triangle inequality: $\forall z, d(x, y) \leqslant d(x, z) + d(z, y)$

Proof. Let $d(x, y) = wE(x, y) + (1 - w)P(x, y)$. Conditions (1) and (2) are intuitively satisfied for the *EP_dis*. It only needs to prove the triangle inequality (3).

$$\begin{aligned} &d(x, z) + d(z, y) - d(x, y) \\ &= wE(x, z) + (1 - w)P(x, z) + wE(z, y) + (1 - w)P(z, y) \\ &-wE(x, y) - (1 - w)P(x, y) \\ &= w[E(x, z) + E(z, y) - E(x, y)] + (1 - w)[P(x, z) + P(z, y) - P(x, y)] \\ &\geqslant 0, \end{aligned} \tag{5}$$

Thus, $\forall z, d(x, y) \leqslant d(x, z) + d(z, y)$.

3.2. Optimization of clustering centers and w

3.2.1. Determination of clustering centers

The distance matrix of *EP_dis* was first calculated for a dataset under an initial w. The furthest pair of data points were selected as the first two centroids, c_1 and c_2. If the number of clusters $k > 2$, the remaining centroids would be determined using the 'Maxmin' technique [55]. The 'min' distance from data point x_i to each centroid could be defined as below:

$$Dis_i = \min_{1 \leqslant g \leqslant j} (EP\_dis(c_g, x_i)) \tag{6}$$

The data point corresponding to the maximum of *Dis* was chosen as the next centroid, defined below:

$$c_{j+1} = arg(max_{1 \leqslant i \leqslant (N-j)} (Dis_i)) \tag{7}$$

Once all centroids were identified, the remaining data points were allocated to the cluster with the nearest centroids. Then,

the k initial centroids would go through a refinement. The data point with the minimum sum of *EP_dis* in on cluster was considered as the new centroid, which was defined as:

$$c_j' = arg\left(min_{1\leqslant i\leqslant n_j}\left(\sum_{t=1}^{n_j} EP\_dis(x_i, x_t)\right)\right), i \neq t, \tag{8}$$

where n_j is the number of data points in V_j. Redistribute all the data points. This process was repeated until the cluster labels for all data points remained unchanged.

Example 1. To illustrate the impact of *EP_dis* on the determination of cluster centers, we synthesized two datasets, D1 and D2, using different mathematical models (Table 1) [68]. D1 comprised 490 samples belonging to six clusters with 30-dimensional features. D2 comprised 400 samples belonging to five clusters with 10-dimensional features. Fig. 1 shows the variation of centroids (marked in solid red circle) in D1 (Figs. 1a-1c) and D2 (Figs. 1d-1f) under different w, respectively. We can see that the centroids of D1 selected when $w = 0.6$ (Fig. 1c) and the centroids of D2 selected when $w = 0.8$ (Fig. 1f) were more reliable than those selected when $w = 0$ or $w = 1$. This example indicated that the weighted distance *EP_dis* helps obtain better cluster centers.

3.2.2. Optimization of w

We used an incremental search strategy to determine the optimal w in *EP_dis*, of which the range is [0, 1], and the incremental step (*w_step*) is set to 0.1. Once we obtained the clusters under a specified w, SS_B/SS_W (Eq. 10 and Eq. 11) could be calculated. The w corresponding to the maximum SS_B/SS_W was regarded as the optimal weight in *EP_dis*.

Table 1
Mathematical models of D1 and D2.

	D1	D2
model 1	$0.1 + sin\left(\frac{j}{3}\right) + \xi(i,j,k)$	$\frac{-exp(j)}{1000} + \xi(i,j,k)$
model 2	$-0.1 + sin\left(\frac{j}{3} - 1\right) + \xi(i,j,k)$	$\frac{j}{6.6} + \xi(i,j,k)$
model 3	$1.2sin\left(\frac{2j}{5} - 2\right) + \xi(i,j,k)$	$\frac{5(j-4)^2}{max(j-4)^2} + \xi(i,j,k)$
model 4	$1.5sin\left(\frac{j}{3} - 3.5\right) + \xi(i,j,k)$	$sin(j) + \xi(i,j,k)$
model 5	$0.5sin\left(\frac{2j}{5} - 2.2\right) + \xi(i,j,k)$	$cos(j) + \xi(i,j,k)$
model 6	$0.6sin\left(\frac{j}{3} - 3.8\right) + \xi(i,j,k)$	

i: the model id; j: the number id of features; k: the number id of samples in each model ξ: the random error.

Example 2. To illustrate the process of optimizing w more explicitly, we tested the performance of clustering on D1 and D2 with the incremental value of w (Fig. 2). Two external validation measures, the *CA* (Clustering Accuracy) [51] and *Rand* [61], were used for evaluating the clustering performance and comparison with the SS_B/SS_W. As expected, we observed that the variation trend of SS_B/SS_W, in general, is consistent with those of *CA* and *Rand* on both datasets. The fluctuation of *CA* appeared to be more fierce than the other two measures. Remarkably, these three measures achieved the highest scores under the same w that did not appear at the endpoints (0.6 for D1 and 0.8 for D2), indicating that the SS_B/SS_W is effectively optimizes the weight w.

3.3. RCH

An appropriate number of clusters is crucial for clustering performance. Milligenet et al. compared 30 internal validation indices comprehensively and found that the *CH* performed the best in estimating the number of clusters [37,38]. *CH* [35] is a clustering internal validation index proposed by Calinski and Harabasz in 1974, which is defined as follows:

$$CH = \frac{\frac{SS_B}{k-1}}{\frac{SS_W}{N-k}}, \tag{9}$$

$$SS_B = \sum_{i=1}^{k} n_i \|c_i - \overline{c}\|^2, \tag{10}$$

$$SS_W = \sum_{i=1}^{k}\sum_{j=1}^{n_j} \|x_j - c_i\|^2, \tag{11}$$

where N is the number of data points; k is the number of clusters; n_j (n_i) is the number of data points in cluster V_j (V_i); c_i is the centroid of cluster i; $\overline{c} = \sum_{i=1}^{N} x_i/N$ is the overall mean of the data points. SS_B (sum of squares between clusters) reflects the extent of inter-cluster separation, and SS_W (sum of squares within clusters) depicts

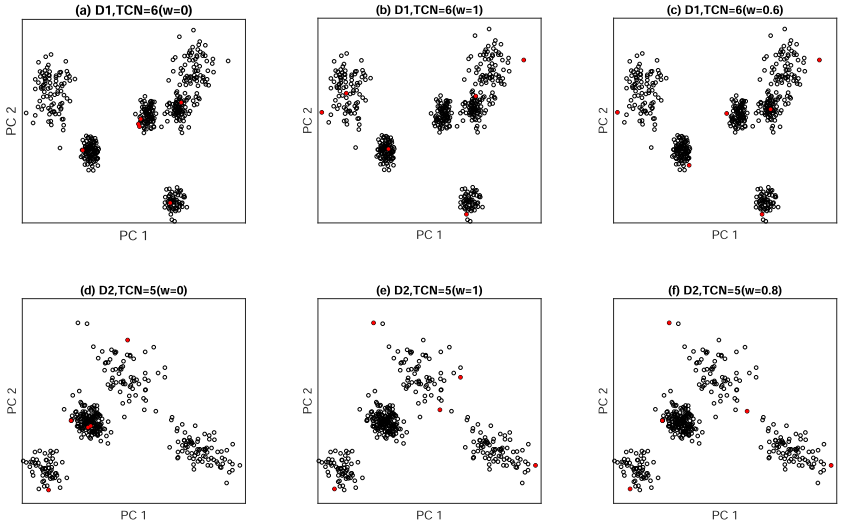

Fig. 1. The variation of centroids under different w for D1 and D2. Each hollow circle represents a sample, and the solid red circles represent centroids.

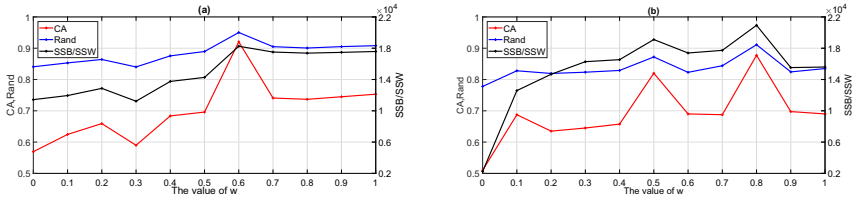

Fig. 2. Clustering performance was evaluated by three measures in optimizing the *w* of *EP_dis* for D1 and D2. The left vertical axis in each subplot represents the values of *CA* and *Rand* indices, and the right axis represents the scores of SS_B/SS_W.

the intra-cluster compactness. The k corresponding to the maximum *CH* is regarded as the optimal number of clusters. From the statistics viewpoint, the essence of *CH* (Eq. 9) is to perform ANOVA (analysis of variance) among different clusters under a specified k. As the degrees of freedom changed, the *CH* values calculated under different k become incomparable (details in Section 1). An intuitive idea to make the *CH* values comparable is to use a threshold, e.g. the *F*-value used in the *F*-test with corresponding degrees of freedom at a significant level, to adjust *CH*. Then, we can design a new index, relative *CH* (*RCH*), to determine the number of clusters, which is defined as below:

$$RCH_k = \frac{CH_k}{F_{(\alpha,k-1,N-k)}}, \tag{12}$$

the k corresponding to the maximum *RCH* is regarded as the optimal number of clusters. The significant level α for the threshold of *F*-test in this study was set to 0.05.

Example 3. An effective internal validation index should have properties performing well in different aspects of clustering. In this example, four two-dimensional Gaussian datasets [31] with different characteristics were used to illustrate the properties of *RCH*. D3 (Fig. 3a) is a dataset composing five well-separated clusters. D4 (Fig. 3b) is a dataset formed by adding 10% noise to the dataset D3. D5 (Fig. 3c) is a dataset composed of two clusters with different densities. D6 (Fig. 3d) is a dataset containing five clusters, in which two pairs of clusters can be regarded as two subclusters. These four datasets represent different properties in clustering, such as monotonicity, noise, density, and subclusters. We compared the *RCH* with four classical internal validation indices, including *CH*, *Sil*, *Gap*, and *XB*. For the monotonicity property, all five indices achieved their best values when the number of clusters estimated equals the true cluster numbers (*TCN*) (Fig. 4a), which may be attributed to their consideration of separation and compactness. By observing the effect of noise on internal validation indices, except for the *Sil* and *XB*, each of the remaining indices obtained the *TCN* (Fig. 4b), indicating that these indices are insensitive to a noise. For clusters in a dataset with different densities, all five indices determined the *TCN* (Fig. 4c). For the effect of subclusters on internal validation indices, only the *RCH* and *CH* acquired the *TCN* (Fig. 4d). The fluctuation of *RCH*, in general, was less severe than the others, partly due to the attribution to adjustment by the *F*-test threshold. Generally, the *RCH* proposed in this study presented a better performance for datasets with different clustering properties. Based on the above analysis, The WeDIV was presented in Algorithm 1.

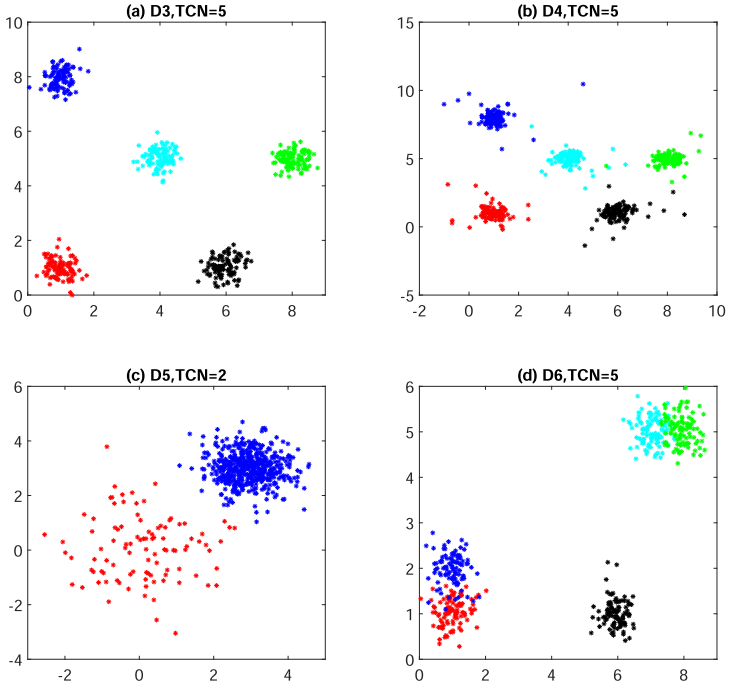

Fig. 3. Four datasets with different cluster properties. (a) clusters are "well separated". (b) clusters with "noise" are well separated. (c) clusters with "different densities". (d) clusters with "subclusters". *TCN* is the true cluster numbers.

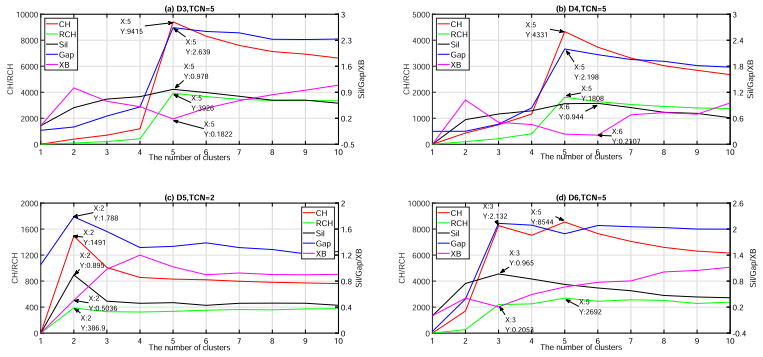

Fig. 4. Performance of five internal validation indices to estimate the cluster numbers for datasets with different cluster properties. The optimal coordinates relating to each index were marked on each subplot separately. *TCN* is the true cluster numbers.

Algorithm 1 WeDIV(X,*w_step*,*NC*)

Input: A dataset $X = \{x_i\}_{i=1}^{N}$, where every feature should be normalized; *w_step*: the incremental step of w; *NC*: the number of clusters given as a constant if the cluster numbers is provided, or being a vector as 2:*MaxNC*.
Output: Clustering result $\{V\}$
1: calculate the scaled E distance and $P = (1 - R)$ distance using Eq. 2 and Eq. 3.
2: **for** k in *NC* **do**
3: $w = 0$
4: **while** $w \leqslant 1$ **do**
5: calculate the $EP\_dis_{i,j}$ using Eq. 4. for all pairs of data points in X.
6: choose the first two centroids c_1 and c_2 with the largest distance of paired data points.
7: **if** $k > 2$ **then**
8: determine the remaining centroids sequentially using Eq. 6 and Eq. 7.
9: **end if**
10: **repeat**
11: assign each non-centroid data point to the corresponding clusters.
12: updata the centroids using Eq. 8.
13: **until** no centroid needs to be updated
14: calculate and record the SS_B/SS_W under w.
15: $w = w + w\_step$.
16: **end while**
17: $optimal\_w = arg(max(SS_B/SS_W))$;
18: **if** *NC* is a constant **then**
19: return $\{V\}_{(NC,optimal\_w)}$
20: **else**
21: $CH_k = max(SS_B/SS_W \ * ((N-k)/(k-1)))$
22: calculate the RCH_k using Eq. 12.
23: $optimal\_wink[k] = optimal\_w$
24: **end if**
25: **end for**
26: $optimal\_k = arg(max(RCH))$.
27: $optimal\_w = optimal\_wink[optimal\_k]$
28: return $\{V\}_{(optimal\_k,optimal\_w)}$.

3.4. Time complexity of WeDIV

We denoted n as the number of samples, m as the feature number of samples, l as the number of iterations to determine the cluster centers, k as the number of clusters, r as the range of k, *w_step* as the incremental step. According to Algorithm 1, the time complexity of WeDIV is O($lkmn/w\_step$) when the number of clusters is provided. The time complexity is O($lkmnr/w\_step$) when the number of clusters needs to be estimated. Since the k and r are typically much smaller than n as well as the *w_step* is a constant, the time complexity of the WeDIV holds approximately O(lmn).

4. Experiments

4.1. Datasets

Sixteen real-world datasets (Table 2) derived from the UCI machine learning repository (http://archive.ics.uci.edu) were used to validate the effectiveness of WeDIV. These datasets represent wide research fields, such as medicine, agriculture, biology, chemistry, and physics.

4.2. Experimental setup

4.2.1. Evaluation metrics

In this study, two external validation measures, *CA* (Clustering Accuracy) and *Rand*, were used to evaluate clustering performance, where the *CA* [51] represents the ratio of correctly predicted data points to the total number of data points defined as follows:

$$CA = \sum_{i=1}^{k} \frac{a_i}{N}, \tag{13}$$

where N is the number of data points, a_i is the number of data points that are correctly predicted in cluster V_i. The *Rand* [61] is defined as follows:

Table 2
Descriptions of the datasets.

Datasets	No. of samples	No. of features	No. of categories
butterfly	70	9	5
wdbc	569	30	2
statlog	2000	36	6
spectf	187	44	2
wine	178	13	3
seed	569	7	3
sensor	5456	24	4
frogs	7195	22	10
dermatology	366	33	6
glassdata	214	9	6
haberman	306	3	2
mcluster	60	4	3
yeast	1484	8	10
zoo	101	16	7
srbct	83	2308	4
waveform	5000	21	3

$$Rand = \frac{a+d}{a+b+c+d} = \frac{a+d}{C_N^2}. \quad (14)$$

Assume that R represents the published clusters and P represents the predicted clusters. Then, the elements in the *Rand* equation can be explained as a is the number of pairs of data points that are in the same R and P; b is the number of pairs of data points that are in the same R but the different P; c is the number of pairs of data points that are in the different R but the same P; d is the number of pairs of data points that are in the different R and P. C_N^2 is the number of data points in a dataset. The ranges of *CA* and *Rand* are [0, 1]. A larger *CA* (or *Rand*) represents a better performance of clustering.

4.2.2. Implementation

The WeDIV clustering algorithm proposed in this study was implemented with Matlab2017a. All of the algorithms, except for the *S*-*k*-means and X-means, were run in Matlab R2017a on a Windows 10 64-bit system with Intel (R) Core (TM) i5-7200U(2.5 GHz, 2.7GHZ) CPUs and 8 GB RAM. The X-means was implemented with the package 'pyclustering' in Python 3.7. The *S*-*k*-means was implemented with the package 'skmeans' in Rstudio 4.0; *i*WMK-means, *S*-*k*-means, R-EM, C-FS, A – $\text{Ward}_{p\beta}$, and RL-FCM source codes were provided by the authors. All clustering results calculated in this study were averaged over 50 independent runs. The default parameters, unless specifically stated, in the program were used as suggested in the companion paper.

4.3. Results and discussion

4.3.1. Role of EP_dis in WeDIV

The properties of distance *EP_dis* have been proven reliable in Section 3.1. To further investigate the role of *EP_dis* in WeDIV, we compared the results of *EP_dis* when using the optimal w with those of Euclidean ($w = 1$) and Pearson distance ($w = 0$) (Table 3). The best accuracies were marked in bold. *EP_dis* with the optimal w reached the best performance on average, which improved the clustering accuracy by 3% (4%) versus Euclidean distance and 8.5% (15%) versus Pearson distance in *Rand* (*CA*), respectively. More specifically, *EP_dis* acquired the best accuracies on 13 (12) of 16 datasets in *Rand* (*CA*); 7 (6) of them presented the best accuracies when the optimal w was neither 1 nor 0. Furthermore, a Wilcoxon signed-rank test was conducted to test the statistical difference between them. As expected, the p-value of the Wilcoxon signed-rank test calculated between *EP_dis* and Euclidean distance attained a significant difference (<0.05) as 9.76*E*-4 (4.19*E*-2) in *Rand* (*CA*); the p-value tested between *EP_dis* and Pearson distance was also significant as 3.35*E*-3 (3.78*E*-3). These results validated our hypothesis that considering the distance with global spatial correlation and local variation trend between data points could capture more intrinsic information to improve clustering.

4.3.2. Comparison with the reference algorithms under the specified cluster numbers

We then investigated if the WeDIV is superior to the reference algorithm when the true cluster numbers (*TCN*) is provided. These algorithms include hierarchical [21], EM [19], FCM [28], *k*-means++ [49], PAM [26], *i*WMK-means [58], and *S*-*k*-means [59]. Table 4 presented the clustering performance of WeDIV and the seven reference algorithms on 16 real datasets, where the best accuracies were marked in bold. WeDIV achieved the best average performance, with *Rand* at 0.779 and *CA* at 0.702. Specifically, WeDIV performed the best on eight datasets (including butterfly, wdbc, statlog, spectf, seed, dermatology, haberman, and zoo); the *S*-*k*-means algorithm performed fairly well on six datasets (including wine, sensor, frogs, glassdata, mcluster, and yeast). The others only acquired the best accuracy on two or fewer datasets. We also applied a one-sided Wilcoxon signed-rank test in Table 4 to investigate if the difference between WeDIV and reference algorithms is statistically significant. The p-values of the hypothesis tests indicated that the clustering accuracies of WeDIV were significantly superior to other algorithms (<0.05) except for *S*-*k*-means (Table 5). The *S*-*k*-means, ranked second-top, is also a practical clustering algorithm; however, the cluster numbers must be provided as parameters. The results demonstrated that WeDIV is an excellent algorithm for specifying the cluster numbers.

Table 3
Clustering accuracies in *Rand*(*CA*) with WeDIV using different distance metrics.

Datasets	Measure	Pearson distance ($w = 0$)	Euclidean distance ($w = 1$)	*EP_dis* (optimal w)
butterfly	*Rand*	**1.00**	**1.00**	**1.00(0)**
	CA	**1.00**	**1.00**	**1.00(0)**
wdbc	*Rand*	0.587	**0.875**	**0.875(1.0)**
	CA	0.710	**0.933**	**0.933(1.0)**
statlog	*Rand*	0.743	**0.859**	**0.859(1.0)**
	CA	0.546	**0.701**	**0.701(1.0)**
spectf	*Rand*	0.497	**0.694**	**0.694(1.0)**
	CA	0.503	**0.813**	**0.813(1.0)**
wine	*Rand*	0.866	0.875	**0.904(0.8)**
	CA	0.893	0.899	**0.927(0.8)**
seed	*Rand*	0.768	**0.888**	**0.888(1.0)**
	CA	0.786	**0.910**	**0.910(1.0)**
sensor	*Rand*	0.592	0.577	**0.602(0.3)**
	CA	0.397	**0.413**	0.375(0.3)
frogs	*Rand*	0.784	0.814	**0.846(0.5)**
	CA	0.473	0.575	**0.610(0.5)**
dermatology	*Rand*	0.931	0.911	**0.960(0.6)**
	CA	0.833	0.809	**0.918(0.6)**
glassdata	*Rand*	**0.704**	0.601	0.677(0.9)
	CA	0.402	0.477	**0.537(0.9)**
haberman	*Rand*	0.499	**0.629**	**0.629(1.0)**
	CA	0.513	**0.755**	**0.755(1.0)**
mcluster	*Rand*	0.566	0.542	**0.573(0.8)**
	CA	0.467	0.450	**0.517(0.8)**
yeast	*Rand*	**0.747**	0.710	**0.747(0.4)**
	CA	0.376	**0.392**	0.379(0.4)
zoo	*Rand*	0.929	0.878	**0.950(0.8)**
	CA	0.802	0.713	**0.881(0.8)**
srbct	*Rand*	**0.663**	0.648	0.650(0.9)
	CA	**0.506**	0.410	0.422(0.9)
waveform	*Rand*	**0.609**	0.597	0.608(0.7)
	CA	**0.570**	0.547	0.552(0.7)
average	*Rand*	0.718	0.756	**0.779**
	CA	0.611	0.675	**0.702**

4.3.3. Contribution of RCH in WeDIV

Relative *CH* plays an important role in WeDIV estimating the optimal number of clusters. To illustrate the effectiveness of *RCH*, the *CH* and *RCH* were employed in WeDIV for comparison (Fig. 5). the *RCH* estimated the correct cluster numbers on nine datasets (including butterfly, wdbc, spectf, statlog, sensor, seed, wine, srbct, and waveform). In comparison, *CH* only exhibited the correct cluster numbers on three datasets such as butterfly, wdbc, and spectf. Except for the butterfly (Fig. 5a) and statlog (Fig. 5c), the maximum value of *CH* for all datasets appears when $k = 2$, which means the *CH* tends to divide a dataset into fewer clusters. In contrast, the optimal k of *RCH* is more abundant. The trend of *RCH* scores was much more stationary than *CH*, probably because of the adjustment of the threshold of the *F*-test to *CH*.

4.3.4. Comparison with the reference algorithms under the estimated cluster numbers

In most clustering scenarios, the cluster numbers are generally unidentified. We also compared the WeDIV with the reference

Table 4
Clustering performance of different algorithms for specifying the cluster numbers.

Datasets	Measure	Hierarchical	*k*-means++	FCM	PAM	EM	*i*WMK-means	*S-k*-means	WeDIV
butterfly	*Rand*	**1.00**	0.908	**1.00**	**1.00**	**1.00**	0.840	0.758	**1.00**
	CA	**1.00**	0.860	**1.00**	**1.00**	**1.00**	0.671	0.486	**1.00**
wdbc	*Rand*	0.655	0.750	0.750	0.750	0.860	0.526	0.776	**0.875**
	CA	0.779	0.854	0.854	0.854	0.924	0.617	0.872	**0.933**
statlog	*Rand*	0.846	0.818	0.853	0.855	0.798	0.786	0.845	**0.859**
	CA	0.678	0.601	0.693	0.691	0.516	0.483	0.649	**0.701**
spectf	*Rand*	0.570	0.605	0.538	0.570	0.633	0.509	0.570	**0.694**
	CA	0.690	0.732	0.642	0.690	0.759	0.578	0.690	**0.813**
wine	*Rand*	0.717	0.701	0.711	0.688	0.933	0.876	**0.976**	0.904
	CA	0.697	0.624	0.685	0.579	0.949	0.904	**0.983**	0.927
seed	*Rand*	0.872	0.873	0.874	0.874	0.857	0.780	0.734	**0.888**
	CA	0.890	0.893	0.895	0.895	0.876	0.771	0.695	**0.910**
sensor	*Rand*	0.588	0.600	0.528	0.590	0.588	0.578	**0.622**	0.602
	CA	0.404	0.410	0.497	0.405	0.421	0.365	**0.579**	0.375
frogs	*Rand*	0.816	0.804	0.777	0.949	0.844	0.281	**0.958**	0.846
	CA	0.561	0.517	0.410	0.766	0.605	0.483	**0.820**	0.610
dermatology	*Rand*	0.908	0.911	0.896	0.906	0.911	0.485	0.881	**0.960**
	CA	0.713	0.756	0.776	0.716	0.765	0.437	0.626	**0.918**
glassdata	*Rand*	0.663	0.648	0.712	0.658	0.639	0.665	**0.815**	0.677
	CA	0.500	0.529	0.491	0.537	0.528	0.355	**0.607**	0.537
haberman	*Rand*	0.502	0.525	0.499	0.499	0.498	0.498	0.578	**0.629**
	CA	0.542	0.561	0.510	0.520	0.503	0.507	0.699	**0.755**
mcluster	*Rand*	0.545	0.574	0.573	0.567	0.537	0.555	**0.810**	0.573
	CA	0.467	0.465	0.467	0.450	0.500	0.450	**0.800**	0.517
yeast	*Rand*	0.731	0.741	0.718	0.736	0.606	0.235	**0.866**	0.747
	CA	0.336	0.372	0.322	0.404	0.443	0.313	**0.550**	0.379
zoo	*Rand*	0.900	0.871	0.830	0.894	0.907	0.690	0.886	**0.950**
	CA	0.782	0.728	0.572	0.792	0.832	0.584	0.733	**0.881**
srbct	*Rand*	0.570	0.583	0.543	**0.693**	–	0.604	0.688	0.650
	CA	0.434	0.467	0.395	**0.627**	–	0.349	0.554	0.422
waveform	*Rand*	**0.694**	0.607	0.603	0.598	0.607	0.583	0.611	0.608
	CA	**0.637**	0.541	0.538	0.545	0.542	0.492	0.540	0.552
average	*Rand*	0.724	0.720	0.713	0.739	0.748	0.593	0.773	**0.779**
	CA	0.632	0.619	0.609	0.654	0.678	0.522	0.680	**0.702**

– the algorithm fails to conduct clustering on the dataset; the best accuracies were marked in bold.

Table 5
The results of the Wilcoxon signed-rank test conducted on WeDIV versus the reference algorithms.

Measure	Hierarchical	*k*-means++	FCM	PAM	EM	*i*WMK-means	*S-k*-means
Rand	2.05*E*-3	2.10*E*-4	3.05*E*-4	8.10*E*-3	6.13*E*-4	1.61*E*-4	4.06*E*-1
CA	3.32*E*-3	1.05*E*-3	1.47*E*-3	6.77*E*-2	1.81*E*-2	1.61*E*-4	4.16*E*-1

algorithms that can estimate the cluster numbers automatically, including *k*-means++/*CH*, X-means [29], R-EM [20], RL-FCM [17], C-FS [47], and A – Ward$_{p\beta}$ [22]. The clustering accuracies and the estimated cluster numbers (k) with different clustering algorithms on 16 real datasets were summarized in Table 6, where the correct k and the best accuracies were marked in boldface, and the accuracies with incorrect k were not listed. Table 6 shows that the *RCH* estimated the correct cluster numbers on nine datasets. Remarkably, their accuracies were also superior to reference algorithms. C-FS obtained the correct cluster numbers on six datasets, while only 2 of them received the best accuracies. X-means presented the correct cluster numbers on five datasets and achieved the best performance only on two datasets. The *k*-means++/*CH*, R-EM, A – Ward$_{p\beta}$, and RL-FCM could only estimate the correct cluster numbers on three or fewer datasets. The R-EM could not perform clustering on srbct, and the RL-FCM also failed on three datasets. The ability to estimate the cluster numbers in each algorithm can be represented by calculating the values of *abs*(*k-TCN*) on all datasets and then ranking them. The discrete ranks obtained by each clustering algorithm are illustrated in Fig. 6, where a lower rank represents better performance for estimating cluster numbers. As can be seen that WeDIV exhibited the lowest median of discrete ranks; C-FS and X-means performed the second, while the RL-FCM and A – Ward$_{p\beta}$ presented much higher ranks than the others. The two algorithms failed because they could only estimate the correct cluster numbers on one (for RL-FCM) or two datasets (for A – Ward$_{p\beta}$). The experiment demonstrated that WeDIV could perform well when the cluster numbers are not provided.

5. Conclusion and future work

In this study, we proposed an improved *k*-means clustering algorithm-WeDIV. The main contributions of WeDIV to *k*-means can be summarized as two aspects. Firstly, the algorithm employed a new integrated distance *EP_dis* instead of a single distance for clustering, which combines the Euclidean and Pearson distance through a weighting process. The *EP_dis* can simultaneously capture the information reflecting the globally spatial correlation and locally variable trend in high-dimensional space. Secondly, a novel internal validation index, namely *RCH*, was designed and involved in WeDIV to estimate the number of clusters. The *EP_dis* was proved reliable in mathematics theoretically and was validated on two simulated datasets. The *RCH* not only inherits the *CH*'s properties but also perpetrates better. We demonstrated the performance of *RCH* on four synthetic datasets. Furthermore, we compared WeDIV with 12 prevailing clustering on 16 UCI datasets. The results indicated that WeDIV outperformed reference algorithms, regardless of specifying the cluster numbers or not. We also

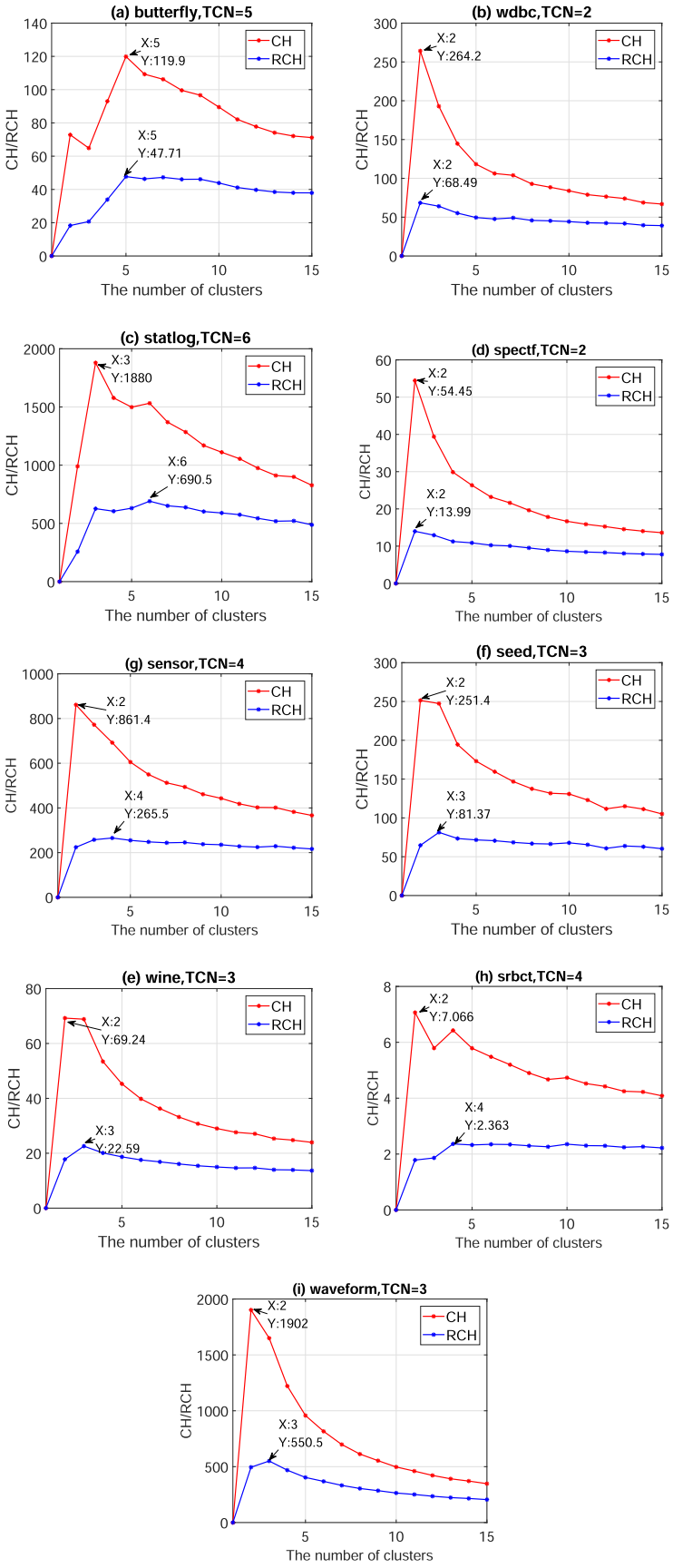

Fig. 5. Performance of *CH* and *RCH* in estimating the cluster numbers. The nine datasets estimated the correct cluster numbers with *RCH*. *TCN* is the true cluster numbers. The maximum of *CH* or *RCH* was marked on each subplot separately.

Table 6
Clustering performance of different algorithms for estimating the cluster numbers automatically.

Datasets	No. of categories	*k*-means++/*CH*			X-means			R-EM			RL-FCM			C-FS			A − Ward$_{p\beta}$			WeDIV		
		k	*Rand*	*CA*	*k*	*Rand*	*CA*	*k*	*Rand*	*CA*	*k*	*Rand*	*CA*	*k*	*Rand*	*CA*	*k*	*Rand*	*CA*	*k*	*Rand*	*CA*
butterfly	5	**5**	0.908	0.860	6	–	–	7	–	–	1	–	–	**5**	**1.00**	**1.00**	9	–	–	**5**	**1.00**	**1.00**
wdbc	2	12,13,15	–	–	15	–	–	**2**	0.842	0.914	569	–	–	**2**	0.501	0.258	15	–	–	**2**	**0.875**	**0.933**
statlog	6	3	–	–	15	–	–	3	–	–	1	–	–	3	–	–	13	–	–	**6**	**0.859**	**0.701**
spectf	2	**2**	0.605	0.732	**2**	0.603	0.729	**2**	0.554	0.668	NAN	–	–	**2**	0.338	0.497	14	–	–	**2**	**0.694**	**0.813**
wine	3	13,14,15	–	–	**3**	0.719	0.702	2	–	–	NAN	–	–	**3**	0.674	0.399	12	–	–	**3**	**0.904**	**0.927**
seed	3	**3**	0.873	0.893	**3**	0.842	0.862	**3**	0.868	0.888	**3**	0.874	0.895	**3**	0.705	0.429	12	–	–	**3**	**0.888**	**0.910**
sensor	4	2	–	–	15	–	–	7	–	–	2	–	–	6	–	–	13	–	–	**4**	**0.602**	**0.375**
frogs	10	2	–	–	15	–	–	3	–	–	11	–	–	3	–	–	2	–	–	7	–	–
dermatology	6	3	–	–	**4,6**	**0.960**	**0.915**	3	–	–	2	–	–	2	–	–	**6**	0.628	0.314	5	–	–
glassdata	6	2,3	–	–	3,7,8,9	–	–	5	–	–	2	–	–	2	–	–	**6**	**0.605**	**0.435**	10	–	–
haberman	2	4	–	–	3	–	–	4	–	–	1	–	–	**2**	**0.501**	**0.389**	11	–	–	6	–	–
mcluster	3	4	–	–	2	–	–	8	–	–	60	–	–	2	–	–	8	–	–	10	–	–
yeast	10	2	–	–	2,12,13,15	–	–	2	–	–	39	–	–	9	–	–	2	–	–	9	–	–
zoo	7	2	–	–	4,5,6,**7**	**0.830**	**0.604**	4	–	–	6	–	–	18	–	–	2	–	–	4	–	–
srbct	4	2	–	–	2	–	–	NAN	–	–	NAN	–	–	7	–	–	5	–	–	**4**	**0.650**	**0.422**
waveform	3	2	–	–	7	–	–	1	–	–	9	–	–	2	–	–	10	–	–	**3**	**0.608**	**0.552**

The best accuracies were marked in boldface; the accuracies with incorrect *k* were not listed.

Fig. 6. The performance ranking of seven clustering methods on sixteen datasets for estimating the cluster numbers. The discrete ranks representing the performance were obtained by calculating the *abs*(*k-TCN*) on all real datasets and then ranking them. A lower rank represents a better estimation of *k*. The mean of their ranks replaces ties. *TCN* is the true cluster numbers.

investigated the role of *EP_dis* in WeDIV on real datasets; a Wilcoxon signed-rank test showed the *EP_dis* used in WeDIV was significantly superior to Euclidean or Pearson distance. Although we comprehensively evaluated the effectiveness of WeDIV in both a theoretical and practical manner, limitations with two aspects still are concerned. Firstly, the time complexity of WeDIV is relatively higher than that of *k*-means due to the intrinsic optimization process of the weight *w* in *EP_dis*. However, this optimization can be easily deployed in a parallel computing environment to make the WeDIV more efficient. Secondly, the influence of redundant features is not considered in WeDIV, which would limit its applications on datasets with high-dimensional. Clustering with high-dimensional data is an important question in unsupervised learning [69,70]. We have conducted a series of research on feature selection in recent years [71–73]. Our future work would mainly focus on developing and employing feature selection methods to improve clustering accuracy. Applying the WeDIV to big data in biology, e.g., single-cell clustering, would also be interesting research work.

Acknowledgements

This work was supported by the National Natural Science Foundation of China [Grant No. 31701164], the Natural Science Foundation of Hunan Province, China [Grant No. 2018JJ3238], the Scientific Research Program of the Educational Department of Hunan Province, China [Grant No. 17A096], the Scientific Research Program of the Educational Department of Hunan Province, China [Grant No. 18A105], and the Scientific Research Program of the Educational Department of Hunan Province, China [Grant No. 18C0171].

References

[1] Kaufman L, Rousseeuw PJ. Finding Groups in Data: An Introduction to Cluster Analysis (DBLP); 2009.
[2] Jain AK, Murty MN, Flynn PJ. Data clustering: a review. ACM Comput Surveys 1999;31(3):264–323.
[3] Fahad A, Alshatri N, Tari Z, Alamri A, Bouras A. A survey of clustering algorithms for big data: taxonomy and empirical analysis. Emerg Top Comput IEEE Trans 2014;2(3):267–79.
[4] Zhang GY, Wang CD, Huang D, Zheng WS, Zhou YR. TW-Co-k-means: Two-level weighted collaborative k-means for multi-view clustering. Knowl-Based Syst 2018;150:127–38.
[5] Zhang YL, Yang Y, Li TR, Fujita H. A multitask multiview clustering algorithm in heterogeneous situations based on LLE and LE. Knowl-Based Syst 2019;163:776–86.
[6] Wang H, Yang Y, Liu B, Fujita H. A study of graph-based system for multi-view clustering. Knowl-Based Syst 2019;163:1009–19.

[7] Tian K, Li JH, Zeng JF, Evans A, Zhang L. Segmentation of tomato leaf images based on adaptive clustering number of K-means algorithm. Comput Electron Agric 2019;165:104962.
[8] Jia CY, Carson MB, Wang XY, Yu J. Concept decompositions for short text clustering by identifying word communities. Pattern Recogn 2018;76:691–703.
[9] Janani R, Vijayarani DS. Text document clustering using Spectral Clustering algorithm with Particle Swarm Optimization. Expert Syst Appl 2019;134:192–200.
[10] Lam YK, Tsang PWM. eXploratory K-Means: A new simple and efficient algorithm for gene clustering. Appl Soft Comput 2012;12(3):1149–57.
[11] Kakushadze Z, Yu W. K-means and cluster models for cancer signatures. Biomol Detection Quantification 2017;13:7–31.
[12] Gan YI, Li N, Zou GB, Xin YC, Guan JH. Identification of cancer subtypes from single-cell RNA-seq data using a consensus clustering method. BMC Med Genomics 2018;11:65–72.
[13] Khanmohammadi S, Adibeig N, Shanehbandy S. An improved overlapping k-means clustering method for medical applications. Expert Syst Appl 2017;67:12–8.
[14] Nithya A, Appathurai A, Venkatadri N, Ramji DR, Palagan CA. Kidney disease detection and segmentation using artificial neural network and multi-kernel k-means clustering for ultrasound images. Measurement 2020;149:106952.
[15] Sarkar JP, Saha I, Maulik U. Rough Possibilistic Type-2 Fuzzy C-Means clustering for MR brain image segmentation. Appl Soft Comput 2016;46:527–36.
[16] Zhang XB, Yang Y, Li TR, Zhang YL, Wang H, Fujita H. CMC: A Consensus Multi-view Clustering Model for Predicting Alzheimer's Disease Progression. Comput Methods Programs Biomed 2020;199:105895.
[17] Yang MS, Nataliani Y. Robust-learning fuzzy c-means clustering algorithm with unknown number of clusters. Pattern Recogn 2017;71:45–59.
[18] Mclachlan GJ, Basford KE. Mixture models: inference and applications to clustering (M. Dekker); 1988.
[19] Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm. J R Stat Soc 1977;39(1):1–22.
[20] Yang MS, Lai CY, Lin CY. A robust EM clustering algorithm for Gaussian mixture models. Pattern Recogn 2012;45(11):3950–61.
[21] Sibson M. SLINK: An optimally efficient algorithm for the single-link cluster method. Comput J 1973;16(1):30–4.
[22] Renato C, Makarenkov V, Mirkin B. A-Ward_pβ) Effective hierarchical clustering using the Minkowski metric and a fast k-means initialisation. Inf Sci 2016;370:343–54.
[23] Wu XD, Kumar V, Quinlan JR, Ghosh J, Yang Q, Motoda H, Mclachlan GJ, Ng A, Liu B, Yu PS, Zhou ZH, Steinbach M, Hand DJ, Steinberg D. Top 10 algorithms in data mining. Knowl Inf Syst 2008;14(1):1–37.
[24] Mittal M, Sharma RK, Singh VP. Modified single pass clustering with variable threshold approach. Int J Innov Comput Inf Control 2015;11(1):375–386.
[25] MacQueen JB. Some methods for classification and analysis of multivariate observations. In: Proceedings of the 5th Berkeley Symposium on Mathematics Statistics and Probability. p. 17.
[26] Park HS, Jun CH. A simple and fast algorithm for K-medoids clustering. Expert Syst Appl 2009;36(2):3336–41.
[27] Bezdek JC. Pattern Recognition With Fuzzy Objective Function Algorithms. Plenum Press; 1981.
[28] Bezdek JC, Ehrlich R, Full W. FCM: The fuzzy c-means clustering algorithm. Comput Geosci 1984;10(2):191–203.
[29] Pelleg D, Moore A. X-means: Extending K-means with Efficient Estimation of the Number of Clusters. Mach Learn 2002:727–34.
[30] Taiyun K, Chen IR, Lin Y, Wang YY, Yang J, Yang P. Impact of similarity metrics on single-cell rna-seq data clustering. Briefings Bioinformatics 2019;20 (6):2316–26.
[31] Liu YC, Li ZM, Xiong H, Gao XD, Wu JJ. Understanding of Internal Clustering Validation Measures. 2010 IEEE International Conference on Data Mining 2010:911–6.
[32] Zhou K, Yang S, Ding S, Luo H. On cluster validation. Syst Eng-Theory Practice 2014;34(9):2417–31.
[33] Davies DL, Bouldin DW. A Cluster Separation Measure. IEEE Trans Pattern Anal Mach Intell 1979;PAMI-1(2):224–7.
[34] Rousseeuw PJ. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 1987;20:13.
[35] Caliński T, Harabasz J. A dendrite method for cluster analysis. Commun Stat 1974;3(1):1–27.
[36] Tibshirani R, Hastie WT. Estimating the Number of Clusters in a Data Set via the Gap Statistic. J R Stat Soc Ser B 2001;63(2):411–23.
[37] Milligan GW, Cooper MC. An examination of procedures for determining the number of clusters in a data set. Psychometrika 1985;50(2):159–79.
[38] Chiang MT, Mirkin B. Intelligent Choice of the Number of Clusters in K-Means Clustering: An Experimental Study with Different Cluster Spreads. J Classification 2010;27(1):3–40.
[39] Hussain SF, Haris M. A k-means based co-clustering (kCC) algorithm for sparse, high dimensional data. Expert Syst Appl 2019;118(15):20–34.
[40] Celebi ME, Kingravi HA, Vela PA. A comparative study of efficient initialization methods for the k-means clustering algorithm. Expert Syst Appl 2013;40:200–10.
[41] Jain AK. Data clustering: 50 years beyond K-means. Pattern Recogn Lett 2010;31(8):651–66.
[42] Bousidis C, Zouzias A, Mahoney MW, Drineas P. Randomized dimensionality reduction for k-means cluster. IEEE Trans Inf Theory 2015;61:1045–62.
[43] Karmitsa N, Bagirov AM, Taheri S. Clustering in large data sets with the limited memory bundle method. Pattern Recogn 2018;83:245–59.
[44] Capo M, Perez A, Lozano JA. An efficient approximation to the k-means clustering for massive data. Knowl-Based Syst 2017;117:56–69.
[45] Bicego M, Figueiredo MAT. Clustering via binary embedding. Pattern Recogn 2018;83:52–63.
[46] Lloyd SP. Least squares quantization in PCM. IEEE Trans Inf Theory 1982;28 (2):129–37.
[47] Rodriguez A, Laio A. Clustering by fast search and find of density peaks. Science 2014;344(6191):1492.
[48] Gonzalez T. Clustering to minimize the maximum intercluster distance. Theor Comput Sci 1985;38(2–3):293–306.
[49] Arthur D, Vassilvitskii S. K-Means++: The Advantages of Careful Seeding. Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms: 1027–1035; 2007.
[50] Hatamlou A. In search of optimal centroids on data clustering using a binary search algorithm. Pattern Recogn Lett 2012;33(13):1756–60.
[51] Kumar Y, Sahoo G. A New Initialization Method to Originate Initial Cluster Centers for K-Means Algorithm. Int J Adv Sci Technol 2014;62:43–54.
[52] Jiang F, Liu GZ, Du JW, Sui YF. Initialization of K-modes Clustering Using Outlier Detection Techniques. Inf Sci 2016;332:167–83.
[53] Kumar KM, Reddy ARM. An Efficient k-Means Clustering Filtering Algorithm Using Density Based Initial Cluster Centers. Inf Sci 2017;418:286–301.
[54] Mittal M, Sharma RK, Singh VP, Kumar R. Adaptive threshold based clustering: a deterministic partitioning approach. Int J Inf Syst Model Design 2019;10 (1):42–59.
[55] Fränti P, Sieranoja S. How much k-means can be improved by using better initialization and repeats? Pattern Recogn 2019;93:95–112.
[56] Wu KL, Yang MS. Alternative c-means clustering algorithms. Pattern Recogn 2002;35(10):2267–78.
[57] Melnykov I, Melnykov V. On K-means algorithm with the use of Mahalanobis distances. Stat Prob Lett 2014;84(1):88–95.
[58] Amorim RCd, Mirkin B. Minkowski metric, feature weighting and anomalous cluster initializing in K-Means clustering. Pattern Recogn 2012;45 (3):1061–75.
[59] Chakraborty S, Das S. k-Means clustering with a new divergence-based distance metric: Convergence and performance analysis. Pattern Recogn Lett 2017;100:67–73.
[60] Meng YF, Liang JY, Cao FY, He YJ. A new distance with derivative information for functional k-means clustering algorithm. Inf Sci 2018;463–464:166–85.
[61] Arima C, Hakamada K, Okamoto M, Hanai T. Modified Fuzzy Gap statistic for estimating preferable number of clusters in Fuzzy k-means clustering. J Biosci Bioeng 2008;105(3):273–81.
[62] Mur A, Dormido R, Duro N, Dormido-Canto S, Vega J. Determination of the optimal number of clusters using a spectral clustering optimization. Expert Syst Appl 2016;65:304–14.
[63] Zhang YQ, Mańdziuk J, Quek CH, Goh BW. Curvature-based method for determining the number of clusters. Inf Sci 2017;415–416:414–28.
[64] Gupta A, Datta S, Das S. Fast automatic estimation of the number of clusters from the minimum inter-center distance for k-means clustering. Pattern Recogn Lett 2018;116:72–9.
[65] James B, James KL. Tests for a Changepoint. Biometrika 1987;74(1):71–83.
[66] Akaike H. A new look at the statistical model identification. Autom Control IEEE Trans 1974;19(6):716–23.
[67] Fulekar MH. Bioinformatics: Applications in Life and Environmental Sciences. Netherlands: Springer; 2009.
[68] Zhang J, Cao Y, Zhang YM. Comparison of cluster analysis methods for gene expression profile. J Nanjing Agric Univ 2014;37(6):1–6.
[69] Qi R, Ma A, Ma Q, Zou Q. Clustering and classification methods for single-cell RNA-sequencing data. Briefings Bioinformatics 2019;4(4):1–13.
[70] Vladimir YK, Tallulash SA, Martin H. Challenges in unsupervised clustering of single-cell RNA-seq data. Nat Rev Genet 2019;20:273–82.
[71] Li Y, Dai Z, Cao D, Luo F, Chen Y, Yuan Z. Chi-mic-share: a new feature selection algorithm for quantitative structure-activity relationship models. RSC Adv 2020;10:19852–60.
[72] Chen Y, Wang L, Li L, Zhang H, Yuan Z. Informative gene selection and the direct classification of tumors based on relative simplicity. BMC Bioinf 2016;17(1):1–16.
[73] Sun C, Dai Z, Zhang H, Li L Yuan Z. Binary Matrix Shuffling Filter for Feature Selection in Neuronal Morphology Classification. Computational and Mathematical Methods in Medicine 2015;(2015-3-29):626975.

Zilan Ning is now pursuing his Ph.D. in Bioinformatics at Hunan Agricultural University, Changsha, China. Her research interest areas are data understanding and algorithm development in pattern recognition and unsupervised machine learning, and their applications to biological big data.

Jin Chen is now pursuing his master's degree in Bioinformatics at Hunan Agricultural University, Changsha, China.

Jianjun Huang is now pursuing master's degree in Bioinformatics at Hunan Agricultural University, Changsha, China.

Umar JIbrilla Sabo is now pursuing his Ph.D. in Bioinformatics at Hunan Agricultural University, Changsha,China.

Zheming Yuan received the Ph.D. degree in agroecology in 2000 from Zhejiang University, China. He is currently a professor at the Department of Bioinformatics, Hunan Agricultural University. He has been the Principal Investigator of bioinformatics research center and the director of Hunan Engineering & Technology Research Center for Agricultural Big Data Analysis & Decision-making, China.

He has authored over 130 scientific papers and has also served as reviewer of many international journals and conferences. His current research interests include clustering algorithms, correlation and predictability of bivariate in complex network of big data, and development of novel classifier.

Zhijun Dai received the Ph.D. degree in Bioinformatics in 2014 from Hunan Agricultural University, Changsha, China. He is currently an assistant professor in the Hunan Engineering & Technology Research Centre for Agricultural Big Data Analysis & Decision-making, Hunan Agricultural University, China. With main research interests in feature extraction, dimension reduction, support vector classification & regression, and their applications to biological big data.