Journal of Advances in Mathematics and Computer Science

30(2): 1-12, 2019; Article no.JAMCS.45837
ISSN: 2456-9968
(Past name: British Journal of Mathematics & Computer Science, Past ISSN: 2231-0851)

Investigating K-means and Kernel K-means Algorithms with Internal Validity Indices for Cluster Identification

Alissar Nasser[1*]

[1]*Faculty of Economic Sciences and Business Administration, Lebanese University, Hadath, Lebanon.*

Author's contribution

The sole author designed, analysed, interpreted and prepared the manuscript.

Article Information

DOI: 10.9734/JAMCS/2019/45837
Editor(s):
(1) Dr. Kai-Long Hsiao, Associate Professor, Taiwan Shoufu University, Taiwan.
Reviewers:
(1) Sameena Naaz, Computer Science and Engineering, Jamia Hamdard, India.
(2) Ohad BarSimanTov, Binghamton University, USA.
Complete Peer review History: http://www.sdiarticle3.com/review-history/45837

Original Research Article

Received: 01 October 2018
Accepted: 23 December 2018
Published: 15 January 2019

Abstract

Clustering is an unsupervised method where the number of clusters is not known by users. Therefore, the outcomes of a clustering algorithm depend on the input number of clusters specified by users. Consequently it is very important to evaluate the result of the clustering algorithms according to the number of clusters and choose the one that optimize a certain criterion. We present in this paper several clustering validity indices used in the literature. Using several synthetic and real datasets, these indices are then compared based on clustering results provided by the well known k-means clustering algorithm and its non-linear version the kernel K-means algorithm. The results showed that none of the validity indices is superior to the others; in the other hand, the kernel k-means failed to improve clustering accuracy of the dataset from the number of clusters perspective.

Keywords: Data mining; clustering algorithms; internal indices; validity indices; k-means; kernel k-means.

1 Introduction

Unsupervised clustering is the process of categorizing objects that share some characteristics into different clusters. So that objects in the same cluster are close to each other and those in different clusters are faraway. Clustering is the subject of active research in several fields such as statistics, pattern recognition, machine learning and market segmentation and others.

**Corresponding author: E-mail: alissar.nasser@gmail.com, alissar.nasser.1@ul.gov.lb;*

Most of the unsupervised clustering algorithms require that the number of classes K be defined a priori. It is obvious that the quality of the result of the clustering depends largely on the number of classes K value. For this purpose, several algorithms have been proposed. A classic approach is to consider the hypothesis that the data come from a mixture of densities [1,2,3]. Roberts et al. [4] show that the minimizing the entropy of partitions generated by the kernel functions can be used to estimate the number of classes. The algorithm Expectation - maximization (EM) is used to model the probability density by Kernel functions. This approach then looks for the number of partitions whose linear combination gives the densities.

The methods of projecting data can be advantageously used to discover the number of classes [5]. In [6] a self-organized network of Kohonen (self-organizing feature map) is used. The final structure of the network after learning allows the visualization of data of high dimensions. This visualization results in a straightforward analysis of the inherent structure of classes in data. Another approach is to observe the projections obtained by the Kernel Principal Components Analysis (KPCA) method [7]. The visualization of data by KPCA provides the user with a means to examine the presence of clusters in data, and by then initialize an automatic classification algorithm like the K-means algorithm.

Noting that the visualization is effective only if the inter-points distances of the projected data is a fairly accurate picture of the inter-point distances of data in multidimensional input space and the number of classes is small with slight overlapping between classes.

In the literature, there are three clustering structures; Hierarchical clustering, partitional clustering structures as well as evaluating the clusters themselves. There is distinction between different criteria for evaluating clustering analysis; the criteria are divided into external, internal and relative indices. The distinction between external and internal indices of cluster validity has often been confused in the literature; an external index evaluates a clustering structure in terms of priori information, such as category labels which have been assigned without reference to the pattern or proximity matrix. An internal index, on the other hand, uses only the proximity matrix itself and information from the cluster analysis. A relative index compares two statistics or clustering methods.

The difficulty with internal indices is their dependence on problem parameters, such as the number of patterns, features and clusters. Square error, for example, naturally decreases as the number of clusters increases, and increase as the number of patterns and the number of features increase [8].

In this paper, we are trying to find out if any of the internal validity indices is suitable to evaluate the clustering capacity of an algorithm and if they can be used as a metric to determine the underlying number of clusters in the datasets. For this purpose, we present six different internal validity indices and two clustering algorithms, namely the K-means algorithm and the non-linear version of it the kernel K-means.

The motivation of the paper comes from the fact that many research papers that exist in the literature compare some chosen validity indices, but none compare the indices with different clustering algorithms.

In the next we will present the clustering validity indices in section 2, and then in section 3 we will present the K-means clustering algorithm and the Kernel K-means algorithm, in section 4 we show the experimental results and conclude in the last section.

2 Clustering Validity Indices

We consider a partition of points in K classes C1, C2 ... CK obtained by an unsupervised clustering algorithm. This section describes the measures called "Internal measures" of validity of unsupervised classification methods. These indices are of two types: parametric and non-parametric. From non-parametric indices, we cite the compactness and isolation indices. Compactness and isolation indices are two internal indices that evaluate the individual clusters; they are defined as the number of edges internal to the cluster and the number of edges linking the cluster to the other clusters at each level of proximity. A valid cluster should have large compactness and small isolation index [8].

Another index is the connectivity proposed by Pauwels and Frederix [8] is based on the assumption that for all pairs of points assigned to the same class for a given partition, the same density must exist along the path connecting these pairs.

In the other hand, parametric evaluation measures assume that the data has a known distribution, for instance Gaussian distribution, with a good interclass separation. The Sum of squares errors is an example of theses parametric indices.

In the following we will present six internal validity indices.

2.1 The sum of squared errors index

This is the simplest validation index. For a partition of K classes, this criterion is calculated as the log of the ratio of within-class dispersion and the between-class dispersion:

$$SS = \log\left(\frac{\sum_{i=1}^{K}\sum_{X\,\epsilon C_i}(X-g_i)^2}{\sum_{i=1}^{K}(G-g_i)^2}\right)$$

Where g_i is the mean or center of the points of class C_i and G is the mean of the centers of the K classes. This index divides compactness by separation so a lower score indicates better clustering.

2.2 The trace index

This trace index introduced by Edwards and Cavalli-Sforza [9], it is used to measure the compactness of the classes for a given partition. We define the within-class dispersion D_w which measures how much the points are scattered around the average c_i of each class:

$$D_w = trace\left(\sum_{i=1}^{K}\sum_{X\,\epsilon C_i}(X-c_i)(X-c_i)^T\right)$$

The best value of the index for different clusters number K is the one that correspond to the greatest difference between two successive slopes.

2.3 The dunn index

The Dunn index introduced by J. C. Dunn in 1974 [10] is a metric for evaluating clustering algorithm. It is an internal index that is the result is based on the clustered data itself. The aim is to identify sets of clusters that are compact, with a small variance between members of the cluster, and well separated, where the means of different clusters are sufficiently far apart, as compared to the within cluster variance.

The Dunn index is defined by:

$$\text{Dunn} = \min_{1\le i\le K}\left\{\min_{1\le j\le K\, i\neq j}\left\{\frac{d_{ij}}{\max_{1\le l\le K}\{S_l\}}\right\}\right\}$$

Where $d_{ij} = \|c_i - c_j\|^2$ the inter-class distance between the centers of is classes i and j, and S_l is the dispersion of the class l given by the equation:

$$S_l = \frac{1}{|C_l|}\sum_{X\epsilon C_l}(X-c_l)^2$$

| Cl | is the cardinal of Cl. For a given assignment of clusters, a higher Dunn index indicates better clustering. One of the drawbacks of using this is the computational cost as the number of clusters and dimensionality of the data increase.

2.4 The davies and bouldin index

This index was introduced by Davies and Bouldin in 1979 [11]. It was originally proposed as a way of deciding when to stop clustering data. The index is plotted against the number of clusters and clustering is stopped when the index is minimized. It was claimed that the index does not depends on the number of clusters or the clustering method.

Given a partition of n objects into K clusters, one first defines the following measure of within-to-between clusters spread for all pairs of clusters (i, k)

$$R_{j,k} = \frac{S_k + S_i}{d_{ik}}$$

Where

$$S_i = \frac{1}{|C_i|}\sum_{X \epsilon C_i}(X - c_i)^2$$

The R_k index for the kth cluster is:

$$R_k = \max_{i \neq k}\{R_{j,k}\}$$

The Davies and Bouldin (DB) index for K classes is calculated as follows:

$$DB = \frac{1}{K}\sum_{k=1}^{K} R_k \qquad \text{For } K > 1$$

S_k is the within-class dispersion of the kth class and d_{ik} is the distance between the two centers ci and ck $d_{ik} = (c_i - c_k)^2$.

The optimum number of classes is the one that minimizes the DB index. This index is equal to zero for the trivial clustering that places each object in an individual cluster. In addition, each cluster should contain a reasonable number of objects. It is not defined for the case when all objects are in the same cluster. One plot DB against K for successive values of K and chooses that partition that minimizes the index.

2.5 Calinski–Harabasz index (CH)

The CH index introduced by Calinski and Harabasz in 1974 [12]. The method is based on a relationship between a "between cluster scatter matrix" and a "within cluster scatter matrix":

$$CH = \frac{\sum_{i=1}^{K}(c - c_i)^2}{\sum_{i=1}^{K}\sum_{X \epsilon C_i}(X - c_i)^2} * \frac{N - K}{K - 1}$$

Like DB, CH uses cluster centers for calculating separation; however, separation is measured according to the center of the data set, rather than particular clusters. The normalization factor, (N-k)/(K-1), will diminish the score as k increases.

2.6 PBM index

The PBM index is a more recent index introduced by Bandyopadhyay et al. [13]. The PBM score is a more recent index, it relates a figure of compactness, measured as the sum of the distances between each point and

its cluster center, to a measure of separation, calculated as the maximum distance between any two cluster centers, normalized over a measure of dispersion, calculated as the sum of the distances between all points:

$$(\frac{1}{K} * \frac{E_w}{E_T} * D_B)^2$$

Where $E_w = \sum_{i=1}^{K} \sum_{X \epsilon C_i} (X - c_i)^2)$ is the sum of the distances between each point and its cluster center, $E_T = \sum_{i=1}^{N} (X_i - C)^2$ is the sum of the distances between each point and the center of the data set, and D_B is the larger distance between the cluster centers. Separation is measured by the greatest distance between any two clusters, so this measure favors clustering that have at least two well-separated clusters. The larger the value of the index the better the result of clustering.

3 Clustering Algorithms

3.1 K-means clustering

K-means clustering [14] is the most commonly used unsupervised machine learning algorithm for partitioning a given data set into a set of k groups (i.e. *K clusters*), where k represents the number of groups pre-specified by the analyst. It classifies objects in multiple clusters such that objects within the same cluster are as similar as possible whereas objects from different clusters are as dissimilar as possible. In k-means clustering, each cluster is represented by its center which corresponds to the mean of points assigned to the cluster.

The K-means algorithm

1. Clusters the data into *K* groups where *K* is predefined.
2. Select *K* points at random as cluster centers.
3. Assign objects to their closest cluster center according to the *Euclidean distance* function or any other distance.
4. Calculate the center or mean of all objects in each cluster.
5. Repeat steps 2, 3 and 4 until the same points are assigned to each cluster in consecutive rounds

K-Means is relatively an efficient method. However, we need to specify the number of clusters, in advance and the final results are sensitive to initialization and often terminates at a local optimum. Unfortunately there is no global theoretical method to find the optimal number of clusters. A practical approach is to compare the outcomes of multiple runs with different *K* and choose the best one based on a predefined criterion. In general, a large K probably decreases the error but increases the risk of overfitting.

3.2 Kernel K-means

Kernel K-means [15] utilizes kernel trick to perform operation in a new feature space F where data samples are more separable. By using a nonlinear kernel function instead of the standard dot product, we implicitly perform K-means in a high-dimensional space F which is non-linearly related to the input space.

The objective function MSE (mean squared error) in the feature space is given by:

$$MSE^{\phi} = \sum_{k=1}^{K} \sum_{x_j \in C_k} \left\| \phi(x_i) - c_k^{\phi} \right\|^2$$

Where c_k^{ϕ} is the centre of the k-th class in the feature space:

$$c_k^{\phi} = \frac{1}{\left|C_k^{\phi}\right|} \sum_{x_j \in C_k} \left\| \phi(x_j) - c_k^{\phi} \right\|^2$$

Note that c_k^{ϕ} is the best representation of a class since:

$$c_k^{\phi} = \arg\min_z \sum_{X \in C_k} \left\| \phi(x) - z \right\|^2 .$$

The Euclidian distance $\phi(x)$ to the centre c_k^{ϕ} is given by:

$$\left\| \phi(x) - c_k^{\phi} \right\|^2 = \phi(x).\phi(x)^T - \frac{2}{\left|C_k\right|} \sum_{a \in C_k} \phi(a).\phi(x)^T + \frac{1}{\left|C_k\right|^2} \sum_{a \in C_k} \sum_{b \in C_k} \phi(a).\phi(b)^T$$

$$\left\| \phi(x) - c_k^{\phi} \right\|^2 = K(x,x) \quad - \frac{2}{\left|C_k\right|} \sum_{a \in C_k} K(a,x) \qquad + \frac{1}{\left|C_k\right|^2} \sum_{a \in C_k} \sum_{b \in C_k} K(a,b)$$

4 Experimentations

In order to compare the six validity measures, we used some synthetic and real datasets. Table 1 summarizes the characteristics of the datasets.

Table 1. Datasets

Dataset	Number of attributes	Number of observation	Number of clusters
Four-gauss dataset	2	100	4
Saturn dataset	2	200	2
Half-ring dataset	2	520	2
Spiral dataset	2	200	2
xclara (artificial)	2	3000	3
Iris (real)	4	150	3
Wine(real)	13	178	3
Energy efficiency(real)	8	768	5

The following graphs show the plot of the data set into 2 dimensions. The four gauss dataset (a) is an easy example for clustering. The datasets Saturn (b), Halfring (c), and Spirals (d) are non-linearly separable and considered as difficult examples for clustering. The iris (e) and xclara (f) datasets are in 3 dimensions dataset but the graphs were plotted using only the first 2 dimensions.

We used in our experimentations, R, the open source programming language for data analysis. First, we calculated the internal validity indices on clustering results using k-means algorithm for several values of K ranging from 2 to 6. Then we used the results of the Kernel K-means with Gaussian kernel function. The value of sigma is estimated automatically from the datasets themselves.

For each dataset, k-means, and kernel k-means were run successively and then the validity indices were obtained and the best value of the index is calculated using the "best" function is R. note that for some

indices the larger the value, the better the index and for some others, the smaller the value, the better the index whereas for others using the first max or first min is considered as the optimal value.

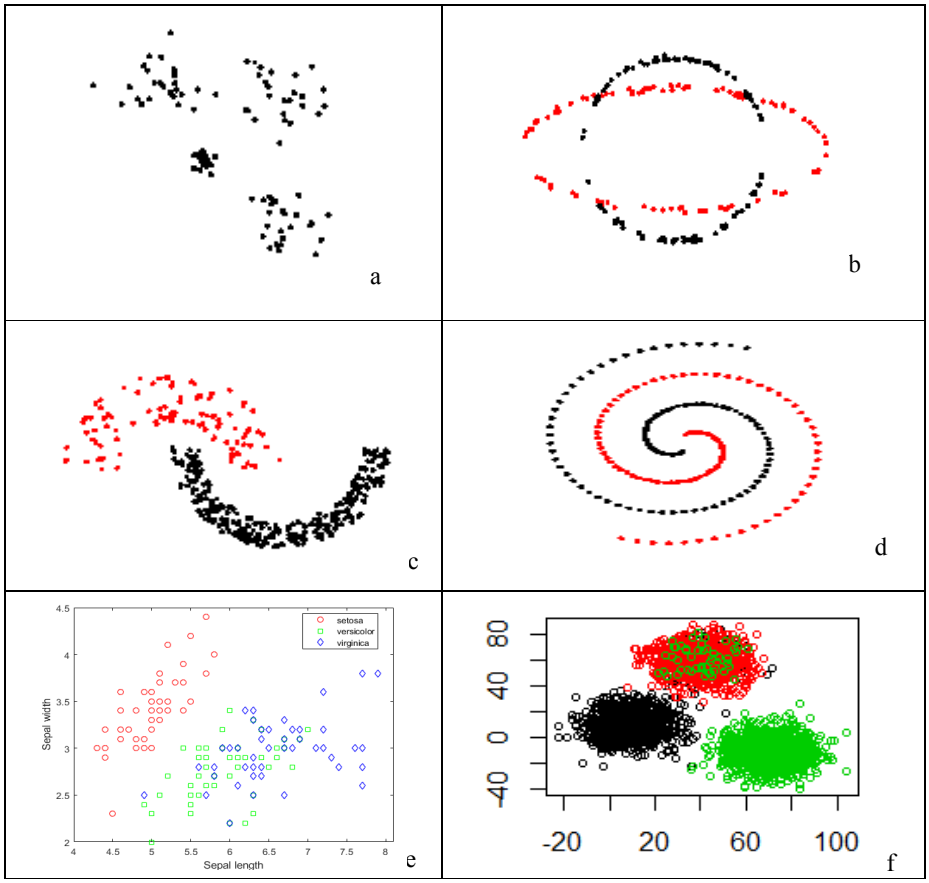

Fig. 1. Data plot of synthetic and real datasets

5 Results for K-means Algorithm

Tables 2 to 9 below summarize the values of the six validity indices compile on K-means results with the gray one represent the best value of the index (Using the "best function in R as noted before).

Table 2. Results of the four Gauss dataset (Expected K = 4 clusters)

Four gauss					
K	**2**	**3**	**4**	**5**	**6**
Index SS	0.0299	0.9835	**2.3565**	2.5345	2.7209
Index trace	961.6692	531.4653	**168.9932**	143.4567	120.5744
Index CH	100.9723	129.6800	**337.7199**	299.4973	285.6351
Index Dunn	0.2013	0.1902	**0.2920**	0.1372	0.1372
Index DB	0.6192	0.7025	**0.3862**	0.5985	0.7597
Index PBM	28.1773	27.0204	**100.1433**	92.6491	74.4333

Source: R programming, assembled by author

Table 3. Results of the Saturn dataset (Expected K = 2 clusters)

Saturn					
K	**2**	**3**	**4**	**5**	**6**
Index SS	-0.537	**0.728**	1.377	1.761	2.163
Index trace	1275.216	**657.678**	407.074	296.257	208.369
Index CH	115.685	204.077	258.913	283.694	**337.394**
Index Dunn	0.023	0.034	**0.053**	0.031	0.029
Index DB	1.123	0.719	0.666	0.647	**0.625**
Index PBM	7.590	8.877	14.606	13.939	**16.978**

Source: R programming, assembled by author

Table 4. Results of the Half-ring dataset (Expected K =2 clusters)

Half -ring					
K	**2**	**3**	**4**	**5**	**6**
Index SS	-0.121	0.916	**2.545**	2.638	2.816
Index trace	131.654	70.971	**18.062**	16.577	14.022
Index CH	459.059	645.994	**2192.744**	1799.920	1717.811
Index Dunn	0.029	0.044	**0.053**	0.041	0.020
Index DB	1.023	0.611	**0.340**	0.656	0.686
Index PBM	0.443	0.586	**2.085**	1.796	1.270

Source: R programming, assembled by author

Table 5. Results of the Spirals dataset (Expected K =2 clusters)

Spiral					
K	**2**	**3**	**4**	**5**	**6**
Index SS	-0.558	**0.232**	0.548	0.969	1.276
Index trace	2114.117	**1470.643**	1218.217	914.690	725.376
Index CH	113.370	124.174	112.966	128.441	**139.032**
Index Dunn	0.013	0.018	0.019	0.019	**0.030**
Index DB	1.198	1.003	1.034	0.930	**0.794**
Index PBM	9.312	7.949	**11.180**	10.257	10.334

Source: R programming, assembled by author

Table 6. Results of Xclara dataset (Expected K =3 clusters)

K	**2**	**3**	**4**	**5**	**6**
Index SS	0.16	**1.98**	2.09	2.21	2.37
Index trace	2309985.00	**611605.90**	555336.50	498587.60	428338.60
Index CH	3530.72	**10826.60**	8047.61	6805.66	6433.54
Index Dunn	0.02	**0.047**	0.01	0.01	0.00
Index DB	0.68	**0.421**	0.85	0.73	0.95
Index PBM	2616.52	**6196.97**	3906.09	3399.77	2.67
Index K-DB					

Source: R programming, assembled by author

Table 7. Results of Iris data set (Expected K =3 clusters)

K	**2**	**3**	**4**	**5**	**6**
Index SS	1.245	2.034	2.144	**2.615**	2.195
Index trace	152.348	**78.851**	71.445	46.446	68.266
Index CH	513.925	**561.628**	415.465	495.541	258.655
Index Dunn	0.077	**0.099**	0.053	0.082	0.053
Index DB	**0.404**	0.662	0.859	0.806	0.956
Index PBM	19.923	**25.175**	17.296	19.118	9.174

Source: R programming, assembled by author

Table 8. Results of Wine dataset (Expected K =3 clusters)

K	2	3	4	5	6
Index SS	0.28	1.00	**1.46**	1.58	1.78
Index trace	17105.15	**10664.58**	7490.12	6801.78	5732.03
Index CH	233.38	238.94	**250.10**	209.74	204.38
Index Dunn	0.04	0.04	**0.08**	0.05	0.06
Index DB	**0.72**	0.76	0.75	0.87	1.00
Index PBM	317.97	548.07	**854.72**	666.85	545.04

Source: R programming, assembled by author

Table 9. Results of energy efficiency dataset (Expected K =5 clusters)

K	2	3	4	5	6
Index SS	0.7	1.2	1.4	**2.5**	1.5
Index trace	3000870.0	2100334.0	1848389.0	**660693.0**	1694725.0
Index CH	1526.5	1253.1	982.7	**2402.1**	655.2
Index Dunn	**0.7**	0.5	0.2	0.4	0.2
Index DB	0.6	0.5	0.5	0.6	**0.4**
Index PBM	27227.0	20539.6	19795.4	**44325.2**	13473.1

Source: R programming, assembled by author

It can be seen that trace index is the best predictor, which is followed by CH and PBM indices. It is normal result as the k-means clustering algorithm is based on the concept of minimizing the sum of distances to the centers of the clusters which is the similar to the trace index.

6 Results for Kernel k-means

The Tables 10 to 17 below summarize the values of the indices obtained from the kernel K-means.

Table 10. Results of four Gauss dataset (Expected K =4 clusters)

Four Gauss					
K	**2**	**3**	**4**	**5**	**6**
Index SS	2.542	1.246	**1.722**	0.931	-0.984
Index trace	142.487	436.320	296.018	**552.191**	1421.159
Index CH	**238.818**	65.329	105.203	47.675	7.029
Index Dunn	**0.1102**	0.0131	0.0226	0.0256	0.0140
Index DB	**0.798**	1.421	2.224	2.092	6.557
Index PBM	**74.731**	33.931	34.366	21.136	2.623

Source: R programming, assembled by author

Table 11. Results of Saturn dataset (Expected K =2 clusters)

Saturn					
K	**2**	**3**	**4**	**5**	**6**
Index SS	0.649	**2.011**	1.788	0.471	1.925
Index trace	693.233	**238.421**	289.650	776.755	257.098
Index CH	74.275	**289.975**	231.826	62.116	266.091
Index Dunn	0.007	0.018	**0.022**	0.008	0.008
Index DB	2.000	0.670	**0.655**	1.797	0.662
Index PBM	5.416	**16.736**	14.084	3.586	14.273

Source: R programming, assembled by author

Table 12. Results of Half-ring dataset (Expected K =2 clusters)

Half ring					
K	**2**	**3**	**4**	**5**	**6**
Index SS	1.755	**2.077**	1.611	1.915	2.130
Index trace	36.604	**27.662**	41.321	31.898	26.377
Index CH	594.615	820.067	515.007	697.499	**865.012**
Index Dunn	0.0008	0.0057	0.0012	**0.0060**	0.0024
Index DB	1.125	1.176	1.285	**1.107**	1.618
Index PBM	0.758	**0.876**	0.627	0.868	0.868

Source: R programming, assembled by author

Table 13. Results of Spirals dataset (Expected K =2 clusters)

Spirals					
K	**2**	**3**	**4**	**5**	**6**
Index SS	0.849	0.863	0.192	**1.044**	0.036
Index trace	996.194	986.145	1503.234	**865.342**	1632.078
Index CH	90.688	92.007	47.012	**110.268**	40.237
Index Dunn	**0.013**	0.011	0.011	0.010	0.013
Index DB	1.102	0.989	2.641	**0.964**	2.548
Index PBM	6.487	8.059	3.070	**8.744**	5.425

Source: R programming, assembled by author

Table 14. Results of Iris dataset (Expected K =3 clusters)

Iris					
K	**2**	**3**	**4**	**5**	**6**
Index SS	2.614	**2.675**	1.841	2.400	2.583
Index trace	46.481	**43.941**	93.314	56.694	47.865
Index CH	393.384	**417.791**	181.495	317.329	381.177
Index Dunn	**0.068**	0.049	0.029	0.064	0.062
Index DB	**1.031**	1.072	1.619	1.685	1.088
Index PBM	12.060	**13.064**	7.437	11.191	11.362

Source: R programming, assembled by author

Table 15. Results of Wine dataset (Expected K =3 clusters)

Wine					
K	**2**	**3**	**4**	**5**	**6**
Index SS	0.567	0.507	0.285	**0.425**	0.216
Index trace	14398.340	14953.830	17074.730	**15731.360**	17755.310
Index CH	**60.659**	57.128	45.759	52.604	42.686
Index Dunn	0.020	0.023	0.017	0.020	0.012
Index DB	2.675	2.091	5.201	**2.054**	4.052
Index PBM	**159.081**	135.848	98.772	99.815	121.532

Source: R programming, assembled by author

Table 16. Results of energy efficiency dataset (Expected K =5 clusters)

EE					
K	**2**	**3**	**4**	**5**	**6**
Index SS	-0.558	**1.917**	-0.140	-0.558	1.917
Index trace	5711166	**1151328**	4804138	5711166	1151328
Index CH	87.252	**1036.394**	132.499	87.252	1036.394
Index Dunn	0.000	0.001	**0.128**	0.000	0.001
Index DB	3.214	0.723	2.683	3.214	**0.723**
Index PBM	2309.552	**29342.090**	3015.490	2309.552	29342.090

Source: R programming, assembled by author

Table 17. Results of Xclara dataset (Expected K =3 clusters)

Xclara					
K	**2**	**3**	**4**	**5**	**6**
Index SS	1.108	**1.809**	1.288	2.269	1.108
Index trace	1249217.0	**708296.6**	1087407.0	471643.6	1249217.0
Index CH	1812.490	3653.971	2171.297	**5787.851**	1812.490
Index Dunn	0.0005	**0.0007**	0.0003	0.0004	0.0005
Index DB	1.902	1.579	1.706	**0.951**	1.902
Index PBM	1098.398	2079.610	1356.160	**2900.165**	1098.398

Source: R programming, assembled by author

It is obvious from the results above that the kernel K-means does not outperform the K-means in terms of providing the six validity indices better clustering distributions. One would expect the Kernel k-means provide better clustering results for the non-linear dataset like the spirals, Saturn and the half-ring. Looking at Table 18, it is clear the first K-means is better than the kernel K-means and second comparing the indices together, the trace index is the best criterion but it detect more correctly the number of clusters using K-means than when using kernel K-means.

Table 18. Correctly clusters per validity index

Validity index	Number of time correctly identified K		Percentage of correctly identified K	
	K-means	**Kernel k-means**	**K-means**	**Kernel k-means**
Index SS	3	3	50%	50%
Index trace	5	2	83.3%	33.3%
Index CH	4	1	66.6%	16.6%
Index Dunn	3	1	50%	16.6%
Index DB	2	0	33.3%	0%
Index PBM	4	1	66.6%	16.6%

Source: Assembled by author

7 Conclusion

When compared to the results of other comparative studies, we conclude that none of the validity indices is able to determine the correct number of clusters within the datasets. Although the Trace or CH indices are able to predict the number of clusters but this finding cannot be generalized to different datasets. In the other hand the kernel K-means algorithm failed to provide better results as it was expected especially for non-linear dataset. Which lead us to conclude that the problem of determining the number of clusters remain a fundamental problem when dealing with clustering algorithms that need to specify K.

8 Limitation of the Study

Although the K-means clustering algorithm is most popular and efficient algorithm because it is very simpler but some limitations in this algorithm makes it somewhat difficult. One of the limitations is the initiation of the centers of classes. To overcome this limitation we run the algorithm many times with different initial centers and kept the one that optimize the results.

Another limitation of our research is the use of the width of the Kernel function. Many alternatives exist in the literature to choose the optimal value of the width. But as we can conclude from this research there is no perfect solution of perfect parameters for all kinds of data.

Competing Interests

Author has declared that no competing interests exist.

References

[1] Jain AK, Duin R, Mao J. Statistical pattern recognition: A review. IEEE Trans. on Pattern Analysis and Machine Intelligence. 2000;22(1).

[2] Celeux G. Modèles probabilistes en classification, Chapitre 6 du livre Modèles pour l'analyse des données multidimensionnelles. édition Economica, J.-J. Droesbeke, B. Fichet, P. Tassi, éditeurs, 1992;165-214.

[3] Redner RA, Walker HF. Mixture densities, maximum likelihood and the EM algorithm. SIAM Review. 1984;26(2):195-239.

[4] Roberts SJ, Everson R, Rezek I. Maximum certainty data partitioning. Pattern Recognition. 1999;33(5):833-939.

[5] Jain AK, Mao J. Artificial neural network for nonlinear projection of multivariate data. In Proc. IEEE Int. Joint. Conf. on Neural Networks. Baltimore, Maryland. 1992;3:335-340.

[6] Su Mu-Chun, Chang Hsiao-Te. A new model of self-organizing neural networks and its application in data projection. IEEE Trans, Neural Network. 2001;12(1):153-158.

[7] Nasser A, Nasr C, Hamad D. Visualization methods for interactive classification. 2nd IEEE International Conference on Information & Communication Technologies: From Theory to Applications - ICTTA'06, Damascus, Syria April, 24-28; 2006.

[8] Jain AK, Dubes R. Algorithms for clustering data. Prentice-Hall, Englewood, Cliffs, NJ; 1988.

[9] Edwards AWF, Cavalli-Sforza L. A method for cluster analysis. Biometrika. 1965;56:362-375.

[10] Dunn JC. Well separated clusters and optimal fuzzy partitions. Journal of Cybernetica. 1974;4:95-104.

[11] Davies DL, Bouldin DW. A cluster separation measure. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-1. 1979;2:224-227.

[12] Calinski T, Harabasz J. A dendrite method for cluster analysis. Communications in Statistics. 1974;3(1):1-27.

[13] Bandyopadhyay S, Pakhira MK, Maulik U. Validity index for crisp and fuzzy clusters. Pattern Recognition. 2004;37:487-501.

[14] MacQueen JB. Some methods for classification and analysis of multivariate observations. Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability", Berkeley, University of California Press. 1967;1:281-297.

[15] Girolami M. Mercer Kernel based clustering in feature space. IEEE Transactions on Neural Networks. 2002;13(4):669–688.

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
http://www.sdiarticle3.com/review-history/45837