OPEN ACCESS
Vietnam Journal of Computer Science
Vol. 7, No. 4 (2020) 417–431

DOI: 10.1142/S2196888820500232

A Criterion for Deciding the Number of Clusters in a Dataset Based on Data Depth

Ishwar Baidari* and Channamma Patil†
Department of Computer Science, Karnatak University
Dharwad, Karnataka 580003, India
**ishwar_sp@yahoo.co.in*
†*channamma.shivu@gmail.com*

Received 18 November 2019
Accepted 4 May 2020
Published 8 July 2020

Clustering is a key method in unsupervised learning with various applications in data mining, pattern recognition and intelligent information processing. However, the number of groups to be formed, usually notated as k is a vital parameter for most of the existing clustering algorithms as their clustering results depend heavily on this parameter. The problem of finding the optimal k value is very challenging. This paper proposes a novel idea for finding the correct number of groups in a dataset based on data depth. The idea is to avoid the traditional process of running the clustering algorithm over a dataset for $\sqrt{n}$ times and further, finding the k value for a dataset without setting any specific search range for k parameter. We experiment with different indices, namely CH, KL, Silhouette, Gap, CSP and the proposed method on different real and synthetic datasets to estimate the correct number of groups in a dataset. The experimental results on real and synthetic datasets indicate good performance of the proposed method.

Keywords: Data depth; depth median; within-cluster depth; between-cluster depth; optimal k.

1. Introduction

Clustering is one of the key methods in unsupervised learning used to identify groups of similar objects in a dataset. Clustering is widely used in pattern recognition, data mining and intelligent information processing. However, the number of clusters to be formed, usually notated as k is a vital parameter for most of the existing clustering algorithms as their clustering results depend heavily on this parameter. Various internal validity indices have been presented in the literature to examine the results of clustering for estimating the optimal clusters in a dataset. The usual approach used by these internal indices is to run the clustering algorithm on a dataset with a

*Corresponding author.

different k values for each run and decide to use the one that maximizes or minimizes the index value. While validating the clustering results using the validity indices, usually the Euclidean distance measure is used and the search range for the clustering numbers is usually set to $k_{\min} = 2$ and $k_{\max} = \sqrt{n}$ according to the commonly used rule.[1,2] As the number of instances (n) in a dataset increase, the $k_{\max}$ value also increases. And iterating the clustering process for $k_{\max}$ times will be lengthy. Distance-based indices are sensitive to the inclusion of unrelated variables and can be affected by high variance clusters.

Therefore, this paper presents a simple method for deciding the number of clusters in a dataset using data depth. The proposed method avoids the traditional process of running the clustering algorithm over a dataset for $\sqrt{n}$ times and further, finds the optimal k value for a dataset without setting any specific search range for k parameter.

The proposed method works on the concept of average difference between within-cluster depth and between-cluster depth. The average difference between every object and the center of cluster k is the within-cluster depth and the minimum value of average difference between every object in cluster k and the center of every other cluster is the minimum between-cluster depth. The average difference between within-cluster depth and between-cluster depth characterizes the separateness of clusters. A small difference indicates less separateness between the clusters and a large difference indicates higher separateness between the clusters. Based on this concept, the method initializes k value to one initially and iteratively increments the value of k by one until the difference falls down. When the difference value of k clusters, drops from the previous $k-1$ clusters, the method stops and concludes $k-1$ as the optimal number of clusters.

The organization of the paper is as follows. The background and related work of internal validity indices and the data depth are presented briefly in Sec. 2. We present the proposed method in Sec. 3. Section 4 details the experimental results of the proposed algorithm and lastly Sec. 5 concludes this paper.

2. Background and Related Work

2.1. *Internal validity indices*

The quality of the clustering results can be deduced by using clustering validity indices. In general, validity indices are categorized as external and internal. While internal indices use the information of the partitioned data, external indices use external information such as labels. For finding the correct number of groups in a dataset, internal indices in conjunction with a clustering algorithm are commonly used. Based on the chosen validity index, either maximum or minimum index value is used to find the optimum k value.

In 1985, Milligan and Cooper[3] compared around 30 internal indices for estimating the correct k value and concluded with the top performers. Dimitriadou *et al.*[4] also proposed a comparison of 15 validity indices for binary datasets.

More studies using clustering validity indices can be found in Milligan and Cooper.[3] In this paper, we compare five internal validity indices, which are CH, KL, Silhouette, Gap and CSP due to best performers with respect to determining k value. We explain each index in the following Secs. 2.1.1–2.1.5.

2.1.1. *Calinski and Harabasz (CH) index*

In an experimental study, Milligan and Cooper[3] compared 30 different indices and the index proposed by Calinski and Harabasz[5] was the best among others. It is based on the average sum of squares between and within-clusters. The Calinski and Harabasz (CH) index[5] is given by

$$\mathrm{CH}(k) = \frac{B(k)/(k-1)}{W(k)/(n-k)}, \tag{1}$$

where k denotes the number of clusters, n denotes the number of instances, and $B(k)$ and $W(k)$ denote the between-cluster and within-cluster sum of squares of the partition, respectively. An optimal number of clusters is then defined as a value of k that maximizes $\mathrm{CH}(k)$. $\mathrm{CH}(k)$ is only defined for k greater than 1 since $B(k)$ is not defined when $k = 1$.

2.1.2. *Krzanowski and Lai (KL) index*

The KL index proposed by Krzanowski and Lai[6] considers two consecutive clusters and finds the traces of the within-clusters matrices. The difference between these two is the KL index value. The high KL index value indicates better clustering results. The KL index is defined as

$$\mathrm{KL}(k) = \left| \frac{\mathrm{DIFF}(k)}{\mathrm{DIFF}(k+1)} \right|, \tag{2}$$

where

- $\mathrm{DIFF}(k) = (k-1)^{2/p}\mathrm{trace}(W(k-1)) - k^{2/p}\mathrm{trace}(W(k))$,

and p denotes the number of features in the dataset and W is the sum of all point-to-point distances of the observations within each cluster. The value of k, maximizing $\mathrm{KL}(k)$, is regarded as specifying the optimal number of clusters. But $\mathrm{KL}(k)$ is not defined for $k = 1$.

2.1.3. *Silhouette index*

Kaufman and Rousseeuw[7] introduced the silhouette index which is constructed to show graphically how well each object is classified in a given clustering output. It is based on the concept of the closeness of a point within the cluster and in the

neighboring clusters. The silhouette width is given as

$$\text{Silhouette} = \frac{\sum_{i=1}^{n} S(i)}{n}, \text{Silhouette} \in [-1, 1], \tag{3}$$

where

- $S(i) = \frac{b(i)-a(i)}{\max\{a(i);b(i)\}}$,
- $a(i) = \frac{\sum_{j\in\{C_r/i\}} d_{ij}}{n_r - 1}$ is the average dissimilarity of the ith object to all other objects of cluster C_r,
- $b(i) = \min\{d_{iC_s}\}$,
- $d_{iC_s} = \frac{\sum_{j\in C_s} d_{ij}}{n_s}$ is the average dissimilarity of the ith object to all objects of cluster C_s

The maximum value of the index is used to determine the optimal number of clusters in the data. $S(i)$ is not defined for $k = 1$ (only one cluster).

2.1.4. *Gap index*

For estimating the optimal number of groups in a dataset Tibshirani *et al.*[8] proposed an approach called gap statistic. The approach is designed for all clustering techniques. The idea in this approach is to compute different clusterings of the data by increasing the number of clusters and compare to clusters of reference data (B) generated with a uniform distribution. The gap index is defined as

$$\text{Gap}(k) = \frac{1}{B}\sum_{b=1}^{B} \log W_{kb} - \log W_k, \tag{4}$$

where B is the number of reference data sets generated using uniform prescription[8] and W_{kb} is the within-dispersion matrix. The optimal number of clusters is chosen via finding the smallest k such that:

$$\text{Gap}(k) > \text{Gap}(k+1)s_{k+1}, (k = 1, \ldots, n-2),$$

where

- $s_k = \text{sd}_k\sqrt{1 + 1/B}$,
- sd_k is the standard deviation of $\log W_{kb}$, $\text{b} = 1, \ldots, \text{B} : \text{sd}_k = \sqrt{\frac{1}{B}\sum_{b=1}^{B}(\log W_{kb} - \bar{l})^2}$,
- $\bar{l} = \sum_{b=1}^{B} \log W_{kb}$

To apply this method, it is important to choose an appropriate reference null distribution.

2.1.5. *Compact-separate proportion (CSP) index*

CSP index is a new index proposed by Zhou *et al.*[9] in 2016. It can evaluate the clustering results produced by agglomerative hierarchical clustering (AHC)

algorithm and determine the optimal number of clusters for linear, manifold, annular, and convex datasets. The CSP index is defined as

$$\text{avgCSP}(k) = \frac{1}{k}\sum_{i=1}^{k}\text{CSP}(i), \tag{5}$$

where

- $\text{CSP}(i) = \frac{\text{sd}(i)-\text{cd}(i)}{\text{sd}(i)+\text{cd}(i)}$
- $\text{cd}(i) = \frac{W(G_i)}{n_i-1}$ is the intra-cluster compactness defined as the average weight of the minimum spanning tree for all samples in the ith cluster ($W(G_i)$).
- $\text{sd}(i) = \min_{1\le j\le c, j\neq i}\{\min\{\text{dist}(x_i, x_j)|x_i \in G_i, x_j \in G_j\}\}$ is the intercluster separation defined as the minimum value of minimum distances between the samples in cluster i and the samples in other clusters.

avgCSP(k) is the average CSP value in the case where data points are clustered into k clusters. The clustering number that corresponds to the maximum average value (avgCSP) is the optimal number of clusters.

2.2. *Data depth*

Data depth is a statistical function which calculates the deepness or the centrality of a point in a data cloud. Larger depth value of a point indicates that it is deeper with respect to the data cloud and lower value indicates its outlyingness. Data depth can also be used to get information about spread, shape, and symmetry of a dataset through depth regions.[10] Statistical depth functions provide an ordering of all points from the center outward in a multivariate dataset. Tukey[11] first introduced halfspace depth. Oja[12] defined Oja depth and several other depth functions which are proposed in the literature such as convex-hull peeling depth[13], simplicial depth[14], regression depth[15], L1 depth[16], and Mahalanobis depth.[17] The four desirable properties introduced by Liu *et al.*[14] for the depth functions are affine invariance, maximality at the center, monotonicity relative to the deepest point and vanishing at infinity. More details regarding the depth functions and their properties can be found in Zuo and Serfling.[17] Mahalanobis depth is one of the well-known depth functions that we use in our proposed method. It satisfies all the four properties of depth functions mentioned above. It is fast and easy to compute and we consider this depth function to be competitive.

2.2.1. *Mahalanobis depth*

The Mahalanobis depth (MD) function is proposed by Liu and Singh[18] and it is based on the well-known Mahalanobis distance defined by Mahalanobis.[19] MD of a point x with respect to a dataset X is defined as

$$\text{MD}(x|X) = [1 + (x-\bar{x})^T\text{Cov}(X)^{-1}(x-\bar{x})]^{-1}, \tag{6}$$

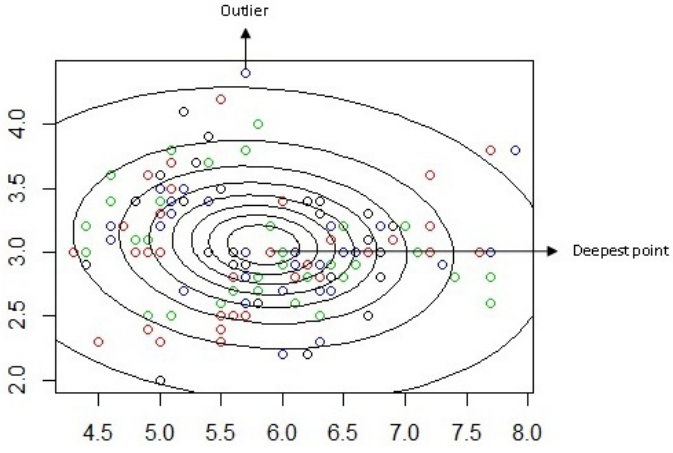


Fig. 1. Mahalanobis depth contours over Iris dataset.

where $\bar{x}$ and $\mathrm{Cov}(X)$ are the mean and covariance matrix of X, respectively. Maximum depth value point is the deepest point in the dataset, larger the value deeper will be the point, and the smaller value indicates the outlyingness, as shown in Fig. 1.

However, this depth function is not robust as it is based on the sample mean and covariance matrix. Therefore, for calculating the depth of each point, we use robust alternatives of mean vector and covariance matrix such as Rousseeuw's minimum covariance determinant (MCD), or the minimum volume ellipsoid (MVE) estimates, which are available in the literature.[20,21] The robust MD of a point x with respect to a dataset X is defined as

$$\mathrm{MD}(x; X) = [1 + (x - X_i)^T C^{-1} (x - X_i)]^{-1}. \tag{7}$$

Here, X_i and C are the MCD-based or MVE-based estimators of the mean vector and the covariance matrix, respectively.

3. Proposed Method

In this section, we propose a new method for finding correct number of clusters in a dataset based on the data depth. In the following Sec. 3.1, we explain the definitions used in the method.

3.1. *Definitions used in the proposed method*

Let $X = \{x_1, x_2 \ldots, x_n\}$ be a dataset with n instances. A data depth function assigns a value between 0 and 1 to each data point in the dataset which specifies the centrality or deepness of that point in the dataset. The depth of each point x_i, $i = 1, 2, \ldots, n$, in the kth cluster is calculated using Eq. (7) and is denoted by $D(x_i|k)$.

Definition 1. *Depth median* (μ_k): We define the maximum depth value in cluster k as the depth median, denoted by μ_k, which is formulated as follows:

$$\mu_k = \max_{1 \le i \le n_k} D(x_i|k), \tag{8}$$

where n_k is the number of points within-cluster k.

Definition 2. *Within-cluster depth* (WD(k)): We define the average difference between each object and the center of cluster k as the within-cluster depth (WD(k)), which is formulated as follows:

$$\mathrm{WD}(k) = \frac{1}{n_k} \sum_{i=1}^{n_k} (\mu_k - D(x_i|k)), \tag{9}$$

where n_k is the number of points within-cluster k and μ_k is the centroid of cluster k.

Definition 3. *Between-cluster depth* (BD(k)): We define the minimum value of average difference between every object in cluster k and the center of every other cluster as the minimum between-cluster depth (BD(k)), which is formulated as follows:

$$\mathrm{BD}(k) = \min_{1 \le j \le m, j \ne k} \left[\frac{1}{n_k} \sum_{i=1}^{n_k} (\mu_j - D(x_i|k)) \right]. \tag{10}$$

Definition 4. *Depth difference* (DD(k)): The difference between within-cluster depth and between-cluster depth of cluster k is defined as the depth difference of cluster k (DD(k)), which is formulated as follows:

$$\mathrm{DD}(k) = \mathrm{WD}(k) - \mathrm{BD}(k). \tag{11}$$

Definition 5. Validity index value (VI(k)): We define the average of depth difference of k clusters as validity index value (VI(k)), which is formulated as follows:

$$\mathrm{VI}(k) = \frac{1}{k} \sum_{i=1}^{k} \mathrm{DD}(i). \tag{12}$$

If the instances of X are well clustered into $k-1$ clusters than $\mathrm{VI}(k) < \mathrm{VI}(k-1)$, so we choose the $k-1$ value where the Average VI of k clusters drop from the previous $k-1$ clusters, which is denoted by k_{opt}. The obtained k_{opt} value is the optimal number of clusters found for the current dataset.

3.2. *Implementation*

- The Algorithm takes only one input that is the dataset X with n instances.
- At the start of the procedure, the k value is set to 1, as the whole dataset is considered as one cluster. And the Validity index (VI) value of one cluster is set to 0, as there is only one cluster and the difference between within-cluster and between-cluster will be 0.

Algorithm 3.1. Estimating the Number of clusters

(1) **Input**: A dataset X with n points $X = \{x_1, x_2, ..., x_n\}$
(2) Set the number of clusters $k \leftarrow 1$.
(3) Set the Validity Index Value $VI(k) \leftarrow 0$.
(4) Find the depth of all points x_i in X, $D(x_i|X)$
(5) Do
  (a) Increment $k \leftarrow k + 1$.
  (b) $range \leftarrow n/k$
  (c) $start \leftarrow 0$
  (d) $end \leftarrow 0$
  (e) Divide the dataset into k partitions with n/k points each with limit $start : end$.
  (f) $start \leftarrow end + 1$
  (g) $end \leftarrow start + range - 1$
  (h) For each partition j in $range$ $start : end$ do:
    i. Find the depth median(centroid) $\mu_j \leftarrow \text{argMax}\,(D(x_i|j))$
    ii. Find Within-cluster depth of partition j $WD(j) = \frac{1}{n_j}\sum_{i=1}^{n_j}(\mu_j - D(x_i|j))$
  (i) For each partition j in $range$ do:
    i. Find Between-cluster depth of cluster j with respect to the nearest cluster : $BD(j) = \min_{1 \le s \le m, s \neq j} \left(\frac{1}{n_j}\sum_{i=1}^{n_j}(\mu_s - D(x_i|j))) \right)$
    ii. Find the difference between within-cluster depth and between-cluster depth of cluster j : $DD(j) = WD(j) - BD(j)$
  (j) Find the average depth difference of k clusters $VI(k) = \frac{1}{k}\sum_{i=1}^{k} DD(i)$
  (k) Repeat steps from a to j Until $VI(k) < VI(k-1)$
(6) Output the optimal number of clusters: $k_{\text{opt}} \leftarrow k - 1$

- The k value is incremented by one and the dataset is divided into k partitions with n/k points in each partition. Actually for forming the clusters here clustering process is not used, instead k partitions will have n/k points set by the range with limit $start : end$ as shown in the algorithm.
- For each partition within-cluster depth is calculated. As defined above within-cluster depth is the average difference between every object and the center of cluster of that partition. The within-cluster depth will give the compactness of the points within the cluster.
- For each partition, the near by partition is found by finding the difference between the depth medians of the clusters. Then, the between-cluster depth with respect to

the nearest cluster is calculated as given in Eq. (10). The between-cluster depth gives the separation of the partition with the nearest partition.

- For each partition, the difference between the within-cluster and the between-cluster depth is defined in Eq. (11).
- The validity index value is calculated for k partitions. If this index value is less than the previous $k-1$ clusters than the process is terminated by declaring $k-1$ as the optimal number of clusters else the process is repeated until the validity condition is satisfied.

4. Experimental Results and Discussions

In this section, we give the detailed experimental comparison of CH index, KL index, Silhouette index, Gap index, CSP index and the proposed index for determining the optimal k value. The proposed method is also compared with our previous work DeD.[22] The experiments are carried out on Intel i3 core processor with 8 GB of RAM running Windows 8.0. The evaluations are conducted in R programming language. For determining the optimal k value using the four indices CH, KL, Silhouette and GAP, we use NbClust function of NbClust[23] package. When adopting NbClust function for validating the clustering results, the Euclidean distance measure and k-means clustering algorithm are used for the experimental datasets. CSP index is also implemented in R with agglomerative hierarchical clustering (AHC) algorithm. In the experiments, the search range for the clustering numbers for the five indices is $k_{\min} = 2$ and $k_{\max} = 10$. Usually, the range is set to $k_{\min} = 2$ and $k_{\max} = \sqrt{n}$ according to the commonly used rule $k_{\max} \leq \sqrt{n}$.[2,1] But in our experiments, we set $k_{\max} = 10$ for the five indices and for the proposed method, we set $k_{\min} = 1$ and the method automatically determines the optimal k value by incrementing the $k_{\min}$ value. We conduct the experiments on 20 different datasets to evaluate the performance of given indices for determining the optimal k value. Table 1 gives the detail of 20 datasets. The Iris, Wine,

Table 1. The characteristics of the real-world and synthetic datasets.

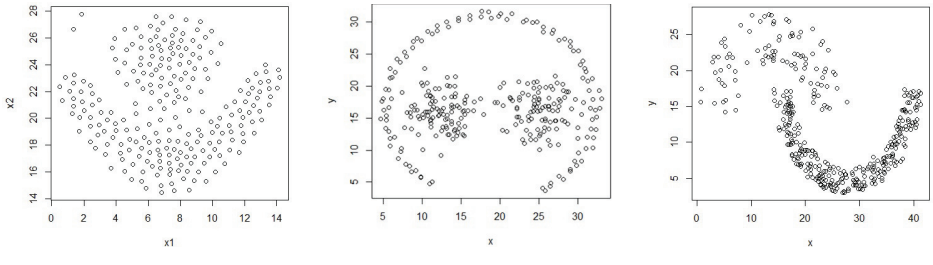
Datasets	No. of instances	No. of attributes	No. of clusters
Iris (IR)	150	5	3
Tae (TA)	151	5	3
Wine (WI)	178	13	3
Seed (SE)	210	8	3
Flame (FL)	240	3	2
Heart (HE)	270	13	2
Pathbased (PA)	300	3	3
Column2C (CO)	310	6	2
Stampout (ST)	340	10	2
Jain (JA)	373	3	2

Table 1. (*Continued*)

Datasets	No. of instances	No. of attributes	No. of clusters
Breast Cancer (BC)	699	10	2
Pima (PI)	768	9	2
Pen (PE)	809	17	2
Ring2 (RI)	1,000	2	2
A1	1,500	2	3
A2	1,500	2	3
B1	1,500	2	3
B2	1,500	2	3
Semicircle2 (SC)	4,811	2	2
Smiley (SM)	5,000	2	4

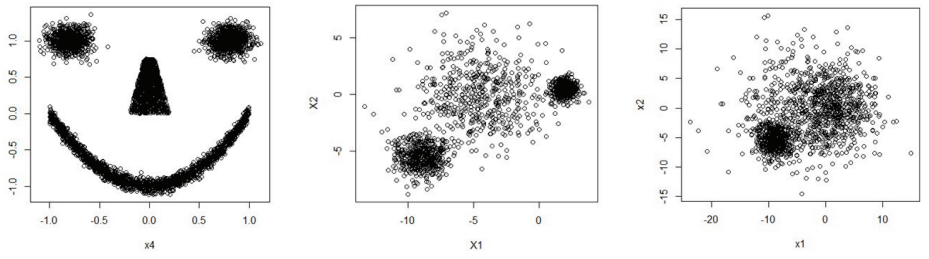
Seed, Vertebral Column (Column_2C), Statlog Heart (Heart), and Teaching Assistant Evaluation (Tae) and Breast Cancer datasets are available at UCI Machine Learning Repository.[24] The Shape datasets Flame, Pathbased and Jain are available at Ref. 25. The Smiley dataset is drawn from mlbench package of *R*. The outlier detection datasets Pima, Pen and Stampout are available at Ref. 26. We generate clustered datasets A1, A2, B1 and B2 with three high variance clusters in two or four dimensions. B1, B2 datasets include two unrelated variables. Semicircle2 and Ring2 datasets comprise 2D random numbers generated by computer simulation. The structure distributions of the shape datasets and synthetic datasets are shown in Fig. 2.

Table 2 shows the estimated k value by using the five indices, DeD method and the proposed index. Performance of the CH and KL index is very poor in comparison to other indices. We observe that KL index does not work well in case of shape datasets and outlier detection datasets. The performance of CH index is also poor in case of shape datasets. All the four indices CH, KL, Silhouette and Gap fail to predict



(a) (b) (c)

Fig. 2. Structure distributions of shape and synthetic datasets. (a) Flame (b) Pathbased (c) Jain (d) Smiley (e) A1 (f) A2



(d) (e) (f)

Fig. 2. (*Continued*)

the optimal k value for synthetic datasets B1 and B2 with unrelated variables. Performance of CSP and DeD method is very good. The proposed method outperforms in all cases compared to other five indices and DeD. Figure 3 depicts all the experimented indices for Smiley dataset.

Table 2. k_{est}: Number of clusters estimated, k is known number of clusters in the datasets.

		k_{est}						
Dataset	k (known)	CH	KL	SIL	GAP	CSP	DeD	Depth
IR	3	3	4	2	2	3	3	4
TA	3	2	8	2	2	3	2	3
WI	3	4	2	2	2	2	2	2
SE	3	3	3	2	2	3	3	3
FL	2	8	5	4	2	3	2	2
HE	2	2	2	2	2	2	2	2
PA	3	2	9	3	2	2	2	3
CO	2	4	6	2	2	2	2	2
ST	2	2	7	2	2	2	2	2
JA	2	10	10	2	2	2	2	2
BC	2	2	2	2	2	2	3	2
PI	2	3	8	2	2	2	2	4
PE	2	3	4	3	2	2	2	2
RI	2	9	2	7	2	2	2	2
A1	3	3	3	3	2	3	3	3
A2	3	2	10	2	2	4	2	3
B1	3	2	2	2	2	3	3	2
B2	3	2	2	2	2	2	2	3
SC	2	10	2	8	2	2	2	2
SM	4	10	3	5	2	6	6	4
Overall		6/20 (30%)	6/20 (30%)	8/20 (40%)	10/20 (50%)	14/20 (70%)	13/20 (65%)	16/20 (80%)

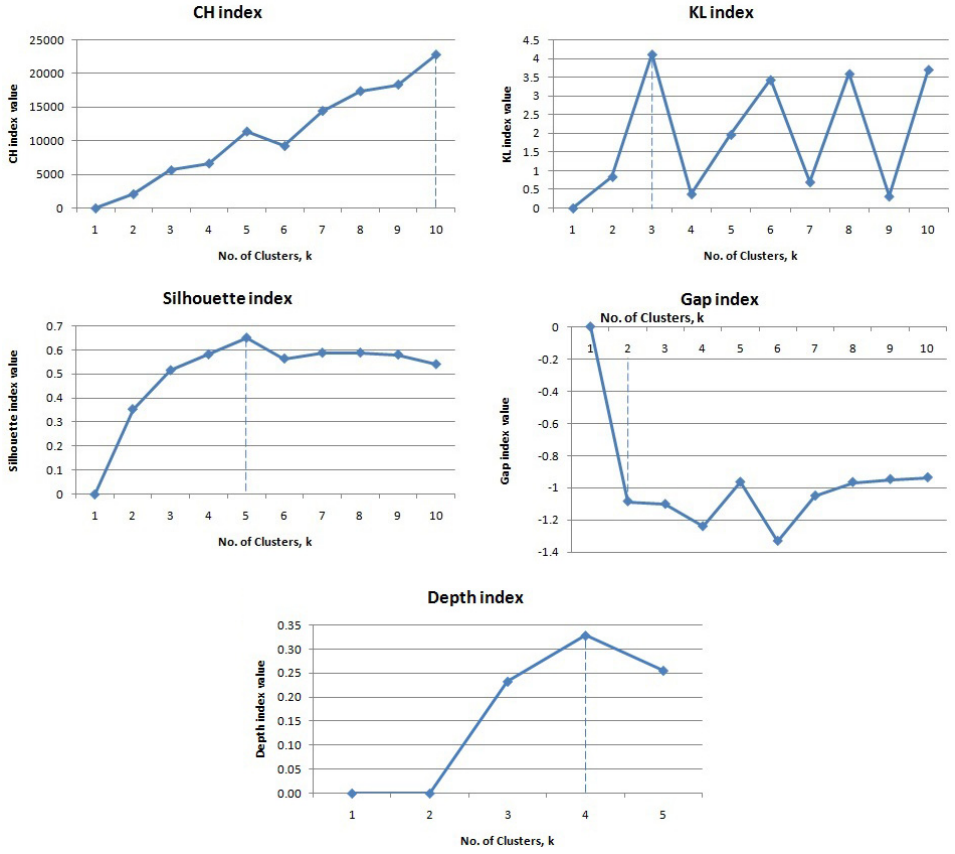


Fig. 3. Estimated k value based on the index values of CH, KL, Silhouette, Gap and Depth indices for Smiley dataset.

Table 3 shows the relative error for determining the optimal k value. The relative error for the estimation of the number of clusters k is given by

$$\mathrm{RE} = \frac{|k - k_{\mathrm{est}}|}{k}, \tag{13}$$

where k_{est} is the estimated number of clusters and k is the known number of clusters in the dataset. From Table 3, it can be seen that KL index depicts the highest relative error which is followed by CH index. Among all the methods, CSP index and the proposed method gives the least relative error.

Table 4 shows the computation time taken by each index for determining the optimal k value for each dataset. The NbClust function is executed 10 times on each dataset and for each index. It can be observed from Table 4 that the Silhouette index has the highest average computation time. The CH, KL and CSP indices have high

Table 3. Relative errors of the estimation of the number of clusters in relation to the known number of classes.

Dataset	CH	KL	SIL	GAP	CSP	DeD	Depth
IR	0.00	0.33	0.33	0.33	0.00	0.00	0.33
TA	0.33	1.67	0.33	0.33	0.00	0.33	0.00
WI	0.33	0.33	0.33	0.33	0.33	0.33	0.33
SE	0.00	0.00	0.33	0.33	0.00	0.00	0.00
FL	3.00	1.50	1.00	0.00	0.50	0.00	0.00
HE	0.00	0.00	0.00	0.00	0.00	0.00	0.00
PA	0.33	2.00	0.00	0.33	0.00	0.33	0.00
CO	1.00	2.00	0.00	0.00	0.00	0.00	0.00
ST	0.00	2.50	0.00	0.00	0.00	0.00	0.00
JA	4.00	4.00	0.00	0.00	0.00	0.00	0.00
BC	0.00	0.00	0.00	0.00	0.00	0.50	0.00
PI	0.50	3.00	0.00	0.00	0.00	1.00	1.00
PE	0.50	1.00	0.50	0.00	0.00	0.00	0.00
RI	3.50	0.00	2.50	0.00	0.00	0.00	0.00
A1	0.00	0.00	0.00	0.33	0.00	0.00	0.00
A2	0.33	2.33	0.33	0.33	0.33	0.33	0.00
B1	0.33	0.33	0.33	0.33	0.00	0.00	0.33
B2	0.33	0.33	0.33	0.33	0.33	0.33	0.00
SC	4.00	0.00	3.00	0.00	0.00	0.00	0.00
SM	1.50	0.25	0.25	0.50	0.50	0.50	0.00
Average	1.00	1.08	0.48	0.18	0.10	0.18	0.10

Table 4. Time, in seconds of 10 runs.

Dataset	CH	KL	SIL	GAP	CSP	DeD	Depth
IR	0.16	0.20	0.46	0.79	0.73	0.28	0.22
TA	0.21	0.26	0.38	0.67	0.79	0.22	0.22
WI	0.23	0.33	0.61	1.16	0.88	0.25	0.22
SE	0.24	0.24	0.64	0.87	0.88	0.27	0.22
FL	0.18	0.21	0.71	0.68	0.86	0.25	0.20
HE	0.28	0.38	0.72	1.1	1.04	0.31	0.23
PA	0.20	0.36	1.11	0.76	0.93	0.24	0.20
CO	0.30	0.4	0.85	1.74	1.48	0.35	0.23
ST	0.28	0.41	1.25	1.12	1.07	0.29	0.22
JA	0.91	0.28	1.45	0.76	0.97	0.29	0.23
BC	1.78	1.63	3.82	2.75	1.01	0.32	0.23
PI	1.53	1.45	4.31	1.87	1.95	0.31	0.23
PE	1.00	1.12	4.23	3.38	2.67	0.33	0.25
RI	1.16	1.1	4.47	1.34	1.90	0.39	0.26
A1	3.35	3.36	10.56	1.73	3.85	0.31	0.22
A2	3.46	3.54	10.66	1.78	3.85	0.31	0.23
B1	3.43	3.70	10.60	2.43	4.29	0.31	0.23
B2	3.50	3.33	10.82	2.35	4.36	0.32	0.23
SC	28.28	28.8	1.3	3.75	29.75	0.52	0.27
SM	35.36	36.12	87.00	3.91	30.20	0.58	0.35
Average	4.29	4.36	7.80	1.75	4.67	0.32	0.21

average computation time followed by GAP. The proposed method has the least average computation time.

5. Conclusion

This paper has presented a novel method for estimating the number of clusters based on data depth. The method is uninfluenced by the inclusion of unrelated variables and is also robust to the dominance of high variance clusters. Our method is simple and comparably efficient in terms of k parameter selection. The existing methods run the clustering algorithm on a dataset with a different k value for each run and decide to use the one that maximizes or minimizes the index value. However, our method does not employ any clustering algorithm and it automatically determines the optimal k value when it satisfies the validation condition. Therefore, the proposed method minimizes the time involved in the clustering process. The experimental results on real and synthetic datasets indicate good performance of the proposed method.

References

1. M. Kim and R. Ramakrishna, Some new indexes of cluster validity, *Pattern Recogn. Lett.* **26**(15) (2005) 2353–2363.
2. N. R. Pal and J. C. Bezdek, On cluster validity for the fuzzy c-means model, *IEEE Trans. Fuzzy Syst.* **3**(3) (1995) 370–379.
3. G. W. Milligan and M. C. Cooper, An examination of procedures for determining the number of clusters in a data set, *Psychometrika* **50**(2) (1985) 159–179.
4. E. Dimitriadou, S. Dolničar and A. Weingessel, An examination of indexes for determining the number of clusters in binary data sets, *Psychometrika* **67**(1) (2002) 137–159.
5. T. Caliński and J. Harabasz, A dendrite method for cluster analysis, *Commun. Stat. Theory Methods* **3**(1) (1974) 1–27.
6. W. J. Krzanowski and Y. Lai, A criterion for determining the number of groups in a data set using sum-of-squares clustering, *Biometrics* **44** (1988) 23–34.
7. L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis* (John Wiley & Sons, 2009).
8. R. Tibshirani, G. Walther and T. Hastie, Estimating the number of clusters in a data set via the gap statistic, *J. Royal Stat. Soc. Series B* **63**(2) (2001) 411–423.
9. S. Zhou, Z. Xu and F. Liu, Method for determining the optimal number of clusters based on agglomerative hierarchical clustering, *IEEE Trans. Neural Netw. Learn. Syst.* **28**(12) (2016) 3007–3017.
10. R. Serfling, Nonparametric multivariate descriptive measures based on spatial quantiles, *J. Stat. Plan Inference* **123**(2) (2004) 259–278.
11. J. W. Tukey, Mathematics and the picturing of data, in *Proc. Int. Congress Mathematicians*, Vol. 2 (1975), pp. 523–531.
12. H. Oja, Descriptive statistics for multivariate distributions, *Stat. Probab. Lett.* **1**(6) (1983) 327–332.
13. W. Eddy, Convex hull peeling, in *COMPSTAT 1982 5th Symp Toulouse 1982* (Springer 1982), pp. 42–47.
14. R. Y. Liu *et al.*, On a notion of data depth based on random simplices, *Ann. Stat.* **18**(1) (1990) 405–414.

15. P. J. Rousseeuw and M. Hubert, Regression depth, *J. Am. Stat. Assoc.* **94**(446) (1999) 388–402.
16. Y. Vardi and C.-H. Zhang, The multivariate l1-median and associated data depth, *Proc. Nat. Acad. Sci.* **97**(4) (2000) 1423–1426.
17. Y. Zuo and R. Serfling, General notions of statistical depth function, *Ann. Stat* **28** (2000) 461–482.
18. R. Y. Liu and K. Singh, A quality index based on data depth and multivariate rank tests, *J. Am. Stat. Assoc.* **88**(421) (1993) 252–260.
19. P. C. Mahalanobis, On the generalized distance in statistics, *Proceedings of National Institute of Sciences (India)* Vol. 2, No. 1 (1936), pp. 49–55.
20. P. J. Rousseeuw, Multivariate estimation with high breakdown point, *Math. Stat. Appl.* **8**(283–297) (1985) 37.
21. P. J. Rousseeuw and A. M. Leroy, *Robust Regression and Outlier Detection* (John Wiley & Sons, 2005).
22. C. Patil and I. Baidari, Estimating the optimal number of clusters k in a dataset using data depth, *Data Sci. Eng.* **4** (2019) 1–9.
23. M. Charrad, N. Ghazzali, V. Boiteau and A. Niknafs, NbClust: An R package for determining the relevant number of clusters in a data set, *Journal of Statistical Software* **61**(6) (2014) 1–36. http://www.jstatsoft.org/v61/i06/.
24. D. Dua and C. Graff, UCI machine learning repository (2017). http://archive.ics.uci.edu/ml.
25. P. Fränti and S. Sieranoja, K-means properties on six clustering benchmark datasets, *Appl. Intell.* **48** (2018) 4743–4759.
26. S. Rayana, ODDS library (2016), http://odds.cs.stonybrook.edu.