Data Science and Engineering (2019) 4:132–140
https://doi.org/10.1007/s41019-019-0091-y

Estimating the Optimal Number of Clusters *k* in a Dataset Using Data Depth

Channamma Patil[1] · **Ishwar Baidari**[1]

Received: 6 July 2018 / Revised: 22 May 2019 / Accepted: 29 May 2019 / Published online: 20 June 2019

Abstract
This paper proposes a new method called depth difference (DeD), for estimating the optimal number of clusters (k) in a dataset based on data depth. The DeD method estimates the k parameter before actual clustering is constructed. We define the depth within clusters, depth between clusters, and depth difference to finalize the optimal value of k, which is an input value for the clustering algorithm. The experimental comparison with the leading state-of-the-art alternatives demonstrates that the proposed DeD method outperforms.

Keywords Data depth · Depth within cluster · Depth between cluster · Depth difference · Average depth · Optimal value k

1 Introduction

Clustering is an unsupervised machine learning technique that partitions the input dataset into clusters in such a way that the objects within a cluster are more similar to each other than to those in other clusters. Several clustering methods are available [5], for cluster analysis. However, a core problem in applying many of the existing clustering methods is that the number of clusters (k parameter) needs to be prespecified before the clustering is carried out. The parameter k is either identified by users based on prior information or determined in a certain way. Clustering results may largely depend on the number of clusters specified. It is necessary to provide educated guidance for determining the number of clusters in order to achieve appropriate clustering results. Since the number of clusters is rarely previously known, the usual approach is to run the clustering algorithm several times with a different k value for each run.

The process of evaluating the partitions produced by clustering algorithms is known as cluster validation, which is an important subject in cluster analysis. The common approach for this evaluation is to use validity indices. Validity indices are typically classified by researchers into two groups, i.e., internal or external. The external indices validate a partition by comparing it with the external information (true cluster labels), whereas the internal indices focus on the partitioned data and measure the compactness and separation of the clusters.

In the literature, many internal indices have been proposed [1, 9, 13, 15] to analyze the clustering results and determine the optimal number of clusters (NC). Most of the internal indices are distance based. A recent proposal is the gap statistic which compares the change in within-cluster dispersion with that expected under an appropriate null distribution [15]. These indices measure the cluster compactness based on average pairwise distance or average center-based distance. Similarly, they measure the separation between the clusters by calculating the pairwise distance between cluster centers. Distance-based indices are sensitive to the inclusion of unrelated variables. Moreover, these indices can be dominated by high variance clusters. Further, they depend on the scales of individual clusters.

In this paper, a novel method called depth difference (DeD) for estimating the optimal number of clusters in a dataset based on data depth is proposed. A depth function (Mahalanobis depth) arranges data by their degree of centrality. High depth value coincides closely with centrality, and low depth value coincides with outlyingness. It focuses on centrality and separation of observation rather than spread. DeD method partitions the data into k partitions (clusters) and calculates the depth of each point within the

✉ Ishwar Baidari
ishwar_sp@yahoo.co.in

Channamma Patil
channamma.shivu@gmail.com

[1] Department of Computer Science, Karnatak University, Dharwad, Karnataka 580003, India

Table 1 Details of notations used

Notation	Description
n	Number of observations
p	Number of variables
q	Number of clusters
X	$\{x_{ij}\}, i = 1, 2, \dots, n, j = 1, 2, \dots, p, n \times p$ data matrix of p variables measured on n independent observations
$\bar{X}$	$q \times p$ matrix of cluster means
$\bar{x}$	Centroid of data matrix X
n_k	Number of objects in cluster C_k
c_k	Centroid of cluster C_k
x_i	p-dimensional vector of observations of the ith object in cluster C_k
$\parallel x \parallel$	$(x^T x)^{1/2}$
$W_q = \sum_{k=1}^{q} \sum_{i \in C_k} (x_i - c_k)(x_i - c_k)^T$	Within-group dispersion matrix for data clustered into q clusters
$B_q = \sum_{k=1}^{q} n_k (c_k - \bar{x})(c_k - \bar{x})^T$	Between-group dispersion matrix for data clustered into q clusters
$T = \sum_{i=1}^{n} (x_i - \bar{x})(x_i - \bar{x})^T$	Total dispersion matrix of the data

cluster. The maximum depth value within a cluster represents the cluster centroid. The method used in this study measures the compactness of a cluster by finding the average difference between depths of points within the cluster and the cluster centroid. It also measures the separation between the clusters and the dataset, by calculating the average difference between the average depth of *k* partitions and the average depth of a dataset. The optimal cluster number is determined by maximizing the value of the depth difference.

The traditional methods [1, 9, 13, 15] for finding the *k* value run the clustering algorithm several times with a different *k* value for each run. Thus, all the partitions are evaluated and the partition that best fits the data is selected. But, DeD does not employ any clustering algorithm for partitioning the data. Thus, it is computationally efficient and it effectively estimates *k* value irrespective of dimensions, scales, and cluster densities of a dataset. The proposed method is also independent of the scales of individual clusters and is thus not dominated by high variance clusters. Further, DeD is robust to the inclusion of unrelated variables. Theoretical research and experimental results indicate good performance of the proposed method.

The paper is organized as follows: In Sect. 2, a brief description of the existing methods used for estimating the number of clusters in a dataset is given. In Sect. 3, the proposed method is presented. The experimental results are presented in Sect. 4, and finally, Sect. 5 concludes the paper with scopes for future applications.

2 Background and Related Work

In the literature, several internal validity indices have been proposed to evaluate the clustering results and determine the optimal NC. Examples include CH index [1], KL index [9], Silhouette index [13], and Gap index [15]. Most of the works have compared different internal indices by running clustering algorithms over different datasets for a range of values of *k* and considering *k* value for the best partitioning.

In 1985, Milligan and Cooper [12] compared 30 internal indices on 108 synthetic datasets with the varying number of clusters, dimensions, and cluster sizes. Authors called these indices as "stopping criteria" and concluded with top performers.

However, in 1999 Gordon [8] categorized these stopping rules into global and local rules. He states that global rules evaluate the measure, G(q), of the goodness of the partition into q clusters, usually based on the within- and between-cluster variability and identify the value of q for which G(q) is optimal. A disadvantage of many of these rules is that there is no natural definition of G(1). Hence, they can provide no guidance on whether the data should be partitioned. Local rules involve examining whether a pair of clusters should be amalgamated (or a single cluster should be subdivided). Unlike global rules, they are thus based on only part of the data and can assess only hierarchically nested partitions. A disadvantage of local rules is that they generally need the specification of a threshold value or significance level, the optimal value of which will depend on the (unknown) properties of the dataset that is under investigation.

In the following subsections, the four internal indices compared in this work are described. The notations used in the following subsections are summarized in Table 1.

2.1 Calinski and Harabasz (CH) Index

Milligan and Cooper [12] have carried out an experimental comparative study of 30 different approaches. Among the 30 different approaches, the approach proposed by Calinski and Harabasz [1] outperforms the others. The Calinski and Harabasz (CH) index [1] is defined as

$$\mathrm{CH}(q) = \frac{\mathrm{trace}(B_q)/(q-1)}{\mathrm{trace}(W_q)/(n-q)} \tag{1}$$

The procedure estimates $\bar{X}$ by maximizing the index CH(q) over q. B_q and W_q are the between-cluster and within-cluster sum of squared errors, calculated as the trace (B_q) and trace (W_q), respectively. $CH(q)$ is only defined for $q > 1$ since trace (B_q) is not defined when $q = 1$.

2.2 Krzanowski and Lai (KL) Index

Friedman and Rubin [7] proposed minimization of $|W_q|$ as a clustering criteria. Concerned with the problem of finding $\bar{X}$ when using this method, Marriott [11] studied property of $|W_q|$ in detail and described an approach based on $q^2|W_q|$. Krzanowski and Lai [9] examined the behavior of Marriot's $q^2|W_q|$ criterion by the Monte Carlo methods. They calculated the sample value of $q^2|W_q|/|T|$ when sampling from a homogeneous, uniform population. The results showed that there was a large discrepancy between the estimated value and the predicted value. Instead, a similar criterion using $q^2 W_q$ demonstrated much better consistency between the estimated value and predicted value. The KL index proposed by Krzanowski and Lai [9] is defined as

$$\mathrm{KL}(q) = \left| \frac{\mathrm{DIFF}_q}{\mathrm{DIFF}_{q+1}} \right| \tag{2}$$

where

- $\mathrm{DIFF}_q = (q-1)^{2/p}\mathrm{trace}(W_{q-1}) - q^{2/p}\mathrm{trace}(W_q)$.

The value of q, maximizing KL(q), is regarded as specifying the optimal number of clusters. But KL(q) is not defined for $q = 1$.

2.3 Silhouette Index

Kaufman and Rousseeuw [13] introduced the Silhouette index which is constructed to show graphically how well each object is classified in a given clustering output.

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

where

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

2.4 Gap Index

Tibshirani et al. [15] proposed an approach to estimate the number of clusters in a dataset via gap statistic. This procedure is designed to be fit for any clustering technique. The idea is to compare the change in W_{qb} as qb increases for original data with that expected for the data generated from a suitable reference null distribution.

$$\mathrm{Gap}(q) = \frac{1}{B}\sum_{b=1}^{B} logW_{qb} - logW_q, \tag{4}$$

where B is the number of reference datasets generated using uniform prescription [15] and W_{qb} is the within-dispersion matrix. The optimal number of clusters is chosen via finding the smallest q such that:

$Gap(q) \geq Gap(q+1) - s_{q+1}, (q = 1, \ldots, n-2),$

where

- $s_q = sd_q\sqrt{1 + 1/B}$,
- sd_q is the standard deviation of $\{logW_{qb}\}, b = 1, \ldots, B$,
- $sd_q = \sqrt{\frac{1}{B}\sum_{b=1}^{B}(logW_{qb} - \bar{l})^2}$,
- $\bar{l} = \frac{1}{B}\sum_{b=1}^{B} logW_{qb}$.

To apply this method, it is important to choose an appropriate reference null distribution. Considering k-means clustering, Tibshirani et al. [15] proved that if $p = 1$, the uniform distribution is most likely to produce spurious clusters based on the gap test among all uni-modal distributions. They also proved that in the multivariate case ($p > 1$), there is no such generally applicable reference distribution: It may depend on the geometry of the particular null distribution.

3 Proposed Method

3.1 Data Depth

Data depth measures a median in a multivariate dataset, which is the deepest point in a given dataset. Tukey [16] proposed a "half space" depth in order to present an idea about multivariate data analysis, based on center outward ordering. Various depth methods are found in the literature,

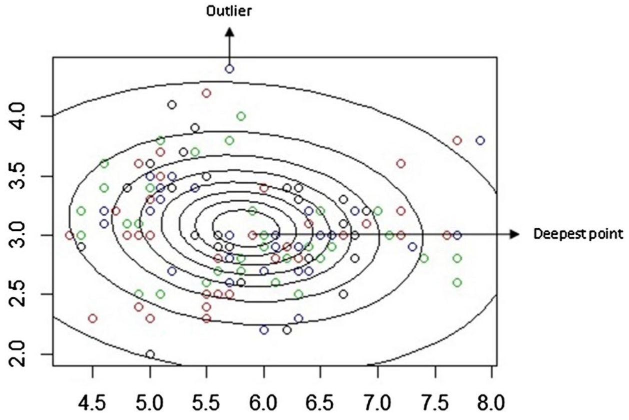

Fig. 1 Mahalanobis depth contours

such as convex-hull peeling depth [4], simplicial depth [10], regression depth [14], and L1 depth [17]. This study uses the Mahalanobis depth function to measure the centrality of a point within a cloud of data because of its fast and easy computability. Data depth 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 point with maximum depth will be the deepest point in the dataset, which is shown in Fig. 1 using the Mahalanobis depth over iris dataset.

The Mahalanobis depth function can be defined as follows:

$$M_D(x;X) = [1 + (x - \bar{x})^T Cov(X)^{-1}(x - \bar{x})]^{-1} \tag{5}$$

where $\bar{x}$ and $Cov(X)$ are the mean and covariance matrix of X, respectively. Maximum depth point is a center point, higher depth value points are near the center, and the lower depth value points are outliers. However, data depth presents globally maximizing depth. Since the mean is sensitive to outliers, the equation to calculate the depth of each point is modified as follows:

$$M_D(x;X_i) = [1 + (x - X_i)^T Cov(X)^{-1}(x - X_i)]^{-1} \tag{6}$$

Here, point X_i is used rather than the mean vector. Thus, each point X_i can be regarded as a center point so that it is possible to calculate data depth from each point with respect to a given dataset.

3.2 DeD Method

This paper defines a theory for the formulation of the depth difference (DeD) method. Let $X = \{x_1, x_2 \dots x_n\}$ be a dataset with n instances. Data depth assigns a value between 0 and 1 to each data point in the dataset which specifies the centrality or deepness of the point in the dataset. The depth of each point x_i in X is calculated using Eq. 6 and is denoted by D_i, for $i = 1, 2, \dots n$.

Definition 1 *Depth median (DM)* An instance x_i in the dataset X is called depth median, if it has maximum depth value in X. Depth median is the deepest point in the dataset X. The depth median is denoted by DM. Therefore, we define depth median as follows:

$$DM = max(D_i) \tag{7}$$

Definition 2 *Depth within cluster* (DW) The depth of each point within a cluster C_k, for $k = 2, 3, \dots 20$, is denoted by D_i^k, for $i = 1, 2, \dots n_k$, where n_k is the number of points within cluster C_k. The depth median of each cluster C_k is represented as DM^k. Hence,

$$\mathrm{DM}^k = max(D_i^k) \tag{8}$$

The average difference between the depths of points within the cluster C_k and the depth median of C_k is denoted by $\triangle^k$, which is formulated as follows:

$$\triangle^k = \frac{1}{n_k} \sum_{i \in C_k} |(D_i^k - DM^k)| \tag{9}$$

The depth within cluster (DW) is defined as the average of $\triangle^k$ of k clusters as follows:

$$\mathrm{DW} = \frac{1}{k} \sum_{i=1}^{k} (\triangle^i) \tag{10}$$

Definition 3 *Depth between cluster* (DB) The average difference between the depths of points within the dataset X and the depth median of X is formulated as follows:

$$\triangle = \frac{1}{n} \sum_{i=1}^{n} |(D_i - DM)| \tag{11}$$

where n is the number of instances in dataset X. The depth between cluster is defined as the difference between $\triangle$ and DW, and it is defined as follows:

$$\mathrm{DB} = \triangle - \mathrm{DW} \tag{12}$$

Definition 4 *Depth Difference* (DeD): The depth difference (DeD) finds the difference between depth within cluster (DW) and depth between cluster (DB). DeD is defined follows:

$$\mathrm{DeD} = \mathrm{DW} - \mathrm{DB} \tag{13}$$

Definition 5 *Optimal* k: The optimal k is the maximum index value of *DeD*. Hence,

$$k = \mathrm{index}(max(\mathrm{DeD})) \tag{14}$$

Algorithm Description

1. Line 3 computes the depth of each point x_i in dataset X using Mahalanobis depth. The depth values are retained in a vector D_i (Eq. 6).
2. Line 4 finds maximum depth or depth median (DM) of the vector D_i of the dataset X (Eq. 7).
3. Line 5 calculates the average difference between the depths of points within the dataset X and the depth median of X (Eq. 11).
4. Lines 6–14 partition the dataset X into k partitions, for $k = 2...20$. Each partition (start: end) represents one cluster C_k, for $k = 2 \ldots 20$. For each cluster C_k, it finds the depth (D_i^k) of each point x_i within the cluster C_k and also finds the depth median (DM^k) of each cluster C_k (Eq. 8).
5. Line 15 computes the average difference between D_i^k and DM^k of the kth cluster which is retained in a vector $\triangle^k$ (Eq. 9).
6. Line 17 calculates the average of $\triangle^k$ of k clusters which is stored in a vector DW (Eq. 10).
7. Line 18 finds the difference between $\triangle$ and DW which is stored in a vector DB (Eq. 12).
8. Line 19 finds the difference between DW and DB which is assigned to vector DeD (Eq. 13).
9. Finally, line 21 finds the index of the maximum value of DeD as the optimal value, which is the k value (Eq. 14).

Algorithm 1 Estimating Number of Clusters

```
1: Input: A dataset $X$ with points $X = \{x_1, x_2, ...., x_n\}$
2: Output: $k$ , The number of clusters estimated
3: $D_i \leftarrow$ Depth of each point in $X$
4: $DM \leftarrow$ Depth median of $X$
5: $\triangle \leftarrow$ Average difference between $D_i$ and $DM$
6: for $k = 2\, to\, 20$ do
7:     $range \leftarrow n/k$
8:     $start \leftarrow 0$
9:     $end \leftarrow 0$
10:    for $j = 1\, to\, k$ do
11:        $start \leftarrow end + 1$
12:        $end \leftarrow start + range - 1$
13:        $D_i^k \leftarrow$ Depth of each point within the cluster $C_k$ (partition start : end)
14:        $DM^k \leftarrow$ Depth median of cluster $C_k$
15:        $\triangle^k \leftarrow$ Average difference between $D_i^k$ and $DM^k$ of $k^{th}$ cluster
16:    end for
17:    $DW \leftarrow$ Average of $\triangle^k$ of $k$ clusters
18:    $DB \leftarrow \triangle - DW$
19:    $DeD \leftarrow DW - DB$
20: end for
21: $k \leftarrow index(max(DeD))$ index of $DeD$ for which $DeD$ is maximum
```

4 Experimental Results

To verify the performance of the DeD algorithm, this study uses NbClust function of NbClust package [2] defined in R, to validate indices such as CH index, KL index, Silhouette index, and Gap index. The parameters used in NbClust function are Euclidean distance, k-means algorithm, and the range $k = 2$ to 20. Experiments are conducted on synthetic datasets, real-world datasets, and one image dataset to test the DeD and to compare it with other indices, such as the CH index, KL index, Silhouette index, and Gap index.

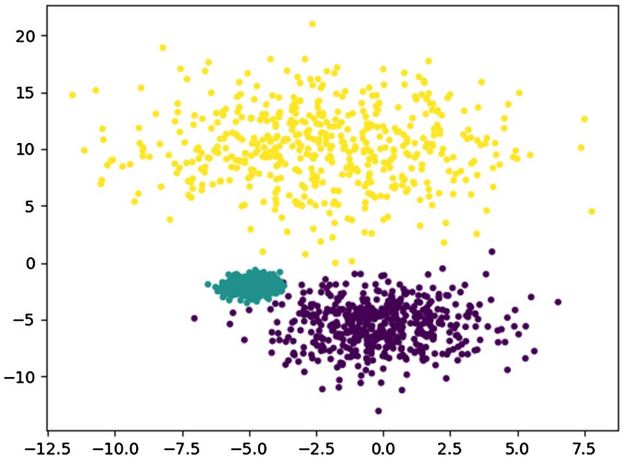

Fig. 2 Dataset with different within-cluster variance

Table 2 Simulation study on synthetic datasets with 3 high variance clusters

Dataset	CH index	KL index	Silhouette index	GAP index	DeD
A1	3	3	5	2	3
A2	3	3	3	2	2
A3	2	10	2	2	3
A4	2	14	2	2	3
A5	2	15	2	2	2
Overall	2/5	2/5	1/5	0/5	3/5

4.1 Synthetic Datasets

These experiments include 10 2-D synthetic datasets with 1500 instances. Structure distribution of a 2-D synthetic dataset is shown in Fig. 2. Among 10 datasets, for 5 datasets 2 unrelated variables were included. The experimental results for the five validity indices used to determine the optimal number of clusters for the 5 synthetic datasets with 3 high variance clusters are shown in Table 2.

The correct number of clusters for the 5 synthetic datasets is 3, where it is observed that DeD achieves the correct optimal NCs for the 3 synthetic datasets; the Silhouette index is effective for 1 dataset; the CH index and KL index are effective for 2 datasets, respectively, and the Gap index fails to predict the optimal NC, shown in Table 2.

Table 3 Simulation study on synthetic datasets with inclusion of 2 unrelated variables

Dataset	CH index	KL index	Silhouette index	GAP index	DeD
B1	2	2	2	2	3
B2	2	2	2	2	2
B3	2	2	2	2	3
B4	2	2	2	2	3
B5	2	2	2	2	2
Overall	0/5	0/5	0/5	0/5	3/5

Table 4 The characteristics of the real datasets

Dataset	No. of instances	No. of attributes	No. of clusters
Face images (FI)	100	90	10
Iris (IR)	150	5	3
Wine (WI)	178	13	3
Seed (SE)	210	8	3
Flame (FL)	240	3	2
Pathbased (PA)	300	3	3
Spiral (SP)	312	3	3
Stampout (ST)	340	10	2
Jain (JA)	373	3	3
R15 (R15)	600	3	15
Breast Cancer (BC)	699	10	2
Pima (PI)	768	9	2
Aggregation (AG)	788	3	7
Pen (PE)	809	17	2
Dim032 (D32)	1024	33	16
Shapes (SH)	5000	3	4
LandArea (LA)	10,546	29	6
Shuttle (SL)	24,917	10	2

Table 3 shows that the DeD achieves the correct optimal NC for the 3 synthetic datasets, and the four indices fail to predict optimal NC.

4.2 Real-World Datasets

The experiments include 18 real datasets, 9 datasets drawn from the UCI Machine Learning Repository [3], 8 datasets from Clustering benchmark datasets [6], and one image dataset drawn from ORL face database. The 18 real datasets and their characteristics are shown in Table 4.

The experimental results from the five validity indices used to evaluate the optimal number of clusters are shown in Table 5. It is observed that the DeD achieves the correct optimal NCs for the 11 real datasets; the Silhouette index is effective for 6 datasets; the CH index is effective for 5 datasets; KL index is effective for 4 datasets, and the Gap index is effective for 7 datasets.

4.3 Relative Error

The experiments for finding the optimal clustering numbers for the datasets allows to analyze the changes in the relative error in the estimation of *k* given by each internal index experimented with:

$$\mathrm{RE} = \frac{|k - k_{\mathrm{est}}|}{k} \tag{15}$$

where k_{est} is the estimated number of clusters and k is the true number of clusters.

The experimental results from the five validity indices used to analyze the changes in the relative error in the estimation of k are shown in Table 6, where it is found that the KL index has a high error rate for FL dataset. In terms of average error rate, KL and CH index is very high. But, the comprehensive performance of the proposed method appears to be outstanding for real-world datasets.

4.4 Adjusted Rand Index (ARI)

We can check whether the estimated value of k is appropriate as the number of clusters or not. We can measure this by using the adjusted Rand index (ARI) given by their respective clustering:

$$\mathrm{ARI} = \frac{\sum_{ij}\binom{n_{ij}}{2} - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] / \binom{n}{2}}{\frac{1}{2}\left[\sum_i \binom{a_i}{2} + \sum_j \binom{b_j}{2}\right] - \left[\sum_i \binom{a_i}{2}\sum_j \binom{b_j}{2}\right] / \binom{n}{2}} \tag{16}$$

where

- $n_{ij} = |S_i \cap S_j|$,
- $a_i = \sum_{j=1}^{k} |S_i \cap S_j|$
- $b_i = \sum_{i=1}^{k} |S_i \cap S_j|$.

In order to compare the clustering results obtained using four internal indices and the proposed method against the external criteria, the ARI index is used here as a performance measure. The ARI validation threshold (θ), $\theta \in [-1, 1]$. When $\theta < 0$ indicates poor matching and when θ equal to 1 indicates perfect matching. The experimental results for ARI index on different datasets are given in Table 7. The ARI index values of the proposed method are comparatively higher than other indices.

Table 5 Optimal clustering numbers for the 18 real datasets

Dataset	k (known)	CH index	KL index	Silhouette index	GAP index	DeD
		k_{est}	k_{est}	k_{est}	k_{est}	k_{est}
FI	10	10	8	12	2	8
IR	3	3	18	2	2	3
WI	3	4	11	2	2	2
SE	3	3	3	2	2	3
FL	2	8	18	4	2	2
PA	3	17	19	3	2	2
SP	3	17	3	20	2	2
ST	2	2	7	2	2	2
JA	2	10	10	2	2	2
R15	15	19	19	16	2	15
BC	2	2	2	2	2	3
PI	2	3	11	2	2	2
AG	7	20	16	4	2	7
PE	2	3	4	3	2	2
D32	16	17	15	15	2	11
SH	4	9	4	4	2	4
LA	6	2	11	2	2	10
SL	2	Err.	Err.	Err.	2	2
Overall		5/18 (28%)	4/18 (22%)	6/18 (33%)	7/18 (39%)	11/18 (61%)

k_{est} number of clusters estimated, k true number of clusters in the dataset, *Err.* used built in function NbClust() from the NbClust package in R, *A runtime error* "cannot allocate vector of size 4.6 Gb"

Table 6 Relative errors of the estimation of the number of clusters in relation to the known number of classes

Dataset	CH index	KL index	Silhouette index	GAP index	DeD
FI	0.00	0.20	0.20	0.80	0.20
IR	0.00	5.00	0.33	0.33	0.00
WI	0.33	2.67	0.33	0.33	0.33
SE	0.00	0.00	0.33	0.33	0.00
FL	3.00	8.00	1.00	0.00	0.00
PA	4.67	5.33	0.00	0.33	0.33
SP	4.67	0.00	5.67	0.33	0.33
ST	0.00	2.50	0.00	0.00	0.00
JA	4.00	4.00	0.00	0.00	0.00
R15	0.27	0.27	0.07	0.87	0.00
BC	0.00	0.00	0.00	0.00	0.50
PI	0.50	4.50	0.00	0.00	0.00
AG	1.86	1.29	0.43	0.71	0.00
PE	0.50	1.00	0.50	0.00	0.00
D32	0.06	0.06	0.06	0.88	0.31
SH	1.25	0.00	0.00	0.50	0.00
LA	0.67	0.83	0.67	0.67	0.66
SL	Err.	Err.	Err.	0.00	0.00
Average	1.21	1.98	0.53	0.34	0.15

Err. used built in function NbClust() from the NbClust package in R, *A runtime error* "cannot allocate vector of size 4.6 Gb"

Table 7 The adjusted Rand index (ARI) for the clustering at each estimated *k* in the datasets

Dataset	CH index	KL index	Silhouette index	GAP index	DeD
FI	0.81	0.78	0.90	0.12	0.78
IR	0.73	0.23	0.54	0.54	0.73
WI	0.13	0.20	0.07	0.07	0.07
SE	0.71	0.71	0.47	0.47	0.72
FL	0.21	0.09	0.43	0.45	0.45
PA	0.22	0.20	0.46	0.40	0.40
SP	0.11	− 0.01	0.11	0.00	0.00
ST	0.10	0.07	0.10	0.10	0.10
JA	0.13	0.13	0.32	0.32	0.32
R15	0.92	0.92	0.88	0.12	0.90
BC	0.84	0.84	0.84	0.84	0.79
PI	0.05	0.04	0.07	0.07	0.07
AG	0.33	0.42	0.76	0.35	0.74
PE	0.06	0.02	0.06	− 0.01	0.34
D32	0.86	0.82	0.82	0.10	0.70
SH	0.65	0.91	0.91	0.49	1.00
LA	0.01	0.07	0.01	0.01	0.06
SL	Err.	Err.	Err.	0.00	0.20
Average	0.38	0.36	0.43	0.27	0.47

Err. used built in function NbClust() from the NbClust package in R, *A runtime error* "cannot allocate vector of size 4.6 Gb"

Table 8 Average runtime (s) of 10 runs

Dataset	CH index	KL index	Silhouette index	GAP index	DeD
FI	0.83	1.21	0.65	3.64	0.60
IR	0.22	0.25	0.60	0.86	0.28
WI	0.26	0.32	0.72	1.16	0.25
SE	0.21	0.27	0.87	0.95	0.27
FL	0.19	0.22	0.95	0.81	0.25
PA	0.22	0.26	1.24	0.82	0.24
SP	0.23	0.26	1.29	0.84	0.29
ST	0.35	0.42	1.48	1.21	0.29
JA	0.24	0.28	1.53	0.88	0.29
R15	0.45	0.50	2.87	0.93	0.29
BC	1.62	1.74	4.73	2.52	0.32
PI	0.91	0.97	4.54	1.86	0.31
AG	0.68	0.76	4.24	1.02	0.28
PE	1.10	1.27	5.00	2.98	0.33
D32	2.03	2.22	7.36	6.54	0.48
SH	30.71	30.85	116.20	3.61	0.58
LA	198	192	610	66	2.28
SL	Err.	Err.	Err.	86	2.14

Err. used built in function NbClust() from the NbClust package in R, *A runtime error* “cannot allocate vector of size 4.6 Gb”

4.5 Computation Times

Some computation times are given in Table 8. This computational experiment was carried out in one of the cores of an Intel i3 in a 64 bits computer with 8 GB of RAM running Windows 8.0 and R. The inbuilt function ‘Sys.time()’ in R is used in terms of computation time metric. The experiments on each dataset were executed 10 times to calculate the average execution time. The average execution times and corresponding datasets are shown in Table 8, where it can be observed that the execution time of all the internal indices increases with respect to the number of objects to be clustered, due to exercising clustering algorithm for finding appropriate *k* value. The average execution time of all internal indices is comparatively higher than the DeD method.

5 Conclusion

This paper has presented a method called DeD for estimating the number of clusters based on data depth. The DeD method is uninfluenced by the inclusion of unrelated variables and is also robust to the dominance of high variance clusters. The proposed method is simple and comparatively efficient, in terms of *k* parameter selection. The existing methods select the *k* value of a dataset by running a clustering algorithm over a dataset, with a set of different values for *k* parameter decided by the user. However, DeD method iterates the DeD computations over a dataset with a range of values of the *k* parameter to finalize the appropriate number of *k* clusters, and also DeD effectively treats the dominance of high variance clusters. Further, this study shows how the distance-based methods are sensitive to the inclusion of unrelated variables.

In case of complicated data, it is hard to select the appropriate parameter for grouping data. But our experimental results demonstrate that DeD is robust to parameter selection than the existing methods. In terms of average relative error, execution time, and the average ARI, measurements on most of the complicated datasets show that the DeD approach outperforms existing methods.

DeD requires prior information for the range of *k*, decided by users, and this paper did not provide an adaptive parameter scheme based on datasets. It is further considered to expand the current algorithm so that it can automatically optimize the *k* parameter without a manual selection process, which results in the improvement of efficiency and accuracy.

Author Contributions IB designed, coordinated this research, and drafted the manuscript. CP carried out experiments and data analysis. The authors read and approved the final manuscript.

Availability of Data and Materials All datasets used for supporting the conclusion of this article are available from UCI Machine Learning Repository at the website of https://archive.ics.uci.edu/ml/datasets.html.

Compliance with Ethical Standards

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Caliński T, Harabasz J (1974) A dendrite method for cluster analysis. Commun Stat Theory Methods 3(1):1–27
2. Charrad M, Ghazzali N, Boiteau V, Niknafs A (2012) Nbclust package: finding the relevant number of clusters in a dataset. UseR! 2012
3. Dua D, Graff C (2017) UCI machine learning repository. http://archive.ics.uci.edu/ml
4. Eddy W (1982) Convex hull peeling. In: COMPSTAT 1982 5th symposium held at Toulouse 1982. Springer, pp 42–47
5. Fahad A, Alshatri N, Tari Z, Alamri A, Khalil I, Zomaya AY, Foufou S, Bouras A (2014) A survey of clustering algorithms for big data: taxonomy and empirical analysis. IEEE Trans Emerg Top Comput 2(3):267–279

6. Fränti P, Sieranoja S (2018) K-means properties on six clustering benchmark datasets. http://cs.uef.fi/sipu/datasets/
7. Friedman HP, Rubin J (1967) On some invariant criteria for grouping data. J Am Stat Assoc 62(320):1159–1178
8. Gordon AD (1999) Classification. Chapman & Hall/CRC Monographs on Statistics & Applied Probability. CRC Press, Boca Raton
9. Krzanowski WJ, Lai Y (1988) A criterion for determining the number of groups in a data set using sum-of-squares clustering. Biometrics 44(1):23–34
10. Liu RY et al (1990) On a notion of data depth based on random simplices. Ann Stat 18(1):405–414
11. Marriott F (1971) Practical problems in a method of cluster analysis. Biometrics 27(3):501–514
12. Milligan GW, Cooper MC (1985) An examination of procedures for determining the number of clusters in a data set. Psychometrika 50(2):159–179
13. Rousseeuw PJ (1987) Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J Comput Appl Math 20:53–65
14. Rousseeuw PJ, Hubert M (1999) Regression depth. J Am Stat Assoc 94(446):388–402
15. Tibshirani R, Walther G, Hastie T (2001) Estimating the number of clusters in a data set via the gap statistic. J R Stat Soc Ser B (Stat Methodol) 63(2):411–423
16. Tukey JW (1975) Mathematics and the picturing of data. Proc Int Congr Math 2:523–531
17. Vardi Y, Zhang CH (2000) The multivariate l1-median and associated data depth. Proc Natl Acad Sci 97(4):1423–1426