

Curvature-based method for determining the number of clusters



Yaqian Zhang^a, Jacek Mańdziuk^{b,a,*}, Chai Hiok Quek^a, Boon Wooi Goh^a

^aSchool of Computer Science and Engineering, Nanyang Technological University, Singapore

^bFaculty of Mathematics and Information Science, Warsaw University of Technology, Warsaw, Poland

ARTICLE INFO

Article history:

Received 15 October 2016

Revised 11 April 2017

Accepted 16 May 2017

Available online 19 May 2017

Keywords:

k-Means clustering

Number of clusters

Cluster analysis

Gap statistic

Hartigan's rule

ABSTRACT

Determining the number of clusters is one of the research questions attracting considerable interests in recent years. Majority of the existing methods require parametric assumptions and substantiated computations. In this paper we propose a simple yet powerful method for determining the number of clusters based on curvature. Our technique is computationally efficient and straightforward to implement. We compare our method with 6 other approaches on a wide range of simulated and real-world datasets. Theoretical motivation underlying the proposed method is also presented.

© 2017 Elsevier Inc. All rights reserved.

1. Introduction

Many clustering algorithms suffer from the limitation that the number of clusters has to be specified by a human user [21,28,34]. However, as Salvador and Chan pointed out [27], in most cases, users do not have sufficient domain knowledge or prior information to select the correct number of clusters to return. Consequently, there have been a number of approaches published in the literature for choosing the right k after multiple runs of k -Means [4,13,16], being the most popular machine learning (ML) clustering algorithm. The notion of a *cluster* is not uniquely-defined as it heavily depends on a particular form of the evaluation function. In order to find the appropriate number of clusters some approaches [27,31,33] construct an evaluation graph where the x-axis is the number of clusters and the y-axis is the corresponding evaluation function value. The properties of such evaluation graph are exploited to identify the right cluster number. One of the basic ideas is to find the *knee/elbow* of the curve. We use within-cluster variance as the evaluation metric and plot the evaluation graph of number of clusters vs. within-cluster variance as shown in Fig. 1.

The evaluation graph is always monotonically decreasing. However, intuitively one would expect much smaller drops for k greater than the true number of clusters because beyond this point, adding more centers simply partitions within groups rather than between groups [31]. For Gaussian clusters presented in Fig. 1, we can visually inspect the *knee* which corresponds to the correct number of clusters.

However, the problem of finding the *knee* of the curve is indeed far from being trivial. The visual inspection method can sometimes be ambiguous especially when there is a high degree of intermix between the clusters. Salvador and Chan [27] proposed to determine the *knee* by finding the pair of lines that most closely fit the curve (with the minimum total

* Corresponding author at: Faculty of Mathematics, Warsaw University of Technology, Koszykowa 75, 00-662, Warsaw, Poland.

E-mail addresses: mandziuk@mini.pw.edu.pl, J.Mandziuk@mini.pw.edu.pl (J. Mańdziuk).

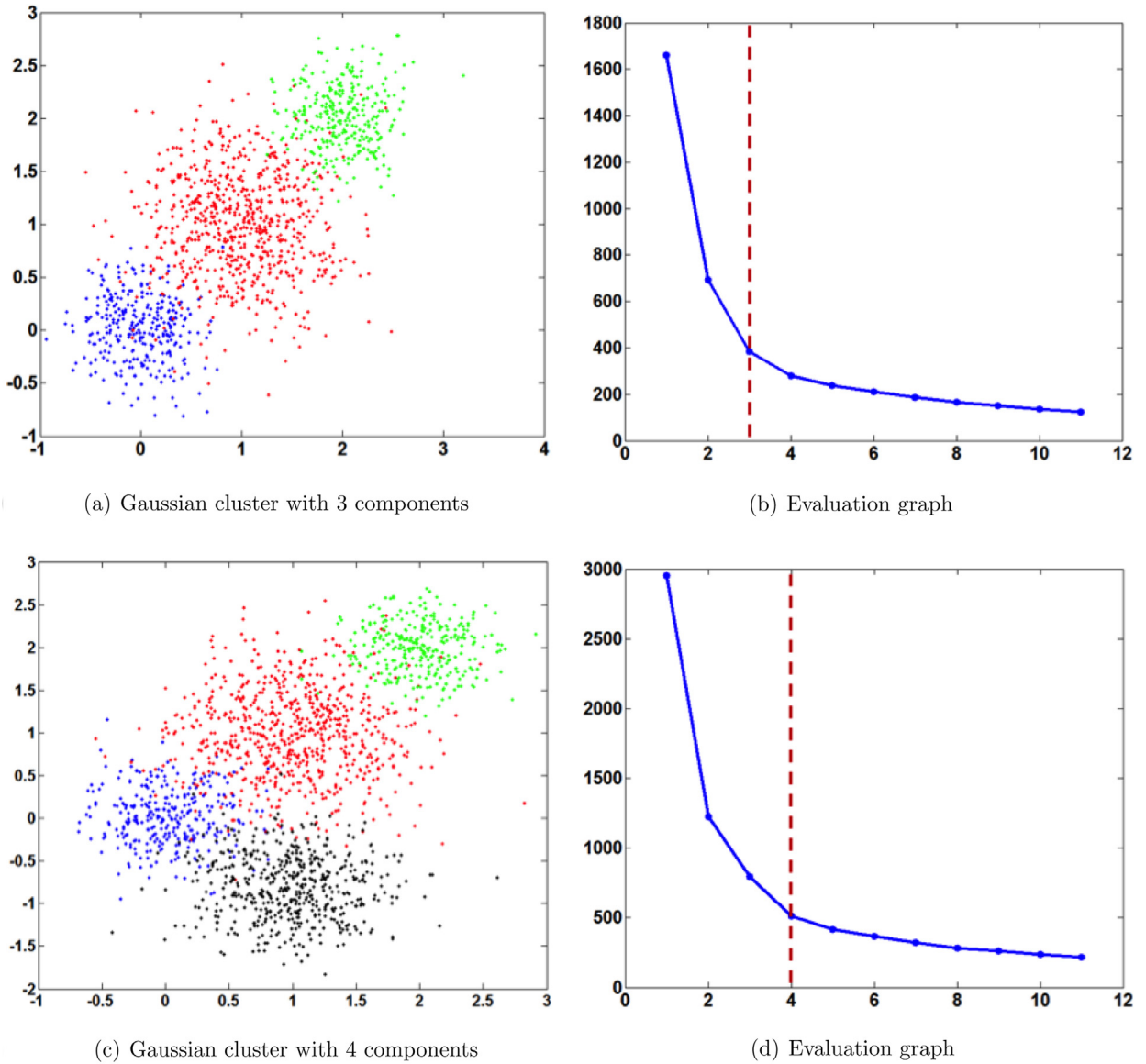


Fig. 1. Visual inspection of the *knee* in the evaluation graph.

root square error) and the intersection of these two lines is returned as the *knee*. However, this method is mainly focused on hierarchical clustering and segmentation algorithms of which the evaluation graph is usually non-smooth and not monotonically decreasing/increasing.

In this paper, we propose a new method to find the *knee* of the evaluation graph by analyzing and exploiting the curvature of the evaluation graph. Actually, both [11] and [27] briefly mentioned the idea of employing the maximum curvature point to identify the number of clusters, however none of them formally defined and applied curvature-based method or discussed the relative challenges and limitations. In this paper, we provide an in-depth discussion on how to use curvature to find the *knee* in the evaluation graph. The within-cluster variance is used as the evaluation metric for the purpose of discussion.

Our main contribution is threefold. Firstly, we exploit curvature to find the *knee* in the evaluation graph to reduce the ambiguity inherent in the process of visual inspection. Secondly, we analyze the challenges and limitations of the curvature-based method. Finally, we propose a new heuristic rule to find the cluster number based on the curvature. The improved method is evaluated on a wide range of synthetic and real-world datasets.

The paper is organized as follows. The next section gives an overview of the related work. Section 3 introduces the curvature-based method to facilitate the visual inspection of the *knee* on the evaluation graph in the task of finding the appropriate number of clusters. Section 4 further analyzes the problem and highlights some challenges related to the use of

curvature to determine the number of clusters. The improvement of the baseline approach based on theoretical analysis of the method is proposed in Section 5. Section 6 contains experimental evaluations of the proposed algorithm and compares our results with those of six other heuristic methods of comparable computational complexity. Section 7 summarizes this study and discusses future research directions.

2. Related work

The issue of determining the clustering number k is a major challenge in cluster analysis. To address this problem, numerous approaches have been suggested over the years. A popular approach is to use an evaluation graph that is constructed by plotting within-cluster variance $J(k)$ for a clustering procedure against the number of clusters k employed. The issue of using the raw $J(k)$ to identify the number of clusters k is that $J(k)$ itself monotonically decreases when k increases. Nonetheless, as Sugar et al. [31] pointed out, all of the requisite information for choosing the correct cluster number is contained in the evaluation graph. Some previous works have analyzed the evaluation graph in the presence and absence of the clusters and trying to find out more sensitive characteristics to determine the cluster number.

Some early efforts proposed heuristic indexes to determine cluster numbers. Calinski et al. [4] suggested an index with F -test form based on within-cluster variance. The method is the best performer in the experiments conducted by Milligan and Cooper [23]. Krzanowski and Lai also derived a criterion using within-cluster variance for choosing clustering number and proposed a plausible stopping rule [16]. In their work, this criterion outperformed Marriott's approach [22], which used within-cluster determinant, rather than within-cluster variance.

Another popular heuristic rule was developed by Hartigan et al. [13] based on the intuition that for $k < k^*$, where k^* is the optimal cluster number, $J(k+1)$ is drastically smaller than $J(k)$, however, for $k > k^*$, $J(k+1)$ and $J(k)$ are not that different. In the experimental study comprising 8 cluster methods [6], Chiang et al. found that the Hartigan's rule can give potentially the best performance in terms of reproducing cluster number k , however the performance deteriorates quickly when the clusters are not well separated.

Besides deriving measurements based on within-cluster measure, some approaches compared the within-cluster cohesion with between-cluster separation. Kaufman and Rousseeuw [15] introduced the concept of silhouette width to measure how well each point is clustered by difference between within-cluster tightness and separation from other groups. This method demonstrated good performance in the experiment conducted by Pollard et al. [24].

Recently, an approach to the problem of estimation of the number of clusters which relies on cross-validation method was proposed by Fu and Perry [10]. The authors consider the task of choosing an optimal value of k as a model selection problem and address it via a form of Gabriel cross-validation. The value of k with the smallest prediction error is selected. The experiments show that the proposed method has competitive performance, especially in high-dimensional settings with heterogeneous or heavy-tailed noise.

In addition, there are some recent studies utilizing model-based measures to examine the *knee* phenomenon. Tibshirani et al. [33] proposed a statistical procedure (gap statistic) to formalize the heuristic process of finding the location of the *knee* on the evaluation graph. The idea is to compare the evaluation graph with its expectation under an appropriate null reference distribution of the data. The method is widely used in bioinformatics community. Unfortunately, it requires heavy computation and may even fail for larger datasets because of the matrix computing problem.

Based on the Gaussian distribution model, Sugar et al. [31] introduced the ideas from the field of rate distortion theory to examine the graph's functional form in both the presence and absence of clustering. From their mathematical derivation and empirical studies, the graph, when transformed to an appropriate negative power, can exhibit a sharp jump at the optimal cluster number. This method is computationally efficient but picking an appropriate transformation power is a non-trivial problem.

Besides exploring the property of evaluation graph which is focused on post-processing the results of clustering algorithms to determine the number of clusters, there are some studies on the clustering methods in which the number of clusters can be automatically founded. A recent method proposed by Rodriguez and Laio [25] is a density-based clustering approach. Cluster centers are identified as points with higher densities than their neighbors and by relatively large distances from the points with higher densities, and the number of clusters arises intuitively after the cluster centers are determined. Tasdemir et al. [32] proposed an automated clustering method for self-organizing maps (SOMs). In this method the number of clusters is determined either by using various cluster validity indices or by prior knowledge on the considered dataset.

Mathematical formulation of the main aspects of the above-mentioned approaches is presented in Section 6.

3. Cluster number at maximum of curvature

For the simplicity of discussion, we use the cluster results of k -Means to compute the within-cluster variance as the evaluation metrics to construct evaluation graph:

$$J(k) = \sum_{j=1}^k \sum_{x_i \in C_j} ||\mathbf{x}_i - \bar{\mathbf{x}}_j||^2 \quad (1)$$

where C_j is the set of samples belonging to class j and $\bar{\mathbf{x}}_j$ is the sample mean of class j .

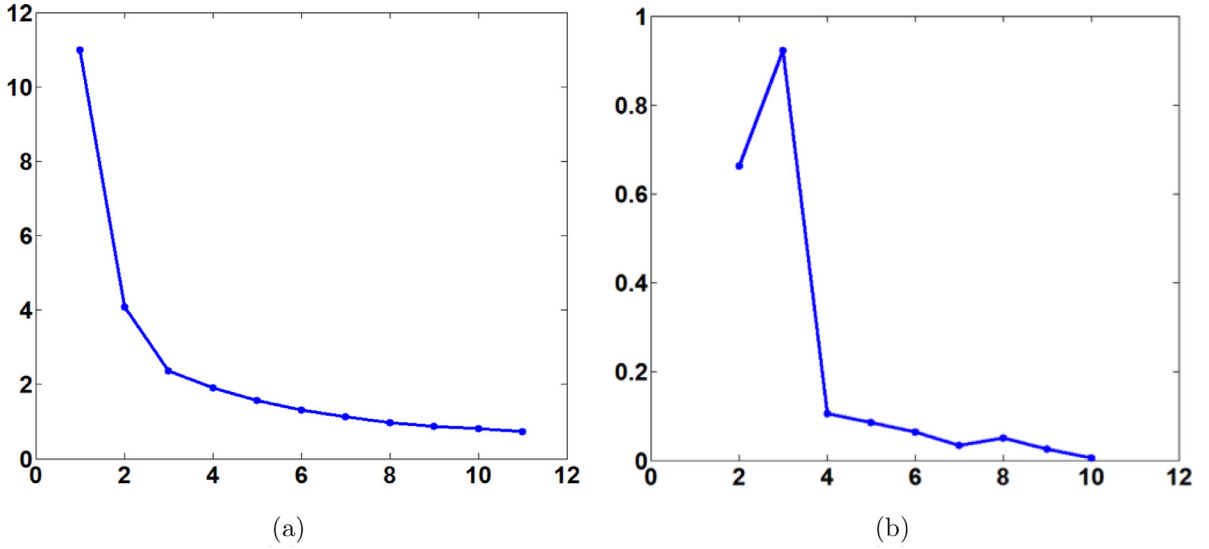


Fig. 2. Data set Seed [5] with real class number equal to 3: (a) Scaled cost function of k -Means; (b) Curvature of the scaled cost function.

We propose to use curvature to identify the *knee* of the evaluation graph (1) in order to reduce the ambiguity stemming from the process of visual inspection. In mathematics, curvature is the amount by which a geometric object deviates from being flat, or straight in the case of a line. So the *knee* in the graph should correspond to the point with the maximum curvature. For a curve explicitly given as $y = f(x)$, the curvature is defined as:

$$\kappa = \frac{|y''|}{(1 + y'^2)^{3/2}} \quad (2)$$

As an example the Curvature method is applied to the real-world dataset Seed [5] from the University of California Irvine Machine Learning Repository [19] (UCI), which contains real application data collected in various fields and is widely used to test the performance of different machine learning algorithms [35,36]. The wheat varieties, Kama, Rosa and Canadian, characterized by measurements of main grain geometric features obtained by X-ray technique, have been analyzed. The within-cluster variance and the corresponding curvature graph are presented in Fig. 2(a) and (b), respectively. As showed in Fig. 2b the true cluster number (equal to 3) in fact corresponds to the maximum curvature point.

4. The challenge in the curvature-based method

While computing the curvature, a critical problem was discovered. It was observed that the position of the maximum curvature point changes when the original data is rescaled. In particular, when the original data is rescaled by a as shown in Eq. (3),

$$\mathbf{x}_i = a\mathbf{x}_i, \quad a \in \mathbb{R}^+ \quad (3)$$

the within-cluster variance exhibits a linear change as indicated in Eq. (4):

$$\begin{aligned} J_a(k) &= \sum_{c=1}^k \sum_{\mathbf{x}_i \in C_j} \|a\mathbf{x}_i - a\bar{\mathbf{x}}_j\|^2 \\ &= a^2 \sum_{c=1}^k \sum_{\mathbf{x}_i \in C_j} \|\mathbf{x}_i - \bar{\mathbf{x}}_j\|^2 \\ &= a^2 J(k) \end{aligned} \quad (4)$$

Curvature is a kind of geometric property of the graph and tightly related to the range of the two axes. In the evaluation graph, the x -axis is the number of clusters, the difference of which is always one and the y -axis is the within-group variance, the range of which lies in a large variety (often much bigger than the range of x). When we rescale the original data, the x -axis remains the same and the y -axis has a linear change as indicated in Eq. (4).

However, when the curvature is computed from the rescaled evaluation graph as in Eq. (5), the change in the curvature is non-linear. For each k , the change of curvature $\beta(k)$ is not only related to a , but also to J' . This non-linear change in the curvature will cause the shift of the maximum curvature point. In fact, it can be easily proven that for $a > 1$ (when the

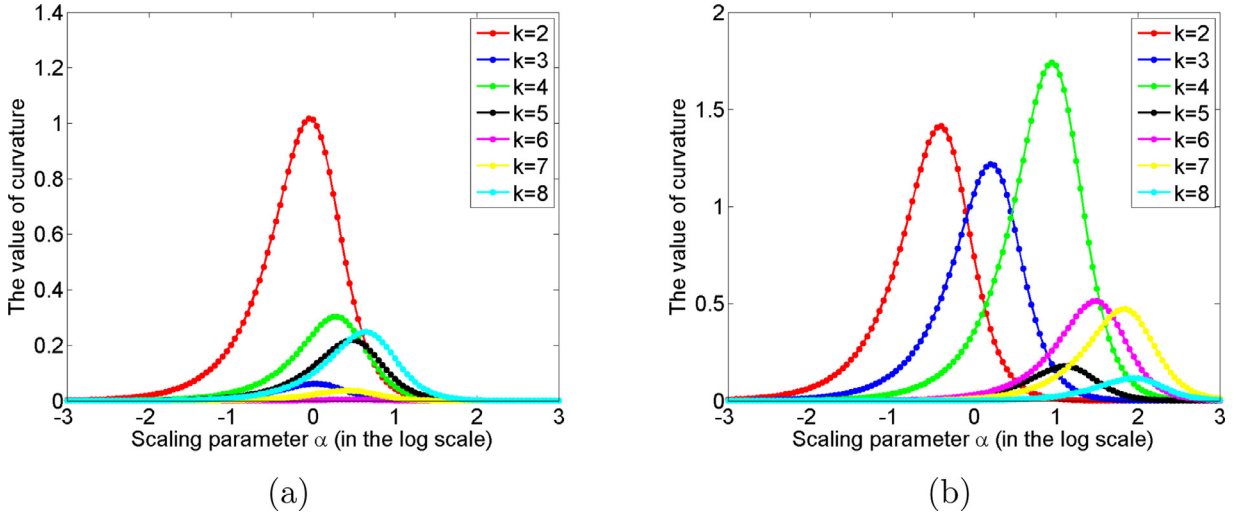


Fig. 3. Plot of curvature-scale parameter for two data sets: (a) *Ionosphere* [29] (the real class number = 2); (b) *Breast tissue* [14] (the real class number = 4).

data is enlarged), the maximum curvature point moves rightwards and for $a < 1$ (i.e. the data is shrunk), the point with maximum curvature moves leftwards.

$$\kappa_a(k) = \frac{|a^2 J''|}{(1 + a^4 J'^2)^{3/2}} = \beta(k) \kappa(k) \quad (5)$$

where

$$\beta(k) = a^2 \left(\frac{1 + a^4 J'^2}{1 + J'^2} \right)^{3/2}$$

As we know, while rescaling the data, the cluster structure in fact remains the same and so does the cluster number. Therefore the raw curvature itself is a poor indicator of cluster number, although it can serve as an effective way to identify the *knee* of the evaluation graph. It should be noted that the traditional *knee* method suffers from the same scaling problem. When the within-cluster variance against k is plotted, the software usually automatically scales the range of axes for representation purpose because the range of within-cluster variance is often much bigger than k (c.f. Fig. 1). When the *knee* of the graph is inspected visually, it is actually being examined under some scaling factor and thus the results may be unreliable.

The behavior of the curvature is further examined and discussed in the next section. Our goal is to eliminate the influence of the scaling factor and at the same time still exploit the usefulness of curvature in the detection of the *knee* on a graph. To this end a new curvature-based index is proposed which does not depend on the scaling factor.

5. Beyond curvature

Firstly, we analyze the impact of the scaling parameter on the change of curvature for each k . For convenience, let us define a scale parameter $\alpha = a^2$. For each point on the graph, we plot curvature κ against the scaling parameter α using two data sets (*Ionosphere* [29] and *Breast tissue* [14]) from the UCI (see Fig. 3). Examining the curve one can observe the following properties:

- (i) All the curves are bell-shaped lines. On each line, the peak occurs between $\alpha = 10^{-3}$ and $\alpha = 10^3$. The curvature approaches zero when $\alpha \ll 10^{-3}$ or $\alpha \gg 10^3$.
- (ii) The location of the peak depends on k .
- (iii) Also, the peak value differs with respect to k .

Furthermore, there is an interesting phenomenon that the k with the highest peak value corresponds to the true cluster number for the two datasets in our experiment. In *Ionosphere* dataset, which has two clusters, the peak value at $k = 2$ is the highest and for the *Breast tissue* dataset the highest peak appears at $k = 4$, which again corresponds to the true number of clusters. In the reminder of this section, this phenomenon is further investigated from the mathematical point of view.

From the analysis presented in Section 3, we find the curvature is related to both cluster number k and scale parameter α as:

$$\kappa(\alpha, k) = \frac{|\alpha J''(k)|}{(1 + \alpha^2 J'(k)^2)^{3/2}} \quad (6)$$

Our goal is to focus on the influence of k and to eliminate the effect of α . So we propose to choose the optimal k by solving the following optimization problem.

$$K = \arg \max_k \max_{\alpha} \kappa(\alpha, k) \quad (7)$$

Let us start with computing $\frac{\partial \kappa}{\partial \alpha}$

$$\frac{\partial \kappa}{\partial \alpha} = \frac{|J''|}{(1 + \alpha^2 J'^2)^{\frac{5}{2}}} (1 - 2\alpha^2 J'^2), \quad \alpha > 0 \quad (8)$$

From Eq. (8) we can see that κ is a concave function with respect to α . For each cluster number k , κ reaches its maximum value if and only if $\alpha = \frac{1}{\sqrt{2}J'}$. The maximum value is denoted as Eq. (9):

$$\max_{\alpha} \kappa(\alpha, k) = \frac{1}{\sqrt{2}(\frac{3}{2})^{\frac{3}{2}}} \times \frac{J''}{J'} \quad (9)$$

Based on Eq. (7) and Eq. (9), we choose the k with the highest peak value to be returned, i.e.

$$K = \arg \max_k \frac{J''(k)}{J'(k)} \quad (10)$$

Now, let us explore the meaning of Eq. (10). Let us define

$$\det_k = J(k-1) - J(k)$$

that describes the decrease of within-cluster variance (i.e. the increase of between-cluster variance) from $k-1$ clusters to k clusters. Since within-cluster variance $J(k)$ is monotonously decreasing, we have

$$\det_k \geq 0, k = 2, 3, \dots$$

For each cluster number k , Eq. (10) can be rewritten as:

$$\begin{aligned} \arg \max_k \frac{J''(k)}{J'(k)} &= \arg \max_k \frac{\det_k - \det_{k+1}}{\det_{k+1}} \\ &= \arg \max_k \left(\frac{\det_k}{\det_{k+1}} - 1 \right) \\ &= \arg \max_k \frac{\det_k}{\det_{k+1}} \end{aligned} \quad (11)$$

We can see that the method is determined by maximizing $\frac{\det_k}{\det_{k+1}}$. Therefore, the Curvature method should rely on computing the maximum ratio of two consecutive decreasing amounts for each k . The method is in favor of bigger \det_k and smaller \det_{k+1} such that a decrease of within-cluster variance from $k-1$ to k is relatively large while from k to $k+1$ is relatively small. This conforms to the *knee* method, which is based on the idea that one should choose a number of clusters so that adding another cluster would not provide much better modeling of the data. Hence the proposed method can be summarized as in Table 1.

Table 1
The Curvature method.

Algorithm
for $k=1:(k_{\max}+1)$
for $t=1:20$
Run the k -Means algorithm and calculate the within-cluster variance $j(k, t)$
Take the minimum within-cluster variance across multiple times, $J(k) = \min_t j(k, t)$
if $k > 1$
Calculate the difference in within-cluster variance, $\det_k = J(k-1) - J(k)$
for $k=2:k_{\max}$
Compute the Curvature index, $r(k) = \frac{\det_k}{\det_{k+1}}$
Return the optimal number of clusters K corresponding to maximum value of $r(k)$, $K = \arg \max_k r(k)$

6. Experimental results

6.1. Baseline approaches

We compared our method with 6 other well-known approaches of comparable computational complexity, mentioned in Section 2.

The CH [4] method, in short, chooses the number of clusters as the argument maximizing Eq. (12) where $J(k)$ is within-cluster variance with k clusters and n is the number of observations.

$$CH(k) = \frac{(J(1) - J(k))/(k - 1)}{J(k)/(n - k)} \quad (12)$$

The approach of Krzanowski and Lai [16] maximizes $KL(k)$ given by Eq. (13):

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

$$DIFF(k) = (k-1)^{2/p} J(k-1) - K^{2/p} J(k) \quad (13)$$

where p is a dimension of the data which is used for adjustment of $DIFF(k)$.

Hartigan et al. [13] proposed choosing the smallest value of k such that $H(k) \leq 10$ in Eq. (14). $H(k)$ is effectively a partial F statistic for testing whether it is worth adding a $(k+1)$ st cluster to the model:

$$H(k) = (n - k - 1) \left[\frac{J(k)}{J(k+1)} - 1 \right] \quad (14)$$

Kaufman and Rousseeuw [15] proposed silhouette width as shown in Eq. (15), measuring how well the i th point is clustered. The term $a(i)$ is the average distance between the i th point and all other observations in its cluster, and $b(i)$ is the average distance to points in the nearest cluster, where nearest is defined as the cluster minimizing $b(i)$. The number of clusters that maximizes the average value of $s(i)$ is chosen:

$$s(i) = \frac{b(i) - a(i)}{\max[a(i), b(i)]} \quad (15)$$

The Gap approach developed by Tibshirani et al. [33] is described in Eq. (16):

$$Gap(k) = \frac{1}{B} \sum_b \log(J_b^*(k)) - \log(J(k)) \quad (16)$$

In this method, B different uniform datasets, each with the same range as the original data, are produced, and the within-cluster variance is calculated for different numbers of clusters. $J_b^*(k)$ is the within-cluster variance for the b th uniform dataset. To avoid adding unnecessary clusters, an estimate S_k of the standard deviation of $\log(W_b^*(k))$ is produced, and the smallest value of k is chosen as the number of clusters, such that

$$Gap(k) \geq Gap(k+1) - S_{k+1}$$

Finally, the Jump method proposed by Sugar et al. [31] maximizes the $Jump(k)$ given in Eq. (17).

$$Jump(k) = J(k)^{-p/2} - J(k-1)^{-p/2} \quad (17)$$

The transformation parameter p is typically chosen as half of the space dimension.

6.2. Experimental results on synthetic data

In this section the performance of the proposed Curvature method is investigated using synthetic data. Firstly, we compare the accuracy of estimating the optimal number of clusters with our method with the other six chosen approaches. Secondly, we examine the ability of the Curvature method to identify hierarchical cluster structure. Lastly, we present the performance of Curvature index when there are different extents of intermix/overlap between clusters.

6.2.1. Estimation of the number of clusters

Some works in the literature have discussed data generation issues for experimental comparison of various methods. Here we borrowed the basic ideas from [6,31] and designed the experiment settings considering the factors of within-cluster spread, between-cluster separation, the number of dimensions, the dependence between dimensions as well as Gaussian/non Gaussian distribution structure. Specifically, we used the following 5 data arrangements (simulations) to test the method:

- (i) Five basic Gaussian clusters in 2 dimensions. This simulation is designed to test the performance on basic Gaussian clusters. One cluster is placed in the middle, and four other clusters are spaced with a separation of 2.5 from the center cluster in each dimension (see Fig. 4(a)).
- (ii) Five elongated clusters in 2 dimensions. This simulation is aimed at investigating the performance of the methods when there was some dependence among the dimensions. Specifically, there is a correlation of 0.7 between the dimensions. The placement of clusters is the same as in the previous case (see Fig. 4(b)).
- (iii) Five clusters with different shapes in 2 dimensions. This simulation is designed to test the effect of differing correlation. The correlations for 5 clusters are 0.0, 0.7, 0.3, 0.3 and 0.7. The placement of clusters is the same as in the two previous cases (see Fig. 4(c)).

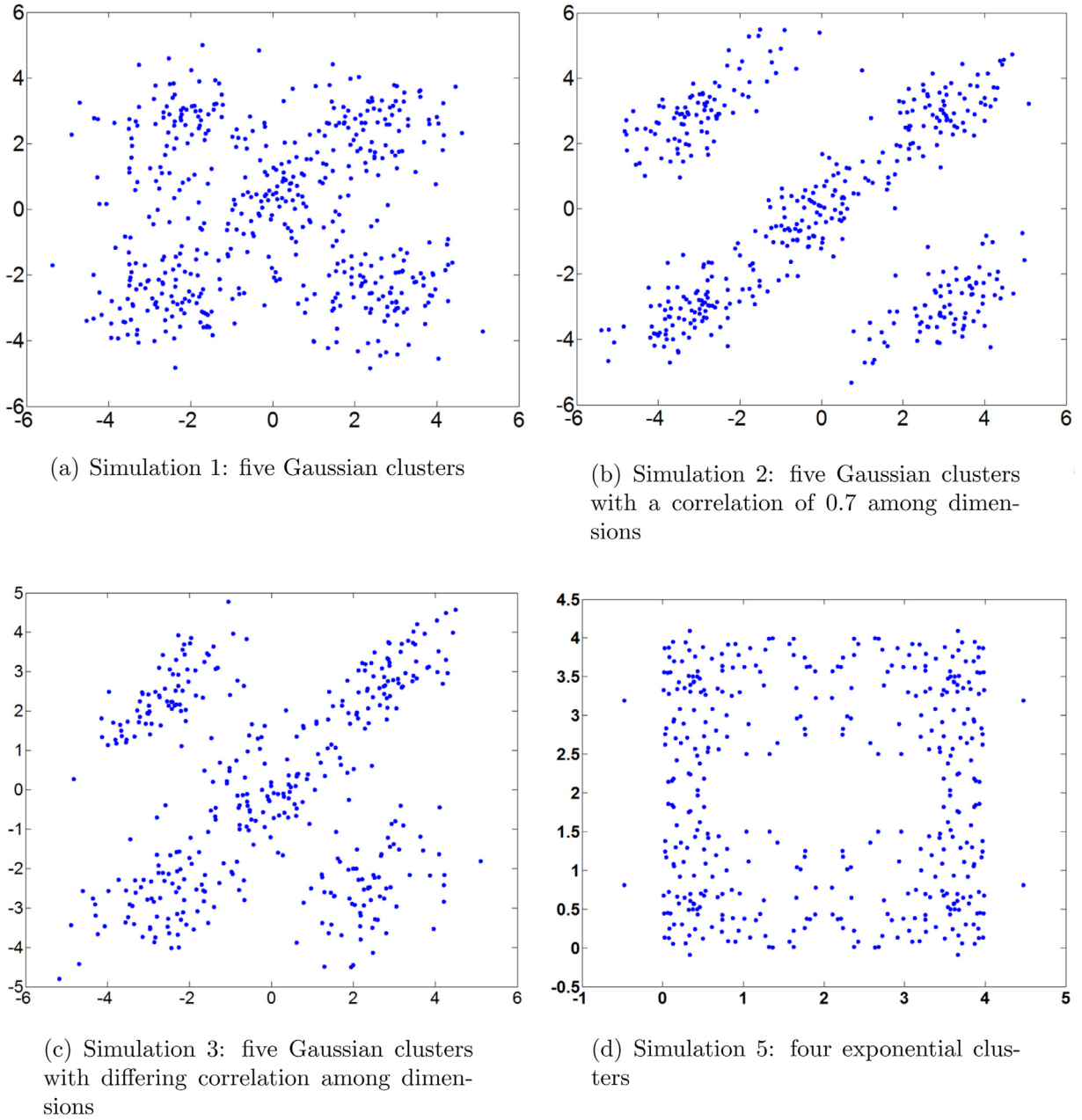


Fig. 4. Data generated in simulations 1,2,3 and 5.

- (iv) Five Gaussian clusters in 10 dimensions. In this simulation, the performance of approaches on highly multivariate data is examined. A basic 10-dimensional mixture of five Gaussian clusters is generated. The five clusters are spaced on a line with a separation of 1.6 in each dimension.
- (v) The last simulation is used to compare the methods on non-Gaussian data. We generated 4 clusters in 2 dimensions using exponential distributions with mean 1 independently in each dimension. The clusters were arranged on a square with sides of length 4 (see Fig. 4(d)).

All the above-described data arrangements have standard deviations of 1 in each dimension. In simulations 1–4, the distances between the centers of the middle clusters and the centers of surrounding clusters are equal to 2.5, 3.0, 2.5 and 1.6, respectively, in each dimension. Each simulation set consists of 400 observations equally divided among clusters.

Initially, for each simulation, 50 datasets were randomly generated. Next, for each dataset we ran k -Means algorithm with 20 restarts and then applied the Curvature method and the 6 other methods to estimate the optimal number of clusters.

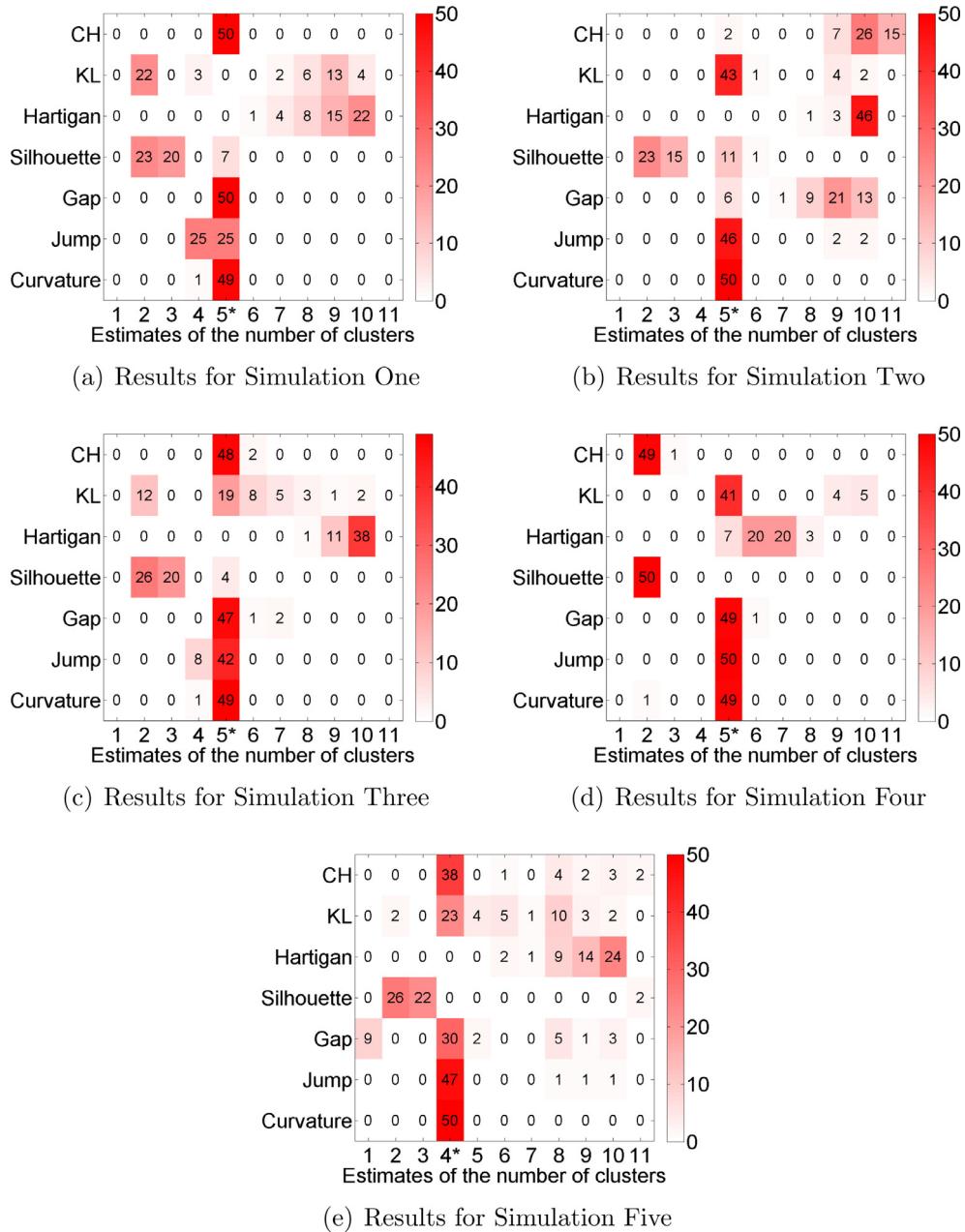


Fig. 5. Results for synthetic data illustrated using heat maps. The numbers represent the respective counts of 50 trials for each method in each simulation. The true number of clusters are highlighted with star (*) symbols.

The results are shown in Fig. 5. Our proposed method (Curvature) achieved at least 98% accuracy in each of the 5 simulations, which was visibly the highest performance score. Among the comparative approaches, the best score was accomplished by the Jump method (Jump), followed by Gap algorithm (Gap).

6.2.2. Detection of hierarchical cluster structure

In Section 6.2.1 we examined the performance of the proposed method in terms of estimating the number of clusters by maximizing the Curvature index. In this section, we further analyze the reasonability of our proposed index by investigating other properties of the index graph aside from the maximum point. In particular, we discuss the information provided by the peak in the index graph as well as the second maximum point.

We designed a simulation with hierarchical cluster structure. More specifically, we produced a two-dimensional mixture of six Gaussian clusters evenly spaced in 3 distinctive groups as shown in Fig. 6. Each group consists of two components

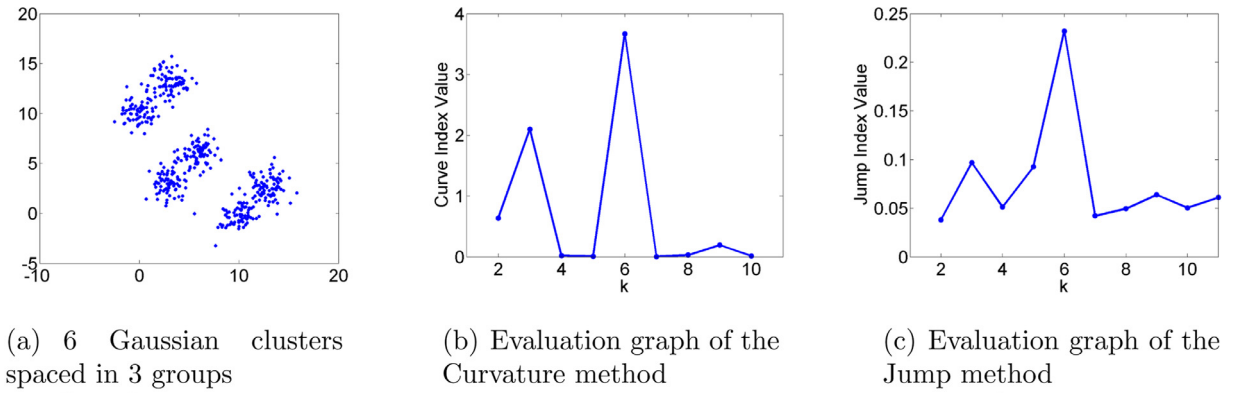


Fig. 6. Performance in the task of detection of hierarchical clusters structure with 6 clusters evenly spaced in 3 groups.

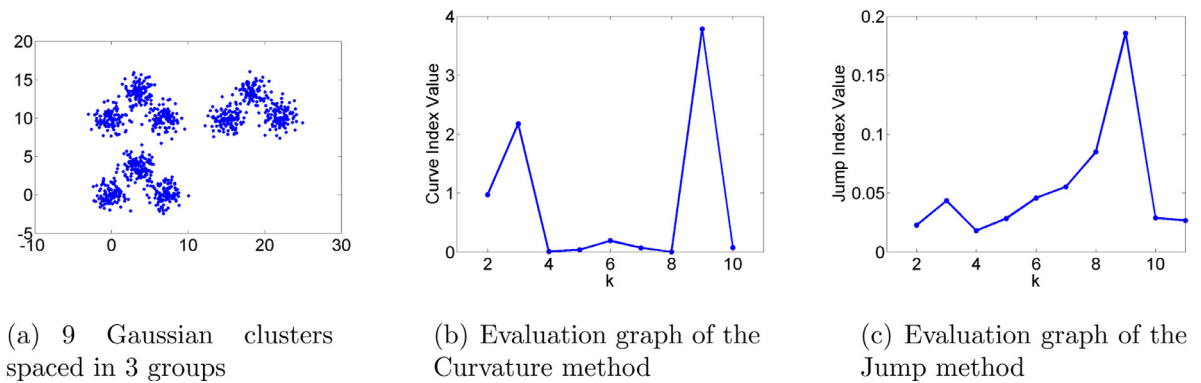


Fig. 7. Performance in the task of detection of hierarchical clusters structure with 9 clusters spaced in 3 groups.

spaced in a line with a separation of 2. In this experiment our method is compared with the Jump algorithm since among 6 approaches the ability to identify cluster structure was reported only for that method.

The results are shown in Fig. 6. Both methods returned number 6 as optimal candidate for the number of clusters and presented two peaks at $k=6$ and $k=3$. Therefore, both of them demonstrated the *in principle* ability to detect hierarchical cluster structure. Note however, that in the case of our method, the two peaks (corresponding to $k=6$ and $k=3$) are very distinctive and a deep valley appears between them (see Fig. 6(b)). The values assigned to other selections of k are penalized as being very small. On the other hand, the two peaks are less distinctive in the Jump method (see Fig. 6(c)). In fact, the peak for $k=3$ is too small to provide any reliable information about hierarchical cluster structure in the practical application. And the value for $k=5$ is even a little higher than that for $k=3$ and is returned as the second candidate. Since the structures within three groups are identical, we believe that $k=5$ is a wrong estimation of the second candidate.

Another two examples of hierarchical cluster structure detection are given in Figs. 7 and 8. In Fig. 7, the simulation consists of 9 clusters spaced in 3 identical groups (see Fig. 7(a)). Each group contains 3 clusters. In the result of Curvature method, there are distinctive peaks at $k=3$ and $k=9$ (see Fig. 7(b)). Nevertheless, in the Jump approach, the values for $k=6, 7, 8$ are all greater than the value at $k=3$, which means the detection of hierarchical cluster structure is unsuccessful (see Fig. 7(c)). Similarly, Fig. 8 shows the results for 6 Gaussian clusters spaced in 3 groups which contain 1, 2, 3 clusters respectively. The means values of 6 clusters are (0, 3), (−1.5, 0), (−3, −12), (0, −12), (3, −12), (10, −5). Again, the Curvature approach identifies a large peak at $k=3$ and a small peak at $k=6$. However, the Jump approach fails to detect the correct number of clusters in this simulation. The experiment results for these two examples provide evidence that our approach is more effective in cluster structure detection than the Jump method.

All in all, the results suggest that some useful information can be obtained from the index graph in our method, for example the distinctive peaks and the second maximum point can be used as an effective hint for the existence of hierarchical clusters.

The above discussion illustrates the potential usage of index graph generated in our method in detection of hierarchical cluster structure. We believe this ability is an additional support for the cluster descriptive characteristics of the Curvature index. For each k , the Curvature index can give a more accurate description of the likelihood of the data containing k clusters compared to the Jump index.

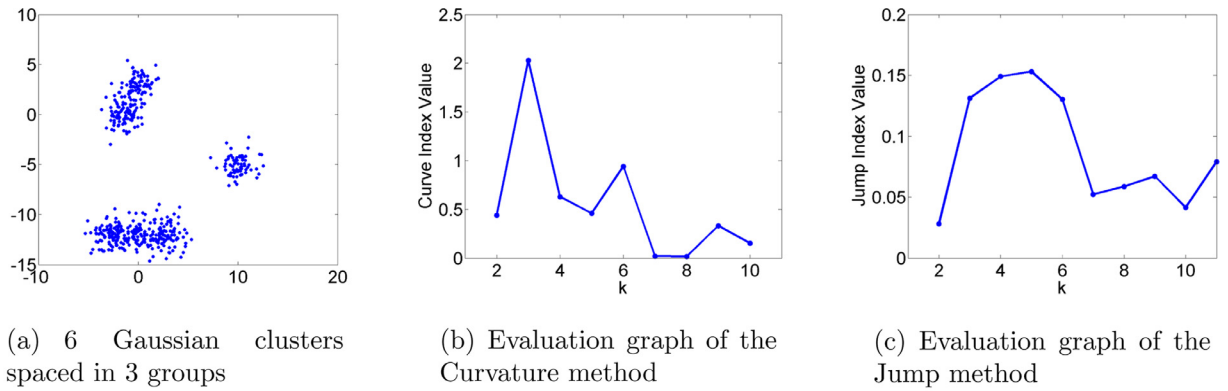


Fig. 8. Performance in the task of detection of hierarchical clusters structure with 6 clusters unevenly spaced in 3 groups.

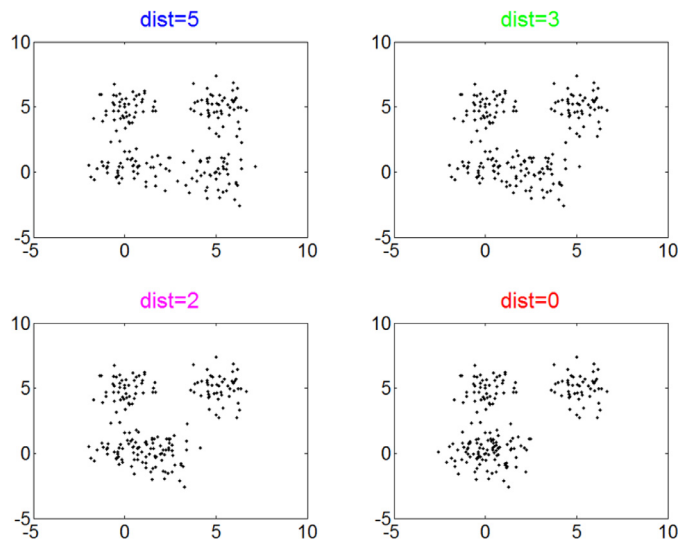


Fig. 9. Simulated compounded datasets.

6.2.3. Performance on compounded data

In this section, we investigate how the Curvature index value changes when the extent of intermix between the clusters varies. For this purpose, we generate four Gaussian clusters in two dimensions which are spaced in a square with the side length of 5. Each cluster has a standard deviation of 1 in each dimension. Then we introduce the intermix between clusters by moving one cluster closer to another. More specifically, we vary the distance between two clusters from 5 to 0. Fig. 9 presents the datasets with distances equal to 5, 3, 2 and 0. It is clear from the figure that the initial dataset contains 4 clusters and the last one contains 3 clusters. The second and the third ones record the transition from 4 to 3 clusters. The Curvature index graphs are shown in Fig. 10. Our Curvature method estimates the first two datasets as consisting of 4 clusters and the other two as consisting of 3 clusters.

Since the definition of a cluster is not precise, we choose not to judge whether the results are correct or not. Instead we analyze the reasonableness of the Curvature index. Firstly, it can be observed from Fig. 10(a) that for all the four datasets the maximum point appears either at $k=3$ or $k=4$ and the values for other k are very low, which indicates the low noise level of the proposed Curvature index. Secondly, as the distance between the two clusters increases, the Curvature index at $k=4$ decreases and Curvature index at $k=3$ increases. This phenomenon corresponds very well to the fact that the data is actually transformed from 4 to 3 clusters. Another interesting observation is that the maximum peaks in the index graph of datasets 1 and 4 are more distinctive than the peaks in datasets 2 and 3. This conforms to the fact that datasets 1 and 4 have clearer cluster structure than the other two datasets.

We examine the performance of the 6 existing approaches using the four datasets presented in Fig. 9. The results are shown in Fig. 11. It can be seen that the maximum points generally appear either at $k=3$ or $k=4$ in the evaluation graphs of CH, KL, Hartigan, Gap and Jump methods. However, in the case of CH, Hartigan and Gap plots, these peaks are not obvious (see Fig. 11(a)(c)(e)). Similar to the Curvature method, KL and Jump plots also have distinctive peaks at $k=3$ and $k=4$, but

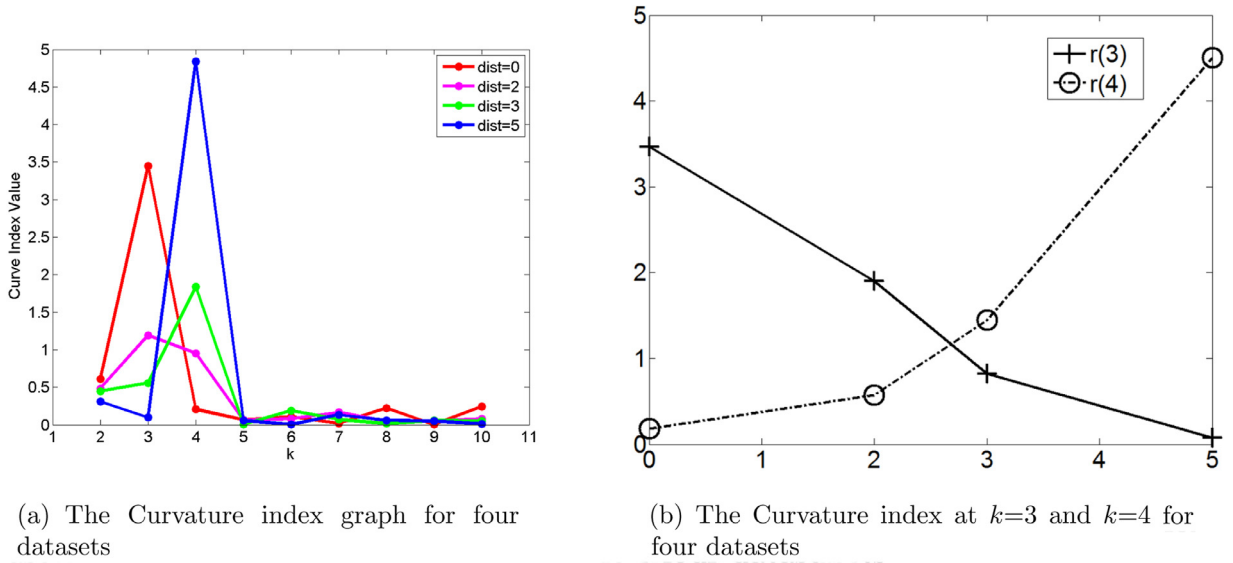


Fig. 10. Curvature index graphs for compounded data.

Table 2

Experimental comparison (first and second candidates) of Curvature method with six other approaches on 20 real-world datasets. A star (*) sign denotes the correct results; a plus (+) sign denotes the data sets, which have two reasonable (alternative) cluster numbers.

Data# set	True# number	CH		KL		Hartigan		Silhouette		Gap		Jump		Curvature	
		1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd	1st	2nd
Spectf [17]	2	2*	3	2*	5	6	7	2*	3	9	5	10	9	2*	3
Ionosphere [29]	2	2*	3	2*	8	8	9	2*	9	9	5	10	9	2*	4
Breast cancer [30]	2	8	9	2*	8	10	8	2*	9	7	5	10	8	2*	3
Parkinsons [20]	2	9	10	4	4	9	10	3	3	8	4	10	9	3	2*
Haberman [12]	2	4	2*	4	4	10	9	2*	4	2*	1	7	4	4	2*
Transfusion [37]	2	9	10	9	7	8	10	2*	3	1	4	9	8	8	9
Magic [3]	2	2*	3	5	5	9	10	2*	3	9	5	9	8	2*	5
Musk [8]	2	3	2*	3	9	8	6	3	6	9	5	9	8	3	10
MiniBooNE [26]	2	2*	3	2*	6	9	10	–	–	–	–	2*	6	2*	3
Skin [2]	2	2*	5	2*	7	8	7	–	–	–	–	10	4	2*	4
Hill [18]	2	8	9	2*	3	10	8	2*	3	6	5	3	7	2*	3
Seed [5]	3	3*	2	3*	2	10	9	2	3*	3*	4	8	9	2	3*
Cardiotocography [1]	3,10 ⁺	3*	7	2	3*	8	5	2	3*	9	4	10*	9	10*	2
Wine [19]	3	10	9	2	7	8	9	2	3*	1	2	10	9	3*	2
Iris [9]	3	3*	4	2	8	8	10	2	3*	9	5	3*	2	2	3*
Sensor [19]	4	2	3	2	4*	10	9	2	3	9	5	10	9	3	4*
Breast tissue [14]	4,6 ⁺	10	9	4*	2	10	9	1	9	4*	5	10	9	4*	2
Vehicle [19]	4	2	6	2	6	9	10	2	3	9	5	10	10	9	2
Winequalityred [7]	6	10	7	2	3	9	7	2	3	1	6*	10	9	7	2
Statlog land [19]	6	3	4	3	4	9	7	2	3	9	5	10	9	3	6*

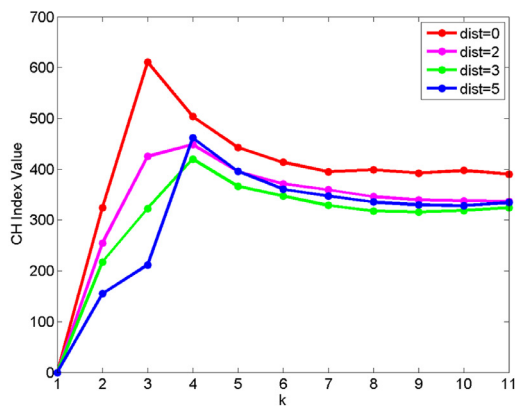
these two methods have much higher noise level than the Curvature method, because the index values are not properly penalized in these approaches for $k > 4$ (see Fig. 11(b), (f) and Fig. 10(a)).

Based on the above results and analysis, we conclude that the Curvature index behaves reasonably and proportionally to increasing intermixing among the clusters.

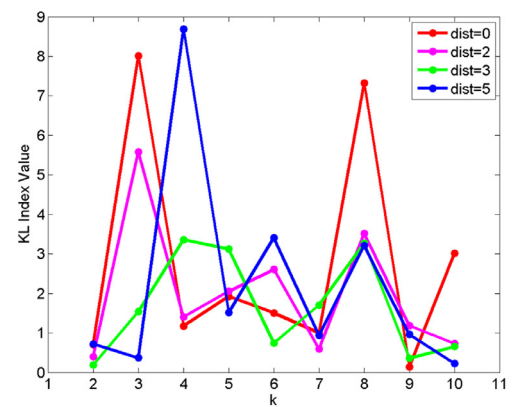
6.3. Experimental results for real-world data

In order to verify the efficacy of the proposed method on non-artificially constructed datasets, we further compared the Curvature method with six other approaches on 20 real-world datasets from the UCI. The detailed results of the experiment are presented in Table 2. The cumulative accuracy in estimating the number of clusters for all tested datasets is given in Table 3.

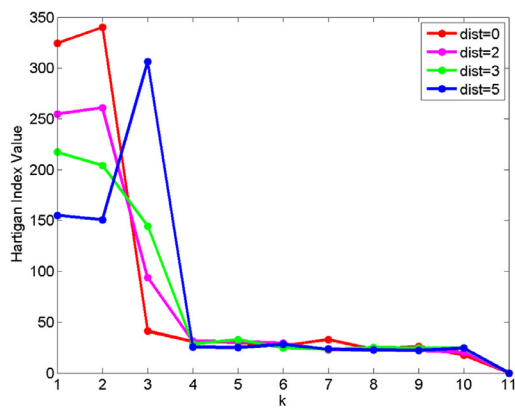
The proposed method also appears to be the most robust among tested algorithms when applied to real-world data sets. It achieved the highest accuracy and was able to produce correct results for 10 out of 20 sets. As the real-world data poses significant challenges to generally all methods, we further present the comparative performance for the correct estimation



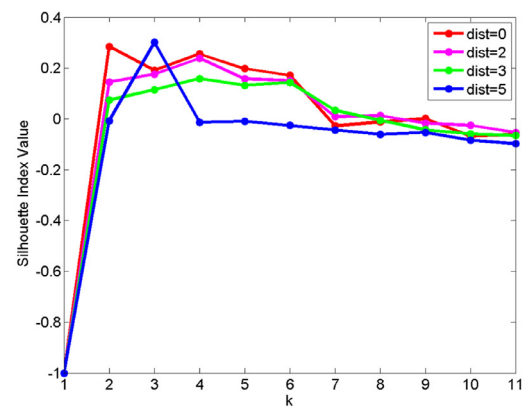
(a) CH Index graph for the four datasets



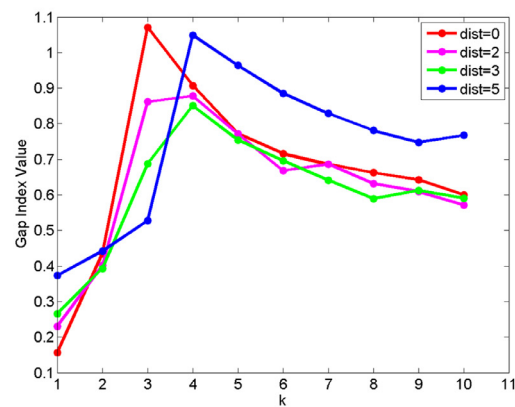
(b) KL Index graph for the four datasets



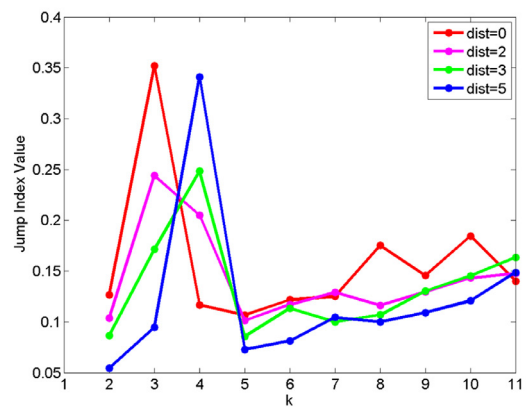
(c) Hartigan Index graph for the four datasets



(d) Silhouette Index graph for the four datasets



(e) Gap Index graph for the four datasets



(f) Jump Index graph for the four datasets

Fig. 11. Performance of the 6 comparative approaches on four datasets presented in Fig. 9.

Table 3

Performance of 7 approaches on 20 real world datasets.

	CH	KL	Hartigan	Silhouette	Gap	Jump	Curvature
<i>First selection accuracy</i>	40%	40%	0	35%	15%	15%	50%
<i>Top-2 selection accuracy</i>	50%	50%	0	55%	20%	15%	80%

Table 4

Performance of 7 approaches on 20 real world datasets depending on the number of clusters.

	CH	KL	Hartigan	Silhouette	Gap	Jump	Curvature
<i>Top-2 selection accuracy ($k < 4$)</i>	67%	53%	0	73%	13%	20%	87%
<i>Top-2 selection accuracy ($k \geq 4$)</i>	0%	40%	0	0%	40%	0%	60%

of the true number of clusters in one of the first two candidates (top-2 selection). In this comparison, the Curvature method achieved 80% accuracy, while the highest accuracy obtained among the other methods was 55%, with the Silhouette method.

In a more detailed comparison of both methods (the Silhouette and the Curvature), we observed that the Silhouette method seems to favor small cluster numbers. In particular, as presented in Table 4, for $k < 4$ its top-2 selection accuracy equals 73.3%, while for $k \geq 4$ the top-2 selection accuracy equals zero. In comparison, the top-2 selection accuracies of the Curvature method were 86.7% and 60%, respectively, suggesting higher robustness and lesser sensitivity to cluster count.

When it comes to the data sets with a large size, such as *MiniBooNE* [26] with 130 064 instances in 50 dimensions or *Skin* [2] with 245 057 instances in 3 dimensions, the Curvature Method has an advantage of computational efficiency. The Silhouette method and the Gap method are unable to handle such datasets due to their matrix computation problem.

7. Discussion

In this work, we proposed a curvature-based method to estimate the optimal number of clusters. The proposed algorithm is computationally efficient and parameter-free. Our comparative evaluation on both synthetic and real-world datasets shows that the proposed Curvature method outperforms six other cluster count estimation algorithms. In addition, empirical results indicate that our method is able to provide reliable information in terms of identifying the underlying hierarchical structure of the data.

Empirical observations suggest that the Curvature method is more suitable for datasets with cluster counts smaller than 10. Beyond that limit, the cluster number yielded by the method is likely to be biased towards a smaller value. Another limitation of the proposed method is that the Curvature index value is undefined for null distributions (i.e. for the case of one cluster in the dataset). One possible remedy is to introduce an additional artificial cluster located far away from the original data. In that case, if the Curvature method returns 2 as the optimal cluster number, the original data can be regarded as coming from a null distribution.

Theoretically, our method can work with virtually any clustering method. However, the within-cluster curve may differ slightly for different clustering methods. In this paper, we focused the experimental results on the highly-popular k -Means algorithm. Investigation of the suitability of the proposed Curvature method for other clustering methods is one of our future research objectives.

Acknowledgements

This paper has benefited from the discussion with Dr. Tu Enmei. We wish to thank him for his inspiration and feedback. This research was supported by Nanyang Technological University Research Scholarship.

References

- [1] D. Ayres-de Campos, J. Bernardes, A. Garrido, J. Marques-de Sa, L. Pereira-Leite, Sisporto 2.0: a program for automated analysis of cardiocograms, *J. Maternal-Fetal Med.* 9 (5) (2000) 311–318.
- [2] R.B. Bhatt, G. Sharma, A. Dhall, S. Chaudhury, Efficient skin region segmentation using low complexity fuzzy decision tree model, in: 2009 Annual IEEE India Conference, IEEE, 2009, pp. 1–4.
- [3] R. Bock, A. Chilingarian, M. Gaug, F. Haki, T. Hengstebeck, M. Jiřina, J. Klaschka, E. Kotrč, P. Savický, S. Towers, et al., Methods for multidimensional event classification: a case study using images from a Cherenkov gamma-ray telescope, *Nucl. Instrum. Methods Phys. Res. Sect. A* 516 (2) (2004) 511–528.
- [4] T. Caliński, J. Harabasz, A dendrite method for cluster analysis, *Commun. Stat.—Theory Methods* 3 (1) (1974) 1–27.
- [5] M. Charytanowicz, J. Niewczas, P. Kulczycki, P.A. Kowalski, S. Łukasik, S. Żak, Complete gradient clustering algorithm for features analysis of x-ray images, in: *Information Technologies in Biomedicine*, Springer, 2010, pp. 15–24.
- [6] M.M.-T. Chiang, B. Mirkin, Intelligent choice of the number of clusters in k -means clustering: an experimental study with different cluster spreads, *J. Classif.* 27 (1) (2010) 3–40.
- [7] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, J. Reis, Modeling wine preferences by data mining from physicochemical properties, *Decision Support Syst.* 47 (4) (2009) 547–553.
- [8] T.G. Dietterich, R.H. Lathrop, T. Lozano-Pérez, Solving the multiple instance problem with axis-parallel rectangles, *Artif. Intell.* 89 (1) (1997) 31–71.
- [9] R.A. Fisher, The use of multiple measurements in taxonomic problems, *Ann. Eugenics* 7 (2) (1936) 179–188.

- [10] W. Fu, P.O. Perry, Estimating the number of clusters using cross-validation, preprint arXiv:1702.02658(2017).
- [11] C. Goutte, P. Toft, E. Rostrup, F.A. Nielsen, L.K. Hansen, On clustering FMRI time series, *NeuroImage* 9 (3) (1999) 298–310.
- [12] S.J. Haberman, Generalized residuals for log-linear models, in: *Proceedings of the 9th International Biometrics Conference*, 1976, pp. 104–122.
- [13] J.A. Hartigan, *Clustering algorithms*, John Wiley & Sons Inc., New York, 1975.
- [14] J. Jossinet, Variability of impedivity in normal and pathological breast tissue, *Med. Biol. Eng. Comput.* 34 (5) (1996) 346–350.
- [15] L. Kaufman, P.J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*, 344, John Wiley & Sons, 2009.
- [16] W.J. Krzanowski, Y. Lai, A criterion for determining the number of groups in a data set using sum-of-squares clustering, *Biometrics* (1988) 23–34.
- [17] L.A. Kurgan, K.J. Cios, R. Tadeusiewicz, M. Ogiela, L.S. Goodenday, Knowledge discovery approach to automated cardiac SPECT diagnosis, *Artif. Intell. Med.* 23 (2) (2001) 149–169.
- [18] F.O. Lee Graham, Hill-Valley data set, <https://archive.ics.uci.edu/ml/datasets/Hill-Valley>.
- [19] M. Lichman, *UCI Machine Learning Repository*, University of California, Irvine, School of Information and Computer Sciences, 2013, <http://archive.ics.uci.edu/ml>.
- [20] M.A. Little, P.E. McSharry, S.J. Roberts, D.A. Costello, I.M. Moroz, Exploiting nonlinear recurrence and fractal scaling properties for voice disorder detection, *BioMed. Eng. OnLine* 6 (1) (2007) 1.
- [21] J. MacQueen, et al., Some methods for classification and analysis of multivariate observations, in: *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1, Oakland, CA, USA, 1967, pp. 281–297.
- [22] F. Marriott, Practical problems in a method of cluster analysis, *Biometrics* 27 (3) (1971) 501–514.
- [23] G.W. Milligan, M.C. Cooper, An examination of procedures for determining the number of clusters in a data set, *Psychometrika* 50 (2) (1985) 159–179.
- [24] K.S. Pollard, M.J. Van Der Laan, A method to identify significant clusters in gene expression data, in: *Proceedings of SCI (World Multiconference on Systems, Cybernetics and Informatics)*, Vol. II, 2002, pp. 318–325.
- [25] A. Rodriguez, A. Laio, Clustering by fast search and find of density peaks, *Science* 344 (6191) (2014) 1492–1496.
- [26] B.P. Roe, H.-J. Yang, J. Zhu, Y. Liu, I. Stancu, G. McGregor, Boosted decision trees as an alternative to artificial neural networks for particle identification, *Nucl. Instrum. Methods Phys. Res. Sect. A* 543 (2) (2005) 577–584.
- [27] S. Salvador, P. Chan, Determining the number of clusters/segments in hierarchical clustering/segmentation algorithms, in: *16th IEEE International Conference on Tools with Artificial Intelligence*, 2004. *ICTAI 2004*, IEEE, 2004, pp. 576–584.
- [28] J. Shawe-Taylor, N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.
- [29] V.G. Sigillito, S.P. Wing, L.V. Hutton, K.B. Baker, Classification of radar returns from the ionosphere using neural networks, *Johns Hopkins APL Tech. Digest* 10 (3) (1989) 262–266.
- [30] W.N. Street, W.H. Wolberg, O.L. Mangasarian, Nuclear feature extraction for breast tumor diagnosis, in: *IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology*, International Society for Optics and Photonics, 1993, pp. 861–870.
- [31] C.A. Sugar, G.M. James, Finding the number of clusters in a dataset: An information-theoretic approach, *J. Am. Stat. Assoc.* 98 (463) (2003) 750–763.
- [32] K. Tasdemir, P. Milenov, B. Tapsall, Topology-based hierarchical clustering of self-organizing maps, *IEEE Trans. Neural Networks* 22 (3) (2011) 474–485.
- [33] R. Tibshirani, G. Walther, T. Hastie, Estimating the number of clusters in a data set via the gap statistic, *J. R. Stat. Soc.: Ser. B (Stat. Methodol.)* 63 (2) (2001) 411–423.
- [34] E. Tu, L. Cao, J. Yang, N. Kasabov, A novel graph-based k-means for nonlinear manifold clustering and representative selection, *Neurocomputing* 143 (2014) 109–122.
- [35] E. Tu, J. Yang, N. Kasabov, Y. Zhang, Posterior distribution learning (PDL): a novel supervised learning framework using unlabeled samples to improve classification performance, *Neurocomputing* 157 (2015) 173–186.
- [36] E. Tu, Y. Zhang, L. Zhu, J. Yang, N. Kasabov, A graph-based semi-supervised k nearest-neighbor method for nonlinear manifold distributed data classification, *Inf. Sci.* 367 (2016) 673–688.
- [37] I.-C. Yeh, K.-J. Yang, T.-M. Ting, Knowledge discovery on RFM model using Bernoulli sequence, *Expert Syst. Appl.* 36 (3) (2009) 5866–5871.