Data & Knowledge Engineering 135 (2021) 101922

Contents lists available at ScienceDirect

Data & Knowledge Engineering

journal homepage: www.elsevier.com/locate/datak

DBHC: A DBSCAN-based hierarchical clustering algorithm

Alireza Latifi-Pakdehi, Negin Daneshpour *

Faculty of Computer Engineering, Shahid Rajaee Teacher Training University, Tehran, Iran

ARTICLE INFO

Keywords:
Clustering
Density based clustering
DBSCAN
Hierarchical clustering

ABSTRACT

Clustering is the process of partitioning objects of a dataset into some groups according to similarities and dissimilarities between its objects. DBSCAN is one of the most important clustering algorithms in the density based approach of clustering. In spite of the numerous advantages of the DBSCAN algorithm, it has two important input parameters, MinPts and Eps, which determining their values is still a great challenge. This problem arises because values of these parameters are heavily dependent on data distribution. To overcome this challenge, firstly features of these parameters are investigated and the data distribution are analyzed. Then a DBSCAN-based hierarchical clustering (DBHC) method is proposed in this paper in order to fix this challenge. For this purpose, DBHC first determines values of these parameters using the notion of k nearest neighbor and k-dist plot. Because most of the real world data is not distributed uniformly, it is needed to be produced several values for the Eps parameter. Then, DBHC executes the DBSCAN algorithm several times based on the number of Eps produced earlier. Finally, DBHC method merges obtained clusters if the number of produced clusters is larger than the number which has estimated by the user. To evaluate the performance of the DBHC method, several experiments were performed on some of benchmark datasets of UCI database. Obtained results were compared with other previous works. The obtained results consistently showed that the DBHC method led to better results in comparison to the other works.

1. Introduction

Clustering is a type of unsupervised learning commonly used in data mining. It is used to organize an input dataset into a finite set of semantically consistent groups based on certain similarity metrics [1,2]. Clustering is a key issue in intelligence science and is widely used in the field of artificial intelligence. So it is always an important concern of the machine learning research [3]. Clustering is used in different fields such as statistics, pattern recognition, machine learning, data mining, and bio-informatics [4–7].

In general, Clustering algorithms can be roughly classified into the following four categories: partitional, hierarchical, density based and grid based [1,8]. Next, we describe hierarchical and density based methods, because we will frequently refer them.

Hierarchical clustering is a well-known clustering method that can be thought of as a set of flat clustering methods organized in a tree structure. These methods construct the clusters by recursively partitioning the data in either a top-down or bottom-up fashion, applicable to different domain regions [9].

Most partitioning methods can find only spherical-shaped clusters and encounter difficulty in discovering clusters of arbitrary shapes. Other clustering methods have been developed based on the notion of density. The general idea of density based methods is to continue growing a given cluster as long as the density (number of objects or data points) in the "neighborhood" exceeds some threshold [8].

* Corresponding author.
E-mail addresses: alireza.latifi@yahoo.com (A. Latifi-Pakdehi), ndaneshpour@sru.ac.ir (N. Daneshpour).

https://doi.org/10.1016/j.datak.2021.101922
Received 30 September 2020; Received in revised form 27 April 2021; Accepted 19 August 2021
Available online 25 August 2021

The most well-known and important density-based clustering algorithms is DBSCAN [10] algorithm. In General, density-based clustering methods are based on the principle that clusters are areas of a high density data space separated by less density regions. DBSCAN algorithm has only two important input parameters called MinPtts and Eps and works as follow: DBSCAN randomly selects an object from dataset and checks whether the Eps-neighborhood of the selected object contains a minimum number (MinPts) of objects [8]. If this condition is satisfied, a new cluster is created, and all identified objects are added in the new cluster. Then, all objects within the cluster are also checked in same way based on the two parameters, so that other objects not previously checked are added to the cluster as far as possible. The above process continues until all objects in given dataset are visited.

DBSCAN efficiently discovers the clusters of arbitrary size, shape and number in a large dataset [11]. Moreover DBSCAN is relatively fast when clustering small and medium datasets [12]. However, as mentioned before, this algorithm has two input parameters called MinPts and Eps. The first parameter specifies the density threshold of dense regions. The second one is the maximum radius of a neighborhood. Both of these parameters are dependent on the data distribution of datasets and it is difficult to estimate them. Wide range of values can be assigned to these parameters, so setting them carelessly may lead to unexpected results.

The DBHC method that is proposed in this paper, first proposes a solution for setting the DBSCAN parameters without user's involvement. The Eps parameter, as mentioned before, determines the neighborhood radius. In most datasets, determining a global value for Eps does not lead to good results. Therefore, in such cases, several values must be considered for Eps parameter. In this paper, the DBHC proposes a method based on k-dist plot for determining values of Eps. k-dist plot depicts distance from each data object to its kth nearest neighbor. After that, DBHC executes the DBSCAN algorithm for each value of Eps. Then it merges the result clusters in bottom-up manner to get optimal level.

Also, with a change points of view, the DBHC method can be seemed as a bottom-up hierarchical clustering method which the earlier levels of clustering process are devolved into the DBSCAN algorithm. Density-based methods are more effective than the hierarchical methods in finding non-spherical clusters. Therefore, by using the DBSCAN algorithm in the structure of the hierarchical algorithms, it is possible to improve hierarchical algorithms ability to find non-spherical clusters.

The rest of this paper is organized as follows. Section 2 introduces some related study about DBSCAN algorithm. Section 3 describes the proposed DBHC method and its algorithms. Section 4 describes the experimental results that demonstrate the effectiveness of our method. Finally, some conclusions are given in Section 5.

2. Related work

The DBSCAN algorithm is the first and the most well-known method of density-based clustering. Other methods are inspired by this method or have improved this algorithm: reducing its running time such as G-DBSCAN [13], NQ-DBSCAN [14], BLOCK-DBSCAN [15] and Dboost [16], enhancing it for clustering special data such as EPDCA [17] and k-DBSCAN [18], applying it for proposing new methods for clustering such as TSCM [19], improving its accuracy such as Revised-DBSCAN [12] or estimating its input parameters. For estimation of its parameter, methods based on grid, k-dist-plot, and so on have been proposed so far:

GRIDBSCAN [20] proposes a new method based on grid concept for addressing parameter issue in three steps. In the first step, appropriate grids are created in that density of each grid is homogeneous. In this level, GRIDBSCAN uses input parameter λ to divide each dimension of data into λ units. In the second step, cells with the same density are merged together and after merging process, the most suitable values for Eps and MinPts in each grid are identified. In this level, GRIDBSCAN uses another input parameter *perc* for identifying two available cells. Finally, in the third step, the DBSCAN algorithm is run to obtain the final result using the obtained parameters from the previous step.

GRPDBSCAN [21] (Grid-based DBSCAN Algorithm with Referential Parameters) uses grid for auto-generating the Eps and MinPts parameters of the DBSCAN algorithm, and such as any grid based method it requires input parameter for generating grid.

HD_DBSCAN [11] generalizes DBSCAN algorithm in two ways. First, important input parameter, Eps, adaptively determined. HD_DBSCAN produces one Eps corresponds to each density region. Second, in the process of finding Eps, HD_DBSCAN checks the data distribution of each dimension and eliminates non-significant dimensions from subspaces of given dataset. Such a strategy is more efficient when we encountered with high dimensional data and makes the HD_DBSCAN effective for subspace clustering. This clustering method detects clusters which hide in subspace of high dimensional datasets. HD_DBSCAN uses notion of kNN and subsequently requires the value of k as input parameter. In addition, HD_DBSCAN does not propose any solution for determining MinPts parameter.

DMDBSCAN [22] selects several values of Eps for different densities according to a k-dist plot such that the number of densities is given intuitively by k-dist plot. DMDBSCAN supposes every sharp change in k-dist plot corresponds to change in density level. DMDBSCAN does not propose any solution for determining MinPts parameter. Also DMDBSCAN uses notion of kNN and subsequently requires the value of k as input parameter. In addition, in Section 3 we demonstrate that every sharp change in k-dist plot does not correspond to change of density.

AutoEpsDBSCAN [23] also supposes every sharp change in k-dist plot corresponds to change in density level. So in order to find all possible Eps values, it calculates the slopes at regular interval in k-dist plot and then finds the difference between slopes. By setting a certain threshold value, every sharp change in slopes is considered as a change in density level. AutoEpsDBSCAN also uses notion of kNN and subsequently requires the value of k as input parameter. In [24], the author proposed a method to determine the value of k.

OPTICS [25] introduces two new Concepts of 'reachability distance' and 'core distance' to produce ordering of points representing the density-based clustering structure. This cluster ordering is equal to density based clustering that can find clusters with varying densities. With aid of this algorithm, there is no need to determine value of Eps parameter.

Ref. [1] in order to distinguish different densities in a dataset, introduces a Density Layer Tree (DLT) that makes it possible to separate dense region from sparse region in a given dataset regardless of the type of it; and then for determining Eps value, proposes two methods called AA-DBSCAN which uses the approximate adaptive ε-distances for each density layer and kAA-DBSCAN, by utilizing Density Layer Tree(DLT). But, it does not propose any solution for determining MinPts parameter. In addition, it defines another MinPts parameter that indicates minimum number of objects contained in a leaf node of DLT. Moreover, for constructing DLT, it requires parent node of dataset as input parameter.

Ref. [26], is an improvement for the DBSCAN algorithm such that the Eps parameter is eliminated and replaced by another parameter called ρ. This parameter is noise ratio in the dataset. In this work, parameters of the original algorithm have reduced, but it is not easy for the user to guess noise ratio.

BDE-DBSCAN [27] proposes a method based on Evolutionary Algorithms to automatically specify appropriate parameter values for Eps and MinPts. The evolutionary algorithm used in this method is the Differential Evolution algorithm. Since the incorrect selection of the Eps parameter can extremely affect the performance of the DBSCAN algorithm, BDE-DBSCAN also utilizes the combination of an analytical-way [28] for estimating Eps and Tournament Selection (TS) method. This analytical-way proposes a DBSCAN-based algorithm to look for natural patterns of arbitrary shape in a dataset by utilizing gamma function and analytical methods.

The proposed work in [29] extends and enhances works based on k-dist plot such as DMDBSCAN [22], AutoEpsDBSCAN [23] and the proposed work in [24] by proposing a new way to determine the Eps radius. It is based on analysis of a knee points which appears in the sorted values of the distance function used in the dataset. First it divides the k-dist plot into some equal parts and then for each of the parts calculates the average value of them (For such operation, we called this method Local-Avg in experiment section). Finally, it calculates the value of Eps by some new relations. But for MinPts parameter, it uses fixed value of six in its experiments section to run the traditional DBSCN method.

EDBSCAN [30] (Efficient Density-Based Spatial Clustering of Applications with Noise) tries to find out a set of pair MinPts and Eps values by two dominant tasks. First, all of objects are subdivided into different multiple cells and for each cell, MinPts and Eps are calculated. Finally obtained MinPts and Eps pairs are merged and DBSCAN algorithm will be run for each merged value of Eps and MinPts.

Ref. [31] proposes a new method for clustering abnormal data by configuring two threshold values of Eps and MinPts which is based on a visual display of the statistical characteristic of the dataset. For finding Eps and MinPts values, it calculates pairwise distance between objects and depicts probability density distribution curve of the distance value.

AEDBSCAN [32] generates epsilons dynamically and applies DBSCAN method to cluster data objects. AEDBSCAN takes suitable MinPts as input parameter. AEDBSCAN first calculates pairwise distance between objects and sorts them in ascending manner. Then it selects top 2*MinPts elements from sorted values, calculates average of them as Eps value and runs DBSCAN algorithm.

Ref. [33] is a grid-based approach to determine parameters of the DBSCAN Algorithm. The authors believe that the value of MinPts parameter equals 4, 5, or 6 for most cases. Also, in this work assumed that three ranges can be defined as a grid size. Based on these assumptions, some fixed threshold values and maximum distances between the elements of the cells, the values of Eps and MinPts parameters are calculated.

KDTDBSCAN [34] solves the problem of input parameters using K-distance graph and KD-Tree method. It calculates distance between each point to all other points and computes average of them and draws averaged k-dist plot in ascending order. For finding Eps values, it calculates slope values at regular intervals. If the difference between slope values is ten percent higher than the previous slope, KDTDBSCAN considers it as Eps value. For each value of Eps, it calculates MinPts by a formula based on the number of points in Eps neighborhood of points in a dataset. In the other words, KDTDBSCAN searches for sharp changes in k-dist plot.

The methods mentioned in this section have tried to provide a way to address issues of determining DBSCAN input parameters, but most of them require another input parameter which in most cases, estimating the value of the new parameter is more difficult to estimate than the DBSCAN algorithm parameters. In this paper, DBHC is proposed based on k-dist plot to overcome this challenge.

3. DBHC method

DBHC method for data clustering consists of three steps. In the first step, we will study the distribution of datasets and the distance between their data objects for several datasets. Then, we find the appropriate value for DBSCAN parameters required to run the DBSCAN algorithm. Since in the first step, several values are determined for Eps, so we have to run the DBSCAN algorithm several times to identify the potential clusters in the second step. In the last step, the clusters produced in the previous step are merged together until certain threshold. We will detail these three steps, in three separate sub-sections.

3.1. Initializing parameters

To find a way to determine the values of Eps and MinPts parameters, let us look at the range of values that can be assigned to these parameters and analyze the behavior of the DBSCAN algorithm in this range. After that, we propose our solution.

The MinPts parameter specifies the density threshold of dense region; so the smaller value of MinPts leads to larger number of clusters, and also the greater value of MinPts leads to smaller number of clusters. The smallest acceptable value that can be assigned to this parameter is 3; because if the value of this parameter is set to 1, it is possible to be created a cluster with density of 1 and that is completely unreasonable. And according to [35], if we choose value of 2 for MinPts, the result of DBSCAN is nearly the result of hierarchical clustering. Therefore, the minimum value can be considered to MinPts is 3.

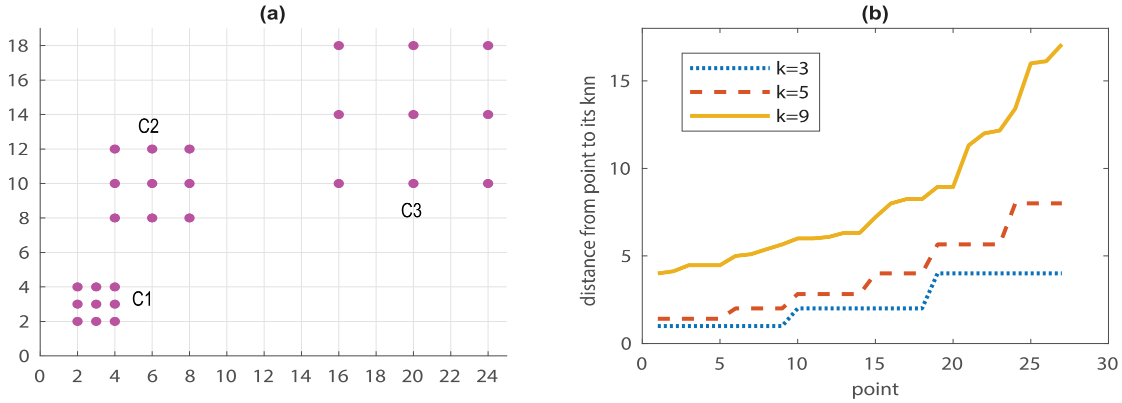


Fig. 1. a: sample dataset. b: k-dist plot of dataset in a.

To determine the upper bound of the MinPts parameter, let us start from highest possible value. Suppose we have a dataset with M objects. If the value of this parameter is set to the M, it is possible to be created only one or none cluster and it is unacceptable. So, the value of this parameter must be smaller than the M. Since the lower bound of this parameter is 3, so the upper bound will not exceed M/3 (if a cluster has more than M/3 objects, a cluster will also be found which has fewer than 3 objects). Assume that the number of clusters is known before the clustering process (as assumed in the proposed method). A more precise upper bound can be considered for this parameter. For this purpose, if M is the number of data objects and C is the number of clusters, the maximum value that can be assigned to MinPts is as follows:

$$MinPts_{max} = \frac{M}{C} \tag{1}$$

Therefore, for the MinPts parameter, the lower bound is 3 and its upper bound can be obtained from Eq. (1).

Another parameter of the DBSCAN algorithm is the Eps parameter which indicates the maximum radius of a neighborhood. The value that can be assigned to this parameter is heavily dependent on the data distribution. Thus, to determine the value of this parameter, the data distribution must be analyzed. In this paper, we utilize the concept of the nearest neighbor and the k-dist graph to analyze the data distribution.

Since a cluster can have at least 3 data objects, so the lower bound of Eps is equivalent to the smallest distance to the second closest object for all objects in the entire dataset. Assume that kNN_{P_i} denotes the distance from object p$_i$ to its kth nearest neighbor. So Eps_{min} which denotes minimum value of Eps can be calculated as Eq. (2):

$$Eps_{min} = min\{dist(P_i, 2NN_{P_i})|i\epsilon(1, M)\} \tag{2}$$

To determine the upper bound of Eps, the easiest way is calculating the maximum distance between all pairs of objects in the entire dataset. So Eps_{max} which denotes maximum value of Eps can be calculated as Eq. (3):

$$Eps_{max} = max\{dist(P_i, P_j)|i, j\epsilon(1, M), i < j\} \tag{3}$$

Although the result of Eq. (3), specifies the upper bound of the Eps value, but it does not applicable in practice. Because by using such a large value, it is possible to produce only one cluster. If the number of clusters is known before the clustering process, we can limit the upper bound of Eps using the Eq. (1). Accordingly, the maximum distance to kth nearest neighbor for all objects can be considered as the upper bound of Eps where k is the maximum value of MinPts (Eq. (1)). So Eps_{max} can be limited as Eq. (4):

$$Eps_{max} = max\{dist(P_i, kNN_{P_i})|i\epsilon(1, M), k = MinPts_{max}\} \tag{4}$$

After investigating the features of these parameters and the range that each of them can be initialized, it is time to use these values and investigate behavior of the DBSCAN algorithm in the obtained ranges. For this purpose, one tool which can help us is k-dist graph. To plot this graph, first, the distance from each point to its k nearest neighbor must be calculated. Then the obtained values are sorted in ascending manner. Finally, a graph is plotted based on these sorted values. For example, Fig. 1(a) is a sample dataset and Fig. 1(b) depicts its k-dist graph. Horizontal axis presents point number and the vertical axis is the distance from each point to its k nearest neighbor.

As shown in Fig. 1(a), the dataset has three clusters with three different densities. To cluster this dataset using the DBSCAN algorithm, setting a global value for the Eps parameter is not enough; because if the Eps is set proportional to the cluster C3, clusters C1 and C2 will be identified as one cluster. Also, if the Eps is set proportional to the cluster C1, the clusters C2 and C3 will be identified as noise. Therefore, it is obvious that setting a global value for the Eps parameter is not enough, and a set of Eps is required. This issue is more common in high-dimensional data approach. As mentioned in [8], real-world, high-dimensional datasets often have very skewed distributions such that their intrinsic clustering structure may not be well characterized by a single set of global density parameters.

Fig. 1(b) is the k-dist plot of sample dataset which is depicted in Fig. 1(a). Despite the dataset is very simple and the data distribution is quite regular, but only for $k = 3$ it is observed that three clusters with three different densities are in the dataset. For larger values of k, there is no significant relationship between dataset representation and its k-dist plot. Of course, this dataset is very regular unlike the real world datasets. It is impossible to represent all of the real world datasets because of their high dimensionality problem in the most cases. So one way to investigate them is depicting their k-dist graph. Let us look at the k-dist plot of real world datasets which are represented in Fig. 2. These plots are depicted with respect to different k. In the experiment section, the characteristics of each dataset is listed in Table 1.

As shown in Fig. 2, the plots of the real dataset, as same as the plot of synthetic dataset shown in Fig. 1(b), do not follow a regular pattern. As the plots show, the distance between a point and its k nearest neighbor not only is not the same for all objects, but also has a significant difference. According to [22–24], Sharp changes (knee points) in the plot can be considered as the point of Eps change. However, as shown in Fig. 2, it is possible that the distance between the points in a dataset has wide range, but there is no Sharp changes (knee points) in its k-dist plot. In addition, in Fig. 1(b), it is observed that there are several Sharp changes (knee points) for $k = 5$, but such a sharp variation of density is not observed in its dataset representation. Therefore, the strategy of finding sharp changes in k-dist plot is not always useful.

Therefore, with aid of the k-dist plot and such strategies which were explained in the previous paragraph and the related work section, it is not possible to find out exactly how many different densities exists in each dataset. So to create a set of Eps, the DBHC first calculates the distance from each point to its k nearest neighbor. Then sorts these distances in ascending manner (same as the manner in k-dist plot drawing). Finally, regardless existence of the knee points, DBHC selects $\sqrt{m}$ numbers of these distances at regular intervals and considers them as a set of Eps. $\sqrt{m}$ Which is dependent on the number of data objects, is considered as the number of Eps, and subsequently, the frequency of running DBSCAN algorithm.

For example, suppose a dataset of 100 objects. Therefore, $\sqrt{100}$= 10 different densities will be assumed in the starting point of DBHC method. If the assumed dataset has more than ten different densities, it is an irregular sparse dataset and a lot of investigations have been done in the context of spare learning such as [36,37]. So $\sqrt{m}$ can be regarded as maximum number of Eps parameter in that ensures all of different densities will have been covered. By utilizing maximum value for number of Eps parameter, number of clusters may increase, so the DBHC method will combine them in merge step. Also obtained results from comparisons between the DBHC method and other methods in the entire of experiment section show this number of Eps is enough.

In case of having many values for Eps, DBSCAN algorithm must be executed once per each Eps value. Each time DBSCAN is executed, some of points are clustered and some of them will be remained to cluster by the next Eps value (This process is explained completely in the second step). At each run, the value of MinPts parameter is set to the minimum value, 3, so that the possible small clusters are also identified (as same as the strategy of Eqs. (1) and (2)). As mentioned in [8], clusters hidden in high-dimensional data are often significantly smaller than conventional clusters in low-dimensional spaces. If the values of these parameters lead to a lot of clusters, the DBHC would merge them in merge step. At the end of clustering process, if the three-member clusters remained beside large clusters, the user could assume them as noise.

By choosing the value of 3 for the MinPts parameter, it is also easy to determine the value of the k parameter (in k-dist plot). When MinPts is set to 3, we have to look for the second nearest neighbor. Therefore, the value of k must also be set to 2, as same as the strategy of Eq. (2). Fig. 3 shows the pseudocode of step 1.

3.2. Identifying primitive clusters

In the previous step, the DBHC method produced several values for the Eps parameter. This situation obliges us to execute the DBSCAN algorithm once per Eps value. Suppose E as a set of Eps; therefore, the DBSCAN algorithm must be run |E| times where || is the cardinality operator. As shown in Fig. 4, each time While-loop is executed, first the smallest Eps is selected, and then the DBSCAN algorithm is initialized using it. Since the small Eps has not the capability to cover all of points, some data objects will definitely remain unclustered. In order to cluster remained points, the clustered points must be removed from the entire dataset and the current Eps must also be removed from the Eps set. Then the remained points are clustered with the next smallest Eps. Based on the above description, Fig. 4 represents the pseudocode of step 2.

As stated in step 1, because of multiple running of the DBSCAN algorithms, it is possible to be created multiple clusters. But in step 3, these clusters will be merged to optimal level. This pattern recalls us the famous divide-and-conquer strategy. In this strategy, the problem is divided into number of sub-problems that are smaller instances of the same problem. Then the solutions to the sub-problems are combined into the solution for the original problem [38]. However, it should be noted that in the proposed method, sub-problems are not independent of each other, because it is not clear how many objects are clustered by DBSCAN algorithms in each execution and how many of them will be remained for the next execution (next Eps).

3.3. Merging the clusters

In the last step, the DBHC method compares the number of clusters generated from the previous step with the number that is estimated by the user. According to the operation of DBHC method, the number of obtained clusters is greater than the actual number. So, the clusters must be merged. For this purpose, DBHC first calculates the center of each cluster. Then, calculates the distance between the centers of each pair of clusters and combines two nearest clusters into a new one. Suppose two clusters C_i and C_j including objects X_i and X_j with the centers of O_i and O_j. The distance between two clusters, which denoted by d() in Eq. (5), is calculated as follows:

$$d(C_i, C_j) = \|\overline{O_i} - \overline{O_j}\|_2 \tag{5}$$

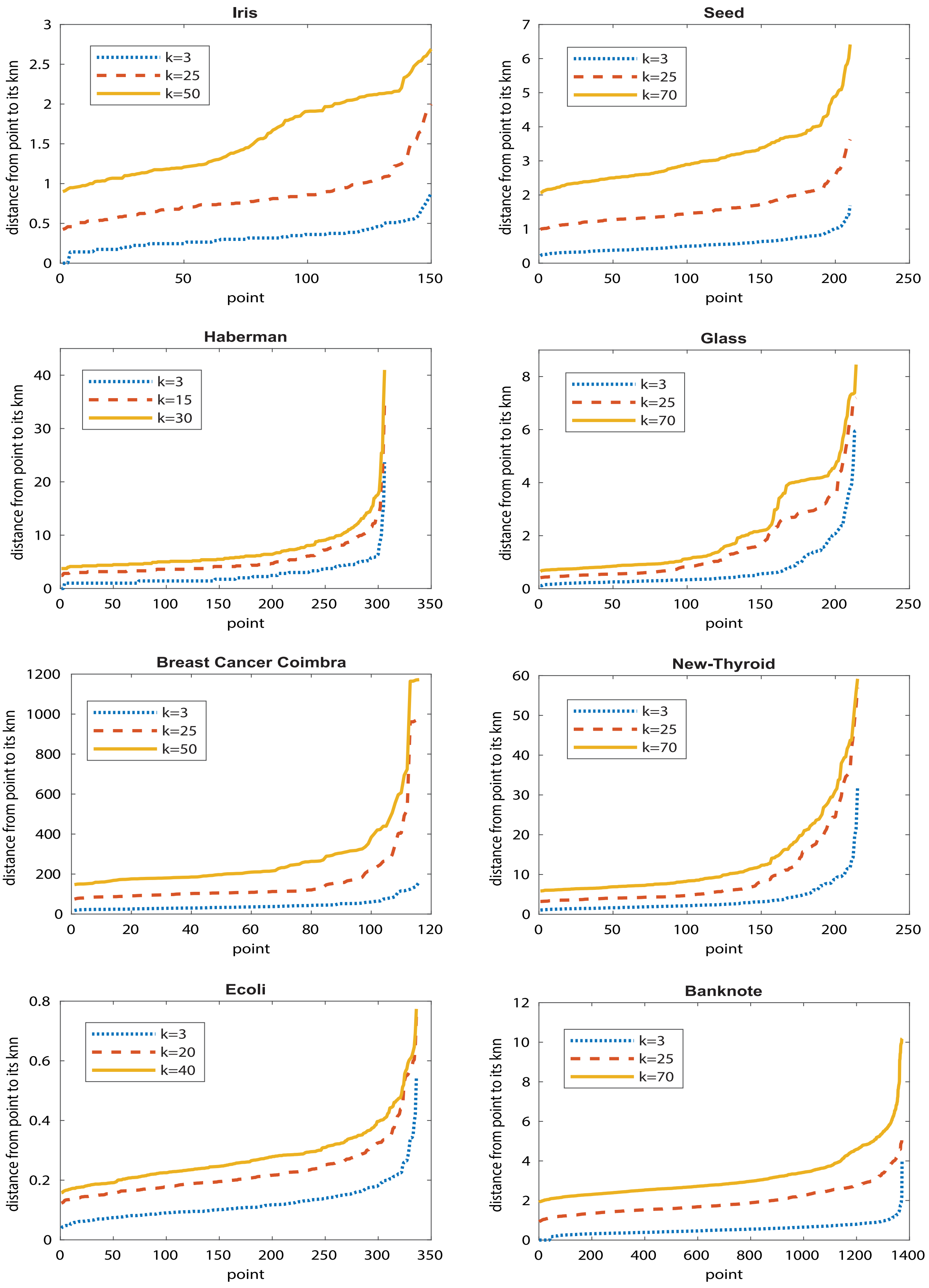
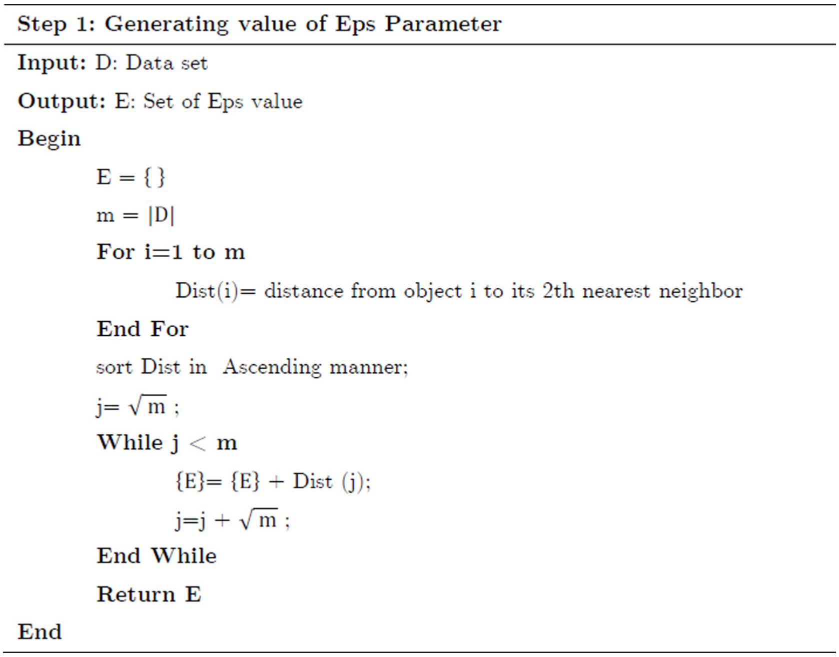
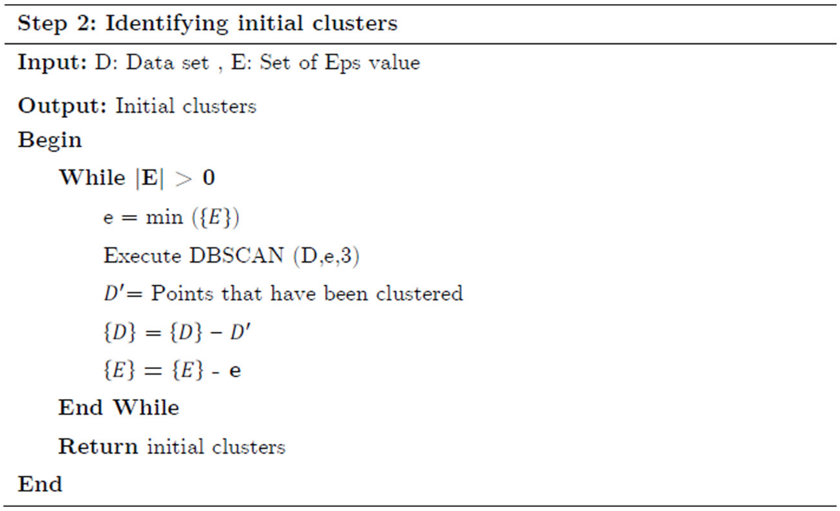


Fig. 2. k-dist plot of some datasets.



```
Step 1: Generating value of Eps Parameter
Input: D: Data set
Output: E: Set of Eps value
Begin
        E = { }
        m = |D|
        For i=1 to m
                Dist(i)= distance from object i to its 2th nearest neighbor
        End For
        sort Dist in  Ascending manner;
        j= $\sqrt{m}$ ;
        While j < m
                {E}= {E} + Dist (j);
                j=j + $\sqrt{m}$ ;
        End While
        Return E
End
```

Fig. 3. Generating value of Eps Parameter.



```
Step 2: Identifying initial clusters
Input: D: Data set , E: Set of Eps value
Output: Initial clusters
Begin
    While |E| > 0
        e = min ({$E$})
        Execute DBSCAN (D,e,3)
        $D'$= Points that have been clustered
        {$D$} = {$D$} − $D'$
        {$E$} = {$E$} - e
    End While
    Return initial clusters
End
```

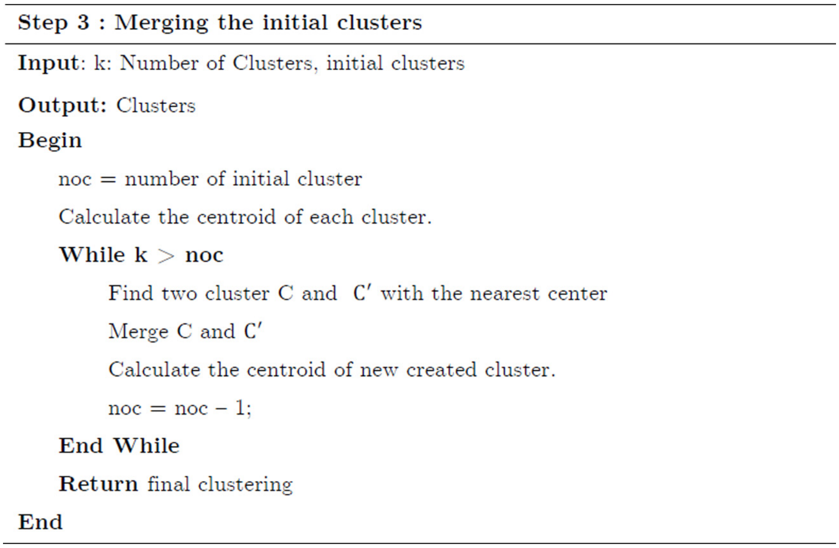
Fig. 4. Identifying initial clusters.

where

$$\overline{O_i} = \frac{1}{|C_i|} \sum_{X_r \in C_i} X_r$$

Hierarchical clustering algorithms require a condition to terminate clustering process. In the DBHC method, this condition is whenever the number of clusters matches the user's desired number. In this situation, clustering process will be terminated. Estimating number of clusters (parameter k in pseudocode of Fig. 5) is much easier than parameters MinPts and Eps. This parameter (k) can be considered as type of flexibility; because if the user is not satisfied with the cluster merging level, it is possible that merging operation continues to the desired level without re-spending. Fig. 5 shows the pseudocode of step 3.

By DBHC method, the need to specify values of DBSCAN parameters is eliminated. Also, with a change points of view, the DBHC method can be seemed as a bottom-up hierarchical clustering method which the earlier levels of clustering process are devolved into the DBSCAN algorithm. Density-based methods are more effective than the hierarchical methods in finding non-spherical clusters.



```
Step 3 : Merging the initial clusters
Input: k: Number of Clusters, initial clusters
Output: Clusters
Begin
    noc = number of initial cluster
    Calculate the centroid of each cluster.
    While k > noc
        Find two cluster C and C′ with the nearest center
        Merge C and C′
        Calculate the centroid of new created cluster.
        noc = noc − 1;
    End While
    Return final clustering
End
```

Fig. 5. Merging the initial clusters.

Therefore, by using the DBSCAN algorithm in the structure of the hierarchical algorithms, it is possible to improve hierarchical algorithms ability to find non-spherical clusters.

3.4. Computational complexity

The DBHC method consists of 3 steps including 3 algorithms: the algorithm of finding the set of Eps (Fig. 3), the initial cluster identification algorithm (Fig. 4), and the merging algorithm (Fig. 5). Therefore, the computational complexity of the proposed method is equal to sum of the complexity of these three algorithms. The complexity of the first algorithm is O(n^2) because of for-loop statement repeating linear operations and sort function. In the second algorithm, the most costly operation is for-loop statement with frequency of $\sqrt{n}$ and at each repeat, the DBSCAN algorithm is executed. Since the complexity of the DBSCAN algorithm is O(nlogn) [8,12], the complexity of this step is O($\sqrt{n}$.nlogn). The complexity of merging step and generally the hierarchical algorithm is O(n^2).

Therefore, the complexity of the DBHC method is equal to O($n^2 +$ $\sqrt{n}$.nlogn+n^2). Since the term n^2 is greater than other terms, the complexity of the DBHC method is O(n^2).

4. Experiment

In this section, performance of the DBHC method is analyzed and compared with other methods. For this purpose, three DBSCAN-based clustering methods, i.e., analytical method [28], OPTICS [25] and Local-Avg [29] are picked up for comparison with the DBHC method. Also AHC (Agglomerative Hierarchical Clustering) and k-means algorithms have been used, because the DBHC method has only one input parameter (k as number of clusters), and the AHC and k-means methods also have only the same input parameter.

Clustering performance was calculated in terms of accuracy measure which is heavily used in clustering context. To calculate accuracy measure, assume dataset $X = \{x_1, x_2, \dots, x_N\}$ as a set of N objects and clustering process partitions X into k groups: $C = \{c_1, c_2, \dots, c_k\}$ and a_i is the number of objects that are correctly assigned to the ith cluster; then accuracy measure (AC) is calculated as follow:

$$AC = \frac{\sum_{i=1}^{k} a_i}{N} \tag{6}$$

Accuracy measure will have larger value for good clustering. This measure has been the most used evaluation metric in many clustering studies. Also, in last subsection, the DBHC is compared with some other mentioned algorithms based on validity indices. For comparison, 9 benchmark real world datasets are also picked up from the UCI machine learning repository [39]. Table 1 summarizes these datasets characteristics.

4.1. Comparison with other parameter estimation methods

To evaluate the performance of the proposed method, the accuracy of the DBHC was computed and reported in Table 2. For comparison, the accuracy of three DBSCAN-based methods, Analytical method, Local-Avg and OPTICS, are included. Datasets used for comparison were chosen from UCI machine Learning Repository and Table 1 shows the characteristic of them. These datasets

Table 1
Properties of datasets.

Dataset	Number of classes	Number of dimensions	Number of samples
Iris	3	4	150
Zoo	7	17	101
Seed	3	7	210
Haberman	2	3	306
Glass	6	9	214
Breast Cancer Coimbra	2	9	116
New-Thyroid	3	5	215
Ecoli	8	21	197
Bupa	2	6	345
Pima	2	8	768
CMC	3	9	1473
Banknote	2	4	1372

Table 2
Comparison with other parameter estimation methods.

Datasets	DBHC	Analytical method	Local Avg	OPTICS
Ecoli	74.40	44.05	43.75	63.39
Iris	90.00	66.67	64.00	80.67
Glass	50.93	43.93	43.93	58.88
Haberman	75.16	73.20	71.90	73.53
Zoo	79.21	66.34	64.36	73.27
New-Thyroid	79.53	70.23	70.70	69.77
Breast Cancer Coimbra	54.31	46.55	51.72	45.69
Seed	90.00	74.29	33.81	88.57
Bupa	55.94	55.65	51.59	49.86
Pima	66.76	63.02	61.46	59.64
CMC	41.21	32.99	38.83	41.21
Banknote	66.25	55.54	40.67	53.57
Avg	68.64	57.71	53.06	63.17

are classification data and in clustering process, labels of data are temporarily eliminated before applying DBHC, Analytical method, Local-Avg and OPTICS. After clustering process, labels of data were used for calculating accuracy of each method. Table 2 represents comparison of these methods.

As mentioned before, the larger value of accuracy measure represents better clustering. As this table shows, except for only one item, Glass dataset, that OPTICS method has better performance, for other datasets, the DBHC method has higher accuracy than the other methods. Also in the last row of Table 2, the average accuracy of each method has been calculated. According to this average value of accuracy, the DBHC method has better performance in compare to Analytical method, Local-Avg and OPTICS.

Table 2 represents comparison between the DBHC and other methods based on accuracy but about time complicity as mentioned earlier in Section 3.4, the DBHC method has increased the computational complexity of DBSCAN from nlogn to n^2. But other compared methods including Analytical method, Local-Avg and OPTICS, do not add such overhead because they do not consider number of values for Eps parameter similar to the DBHC method. If they offered number of values for the Eps parameter, the frequency of running DBSCAN algorithm would change and subsequently the time complexity of their solution would growth. Also OPTICS method has not present any solution for estimating MinPts value (in the experiment section, OPTICS method was run with considering MinPts value from 3 to 15 and the best result registered in Table 2) and Local-Avg method uses fixed value of six and Analytical method uses $\sqrt{m}$ or log(m) without any refinement step.

4.2. Comparison with other methods based on parameter k

In this subsection, the accuracy of DBHC is compared to AHC and k-means algorithms, because the DBHC method has only one input parameter (k as number of clusters), and the AHC and k-means methods also have only the same input parameter. In addition, a major advantage of partitional clustering algorithms (such as k-means) is that they can gradually improve the clustering quality through an iterative optimization process [40,41]. Also, many of the recent proposed data clustering algorithms typically compare their performance to these fundamental clustering algorithms [40].

In executing time, parameter k (number of clustering) of all three methods ACH, k-means and DBHC must be initialized. To set the value of k, the correct values that given in Table 1 are used. The accuracy of the DBHC and ACH and k-means are compared and the results are shown in Table 3.

According to Table 3, the DBHC method has a much higher accuracy than other methods. Also in the last row of Table 3, the average accuracy of each method is calculated. According to this average value, the DBHC method has better performance in compare to ACH and k-means.

Table 3
Comparison with other methods based on parameter k.

Dataset	DBHC	k-means	AHC
Ecoli	74.40	70.54	66.37
Iris	90.00	89.33	84.00
Glass	50.93	48.13	37.85
Haberman	75.16	52.29	73.53
Zoo	79.21	75.25	75.24
New-Thyroid	79.53	78.14	71.16
Breast Cancer Coimbra	54.31	50.86	53.45
Seed	90.00	89.05	81.90
Bupa	55.94	55.07	55.65
Pima	66.76	66.02	65.89
CMC	41.21	39.71	37.41
Banknote	66.25	61.22	66.25
Avg	68.64	64.63	64.06

4.3. *Influence of parameter k (number of clusters)*

In order to perform better evaluation of proposed method, the influence of the variation of input parameter k is examined. In the previous sub-section, calculated accuracies (numbers reported in Table 3) are based on the correct k that given in Table 1. To observe the behavior of the DBHC and other methods for other values of k, the accuracy of the DBHC method was recalculated for $k = 2,\ldots,9$ on different datasets.

Beside to the DBHC method, the results of k-means and ACH methods are also included, because they have exact one input parameter k. The results are presented in Figs. 6, 7. Indeed, each plot in this figure, represents the accuracy measure with respect to the value of k, number of clusters, for each dataset. As shown in Figs. 6, 7, the DBHC method is more accurate than the other methods for most values of k. This experiment shows that if the user does not know the correct value of k, the DBHC method is still comparable to other methods and better in most cases.

4.4. *Comparison based on validity index measure*

In this subsection, to compare the performance of DBHC with other methods, two validity indices are utilized. Validity indices are used for measuring the quality of a clustering result comparing to other ones which were created by other clustering algorithms, or by the same algorithms but using different parameter values [42]. In this section, DB [43] and SI [44] are used for comparison.

Assume dataset $X = \{x_1, x_2, \ldots, x_N\}$ as a set of N objects. Clustering process partitions X into k groups: $C = \{c_1, c_2, \ldots, c_k\}$. Also $d_e(x_i, x_j)$ denotes Euclidean distance between objects x_i, x_j and $\overline{c_k}$ is the centroid of cluster c_k.

Davies–Bouldin index (DB) is probably one of the most used indices in CVI comparison studies. It estimates the cohesion based on the distance from the points in a cluster to its centroid and the separation based on the distance between centroids [45]. DB index will have a small value for a good clustering. It is defined as:

$$DB(C) = \frac{1}{K}\sum_{c_k \in C} max_{C_l \in C\backslash c_k}\{\frac{S(c_k) + S(c_l)}{d_e(\overline{c_k}, \overline{c_l})}\} \tag{7}$$

where

$$S(c_k) = \frac{1}{|c_k|}\sum_{x_j \in c_k} d_e(x_i, \overline{c_k})$$

Silhouette Index (SI) is a normalized summation-type index. The cohesion is measured based on the distance between all the points in the same cluster and the separation is based on the nearest neighbor distance [45]. SI index will have a large value for a good clustering. It is defined as:

$$SI = \frac{1}{N}\sum_{c_k \in C}\sum_{x_i \in c_k}\frac{b(x_i, c_k) - a(x_i, c_k)}{max\{a(x_i, c_k) - b(x_i, c_k)\}} \tag{8}$$

where

$$a(x_i, c_k) = \frac{1}{|c_k|}\sum_{x_j \in c_k} d_e(x_i, x_j)$$

and

$$b(x_i, c_k) = min_{c_l \in C\backslash c_k}\{\frac{1}{|c_l|}\sum_{x_j \in c_l} d_e(x_i, x_j)\}$$

The performance of DBHC, ACH and k-means are validated via DB and SI indices and the results are reported in Table 4. This table shows that the DB Index has smaller value and SI index has a higher value for the DBHC method than the other methods across all the datasets. As mentioned before, a good clustering has a small DB index and large SI index values. Therefore, as Table 4

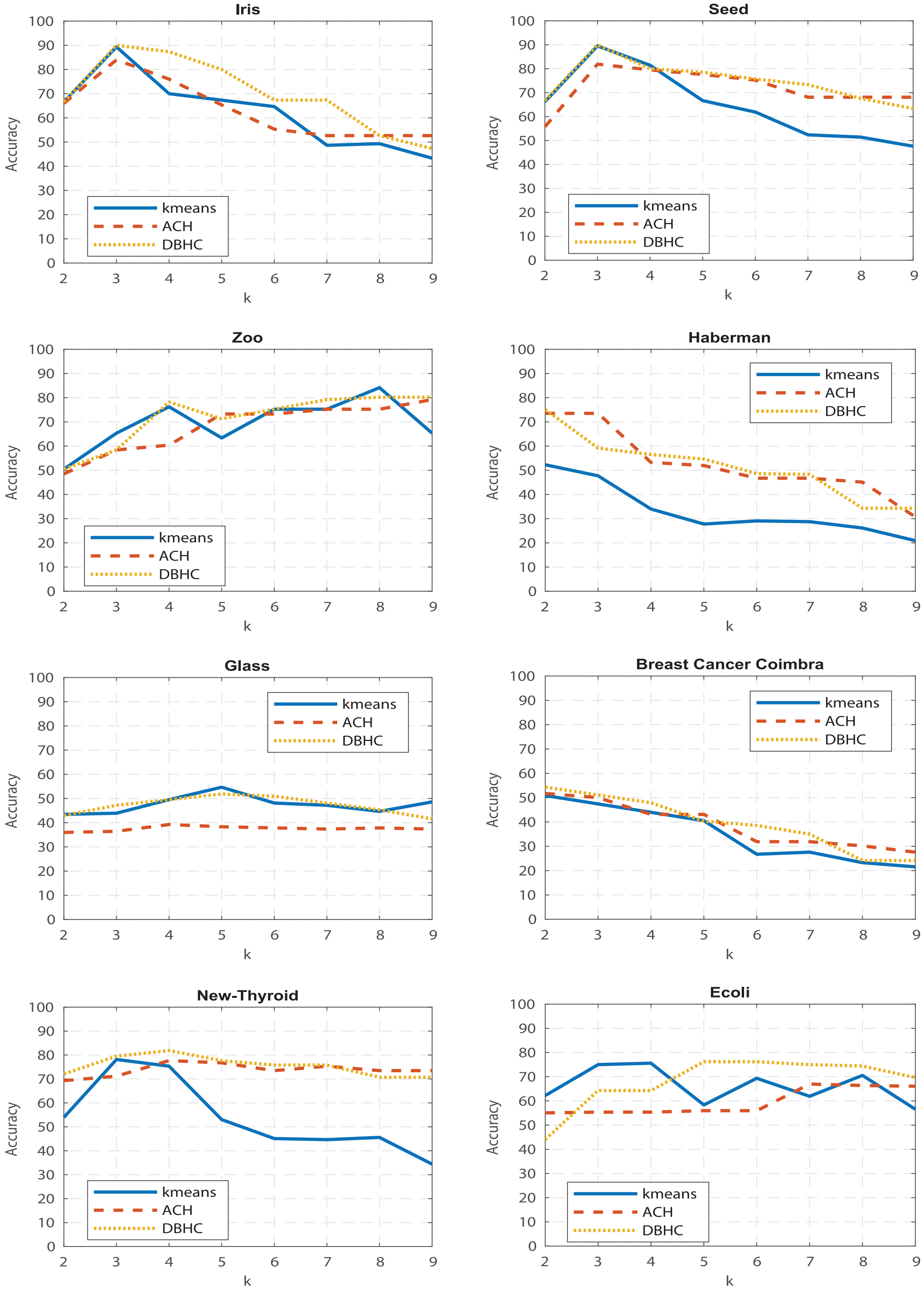


Fig. 6. Influence of parameter k.

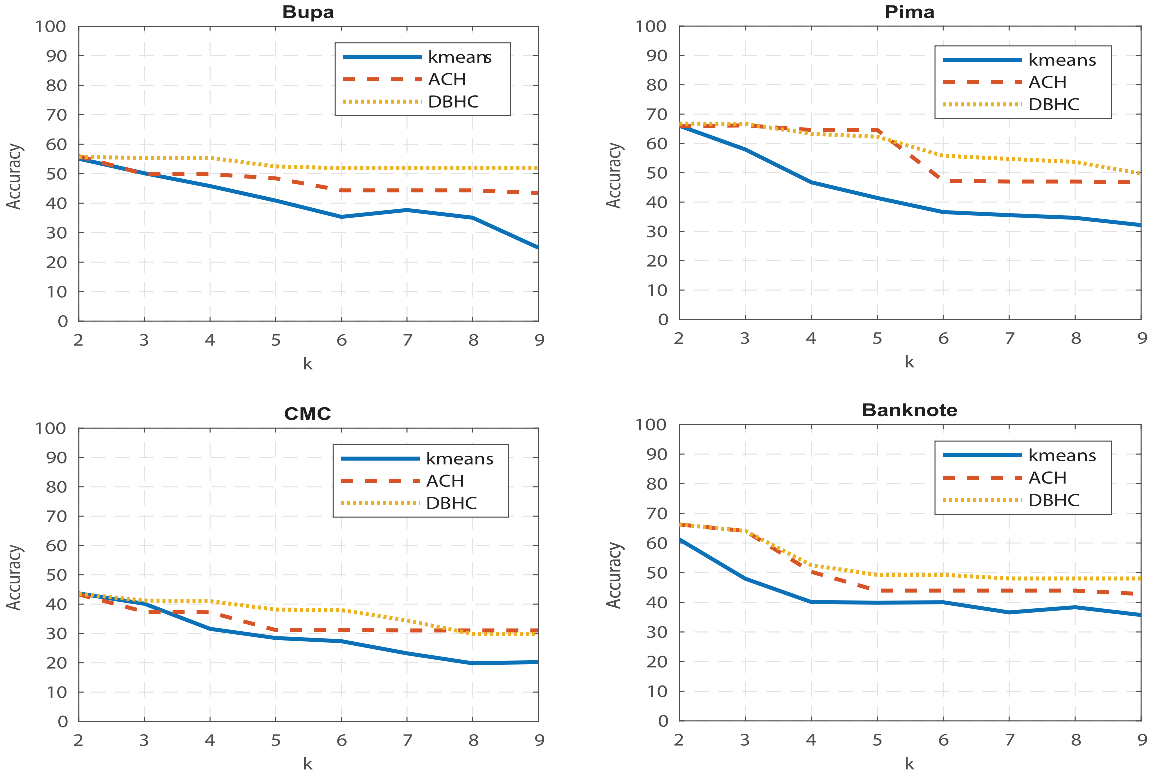


Fig. 7. Influence of parameter k.

Table 4
Comparison by cluster validity indices.

Dataset	VI	DBHC	k-means	AHC
Zoo	DB SI	0.937 0.6209	0.96672 0.59925	2.0995 0.2431
New-Thyroid	DB SI	0.5215 0.6724	0.9537 0.5625	1.4609 −0.6086
Breast Cancer Coimbra	DB SI	0.6109 0.793	0.6446 0.7317	0.404 0.8373
Ecoli	DB SI	0.9323 0.4782	1.1912 0.446	1.5352 0.2936
Iris	DB SI	0.6542 0.7347	0.82993 0.6117	3.5488 0.6866
Haberman	DB SI	0.9527 0.7806	0.9688 0.566	1.0177 0.7518
Seed	DB SI	0.7577 0.6185	0.7533 0.6632	0.7918 0.6038
Bupa	DB SI	0.6654 0.9211	0.7727 0.8216	0.6876 0.8686
Pima	DB SI	0.3428 0.8997	0.7134 0.7488	0.3571 0.8806
CMC	DB SI	0.81238 0.58552	0.7671 0.642	0.7437 0.5721
Banknote	DB SI	0.67646 0.65105	0.8702 0.6352	0.6789 0.6509

shows, DBHC has acceptable result. In this comparison, the only necessary parameter, k (number of clusters), was initialized by the correct values that given in Table 1 for all DBHC, k-means and ACH methods.

5. Conclusion

In this paper we proposed a DBSCAN-based hierarchical clustering algorithm (DBHC) to overcome the challenge of DBSCAN input parameters. We first investigated characteristics of these parameters, before determining the values of them. For the Eps parameter, DBHC produced a set of values and subsequently run the DBSCAN algorithm with respect to the count of Eps values produced earlier. Then merged the obtained clusters together until the number of clusters reached to the desired user level. This was the only input parameter that was determined by the user. To evaluate the efficiency of DBHC method and compare it with other methods, the proposed method was tested on real world datasets. The experiments showed that the DBHC method has a better performance than the other methods.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] J.-H. Kim, J.-H. Choi, K.-H. Yoo, A.J.T.J o. S. Nasridinov, AA-DBSCAN: an approximate adaptive DBSCAN for finding clusters with varying densities, J. Supercomput. 75 (1) (2018) 1–28.

[2] Y. Lv, et al., An efficient and scalable density-based clustering algorithm for datasets with complex structures, Neurocomputing 171 (2016) 9–22.

[3] J. Jia, X. Xiao, B. Liu, L. Jiao, Bagging-based spectral clustering ensemble selection, Pattern Recognit. Lett. 32 (10) (2011) 1456–1467.

[4] E. Akbari, H.M. Dahlan, R. Ibrahim, H. Alizadeh, Hierarchical cluster ensemble selection, Eng. Appl. Artif. Intell. 39 (2015) 146–156.

[5] F.J. Quintana, G. Getz, G. Hed, E. Domany, I.R. Cohen, Cluster analysis of human autoantibody reactivities in health and in type 1 diabetes mellitus: a bio-informatic approach to immune complexity, J. Autoimmun. 21 (1) (2003) 65–75.

[6] L. De Angelis, J.G. Dias, Mining categorical sequences from data using a hybrid clustering method, European J. Oper. Res. 234 (3) (2014) 720–730.

[7] J. Sun, W. Chen, W. Fang, X. Wun, W. Xu, Gene expression data analysis with the clustering method based on an improved quantum-behaved particle swarm optimization, Eng. Appl. Artif. Intell. 25 (2) (2012) 376–391.

[8] J. Han, J. Pei, M. Kamber, Data Mining: Concepts and Techniques, Elsevier, 2011.

[9] Z. Nazari, D. Kang, M.R. Asharif, Y. Sung, S. Ogawa, A new hierarchical clustering algorithm, in: 2015 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS), IEEE, 2015, pp. 148–152.

[10] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, A density-based algorithm for discovering clusters in large spatial databases with noise, in: Kdd, vol. 96, (34) 1996, pp. 226–231.

[11] S. Jahirabadkar, P. Kulkarni, Algorithm to determine ε-distance parameter in density based clustering, Expert Syst. Appl. 41 (6) (2014) 2939–2946.

[12] T.N. Tran, K. Drab, M. Daszykowski, Revised DBSCAN algorithm to cluster data with dense adjacent clusters, Chemometr. Intell. Lab. Syst. 120 (2013) 92–96.

[13] K.M. Kumar, A.R.M. Reddy, A fast DBSCAN clustering algorithm by accelerating neighbor searching using groups method, Pattern Recognit. 58 (2016) 39–48.

[14] ewang Chen, Shengyu Tang, Nizar Bouguila, Cheng Wang, Jixiang Du, HaiLin Li, A fast clustering algorithm based on pruning unnecessary distance computations in DBSCAN for high-dimensional data, Pattern Recognit. 83 (2018) 375–387.

[15] Yewang Chen, Lida Zhou, Nizar Bouguila, Cheng Wang, Yi Chen, Jixiang Du, BLOCK-DBSCAN: Fast clustering for large scale data, Pattern Recognit. 109 (2021).

[16] Y. Zhang, X. Wang, B. Li, W. Chen, T. Wang, K. Lei, Dboost: a fast algorithm for dbscan-based clustering on high dimensional data, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, Springer, 2016, pp. 245–256.

[17] Xingxing Liu, Qing Yang, Ling He, A novel DBSCAN with entropy and probability for mixed data, Cluster Comput. 20 (2017) 1313–1323.

[18] Nahid Gholizadeh, Hamid Saadatfar, Nooshin Hanafi, K-DBSCAN: An improved DBSCAN algorithm for big data, J. Supercomput. (2020).

[19] Mingyang Li, Xinhua Bi, Limin Wang, Xuming Han, A method of two-stage clustering learning based on improved DBSCAN and density peak algorithm, Comput. Commun. 167 (2021) 75–84.

[20] O. Uncu, W.A. Gruver, D.B. Kotak, D. Sabaz, Z. Alibhai, C. Ng, Gridbscan: Grid density-based spatial clustering of applications with noise, in: 2006 IEEE International Conference on Systems, Man and Cybernetics, vol. 4, IEEE, 2006, pp. 2976–2981.

[21] H. Darong, W. Peng, Grid-based DBSCAN algorithm with referential parameters, Physics Procedia 24 (2012) 1166–1170.

[22] M.T. Elbatta, W.M. Ashour, A dynamic method for discovering density varied clusters, Int. J. Signal Process., Imag. Process. Pattern Recognit. 6 (1) (2013) 14.

[23] M.N. Gaonkar, K. Sawant, AutoEpsDBSCAN: DBSCAN with Eps automatic for large dataset, Int. J. Adv. Comput. Theory Eng. 2 (2) (2013) 11–16.

[24] K. Sawant, Adaptive methods for determining DBSCAN parameters, Int. J. Innov. Sci., Eng. Technol. 1 (4) (2014).

[25] M. Ankerst, M.M. Breunig, H.-P. Kriegel, J. Sander, Optics: ordering points to identify the clustering structure, in: ACM Sigmod Record, vol. 28, (2) ACM, 1999, pp. 49–60.

[26] J. Esmaelnejad, J. Habibi, S.H. Yeganeh, A novel method to find appropriate ε for DBSCAN, in: Asian Conference on Intelligent Information and Database Systems, Springer, 2010, pp. 93–102.

[27] A. Karami, R. Johansson, Choosing dbscan parameters automatically using differential evolution, Int. J. Comput. Appl. 91 (7) (2014).

[28] M. Daszykowski, B. Walczak, D. Massart, Looking for natural patterns in data: Part 1. Density-based approach, Chemometr. Intell. Lab. Syst. 56 (2) (2001) 83–92.

[29] Artur Starczewski, Andrzej Cader, Determining the eps parameter of the DBSCAN algorithm, in: 18th International Conference on Artificial Intelligence and Soft Computing (ICAISC), Springer, 2019, pp. 420–430.

[30] Priyanka Sharma, Yogesh Rathi, Efficient density-based clustering using automatic parameter detection, in: International Congress on Information and Communication Technology, Advances in Intelligent Systems and Computing, Springer, 2016, pp. 433–441.

[31] Jin yu Song, Yi ping Guo, Bin Wang, The parameter configuration method of DBSCAN clustering algorithm, in: 5th International Conference on Systems and Informatics (ICSAI), IEEE, 2018, pp. 1062–1070.

[32] Vidhi Mistry, Urja Pandya, Anjana Rathwa, Himani Kachroo, Anjali Jivani, AEDBSCAN—Adaptive epsilon density-based spatial clustering of applications with noise, in: Progress in Advanced Computing and Intelligent Engineering, Springer, 2021, pp. 213–226.

[33] Artur Starczewski, Andrzej Cader, Grid-based approach to determining parameters of the DBSCAN algorithm, in: Artificial Intelligence and Soft Computing (ICAISC), Springer, 2020, pp. 555–565.

[34] N. Valarmathy, S. Krishnaveni, A novel method to enhance the performance evaluation of DBSCAN clustering algorithm using different distinguished metrics, J. Mater. Today: Proc. (2020).
[35] C. Braune, S. Besecke, R. Kruse, Density based clustering: Alternatives to DBSCAN, in: Partitional Clustering Algorithms, Springer, 2015, pp. 193–213.
[36] Jing Gu, Licheng Jiao, Shuyuan Yang, Jiaqi Zhao, Sparse learning based fuzzy c-means clustering, Knowl.-Based Syst. 119 (2017) 113–125.
[37] P. Li, X. Deng, L. Zhang, et al., Sparse learning based on clustering by fast search and find of density peaks, Multimedia Tools Appl. 78 (2019) 33261–33277.
[38] T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, Introduction to Algorithms, MIT Press, 2009.
[39] K. Bache, M. Lichman, UCI machine learning repository, 2013.
[40] C.C. Aggarwal, C.K. Reddy, Data Clustering: Algorithms and Applications, CRC Press, 2013.
[41] P. Berkhin, A survey of clustering data mining techniques, in: Grouping Multidimensional Data, Springer, 2006.
[42] M. Charrad, Y. Lechevallier, M.B. Ahmed, G. Saporta, On the number of clusters in block clustering algorithms, in: FLAIRS Conference, 2010.
[43] D.L. Davies, D.W. Bouldin, A cluster separation measure, IEEE Trans. Pattern Anal. Mach. Intell. 1 (2) (1979) 224–227.
[44] P. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, 20, 1987, pp. 53–65.
[45] O. Arbelaitz, I. Gurrutxaga, J. Muguerza, J.M. PéRez, I.J.P.R Perona, An extensive comparative study of cluster validity indices, 46, (1) 2013, pp. 243–256.

Alireza Latifi-pakdehi received his B.S. in Computer Engineering from Imam Khomeni International University, Qazvin, Iran and M.Sc. degree in computer engineering-software from Shahid Rajaee Teacher Training University, Tehran, Iran. His research interests include data mining and machine learning.

Negin Daneshpour is an assistant professor in the Computer Engineering faculty of Shahid Rajaee Teacher Training University, Tehran, Iran. She received her B.S. degree in computer engineering hardware from the department of electronics and computer engineering at Shahid Beheshti University, Iran, where she graduated summa cum laude in 1999. She received an MS degree and Ph.D. in computer engineering-software from the Department of Computer Engineering and Information Technology at the Amirkabir University of Technology, Iran, in 2002 and 2010, respectively. Her research interests focus on data analysis and management, data mining, and data preprocessing.