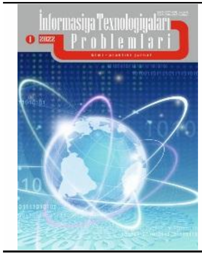


Available online at www.jpit.az

13 (1)
2022

Modification of the DBSCAN algorithm for big data clustering

Aygul F. Fakhraddingizi

Institute of Information Technology, Azerbaijan National Academy of Sciences, B. Vahabzade str., 9A, AZ1141 Baku, Azerbaijan

aygul.fexreddin@gmail.com

ARTICLE INFO

http://doi.org/10.25045/jpit.v13.i1.04

Article history:
Received 14 July 2021
Received in revised form 23 September 2021
Accepted 1 December 2021

Keywords:
Big data
Clustering algorithms
Density-based clustering
DBSCAN algorithm

ABSTRACT

The development of Information and Communication Technologies (ICT) has led to the rapid growth of digital information and the consequent emergence of the concept of big data. Therefore, there is a need to delve into big data and its essence, the possibilities and problems of analytical technologies. Clustering is one of the main methods of analyzing big data. The main purpose of clustering is to separate data into clusters according to certain characteristics. When clusters come in different sizes, densities, and shapes, the problem of detection arises. The article explores the density-based DBSCAN clustering algorithm for working with big data. One of the main features of this algorithm is to create an effective cluster by detecting the noise points in big data. During the implementation of the algorithm, real dataset containing noise points were used. Metrics such as adjusted rand index, homogeneity, Davis-Boldin index were used to evaluate the results of the experiment. The proposed method was more effective than the traditional DBSCAN algorithm in detecting noise points.

1. Introduction

Since the early 21st century, the digital data generated by devices and technologies, that is computers, mobile phones, the Internet, sensor networks, Earth's artificial satellites, space telescopes, cloud computing, etc. is exponentially growing each year. As a result, the concept of "big data" has emerged, representing a new era in data processing, management, storage and use. (Aliguliyev, 2014; Aliguliyev, Hajirahimova, & Aliyeva, 2016). As a phenomenal event, Big data has exposed the scientific community to a number of problems, creating a new research paradigm, along with revolutionary changes in the economic development of society. A need for using new technologies to process this data has emerged. In other words, the term big data refers to information that is complex in terms of volume and variety, however it is difficult to acquire new knowledge in real time as it is impossible to manage this data through traditional processing technologies. Valuable data and useful knowledge can be acquired through Big data analysis. However, the development of Big Data causes security and confidentiality risks as well (Alguliyev, & Imamverdiyev, 2014). In additon, it causes effortless leak of personal data and occurance of difficulties in data classification during collection, storage and use of data. Ensuring security and confidentiality of big data has become one of the most topical issues in the current research phase. (Alguliyev, Aliguliyev, & Sukhostat, 2020). Clustering is one of the main methods of big data analysis (Fakhraddingizi A., 2019). This article conducts an experiment with the application of DBSCAN clustering algorithm. The results are evaluated through various evaluation indices. Subsequent parts of the article are structured as follow: Section 2 provides a brief overview of the work related to the problem under study, and Section 3 classifies the clustering algorithms. Finally, description of the proposed method and analysis of the experiments are included in Section 4 and Section 5.

2. Related works

Several density-based clustering algorithms are currently available. These algorithms differ from each other for their numerous advantages and disadvantages. For example, one of these algorithms is the OPTICS algorithm studied by Ankerst (Zhou, Pan, Wang, & Vasilakos, 2017). Hence, this algorithm eliminates a number of weaknesses of DBSCAN, for example, the problem of detecting clusters that are important in data with different densities. This algorithm is used to find a density-based cluster in spatial data by constructing a wide sequence from a dataset. Another density-based algorithm is VDBSCAN offered by Liu (Liu, Zhou, Wu, 2007). This algorithm is created to analyze a dataset with various densities to solve DBSCAN's density problem. The basic idea of VDBSCAN is to use some methods to select several values of the epsilon parameter before adopting the traditional DBSCAN algorithm. It is possible to determine clusters with different densities using different values of Eps. Another LDBSCAN (Duan, Xu, Guo, Lee, Yan, 2007) algorithm is based on local density. This method eases selecting the appropriate parameters, but it also takes advantage of the local outlier factor (LOF) used in the anomaly detection to sense noise points compared to other density-based clustering algorithms. AUTOEPSDBSCAN is an advanced algorithm and automatically selects input parameters (Gaonkar, & Sawant, 2013). Experimental results show that the AUTOEPSDBSCAN algorithm can detect clusters of different shapes and sizes in big data consisting of points with noise and outlier points. All mentioned algorithms are used in solution of problems related to big datasets. These algorithms use the Eps parameter to determine clusters, therefore one cluster can be denser compared to others at the same value of Eps, based on different densities.

3. Clustering algorithms and their classification

"Machine learning" is an important field of Artificial Intelligence used in big data. The key objective of Machine Learning includes knowledge detection, correct decision-making and data analysis (Andrieu, De Freitas, Doucet, & Jordan, 2003; Berkhin, Kogan, Nicholas, 2006). Machine learning algorithms are categorized based on supervised, unsupervised and semi-supervised learning methods. Machine learning algorithms can also be divided as classification, clustering, regression, density evaluation etc. Decision tree, artificial neuron networks, SVM, Bayes networks, genetic algorithms and others can be listed as Machine learning algorithms. Supervised learning algorithms include Naive Bayes, SVM (Support Vector Machine) and maximal entropy method (MaxENT), etc.

Clustering algorithms refers to unsupervised learning methods. Unsupervised learning algorithms compare ungrouped dataset on their characteristics and classify them by dividing them into relevant groups. Thus, unsupervised learning algorithms unite similar objects in one group. Therefore, it finds common points by interpreting dataset information and acquires similar information by grouping them. Unsupervised learning algorithms increase similarity among the objects within cluster when similarity/difference measurements are provided. However, inter-cluster similarities significantly differ from each other. A special objective function is utilized here. Unsupervised learning algorithms include clustering (k-means, density-based, hierarchical etc.), self-organizing maps (SOM) etc. (Fahad et al., 2014). Clustering algorithms have emerged as an alternative, more powerful meta-learning tool to accurately analyze big data with the application of new technologies. As noted, several clustering algorithms are currently available. Below, is a brief outline of some of them:

Groups are immediately determined in partitioning-based clustering algorithms. Initial groups are determined and re-distributed towards are union. In other words, partitioning-based algorithms divide data objects into several sections, where each section is a set.

In other words, partitioning-based algorithms perform the task of dividing data objects by the number of sections, each section being called a "cluster". (Sajana, Rani, & Narayana, 2016; Zhao, Ma, & He, 2009).

The role of distance metrics is different in all algorithm types. In partitioning-based clustering methods, the template points selected in different iterations of the distance metric can be actual points, such as the centroid of the cluster (if no data are available).

Hierarchical clustering algorithms are developed to overcome some of the shortcomings associated with partitioning-based clustering methods. Obviously, partitioning-based clustering algorithms generally store a user-defined K parameter to reach quality clusters, and this algorithm is uncertain. Hierarchical algorithms are developed to create an identifiable and accessible mechanism during clustering of data objects. (Chana, & Arora, 2014; Chandra, & Anuradha, 2011). This algorithm hierarchically organizes the data depending on the proximity degree. Proximity is reached in intermediate nodes. As the hierarchy continues, initial cluster is gradually divided into several groups. Hierarchical algorithms can be classified as agglomerative (a bottom-up approach: each observation begins in its own cluster and merges in pairs as it progresses toward the largey) or divisive (a top-down approach: all observations begin in one group and the divisions are recursive as the hierarchy descends) methods. The agglomerative method begins by taking a cluster on the bottom surface (which contains only one data object in the cluster) and continues to combine two clusters during each iteration to establish a bottom-up hierarchy of clusters. Hence, in order to cluster this way each cluster starts with one object and recursively combines two or more matching clusters. On the other hand, the separation method begins with all the data objects in a giant macro-cluster and is continuously divided into two groups, forming a cluster hierarchy from top to bottom. Specifically, in this method, clusters start from a single dataset as a single cluster and recursively divide the most appropriate cluster. The process continues until it reaches the stop criterion. (Karypis, Han, & Kumar, 1999) However, the hierarchical method has a significant drawback, as it cannot be reversed after a step is executed (merger or division).

4. Density-based clustering algorithms

Many clustering algorithms are assumed to be originated from the probabilistic distribution of a particular data type. This is especially relevant to Expectation Maximization (EM) and k-means clustering algorithms. According to this hypothesis, these algorithms form spherical sets and do not work well in datasets with convex forms of actual sets. Convex sets naturally occur in spatial data, i.e., in two- or three-dimensional spaces that are different from the real world. Spatial points may get a random shape due to restrictions imposed by geographical objects such as mountains and rivers. In this case, algorithms, such as k-means, will result in incorrect performance by splitting or merging real groups. This observation results in discovery of clusters with random shapes. There is a need for efficiency in gradually growing real datasets. Detection and elimination of noise and deviations is required during clustering of big datasets (Alguliyev, Aliguliyev, & Abdullayeva, 2019). A paradigm of density-based clustering algorithms is offered to meet all these requirements. Density-based clustering can be considered a non-parametric method since there is no hypothesis about the number of clusters or their distribution. Density-based clusters are the dense areas in a data area separated by sparser areas. In addition, the density between the additional noise fields is considered to be lower than the density in any cluster. Due to their nature, dense fields in the data field can be of any shape (Dharni, & Bnasal, 2013; El-Sonbaty Y, Ismail, Farouk, 2004). Given an index structure supporting field queries, density-based clusters are calculated efficiently by performing the most field queries per dataset object. Sparse fields in the data field are considered as noise and do not pertain to any cluster. Note that there are some ideas in the literature about density-based clustering algorithms. First, non-dense points with less than k neighbors at r distance are removed. Second, a single connection method is used to cluster the remaining points. Finally, according to some criteria, non-dense points are determined in one of the clusters. The relationship between density-based clustering and mean-shift clustering paradigms can also be noted. When creating a density-based clustering algorithm, several key questions arise (Parimala, Lopez, & Senthilkumar, 2011):

- How to evaluate the density?
- How to establish the connection?
- Which data structures support efficient

implementation of the algorithm?

Next chapter presents density-based clustering algorithms and discusses the ways to answer these questions.

4.1. DBSCAN algorithm

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a clustering algorithm proposed by Martin Ester, Hans-Peter Kriegel, Jörg Sander and Xiaovei Xu in 1996 (Ester, Kriegel, Sander, & Xu, 1996). This algorithm is a non-parametric algorithm forming the density-based cluster: providing a set of some spatial points, it brings together closely located points (points with close neighbors), but marks the single points in low-density (sparse) areas (the nearest neighbors are very far) as outlier points. DBSCAN is one of the most referred algorithms in scientific literature (Cassisi et al., 2013).

The DBSCAN algorithm evaluates the density by calculating the number of points in a neighborhood of constant radius, and if any two points are in each other's neighborhood, it considers these points being connected to each other. Two main parameters of DBSCAN algorithm are available: Eps (Epsilon) - the distance defining the neighborhoods. Two points are considered neighbors when the distance between those two points is less than or equal to Eps (Rahmah, & Sitanggang, 2016).

$MinPts$ (Minimum Points) is the number of minimum data points to define a cluster. Based on these two parameters, the density-based clustering algorithm divides the points into three different types of points:

- core points, i.e., points located in close proximity ($|NEps(p)| \geq MinPts$); if Eps in the neighborhood radius consists of at least $MinPts$ point, i.e., density in the neighborhood must surpass a certain limit, this point is called a *core point*.
- border points, i.e., points belonging to any cluster, but not in close proximity; If a point can be reached from the base point and there are fewer points in the surrounding area than the MinPts point, this point is called a *border point*.
- noise points do not belong to any cluster; if one point is not a primary point and cannot be reached from any point, then it is evaluated as a *noise point*. (Moreira, Santos, & Carneiro, 2005).

4.2. Analysis of DBSCAN algorithm

Let's suppose a dataset consisting of D data points is given. Assume that there is a distance $dist(p,q)$ function for paired points. Eps neighborhood of a p point denoted by $NEps(p)$ is defined as $NEps(p) = \{ q \in D | dist(p,q) \leq Eps\}$.

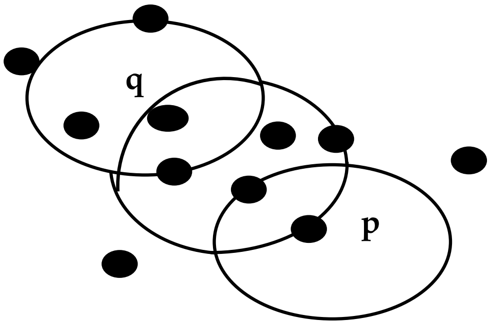
Definition 1. If (1) $p \in NEps(q)$ and (2) $|NEps(q)| \geq MinPts$, then point p is directly *density-reachable point* from q, based on Eps and $MinPts$ parameters.

Definition 2. In $p_1, \ldots, p_n$, $p_1 = q$, $p_n = p$ point sequence, as point p_{i+1} is directly density-reachable from point p_i, point p is *density – reachable point* from point q based on Eps and $MinPts$ parameters. Reachable density is a direct canonic extension of the directly density-reachable point. As this connection is not transient, another connection is applied.

Definition 3. If based on Eps və $MinPts$ paremeters, point o is a density-reachable point from both p and q points, then based on those parameters point p is *density-connected* point from point q. This connection is depicted in Figure 1. Hence, if point q is density-reachable point, however, point q is not density-reachable from point p, then points a and c are density-connected points from point b. Intuitively, density-connected set of points is the maximum of density-reachable points (Sharma, Sharma, & Soni, 2017).

Formally, based on Eps and $MinPts$ parameters, cluster C is a non-empty subset of D that meets following conditions.

1. For $\forall\, p, q$, if $p \in C$ and based on Eps and $MinPts$ parameters, if point q is density-reachable from point p, then $q \in C$ (maximum).
2. For $\forall\, p, q \in C$, point q is density-connected to point p based on Eps and $MinPts$ parameters.



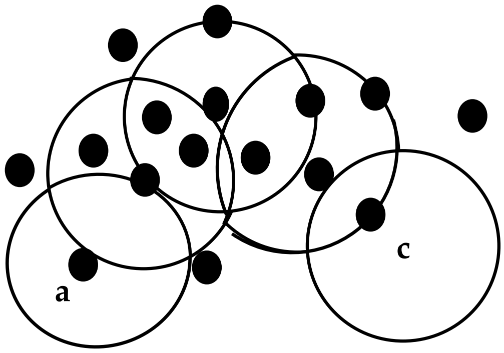


Figure 1. Density-reachable and density-connected points

Let's suppose $\mathrm{C}_1, \dots, \mathrm{C}_\mathrm{k}$ are the clusters of D dataset based on Eps and MinPts parameters. Any noise point that does not belong to any C_i cluster in D dataset is defined as $ise = \{p \in D | \ p \notin \ C_i \forall i\}$

For example, in Figure 1, the points, q and b are central points, while points p, a və c are border points.

Density-based clustering algorithms have two important features allowing effective calculations. Suppose that point p is the central point of D and all p points included into O set pulled from D are density-reachable based on Eps and $MinPts$ parameters. O set is a cluster according to Eps and $MinPts$ parameters. Assume that C is a cluster included in D. All points of C are density-reachable points from the central points of this cluster. Therefore, cluster C consists of all density-reachable points from any central point of this cluster. Thus, in accordance with Eps and $MinPts$ parameters, cluster C is uniquely defined with one of any central points. This forms the basis of the DBSCAN algorithm (Shah, 2012).

To find a cluster, DBSCAN algorithm starts with a p point included in an arbitrary dataset. In accordance with Eps and $MinPts$ parameters, all density-reachable points are taken from p, if necessary, field queries for p are initially performed to find direct or indirect neighbors of p. If p is a central point, this process creates a cluster based on Eps and $MinPts$ parameters. If p is not a central point, and there is no density-reachable point from p, then DBSCAN defines the point p as noise point and the same process is applied for following points in the dataset. If p is in fact the border point of any cluster C, then all density-reachable points from any center point C are grouped together and then assigned to cluster C. Algorithm is finalized after clustering all points and detecting all noise points (Xiong, Chen, Zhang, & Zhang, 2012).

Standard DBSCAN applications are applied in spatial indexes, such as $R-tree$ or $X-tree$ that efficiently support field inquiries of a point located closely to Eps. In worst case scenario, DBSCAN dataset perfoms field inquiry based on dataset point. For DBSCAN, this causes $O(nlogn)$ performance complexity, here, $n-$ is the number of dataset points. Unfortunately, spatial indexes do not provide good results for big data, i.e., field inquiries performance is deteriorated from $O(nlogn)$ to $O(n)$ and DBSCAN performance complexity becomes $O(n^2)$ for data. On the other hand, if there is a grid-based data structure supporting $O(1)$ field inquiries, then DBSCAN performance complexity is reduced to $O(n)$. Note that $O(nlogn)$ performance complexity can be enlarged for big dataset. (Kroger, Kriegel, & Kailing, 2004).

The primary idea of density-based clustering can be generalized in several ways. First, as long as the neighborhood definition is based on a symmetrical and re-existing predicate $NPred(p, q)$, any neighborhood concept can be used instead of distance-based Eps n neighborhood. If p is the neighborhood point of N, then the set of all q points is defined as $NPred(p, q)$. Second, instead of counting the elements in only one neighborhood, in order to determine whether N is dense in the neighborhood, we can use general $MinWeight(N)$ predicate, if $MinWeight$ is monotonous in N. Finally, not only points, but also polygons distributed in the space can be clustered. The GDBSCAN algorithm for finding clusters based on generalized density is a simple modification of the DBSCAN algorithm (Kaufman, & Rousseeuw, 1990).

Stages of DBSCAN algorithm: performance process of DBSCAN algorithm can be interpreted as follows:

Algorithm starts at an arbitrary point and neighborhood information is obtained from $\varepsilon(Eps)$ parameter. If this point is located in ε neighborhood of $MinPts$ parameters, then it forms a cluster. Otherwise, this point is marked as a noise point. This point can later be located in ε neighborhood of a different point and therefore, becomes a part of a cluster. Here, the notion of density-reachable and density-connected points is important. If a point is found to be main, then the points in the ε neighborhood can be considered a cluster. Hence, if all points discovered within ε neighborhood are foremost, then these points are included along with points located in the neighboring areas. Abovementioned process continues until density-connected cluster is completely discovered. Process is restarted with a new point that can be part of a new cluster or marked as a noise point.

5. Experiments

For the analysis of the DBSCAN algorithm, input parameters are initially entered as shown below.

a) Dataset input

Here, different datasets are used. Most datasets pertain to categorized attributes. Note that most of the dataset was obtained from UCI Machine Learning Repository and Kaggle websites.
(https://archive.ics.uci.edu/ml/datasets.php, https://www.kaggle.com/).

b) Tool used (Python)

After reading the dataset, the next step is the Python programming language used to run all of these datasets. (Python - a high-level programming language with interpretive, object-oriented, dynamic semantics.)

c) application of DBSCAN algorithm

One of the important steps of this algorithm is to apply the DBSCAN algorithm to that dataset after entering the entire dataset in the Python programming language.

d) Calculation of parameters

Before applying the DBSCAN algorithm to a dataset, the value of the parameters must be configured to provide a relatively different result from the standard value of the parameter used. Proper selection of parameter values is an important issue of the DBSCAN algorithm.

e) Obtaining results

At this stage, effectiveness of the algorithm is evaluated and the clusters formed using Eps and $MinPts$ parameters, erraneously clustered cases, time calculation and noise points are analysed.

f) Visualization

After obtaining results, different datasets are visualized using graphic images.

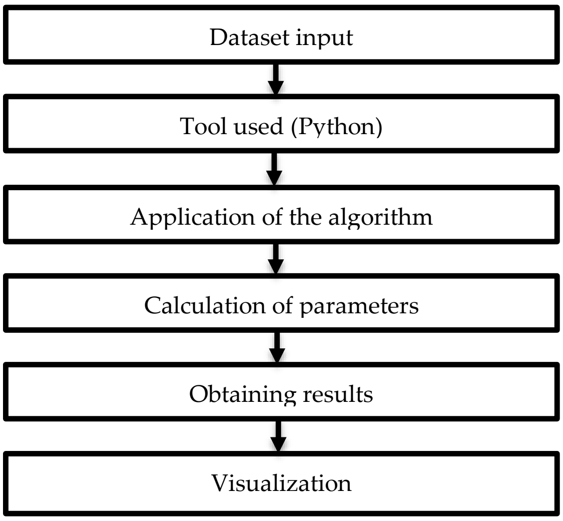


Figure 2. Structure of the applied algorithm

Different types of datasets are used to evaluate the effectiveness of the improved DBSCAN algorithm. The DBSCAN algorithm is applied to all datasets and the datasets are developed in the Python programming language. The description of the datasets used is as shown in Table 1.

Table 1. Dataset characteristics

Title	Number of points	Number of attributes	Application field
Mall Customer	200	5	Business
Wholesale	440	36	Business
Loan	614	16	Social
Live	7050	16	Business
Churn	10000	14	Social
Online Shoppers	12330	18	Business
Adult	48842	15	Census Bureau
Bank – marketing	45211	17	Finance
Diabetic	101766	50	Medicine

As noted, Eps and $MinPts$ parameters must be selected correctly in order to obtain “quality” results from DBSCAN algorithm. First, a correlation coefficient is calculated between the attributes in the dataset. A graph is constructed between the two attributes with strongest correlations. $k-NN$ ($k-$ nearest neighbors) method is used to determine the optimal value of Eps parameter. In $k-NN$ template recognition, $k-$ nearest neighbors algorithm is a non-parametric method used for classification, regression and clustering (Figure 3). The $MinPts$ parameter is set to 5 by default, and then all the points around this value are checked and several evaluation indices are used to analyze the results of the experiment.

Silhouette score, Davies-Bouldin index (1), the Purity index (2), adjusted Rand index (3), and the Homogeneity index are used to evaluate the results of the experiment. Moreover, although the study found the optimal Eps value using $k-NN$, all points defined around the Eps parameter in order to find a good cluster are checked.

Note that the silhouette coefficient calculates the points within a cluster and the distance between the nearest clusters. For example, a cluster with many data points close to each other (high density) will have a higher silhouette coefficient than those of clusters located at a distance from each other. The Silhouette coefficient varies from -1 to 1. -1 is the worst possible value, and 1 is the best maximum value. When the Silhouette coefficient equals to 0, overlapping clusters are proposed.

Davies-Bouldin index is defined as the average size of each cluster with the most similar cluster. The similarity here is the ratio of the distances within the cluster to the distances between the clusters. Therefore, better results will be obtained on farther located clusters. Minimum value of this index is zero and low values of the indexes are considered to be better for clustering.

Davies-Bouldin index is calculated as follows:

$$DB = \frac{1}{n}\sum_{i=1}^{n} \underbrace{max}_{j\neq i}\left(\frac{\sigma_i+\sigma_j}{d(c_i,c_j)}\right), \tag{1}$$

Here, $n-$ is the number of clusters, c_i- center of i^{th} cluster, σ_i- an average distance of cluster i with c_i center and the distance between $d(c_i,c_j)$ c_i and c_j centroids.

Formula (2) is used to calculate the Purity index:

$$Purity = \frac{1}{n}\sum_{i=1}^{k} max_j\left|C_i \cap A_j\right|, \tag{2}$$

Here, $n-$ is the number of objects, $k-$number of clusters, C_i- i^{th} cluster, A_j- j-$^{\text{th}}$ class. The higher the value of the Purity index, the more effective the algorithm.

Let's review the n set of objects in $S=\{O_1,O_2,\dots,O_n\}$ form. Suppose that $U=\{u_1,u_2,\dots,u_R\}$ and $V=\{v_1,v_2,\dots,v_C\}$ are two different subsets of set S. Then, for $1\le i\neq i'\le R$ and $1\le j\neq j'\le C$, it is $\cup_{i=1}^{R} u_i = S = \cup_{j=1}^{C} v_j$; $u_i\cap u_i = \emptyset = v_j \cap v_j$. Suppose that n_{ij} demonstrates the general number of objects in classes u_i and v_j. The Rand Index (Hubert, & Arabie, 1985;) is based on how the object pairs of matching dimensions between U and V are categorized in the $R\times C$-sized random data table.

Table 2. Experiment results

Database	Eps	Min Pts	ARI	Silhouette score	Purity	Homo-genicity	Davies-Bouldin index
Mall Customer (200 x 5)	0.04	5	0.046	0.046	---	---	4.54
	0.05	2	0.049	0.146	---	---	2.90
	0.06	5	0.048	0.181	---	---	3.19
	0.04	4	0.052	0.062	---	---	2.95
	0.04	**3**	**0.047**	**0.088**	**---**	**---**	**1.85**
Wholesale (440 x 36)	0.04	5	0.022	0.441	---	---	0.77
	0.05	**5**	**0.002**	**0.745**	**---**	**---**	**0.72**
	0.03	5	0.003	0.670	---	---	0.87
	0.04	4	0.019	0.724	---	---	0.78
	0.04	6	0.006	0.702	---	---	0.85
Loan (614 x 16)	**0.07**	**5**	**0.001**	**0.699**	**0.68**	**0.84**	**1.47**
	0.03	5	0.001	0.554	0.68	0.74	1.47
	0.05	4	-0.002	0.473	0.68	0.78	0.70
	0.05	5	-0.003	0.452	0.68	0.84	0.72
	0.05	6	-0.001	0.725	0.68	0.68	1.00
Live	0.02	5	0.021	0.719	0.62	0.90	2.36

(7050 x 16)	0.02	9	0.022	0.720	0.62	0.80	1.96
	0.02	7	0.021	0.719	0.62	0.80	1.78
	0.03	**7**	**0.021**	**0.671**	**0.62**	**0.92**	**1.74**
	0.01	7	0.036	0.697	0.62	0.72	0.93
Churn (10000 x14)	0.01	10	0.019	-0.238	0.79	0.74	2.34
	0.02	10	-0.007	-0.311	0.79	0.86	5.75
	0.03	10	-0.013	0.162	0.79	0.76	28.50
	0.02	8	-0.006	-0.268	0.79	0.82	7.68
	0.02	12	-0.005	-0.278	0.79	0.81	6.75
Online Shoppers (12330 x 18)	0.01	10	-0.068	0.578	0.85	0.961	1.043
	0.01	14	-0.073	0.705	0.85	0.969	1.058
	0.01	8	-0.068	0.583	0.85	0.965	1.412
	0.02	**10**	**-0.056**	**0.787**	**0.85**	**0.967**	**1.045**
	0.009	10	-0.072	0.574	0.85	0.964	1.099
Adult (48842 x15)	0.02	10	0.0	0.631	1.0	1.0	0.988
	0.03	**10**	**0.0**	**0.607**	**1.0**	**1.0**	**0.804**
	0.01	10	0.0	0.608	1.0	1.0	1.336
	0.01	7	0.0	0.580	1.0	1.0	1.411
	0.01	12	0.0	0.624	1.0	1.0	1.434
Bank marketing (45211 x 17)	0.02	12	0.011	0.84	0.88	0.88	1.05
	0.02	10	0.009	0.84	0.88	0.88	1.06
	0.03	10	0.005	0.83	0.88	0.94	1.20
	0.01	**10**	**0.025**	**0.70**	**0.88**	**0.94**	**1.07**
	0.02	8	0.008	0.70	0.88	0.81	1.90
Diabetic (101766 x 50	0.02	12	0.0	0.10	1.0	1.0	1.63
	0.03	**12**	**0.0**	**0.31**	**1.0**	**1.0**	**0.99**
	0.02	14	0.0	0.01	1.0	1.0	1.88
	0.01	12	0.0	-0.50	1.0	1.0	1.58
	0.02	10	0.0	0.02	1.0	1.0	1.05

In particular, there are four different types defined among $\binom{n}{2}$ pairs:

1. objects in the pair are placed in the same class in U and V;

2. objects in the pair are placed in different classes in U and V;

3. objects in the pair are placed in different classes in U and same classes in V;

4. objects in the pair are placed in the same class in U and different classes in V;

Typically, the first and second types express the correspondence between the objects in a pair, and the third and fourth types express the discrepancy between objects in a pair. Obviously, if A demonstrates the overall number of correspondences and D demonstrates the overall number of discrepancies, then $A + D = \binom{n}{2}$.

Thus,

$$A = \binom{n}{2} + \sum_{i=1}^{R}\sum_{j=1}^{C} n_{ij}^2 - \frac{1}{2}\left(\sum_{i=1}^{R} n_{i.}^2 + \sum_{j=1}^{C} n_{.j}^2\right) = \binom{n}{2} + 2\sum_{i=1}^{R}\sum_{j=1}^{C}\binom{n^{ij}}{2} - \left(\sum_{i=1}^{R}\binom{n_{i.}}{2} + \sum_{j=1}^{C}\binom{n_{.j}}{2}\right). \quad (3)$$

Table 2 below provides a detailed description of the data and parameters after application of all abovementioned to various datasets. The following experiments describe the initial clustering for different datasets, the final evaluation using the $k - NN$ method and the DBSCAN algorithm.

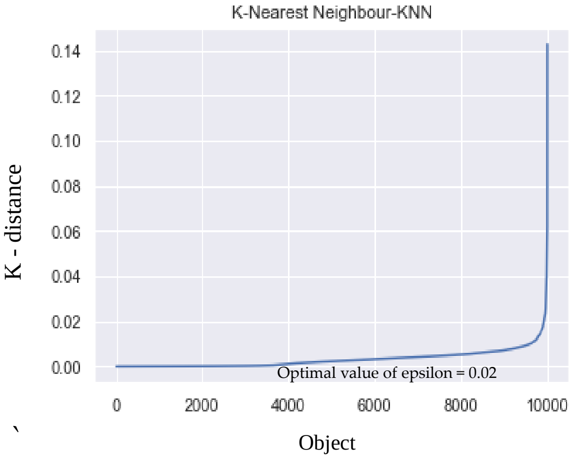


Figure 3. Finding optimal value of epsilon using k-NN

Based on the figure, note that the optimal value of the Eps parameter is found by applying the k-NN method. The value of the MinPts parameter is considered to be less than 6 when the dataset is small, and 7 and higher when the dataset is large. Evaluations are made using metrics based on the values of the parameters found. To obtain more efficient cluster, the values of these parameters are changed by increasing or decreasing the values set by a few units. Note that the minimum value of the Davies-Bouldin index used for evaluation, and the maximum value of the Homogeneity and Purity index indicate better performance. Table 2 creates the clusters mainly for the categorized data according to the Homogeneity index, and for uncategorized data (the first two datasets) according to the Davies-Bouldin index. If the Homogeneity or Purity index in the categorized data are the same in all cases, then here, a cluster is also created, taking into account the Davies-Bouldin index.

Conclusion

This article studied the concept of big data and theoretically and comparatively analyzed the clustering algorithms for big data clustering. The density-based DBSCAN algorithm was applied to analyze big data and accurately detect noise points; its practical significance for big data was determined. Various metrics (Silhouette score, Adjusted Rand index, Purity index and Homogeneity index, etc.) were used to increase the efficiency during evaluation of the DBSCAN algorithm. According to the results of the experiments, the DBSCAN algorithm demonstrated high perfomrance for various indices. As mentioned, although the algorithm is sensitive to the selection of two main parameters (Eps, $MinPts$), the study yielded quality clusters. Consequently, proposed method yielded more effective performance in detecting noise points in comparison with the traditional DBSCAN algorithm.

References

Alguliyev, R. M., Aliguliyev, R. M., & Sukhostat, L. V. (2019). Efficient algorithm for big data clustering on single machine. CAAI Transactions on Intelligence Technology, *5*(1), 9-14. https://doi.org/10.1049/trit.2019.0048

Alguliyev R.M., Aliguliyev R.M., Abdullayeva F.J. (2019). Privacy-preserving deep learning algorithm for big personal data analysis. Journal of Industrial Information Integration, 15, 1-14. https://doi.org/10.1016/j.jii.2019.07.002

Alguliyev, R., Aliguliyev, R., & Sukhostat, L. (2017). Anomaly detection in Big databased on clustering. *Statistics, Optimization & Information Computing*, *5*(4), 325-340. https://doi.org/10.19139/soic.v5i4.365

Alguliyev, R., & Imamverdiyev, Y. (2014). Big data: Big promises for information security. In *2014 IEEE* 8th International Conference on Application of Information and Communication Technologies (AICT) (pp. 1-4). IEEE. 10.1109/ICAICT.2014.7035946

Alguliyev R., & Hajirahimova M. (2014). "BIG DATA" PHENOMENON: CHALLENGES AND OPPORTUNITIES. Problems of information technology, *5*(2), 3-16. https://jpit.az/en/journals/120

Aliguliyev R., Hajirahimova M., & Aliyeva A. (2016). Current scientific and theoretical problems of Big data. Problems of information society, (2), 37-49 (in Azerbaijani). https://jpis.az/az/journals/138

Andrieu, C., De Freitas, N., Doucet, A., & Jordan, M. I. (2003). An introduction to MCMC for machine learning. *Machine learning*, *50*(1), 5-43. https://doi.org/10.1023/A:1020281327116

Berkhin, P. (2006). A survey of clustering data mining techniques. In *Grouping multidimensional data* (pp. 25-71). Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-28349-8_2

Cassisi, C., Ferro, A., Giugno, R., Pigola, G., & Pulvirenti, A. (2013). Enhancing density-based clustering: Parameter reduction and outlier detection. *Information Systems*, *38*(3), 317-330. https://doi.org/10.1016/j.is.2012.09.001

Chana I.A., & Arora S. (2014). Survey of clustering techniques for big data analysis. 5th International Conference - Confluence the Next Generation Information Technology Summit, 59-65. 10.3233/JIFS-202503

Chandra, E., & Anuradha, V. P. (2011). A survey on clustering algorithms for data in spatial database management systems. International Journal of Computer Applications, *24*(9), 19-26.

Dharni, C., & Bnasal, M. (2013). An improvement of DBSCAN Algorithm to analyze cluster for large datasets. In 2013 IEEE international conference in MOOC, innovation and technology in education (MITE) (pp. 42-46). IEEE. 10.1109/MITE.2013.6756302

Duan, L., Xu, L., Guo, F., Lee, J., & Yan, B. (2007). A local-density based spatial clustering algorithm with noise. Information systems, 32(7), 978-986. https://doi.org/10.1016/j.is.2006.10.006

El-Sonbaty, Y., Ismail, M. A., & Farouk, M. (2004). An efficient density based clustering algorithm for large databases. In 16th IEEE international conference on tools with artificial intelligence (pp. 673-677). IEEE. 10.1109/ICTAI.2004.27

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In kdd (Vol. 96, No. 34, pp. 226-231).

Fahad, A., Alshatri, N., Tari, Z., Alamri, A., Khalil, I., Zomaya, A. Y., ... & Bouras, A. (2014). A survey of clustering algorithms for big data: Taxonomy and empirical analysis. IEEE transactions on emerging topics in computing, *2*(3), 267-279. 10.1109/TETC.2014.2330519

Fakhraddingizi A. (2019). Fundamental issues of data security in big data technologies. Actual multidisciplinary scientific-practical problems of information security, V republic conference, 226-228. (in Azerbaijani).

Gaonkar, M. N., & Sawant, K. (2013). AutoEpsDBSCAN: DBSCAN with Eps automatic for large dataset. International Journal on Advanced Computer Theory and Engineering, 2(2), 11-16.

https://archive.ics.uci.edu/ml/datasets.php

https://www.kaggle.com/

https://www.kdnuggets.com/2017/04/42-vs-big-data-data-science.html

Hubert, L., & Arabie, P. (1985). Comparing partitions. Journal of classification, 2(1), 193-218. https://doi.org/10.1007/BF01908075

Karypis, G., Han, E. H., & Kumar, V. (1999). Chameleon: Hierarchical clustering using dynamic modeling. Computer, 32(8), 68-75. 10.1109/2.781637

Kaufman, L., & Rousseeuw, P. J. (2009). Finding groups in data: an introduction to cluster analysis (Vol. 344). John Wiley & Sons. https://books.google.az/books

Kailing, K., Kriegel, H. P., & Kröger, P. (2004, April). Density-connected subspace clustering for high-dimensional data. In Proceedings of the 2004 SIAM international conference on data mining (pp. 246-256). Society for Industrial and Applied Mathematics. 10.1137/1.9781611972740.23

Liu, P., Zhou, D., & Wu, N. (2007). VDBSCAN: varied density based spatial clustering of applications with noise. In 2007 International conference on service systems and service management (pp. 1-4). IEEE. 10.1109/ICSSSM.2007.4280175

Moreira, A., Santos, M. Y., & Carneiro, S. (2005). Density-based clustering algorithms–DBSCAN and SNN. University of Minho-Portugal, 1-18. http://get.dsi.uminho.pt/local/download/SNN&DBSCAN.pdf

Parimala, M., Lopez, D., & Senthilkumar, N. C. (2011). A survey on density based clustering algorithms for mining large spatial databases. International Journal of Advanced Science and Technology, 31(1), 59-66. 10.1.1.643.6121

Rahmah, N., & Sitanggang, I. S. (2016). Determination of optimal epsilon (eps) value on dbscan algorithm to clustering data on peatland hotspots in sumatra. In IOP conference series: earth and environmental science (Vol. 31, No. 1, p. 012012). IOP Publishing. 10.1088/1755-1315/31/1/012012

Sajana, T., Rani, C. S., & Narayana, K. V. (2016). A survey on clustering techniques for big data mining. Indian journal of Science and Technology, 9(3), 1-12. 10.17485/ijst/2016/v9i3/75971

Shah, G. H. (2012). An improved DBSCAN, a density based clustering algorithm with parameter selection for high dimensional datasets. In 2012 Nirma university international conference on engineering (NUiCONE) (pp. 1-6). IEEE. 10.1109/NUICONE.2012.6493211

Sharma, S., Sharma, A. K., & Soni, D. (2017). Enhancing DBSCAN algorithm for data mining. In 2017 International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS) (pp. 1634-1638). IEEE. 10.1109/ICECDS.2017.8389724

Uncu, O., Gruver, W. A., Kotak, D. B., Sabaz, D., Alibhai, Z., & Ng, C. (2006, October). GRIDBSCAN: GRId density-based spatial clustering of applications with noise. In 2006 IEEE International Conference on Systems, Man and Cybernetics (Vol. 4, pp. 2976-2981). IEEE. 10.1109/ICSMC.2006.384634

Xiong, Z., Chen, R., Zhang, Y., & Zhang, X. (2012). Multi-density dbscan algorithm based on density levels partitioning. Journal of Information and Computational Science, 9(10), 2739-2749.

Zhou, L., Pan, S., Wang, J., & Vasilakos, A. V. (2017). Machine learning on big data: Opportunities and challenges. Neurocomputing, 237, 350-361. https://doi.org/10.1016/j.neucom.2017.01.026

Zhao, W., Ma, H., & He, Q. (2009). Parallel k-means clustering based on mapreduce. In IEEE international conference on cloud computing (pp. 674-679). Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-10665-1_71.