ISSN 2007-9737

An Adaptative Eps Parameter of DBSCAN Algorithm for Identifying Clusters with Heterogeneous Density

Nasereddine Amroune[1,2,*], Maklouf Benazi[1,2], Lamri Sayad[1,2]

[1] Mohamed Boudiaf University of M´sila,
Department of computer science, M´sila,
Algeria

[2] University of M’sila,
Laboratory of Informatics and its Applications of M´sila,
Algeria

{nasereddine.amroune, maklouf.benazi, lamri.sayad}@univ-msila.dz

Abstract. Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is one of the most important data clustering algorithms. Its importance lies in the fact that it can recognize clusters of arbitrary shapes and is not affected by noise in the data. To identify clusters, DBSCAN needs to specify two parameters: the parameter Eps, representing the radius of the circle to identify the neighborhood of each observation. The second parameter of DBSCAN is minpts, which represents the minimum size of the neighborhood for a point to be a seed in a cluster and not a noise. However, the task of determining the adequate value of Eps parameter is not easy and represents a major issue when applying DBSCAN since the accuracy of this algorithm highly depends on the values of its parameters. In this paper, we present a new version of DBSCAN where we need only to specify the minpts parameter, then we use k-nearest neighbors (kNN) algorithm to calculate the value of Eps automatically for every point in the data. This technique not only reduces the number of parameters by eliminating Eps which has been very difficult to determine, but also gives DBSCAN the ability to detect clusters with heterogeneous density. The experimental results show that the proposed method is more efficient and more accurate than the original DBSCAN algorithm.

Keywords. Data mining, clustering, density-based algorithms, DBSCAN algorithm, eps parameter, k-nearest neighbors.

1 Introduction

Over the last decades, unsupervised classification, or more precisely data clustering, has become one of the most important techniques in machine learning. This technique allows to identify significant patterns in the dataset based on their similarity in order to better understand them and make more accurate predictions.

It consists in grouping unlabeled objects of a dataset in subgroups called clusters. objects in each cluster should be as similar as possible to each other and as dissimilar as possible to members of other clusters.

The clustering technique has applications in many areas of the real world. for example, astronomy [1], biology [2], engineering [3], computer science [4], marketing [5], insurance [6], medicine [7], etc. In the literature, many clustering methods and algorithms have been proposed. They can be grouped into four categories.

– Partitioning methods where objects are divided into k distinct groups so that the objects in each group are as close as possible to the center of its group and as far away as possible from the centers of the other groups.
– The typical examples of this category are k-means [8] and k-median [9].

- Hierarchical methods are types of approaches where we start with one element in each cluster, then we merge iteratively the two most similar clusters until only one is left like in SLINK [10], or conversely, we start with one cluster and split it until each element has its own cluster like in CLINK [11]. In this type of clustering, it is necessary to provide another method or way to determine at what level to stop the merge or división process.
- Model-based methods: In this category, the cluster is seen as a distribution in a mixture of distributions (e.g., gaussian mixture) and the objective of clustering is to find the original models that generate these distributions. such as the work proposed in [12].
- Density-based methods: in this type of clustering, the cluster is seen as a set of high-density objects separated by low-density objects.
- Algorithms of this type can easily identify clusters of arbitrary shape as they can also handle noise and outliers in the data. The two best known algorithms in this family are DBSCAN [13] and OPTICS [14].

DBSCAN Density-Based Spatial Clustering of Applications with Noise is a density-based clustering algorithm that can identify clusters of arbitrary shapes and is not affected by noise in the data.

However, DBSCAN has two drawbacks: to identify clusters, it needs two parameters, Eps and minpts, which are very difficult to specify, and its inability to detect clusters with heterogeneous density. In this article, we present a new version of DBSCAN which can identify clusters of arbitrary shapes with heterogeneous density, and which only requires the specification of the minpts parameter. the algorithm then uses the k-nearest neighbors (kNN) algorithm to automatically calculate Eps for all points in the data.

The rest of this paper is organized as follows: In section 2, we present the basic concept of DBSCAN algorithm. Then, some variants of DBSCAN algorithm are explained in section 3. Section 4 is devoted to presenting our approach in detail. To show the performance of our method, section 5 gives some results of the application of our method on different datasets well known in the clustering world. Finally, section 6 concludes the paper and gives some perspectives.

2 DBSCAN Algorithm

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm, that can identify clusters with arbitrary shapes and different sizes. Unlike k-means, clusters are not necessarily spheroidal and there is no need to know the number of clusters in advance.

Algorithm 1 DBSCAN algorithm

```
Input:  X: dataset,
        minpts: minimum points,
        Eps: maximum distance
Output: Labels
1. Find sets of points within the epsilon neighborhood
   of each point in X.
2. Select the first unclassified point
   a. If the number of neighbors is less than minpts,
      then label the point as a noise point. Go to step 2.
   b. Otherwise, label the current point as a core point
      belonging to cluster C. and insert all neighbors
      into queue.
3. Iterate over all unclassified points in the queue
   a. If the current point is labeled as noise or the
      number of neighbors is less than minpts, than label
      the current point as a border point belonging to C
      cluster
   b. Otherwise, label the current point as a core point
      belonging to cluster C. and add all its neighbors
      into the queue.
4. Select the next unclassified point in X, and increase
   the cluster count by 1.
5. Repeat steps 2–4 until all points in X are labeled.
```

Clusters are formed by the identifying points that are density-connected. The algorithm uses two parameters, minpts and eps, to define density in data. It can identify three types of points: core points, border points, and noise points [13], which make it more suitable for processing noisy. To assign a point to a cluster, the point must have at least minpts in its radius neighborhood (eps), in this case, we call it core point.

Or, the point is within the radius neighborhood of another core point, in this case, we call it border point. Otherwise, it will be considered a noise point. The original DBSCAN algorithm is presented below.

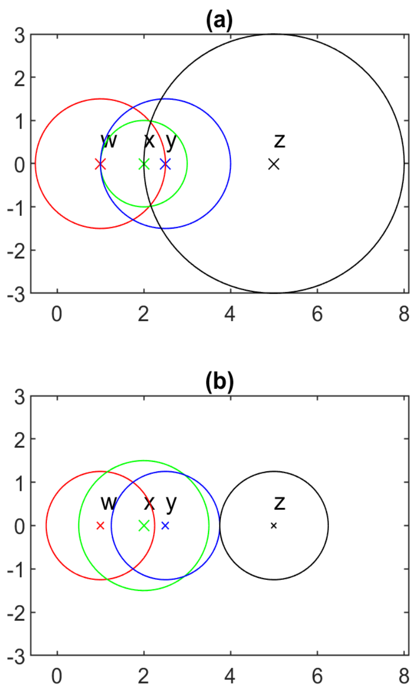


Fig. 1. Example to illustrate the results obtained by applying formula 1

To identify clusters, DBSCAN depends heavily on the parameter eps, which represents the maximum distance to search for neighbors.

In addition to being difficult to guess, using a global value for eps can lead DBSCAN to not find the right clusters, especially when data have clusters with different densities.

When taking eps with small values, low density clusters may be considered as noise. However, if eps take a large value, DBSCAN will merge high density clusters.

3 Related Work

Recent research has been focused on using a single parameter with the widely used clustering algorithm, DBSCAN, to reduce the complexity associated with parameter tuning while maintaining high-quality clustering results.

Various papers have proposed different approaches for using a single parameter with DBSCAN, and this approach has gained attention from the research community. In this section, the recent papers that explore these approaches and their potential to achieve high-quality clustering results will be reviewed and discussed.

A. Bryant and K. Cios [15] have developed a new clustering algorithm called RNN-DBSCAN, which estimates observation density using reverse nearest neighbor counts and employs a DBSCAN-like approach based on k-nearest neighbor graph traversals through dense observations.

This algorithm offers two significant benefits: it reduces problem complexity to a single parameter and enhances the ability to handle large variations in cluster density. However, it may not be effective in distinguishing adjacent clusters with different densities, which could limit its usefulness in some situations.

Hu et al. [16] proposed a new density-based clustering algorithm called KR-DBSCAN. The algorithm is based on the reverse nearest neighbor and influence space. KR-DBSCAN distinguishes adjacent clusters with different densities, reduces computational load, and identifies boundary objects and noise objects.

The algorithm defines a new cluster expansion condition using the reverse nearest neighborhood and its influence space and adds core objects to the cluster by breadth-first traversal.

However, KR-DBSCAN is more complex than standard DBSCAN due to the use of influence space, which can lead to higher computational cost for large databases. In addition to these approaches, other papers have proposed novel mechanisms to find the best value of epsilon for a given dataset.

For instance, the Multi-verse optimizer MVO algorithm [17] was used to find the Eps interval corresponding to the highest accuracy of DBSCAN. The AE-DBSCAN algorithm [18] uses the first slope, which is greater than the mean plus standard deviation of all non-zero slopes, as Eps.

AutoEpsDBSCAN [19] uses the kdist graph to select several values of the Eps parameter for different densities in the dataset before applying traditional DBSCAN. Overall, the research reviewed in this section indicates that using a single parameter with DBSCAN can be a promising alternative to the traditional two-parameter approach for DBSCAN.

Different approaches have been proposed, each with its strengths and limitations, and the choice of the appropriate approach depends on the characteristics of the dataset and the desired clustering quality and efficiency.

4 Modified DBSCAN

In this section, we explain our proposed algorithm, which can be carried out in two steps. In the first step, the eps values are automatically computed for every point in the data. To do this, we calculate the k-distances of all its k nearest points and then, we calculate the average according to the following formula:

$$eps(i) = \frac{1}{k}\sum kdist(j) \quad (j \in N(i)), \tag{1}$$

where i is the point, whose eps will be calculated, k is the number of points in a neighborhood of i, in DBSCAN algorithm k is called minpts, N(i) is the set of nearest neighbors of the point i, and kdist is the function that calculates the distance between a point j and its kth nearest neighbor.

Algorithm 2 Computing eps and set of neighbors

```
Input: X: Dataset, k: minimum points.
Output: eps,N:set of neighbors of each point in X
        Calculate k-distance values of all points in X
for each observation $x_i$ in X
    Calculate eps($x_i$),theaverage k-distance of k nearest neighbors of $x_i$.
    Find N($x_i$), set of neighbors of $x_i$ within eps($x_i$)
    Mark $x_i$ as unclassified
end for
```

The second step is to use DBSCAN to find clusters using eps computed in the first step.For a better understanding, the following example illustrates the results obtained by applying formula 1 to a simple mono-dimensional dataset of four points.

Fig 1.a shows the circles corresponding to the kth nearest neighbors for each point, here k=3, and Fig 1.b shows circles whose radius are equal to the average k-distance of their k nearest neighbors (formula 1).

We can see that radius of the point "x" is extended to be core point, while the radius of the other three points have been reduced to get fewer points in their neighborhoods (less than k), making the point "w" and the point "y" borders points because they fall within the neighborhood of "x" and make "z" as noise point.

Algorithm 3 Our DBSCAN algorithm

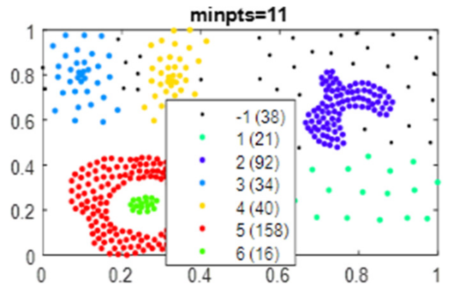
```
Input:  X: Dataset,
        N: Set of neighbors,
        minpts: minimum points.
Output: Labels
$C = 0$
for each observation $x_i$ in $X$
    Select the next unclassified point in $X$
    if $|N(xi)| <$ minpts
        Mark $x_i$ as noise.
        Go to the next observation in $X$
    end if
    $C = C + 1$ // start a new cluster
    Mark $x_i$ as a core point belonging to cluster $C$
    Insert $N(x_i)$, neighbors of $x_i$, into queue $Q$
    while $Q$ is not empty
        Select qi the next unclassified point in $Q$
        if $q_i$ is marked as noise or $|N(q_i)| <$ minpts
            Mark $q_i$ as a border point belonging to cluster $C$.
        else
            Mark $q_i$ as a core point belonging to cluster $C$.
            Insert $N(q_i)$, neighbors of $q_i$, into queue $Q$.
        end if
    end while
end for
```

5 Experimental Evaluation

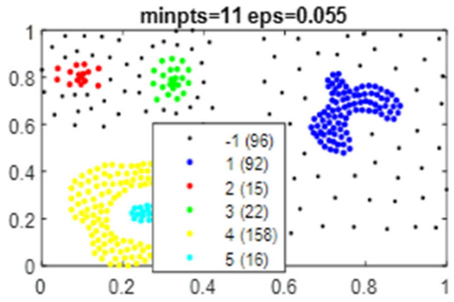
In the previous section, we presented a novel version of the DBSCAN algorithm that utilizes only the 'minPts' parameter for clustering.

In this section, we aim to provide empirical evidence of the effectiveness of our new algorithm by conducting a comparative analysis with the original DBSCAN algorithm, which uses two parameters.

To assess the effectiveness of our proposed algorithm, we utilized three synthetic 2-D datasets of different shapes and different densities. The first dataset, named 'Compound', comprises 399 points divided into six clusters with varying densities,

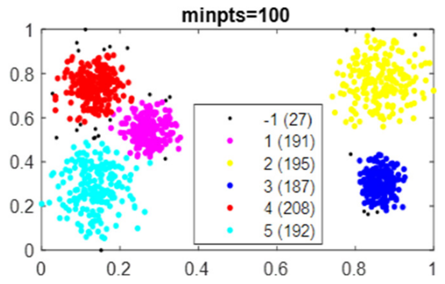


(a)

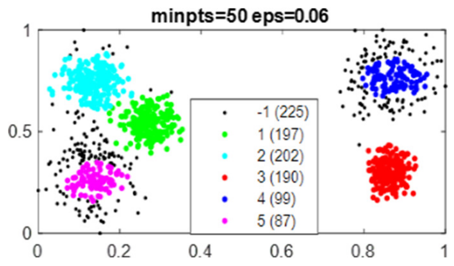


(b)

Fig. 2. Our-DBSCAN (a) and DBSCAN (b) clustering results for the compound dataset



(a)



(b)

Fig. 3. Our-DBSCAN (a) and DBSCAN (b) clustering results for the asymmetric dataset

posing a challenge to accurate clustering. The second dataset, 'asymmetric', is composed of 1000 points divided into five clusters.

The third dataset, 'hard', includes 1500 points divided into three clusters of different densities.

Figure 2 displays the clustering results of the first dataset, where our method successfully identified 6 distinct clusters with only 38 data points labeled as noise. In contrast, the original DBSCAN algorithm detected only 5 clusters and classified the sixth cluster as noise, resulting in 96 noise points.

Figure 3 illustrates the clustering results obtained by two algorithms. our algorithm successfully identified five distinct clusters with 27 points labeled as noise. On the other hand, the original DBSCAN algorithm could barely detect five clusters with 225 points marked as noise.

If we increase the value of epsilon, it will merge the two groups in the upper left, while decreasing the value of epsilon will classify the group in the upper right as noise.

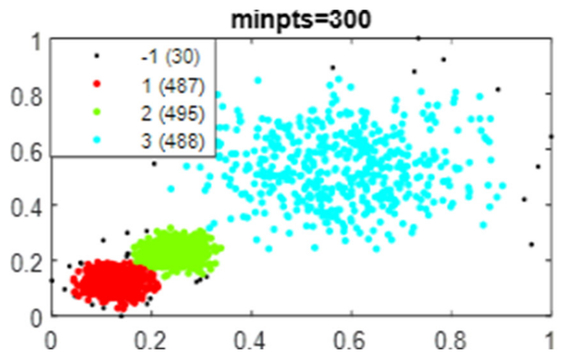
In reference to the hard database shown in Figure 4, it is not possible to determine a suitable value for the parameter eps that would allow us to identify the three clusters using the original DBSCAN algorithm.

This is because the algorithm relies on a fixed global value for eps, whereas our modified algorithm was able to successfully detect the three distinct clusters with only 30 points labeled as noise.

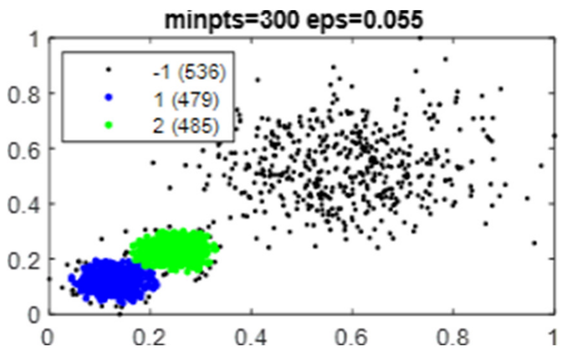
To further assess the effectiveness of our approach, we conducted experiments on widely-used datasets such as iris and seed. Since these datasets come with true labels, we were able to measure the accuracy of our method and compare it with that of the original DBSCAN algorithm.

Table 1 provides a summary of the results we obtained. As per the data presented in Table 1, it is evident that the proposed method exhibits better performance than the traditional DBSCAN algorithm on the iris and seed datasets. For the iris dataset, our DBSCAN algorithm achieved an accuracy of 0.86 compared to 0.67 with the traditional DBSCAN algorithm.

Additionally, our algorithm was able to detect three clusters with 17 noise points compared to two clusters with no noise points in the traditional algorithm.



(a)



(b)

Fig. 4. Our-DBSCAN (a) and DBSCAN (b) clustering results for the Asymmetric dataset

Table 1. Performance of our method on Real-World datasets

Dataset Algorithm		Iris	Seed
DBSCAN	Accuracy	0.6667	0.5905
	NMI	0.7612	0.5536
	Cluster/Noise	2/0	3/81
	Minpts/eps	12/0.025	14/0.19
Our DBSCAN	Accuracy	0.8600	0.8048
	NMI	0.7376	0.6363
	Cluster/noise	3/17	5/15
	Minpts	12	14

This indicates that our algorithm is better at detecting the underlying structure in the iris dataset and is able to handle noisy data more effectively.

Similarly, for the seed dataset, our algorithm achieved an accuracy of 0.8048 compared to 0.5905 with the traditional DBSCAN algorithm. Our algorithm was also able to detect five clusters with only 15 noise points compared to three clusters with 81 noise points in the traditional algorithm.

This again shows that our algorithm is more robust and accurate in detecting the underlying structure in the data and is more effective in dealing with noisy data. Furthermore, our algorithm achieved a higher accuracy on both datasets even though the traditional algorithm achieved higher NMI on the iris dataset.

This suggests that our algorithm is more effective at finding meaningful clusters rather than just maximizing the agreement with the ground truth. Overall, the results suggest that the proposed DBSCAN algorithm is superior to the traditional DBSCAN algorithm in terms of accuracy, the number of clusters detected, and handling noisy data.

6 Conclusion

DBSCAN is a density-based clustering algorithm that can find clusters of arbitrary shapes and sizes. To discover the clusters, the algorithm depends on two parameters, minpts and eps. The problem with eps is that it is difficult to determine its value and takes a single global value for all data, which does not allow DBSCAN to discover clusters of different densities.

To solve these two problems, we proposed in this paper a new version of DBSCAN that can determine the eps value automatically and locally for each observation in the data. Experimental results show that this new version of DBSCAN can help identify good quality clusters.

As a perspective, we will try to figure out how to automatically determine the minpts value from the data, so that the algorithm will be completely parameter free.

References

1. **Yu, H., Hou, X. (2022).** Hierarchical clustering in astronomy. Astronomy and Computing, Vol. 41, pp. 100662. DOI: 10.1016/j.ascom.2022.100662.

2. **Ye, X., Ho, J. W. K. (2019).** Ultrafast clustering of single-cell flow cytometry data using

flowgrid. BMC Systems Biology, Vol. 13, No. 35. DOI: 10.1186/s12918-019-0690-2.

3. **Pham, D. T., Afify, A. A. (2007).** Clustering techniques and their applications in engineering. Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science, Vol. 221, No. 11, pp. 1445–1459. DOI: 10.1243/09544 062jmes508.
4. **Lu, X., Zhou, Y., Qiao, L., Yu, W., Liang, S., Zhao, M., Zhao, Y., Lu, C., Chi, N. (2019).** Amplitude jitter compensation of PMA-8 VLC system employing time-amplitude two-dimensional re-estimation base on density clustering of machine learning. Physica Scripta, Vol. 94, No. 5, pp. 055506. DOI: 10.10 88/1402-4896/ab0a9f.
5. **Ližbetinová, L., Štarchoň, P., Lorincová, S., Weberová, D., Průša, P. (2019).** Application of cluster analysis in marketing communications in small and medium-sized enterprises: an empirical study in the Slovak Republic. Sustainability, Vol. 11, No. 8, pp. 2302. DOI: 10.3390/su11082302.
6. **Gramegna, A., Giudici, P. (2020).** Why to buy insurance? an explainable artificial intelligence approach. Risks, Vol. 8, No. 4, pp. 137.DOI: 10.3390/risks8040137.
7. **Zhang, X., Wang, D., Chen, H. (2019).** Improved biogeography-based optimization algorithm and its application to clustering optimization and medical image segmentation. IEEE Access, Vol. 7, pp. 28810–28825. DOI: 10.1109/access.2019. 2901849.
8. **Likas, A., Vlassis, N., Verbeek, J. J. (2003).** The global k-means clustering algorithm. Pattern Recognition, Vol. 36, No. 2, pp. 451–461. DOI: 10.1016/s0031-3203(02) 00060-2.
9. **Kohonen, T. (1985).** Median strings. Pattern Recognition Letters, Vol. 3, No. 5, pp. 309–313. DOI: 10.1016/0167-8655(85)9 0061-3.
10. **Müllner, D. (2011).** Modern hierarchical, agglomerative clustering algorithms. DOI: 10. 48550/ARXIV.1109.2378.
11. **Defays, D. (1977).** An efficient algorithm for a complete link method. The Computer Journal, Vol. 20, No. 4, pp. 364–366. DOI: 10.1093/com jnl/20.4.364.
12. **Yousri, N. A., Kamel, M. S., Ismail, M. A. (2009).** A distance-relatedness dynamic model for clustering high dimensional data of arbitrary shapes and densities. Pattern Recognition, Vol. 42, No. 7, pp. 1193–1209. DOI: 10.1016/j. patcog.2008.08.037.
13. **Ester, M., Kriegel, H., Sander, J., Xu, X. (1996).** A density-based algorithm for discovering clusters in large spatial databases with noise. Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, pp. 226–231.
14. **Ankerst, M., Breunig, M. M., Kriegel, H., Sander, J. (1999).** Optics: ordering points to identify the clustering structure. ACM SIGMOD Record, Vol. 28, No. 2, pp. 49–60. DOI: 10.11 45/304181.304187.
15. **Bryant, A., Cios, K. (2018).** RNN-DBSCAN: A density-based clustering algorithm using reverse nearest neighbor density estimates. IEEE Transactions on Knowledge and Data Engineering, Vol. 30, No. 6, pp. 1109–1121. DOI: 10.1109/tkde.2017.2787640.
16. **Hu, L., Liu, H., Zhang, J., Liu, A. (2021).** KR-DBSCAN: A density-based clustering algorithm based on reverse nearest neighbor and influence space. Expert Systems with Applications, Vol. 186, pp. 115763. DOI: 10.10 16/j.eswa.2021.115763.
17. **Lai, W., Zhou, M., Hu, F., Bian, K., Song, Q. (2019).** A new DBSCAN parameters determination method based on improved MVO. IEEE Access, Vol. 7, pp. 104085–104095. DOI: 10.1109/access.20 19.2931334.
18. **Ozkok, F. O., Celik, M. (2017).** A new approach to determine EPS parameter of DBSCAN algorithm. International Journal of Intelligent Systems and Applications in Engineering, Vol. 4, No. 5, pp. 247–251. DOI: 10.18201/ijisae.2017533899.
19. **Gaonkar, M. N., Sawant, K. (2013).** AutoEpsDBSCAN: DBSCAN with EPS automatic for large dataset. International Journal on Advanced Computer Theoryand Engineering, Vol. 2, No. 2, pp. 2319–2526.

Article received on 01/05/2023; accepted on 18/04/2024.
Corresponding author is Nasereddine Amroune.