Challenges for DBSCAN: Closely Adjacent Clusters and Varying Densities

Anne-Katrin Link
Institute of Technology Blanchardstown
Dublin 15

15th December 2018

Abstract

DBSCAN is a very popular density-based clustering algorithm. However, DBSCAN struggles with identifying clusters in data sets that contain bordering, nested or overlapping clusters of varying or similar densities. Only one density can be set each time the algorithm is run, and the model is non-deterministic, potentially leading to different results each time the model is run. A case study is presented, and our analysis of literature showed that algorithms with the concept of border points should be avoided for data sets with bordering, nested or overlapping clusters, and alternative algorithms are outlined. Hierarchical density-based clustering methods such as OPTICS or HDBSCAN, or alternative density-based methods, such as SNN clustering and fuzzy clustering, could help with solving these issues. We suggest that future research should focus on investigating the optimization of density-based models for data sets with bordering, overlapping, or nested clusters.

Keywords: clustering, HDBSCAN, DBSCAN, OPTICS, shared nearest neighbour clustering, SNN clustering, fuzzy clustering, deterministic, unsupervised learning, varying densities, nested clusters

1 Introduction

In supervised machine learning, an algorithm ”learns” patterns based on previously determined relationships or classes within a training data set, and it uses these findings to make predictions in test data sets (Tan et al., 2005). In contrast, unsupervised machine learning establishes groups of unlabelled data points without prior training or response variable. These methods are commonly used in data exploration to discover unknown patterns or structures.

1.1 Clustering

Clustering, an unsupervised machine learning method, is often used in spatial database technology, data mining, marketing, and in different fields within science (Han et al.,

2011). The algorithms detect clusters by grouping data points that are similar, which in turn are dissimilar to data points in other clusters (Han et al., 2011). Similarity or closeness is a measure of how much alike two data points as determined by a distance metric. For continuous data, Euclidean distance is most commonly used, which is a metric to measure distance in a straight line (Tan et al., 2005).

1.2 Types of Clustering Methods

Clustering algorithms, including hierarchical, partitioning, grid-based, and density-based clustering algorithms, differ in the following characteristics (Han et al., 2011):

- Hierarchical models create nested clusters in the shape of a hierarchical tree, where each tree node stands for a cluster.
- In Partitioning models, the data is split into k partitions, each representing a cluster. A popular example is K-means clustering. The user needs to be familiar with the data set and know the number of clusters before the model is run.
- Grid-based models separate the space with data points into cells which make up a grid structure, on which the clustering is performed.
- Density-based models search for dense areas of data points and bring them together into clusters.

1.3 Importance and Objective

The algorithm "Density-Based Spatial Clustering of Applications with Noise" (DBSCAN) (Ester et al., 1996) does not make any specific assumptions for a data set (Tan et al., 2005). At the same time, it cannot differ between densities of different clusters or distinguish between closely adjacent clusters (Tan et al., 2005; Ertöz et al., 2003; Kriegel et al., 2011). This paper will elaborate on this issue in the following sections: Section 2 explains DBSCAN. Section 3 discusses how DBSCAN handles data sets with bordering clusters of varying or similar densities. Subsequently, Section 4 provides possible alternatives to DBSCAN and provides suggestions for future research. Finally, Section 5 concludes on the paper.

2 DBSCAN

DBSCAN is a density-based clustering algorithm created in 1996 by Ester, Kriegel, Sander and Xu. Commonly used and awarded with the "Test-of-Time" award at the data mining conference SIGKDD in 2014 (Schubert et al., 2017), it can detect patterns that other algorithms cannot find, such as clusters of arbitrary shapes (Ester et al., 1996). Moreover, unlike in other clustering algorithms, the number of clusters does not need to be entered before the model is run, and it can detect outliers (Ester et al., 1996). In DBSCAN, density is defined as "the number of data points in a unit n-dimensional space", with the term "dimension" referring to the attributes within a data

set (Kotu and Deshpande, 2014). DBSCAN's basic concept is that a cluster is a dense group of data points, close to other points that are also part of the cluster, while points in low-density areas are outliers (Ester et al., 1996).

2.1 Clustering Process

According to Tan et al. (2005) and Kotu and Deshpande (2014), there are three main steps in the algorithm. Firstly, a threshold density is defined by parameter tuning. The second step is the identification of data point classes, after which the data clustering process completes the process in the third step.

2.1.1 Step 1: Parameter Tuning

Before the algorithm scans the data set, two user-defined parameters need to be set to ensure that the algorithm can identify a high-density area (Ester et al., 1996):

- Epsilon, a fixed threshold radius around a data point to define an area called "neighbourhood", and
- MinPts, a threshold number of data points within a neighbourhood as defined by Epsilon

Since it can be challenging to find the right settings for the parameters Epsilon and MinPts, a k-distribution graph (Figure 1), which stems from the k-nearest neighbour algorithm (Kotu and Deshpande, 2014), helps to decide where to set the thresholds.

Figure 1: K-Distance Graph of the Iris Data Set, K=4 (Kotu and Deshpande, 2014)

In this graph, k-distance values for all individual data points in the data set are calculated and sorted in a distribution graph in descending order. In Figure 1, points that are further to the right as part of the cluster (high-density), and points further on the

left as outliers (low-density). For a point within a cluster, the distances will be shorter (the value will be smaller), compared to a point that is not part of the cluster, where the distances will be longer (the value will be larger). The optimal value for Epsilon is at a steep change in the graph's curve, indicating the border between the points within the cluster and the outliers (Schubert et al., 2017). Ester et al. (1996) recommended using MinPts = 4 for all databases in a 2-dimensional space. For multiple dimensions, Sander et al. (1998) and Schubert et al. (2017) calculated MinPts as $MinPts = 2 * dim$ (with dim defined as the number of dimensions in the data set). For extensive data sets, or data with many outliers or duplicates, a larger value for MinPts may need to be selected (Schubert et al., 2017).

2.1.2 Step 2: Data Point Classification

In the second step, the algorithm classifies each data point with Epsilon and MinPts into one of three categories: core point, border point and noise point (Tan et al., 2005) (Figure 2).

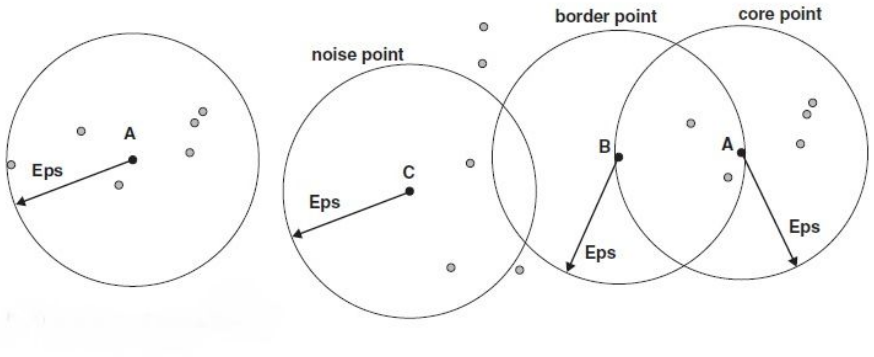

Figure 2: Core Points, Border Points, and Noise Points, with MinPts = 5 (Tan et al., 2005)

The data points are defined as follows (Tan et al., 2005):

- "Core points" have at least MinPts within their neighbourhood with radius Epsilon where the centre is the respective data point. Core points are the data points in high-density areas.
- "Border points" are within the radius of a cluster's core point, but they do not have MinPts in the neighbourhood. Located at the border between the low-density and the high-density area, they are part of the high density area (=cluster).
- Data points that are not border or core points are "noise points". They are further away from the cluster than border points within the low-density area.

2.1.3 Step 3: Clustering Process

The clustering process follows a set of steps. Figure 3 shows the simplified DBSCAN algorithm (Tan et al., 2005).

1: Label all points as core, border, or noise points.
2: Eliminate noise points.
3: Put an edge between all core points that are within *Eps* of each other.
4: Make each group of connected core points into a separate cluster.
5: Assign each border point to one of the clusters of its associated core points.

Figure 3: Simplified DBSCAN Algorithm Steps (Tan et al., 2005)

Firstly, DBSCAN selects an arbitrary point and discards it if it is a noise point unless it is later on found to be part of a cluster. In this case, the label can change to border point. Once it finds a core point, it checks all points in the core point's neighbourhood and classifies them as either core or border point. Then the cluster grows by connecting the core points that are in each other's neighbourhood into a cluster until no more core or border points are found in the neighbourhood. DBSCAN then checks all other points in the same way, until all points are part of a cluster or identified as outliers. The algorithm only needs one single scan over the data set (Tan et al., 2005).

2.2 Model Evaluation

The algorithm' performance can be evaluated by using average distances between cluster points (Tan et al., 2005). The average distances between all data points within a cluster and the overall average of distances between the data points in all clusters are indicators for how well the algorithm performed and how well the clusters were formed (Kotu and Deshpande, 2014).

2.3 Limitations

As previously mentioned, there are many advantages to choose DBSCAN over other clustering algorithms, in particular, if there are arbitrary shapes of dense points in the data and outliers are common. Disadvantages and challenges of DBSCAN include (Tan et al., 2005):

- Only one value can be set for each Epsilon and MinPts, and the data needs to be understood to tune the parameters well.
- DBSCAN has problems with scaling highly dimensional data because it is harder to define density.
- For running DBSCAN on large data sets, speed can be an issue.
- The algorithm cannot differ between varying density.
- DBSCAN is not entirely deterministic.

3 DBSCAN Challenges with Bordering Clusters

3.1 Bordering Clusters of Different Densities

DBSCAN cannot distinguish between the densities of two or more clusters of different densities (Tan et al., 2005). With precisely one setting for each radius Epsilon and MinPts, DBSCAN can only differentiate between high-density areas (clusters) and low-density areas (noise) in the data set (Ester et al., 1996). To counterbalance this issue, Ester et al. (1996) recommended setting Epsilon according to the ”thinnest” (least dense) cluster in the data set which is not considered noise. DBSCAN can then find clusters of multiple densities, including the thinnest and the denser clusters in the data set, and separate them from noise points (Ester et al., 1996) if the clusters are separated. Ester et al. (1996) stated that the gap between the thinnest cluster and a similarly dense or denser cluster needs to be larger than Epsilon to identify them as individual clusters. However, in a data set with bordering clusters of different densities, the algorithm does not perform well (Ertöz et al., 2003; Kriegel et al., 2011). Figure 4 visualizes this problem.

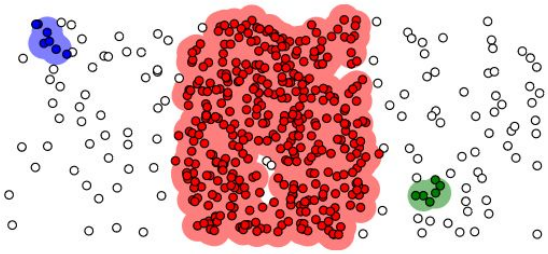

Figure 4: DBSCAN run with Epsilon = 0.54 and MinPts = 4. Core Points and Border Points are Coloured; Noise Points are Uncoloured (Harris, 2015)

Figure 4 shows three bordering clusters of varying densities. If Epsilon is set low in DBSCAN, the thinner clusters are identified as noise points (Tan et al., 2005). Contrarily, Figure 5 shows the same data set displayed in Figure 4, this time with a high Epsilon value.

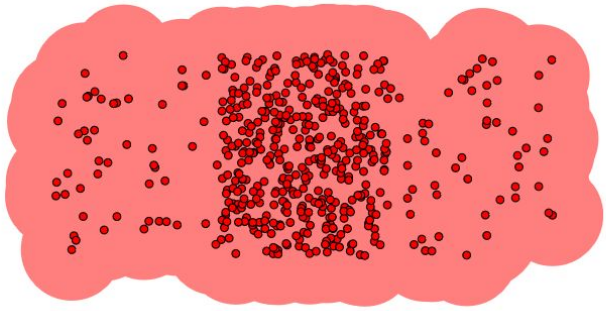

Figure 5: DBSCAN run with Epsilon = 1.98 and MinPts = 4. Core Points and Border Points are Coloured; Noise Points are Uncoloured (Harris, 2015)

The clustering result in Figure 5 shows that with a higher Epsilon, i.e., at the value that is needed to determine the thinnest cluster, all thinner to more dense clusters are considered one single high-density area, and they are merged into one big cluster (Ester et al., 1996; Tan et al., 2005). Similarly to varying densities, it can be a challenge for DBSCAN if the data set contains bordering clusters where all clusters are of similar density (Tan et al., 2005).

3.2 Bordering Clusters of Similar Densities

In addition, DBSCAN cannot distinguish between two or more clusters of similar densities if they are not clearly separated (Tan et al., 2005). The algorithm merges clusters A and B into one cluster if only one single core point of cluster A is in the neighbourhood of another core point of cluster B.

Another challenge is that DBSCAN is not entirely deterministic (Tan et al., 2005). A non-deterministic model bases at least one of its steps on a random value, and the same model input can give different model results (Tan et al., 2005). In DBSCAN, the starting point and the order in which DBSCAN processes the data is selected arbitrarily (Tan et al., 2005). If a border point is in the neighbourhood of two different clusters, the point belongs to both clusters (Ester et al., 1996). In DBSCAN, a border point is assigned to the core point (and its cluster) which is first processed by the algorithm, even if it is also found to be a border point to other clusters later on in the scan (Ester et al., 1996). Border points of multiple clusters can potentially lead to different results each time DBSCAN processes the data (Ester et al., 1996).

Furthermore, as a result of DBSCAN's non-deterministic nature, we cannot assume that all clusters have at least MinPts assigned to them, and in an extreme case scenario,

DBSCAN could potentially form clusters that only contain one single core point (Schubert et al., 2017). For example, a core point could have multiple border points in its cluster, which could also be part of other nearby clusters. The border points could first be assigned to other clusters, depending on the processing order (Tan et al., 2005). Then the remaining core point forms a cluster by itself with only one assigned point, even with the parameter setting of MinPts >1(Schubert et al., 2017). Ester et al. (1996) and Schubert et al. (2017) discussed that a border point could be part of multiple clusters, but they pointed out that this would not happen very often. However, recent research in the medical field indicated that this is not a rare case (Tran et al., 2013).

4 Alternatives to DBSCAN

Based on the findings of this literature review, for the analysis of bordering or nested clusters of varying or similar density, it is advisable to avoid models that use a concept of border points. Table 1 shows an overview of selected alternative density-based methods, including their performance criteria for different data sets.

Table 1: Comparison of Selected Density-Based Algorithms and Performance Criteria

Algorithm	**Dimensionality**	**Can handle clusters of different densities?**	**Can handle overlapping clusters?**
DBSCAN	Low	No	No
HDBSCAN	Low to medium	Yes	No
OPTICS	Low to medium	Yes	No
SNN density-based Clustering	High	Yes	No
Fuzzy Clustering	High	Yes	Yes

Instead of DBSCAN, Schubert et al. (2017) recommended hierarchical density-based clustering methods such as HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) (Campello et al., 2013, 2015), or OPTICS (Ordering Points To Identify the Clustering Structure)(Ankerst et al., 1999), which perform better in low or medium dimensional data set with varying densities. OPTICS creates a reachability plot based on distances, that helps to separate the clusters of different densities (Ankerst et al., 1999). HDBSCAN can detect clusters of varying densities, and it is more robust in parameter tuning with less user input than DBSCAN (Campello et al., 2013, 2015). It uses the parameter "minimum cluster size" instead of Epsilon, which defines how big a cluster needs to be to qualify as a cluster (Campello et al., 2013, 2015). Alternatively to DBSCAN, Tan et al. (2005) suggested applying an SNN (Shared Nearest Neighbour) density-based clustering technique (Ertöz et al., 2003). SNN clustering is a more flexible approach, where an SNN similarity measure is applied to data sets with clusters of varying densities before DBSCAN is run (Ertöz et al., 2003). In addition, it can handle high- dimensional data (Tan et al., 2005). However, similarly to DBSCAN, OPTICS, and HDBSCAN, SNN clustering cannot distinguish

between overlapping or bordering clusters. Advances in recent research were made with fuzzy clustering methods to define clusters that are poorly separated (Ienco and Bordogna, 2018; Bordogna and Ienco, 2014; Du et al., 2018; Chen et al., 2018; Javadian et al., 2017; Tan et al., 2005). To sum up, future research should focus on adaptations to hierarchical density-based algorithms and fuzzy clustering methods to address data with clusters of varying densities and not clearly separated clusters.

5 Conclusion

In this study, we summarized the density-based algorithm DBSCAN. DBSCAN has difficulties with identifying clusters of varying densities and with separating bordering clusters. We reviewed the implications on the model results. Alternative density-based algorithms are OPTICS, HDBSCAN, SNN clustering, and fuzzy clustering. Future research could investigate the performance of adaptations of existing algorithms.

References

Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. (1999). Optics: Ordering points to identify the clustering structure. *SIGMOD Rec.*, 28(2):49–60.

Bordogna, G. and Ienco, D. (2014). Fuzzy core dbscan clustering algorithm. In *International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems*, pages 100–109. Springer.

Campello, R. J., Moulavi, D., and Sander, J. (2013). Density-based clustering based on hierarchical density estimates. In *Pacific-Asia conference on knowledge discovery and data mining*, pages 160–172. Springer.

Campello, R. J., Moulavi, D., Zimek, A., and Sander, J. (2015). Hierarchical density estimates for data clustering, visualization, and outlier detection. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 10(1):5.

Chen, Y., Tang, S., Bouguila, N., Wang, C., Du, J., and Li, H. (2018). A fast clustering algorithm based on pruning unnecessary distance computations in dbscan for high-dimensional data. *Pattern Recognition*.

Du, M., Ding, S., and Xue, Y. (2018). A robust density peaks clustering algorithm using fuzzy neighborhood. *International Journal of Machine Learning and Cybernetics*, 9(7):1131–1140.

Ertöz, L., Steinbach, M., and Kumar, V. (2003). Finding clusters of different sizes, shapes, and densities in noisy, high dimensional data. In *Proceedings of the 2003 SIAM international conference on data mining*, pages 47–58. SIAM.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. pages 226–231. AAAI Press.

Han, J., Pei, J., and Kamber, M. (2011). *Data mining: concepts and techniques*. Elsevier.

Harris, N. (2015). *Naftali Harris' Blog*. Online at https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/ Accessed: 2018-12-06.

Ienco, D. and Bordogna, G. (2018). Fuzzy extensions of the dbscan clustering algorithm. *Soft Computing*, 22(5):1719–1730.

Javadian, M., Shouraki, S. B., and Kourabbaslou, S. S. (2017). A novel density-based fuzzy clustering algorithm for low dimensional feature space. *Fuzzy Sets and Systems*, 318:34–55.

Kotu, V. and Deshpande, B. (2014). *Predictive Analytics and Data Mining: Concepts and Practice with RapidMiner*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1st edition.

Kriegel, H.-P., Kröger, P., Sander, J., and Zimek, A. (2011). Density-based clustering. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):231–240.

Sander, J., Ester, M., Kriegel, H.-P., and Xu, X. (1998). Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data mining and knowledge discovery*, 2(2):169–194.

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., and Xu, X. (2017). Dbscan revisited, revisited: Why and how you should (still) use dbscan. *ACM Trans. Database Syst.*, 42(3):19:1–19:21.

Tan, P.-N., Steinbach, M., and Kumar, V. (2005). Introduction to data mining. 1st.

Tran, T. N., Drab, K., and Daszykowski, M. (2013). Revised dbscan algorithm to cluster data with dense adjacent clusters. *Chemometrics and Intelligent Laboratory Systems*, 120:92 – 96.