Assignment 3: Clustering Algorithm Self-Study --- Joanna Orzechowska

1. Algorithm Overview (20%)

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a clustering algorithm. As the name suggests, it's a density based algorithm which means it defines a cluster as a data that is packed closely together, regardless of their shape, surrounded by the regions of lower density. Outliers in this algorithm are identified as a noise, also based on density.

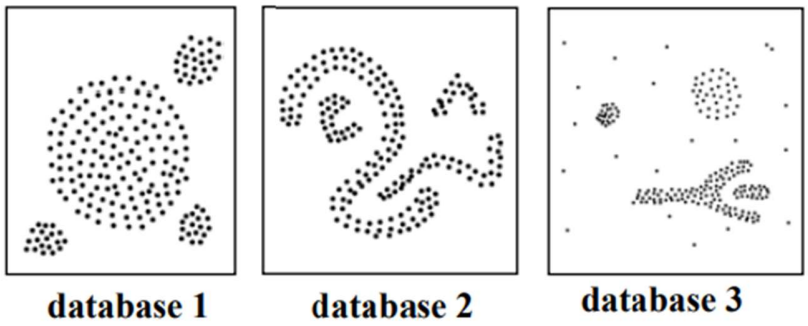

Figure 1. Sample databases (Ester et al., 1996)

Traditional clustering algorithms identify the center if the cluster and then forms the cluster spherically around. That only works for spherical data (as in figure 1, database 1). DBSCAN can cluster data of different shapes (figure 1, database 2, 3) making it well adapted to real-world data.

To understand how DBSCAN identifies clusters, we should define a key parameters:

- eps – the radius of the neighbourhood around the point, a maximum point in between two samples to be considered of the same cluster. It's a float, with a default value of 0.5 however it is important to choose it based on the database.
- min_samples – the minimum number of samples to be in the same neighbourhood to be considered a core sample. It's an integer, with a default value of 5.

The algorithm picks a point and if it is a core sample in a high density region, it expands the cluster from that point, using the eps as a radius looking for other samples. The points in a cluster are density-connected. The points not connected to cluster by density (within the eps parameter) become outliers. The algorithm is visualised in figure 2.

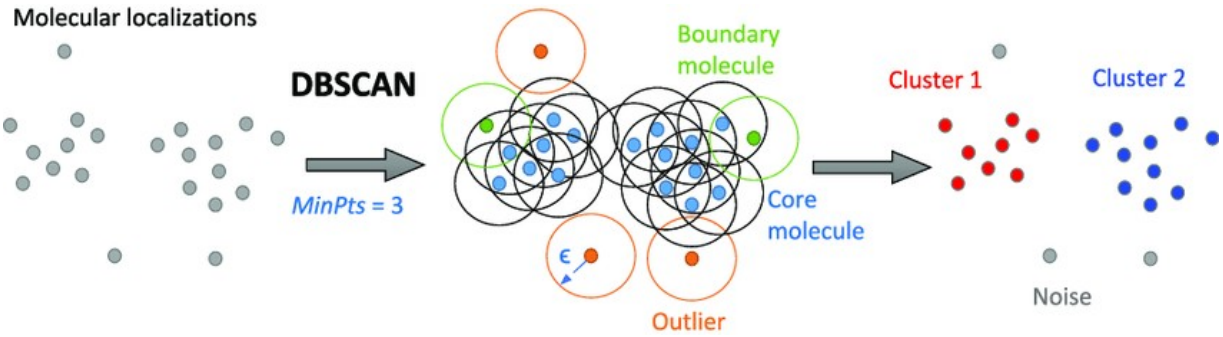

Figure 2. Visualization of DBSCAN algorithm work (Khater et al., 2020)

DBSCAN offers several advantages including: effectiveness for data of irregular shape and with similar density, identification of noise as outliers and no predefined number of clusters. However it has limitations like struggling with varying density, sensitivity to parameter choice and can have high memory usage for large datasets.

2. Algorithm Comparison (40%)

As shown on figure 3, DVSCAN outperforms k-Means and Hierarchical Clustering in the irregularly shaped datasets (visualized by make_circles and make_moons datasets, rows 1 and 3 on figure 3). However it had problems with a dataset with varying densities (make_blobs, row 2 on figure 3).

The struggles with varying densities have their origin in the eps parameter which is defined by the user to the whole dataset and cannot be adjusted to specific cluster. The results of eps value change are shown in figure 4 where the result ranges from most data being identified as outliers with multiple small clusters (for eps = 0.3) to 3 clusters with the least densely populated also having a lot of outliers that in other methods were all included in the cluster.

Another struggle, connected to the one mentioned before, is manual assignment of the eps value and how minimal changes can render clustering ineffective. For example while changing the value of eps from 0.2 to 0.3, the make_circles dataset was shown as a single cluster. The same effect was given by changing the eps parameter from 0.3 to 0.5 in the make_moons dataset

For those reasons I would choose the DBSCAN algorithm to irregularly shaped data. However, I would like to learn more about the data and through the trial and error adjust the parameters for the best result. For spherical data with varied density, I would choose the hierarchical clustering.

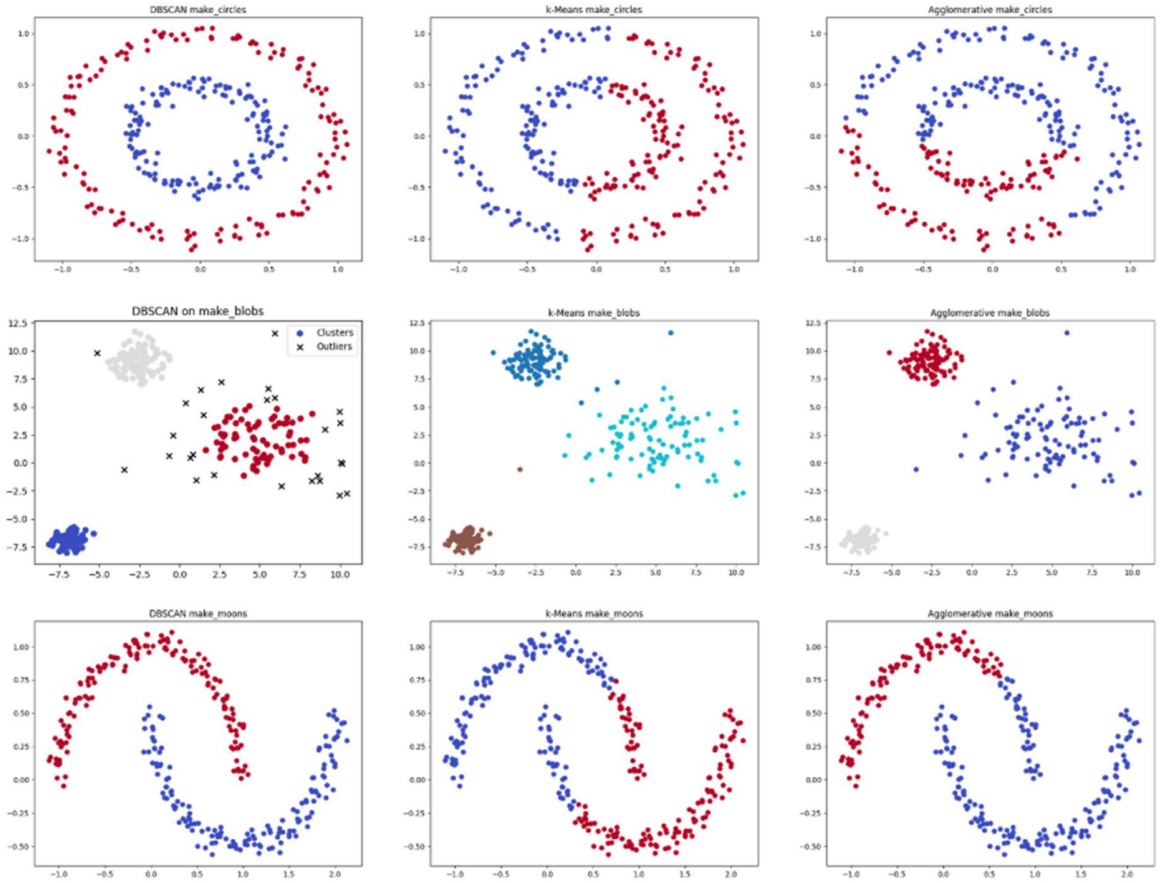

Figure 3. Comparison of clustering algorithms: DBSCAN – column 1, k-Means – column 2, Agglomerative clustering – column 3, as user on sklearn databases: make_circles – row 1, make_blobs – row 2, make_moons – row 3.

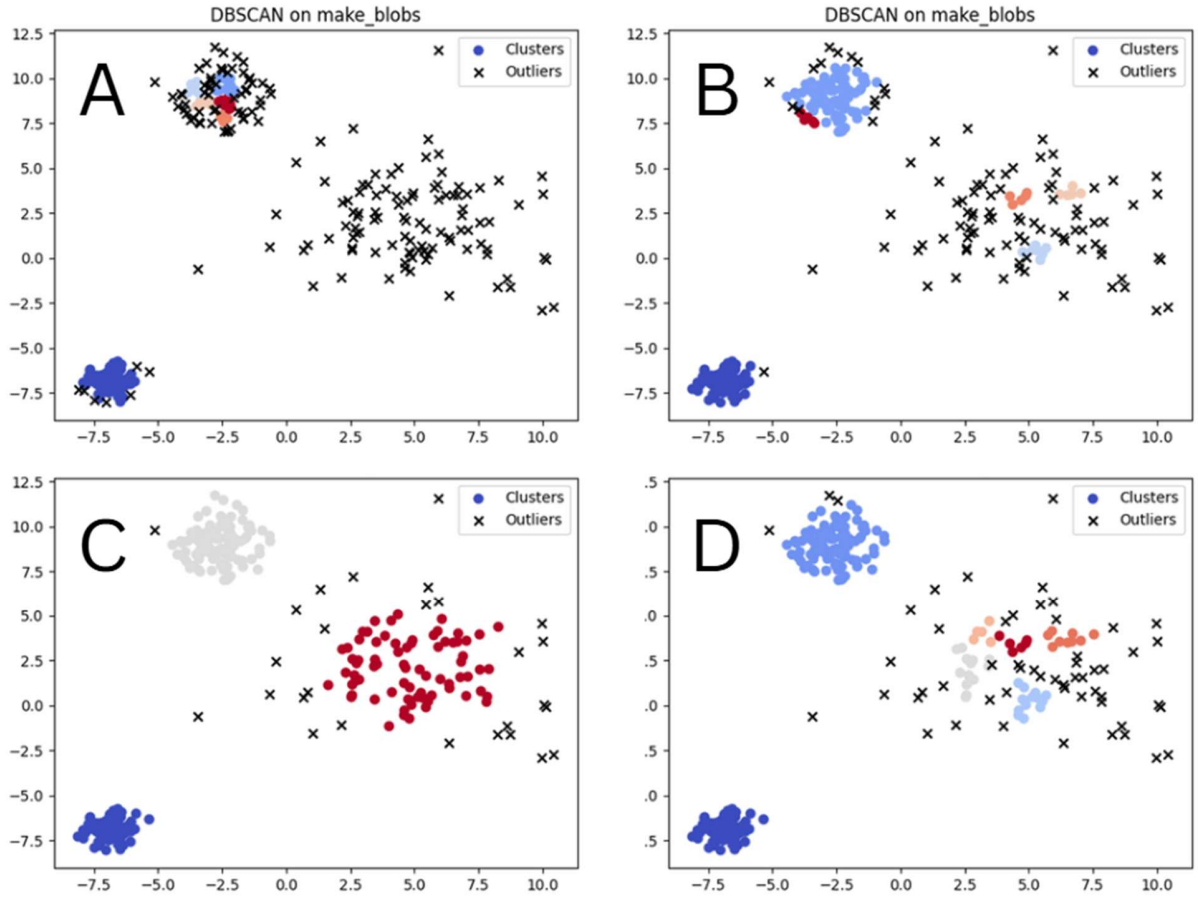

Figure 4. The effect of eps parameter on DBSCAN clustering of data with various density. A. eps = 0.3, B. eps = 0.5, C. eps = 1, D. eps = 0.7

3. Table Update (20%)

Compare and contrast characteristics for all three algorithms:

Feature	k-Means	Hierarchical Clustering	DBSCAN
Definition	Partitioning algorithm that assigns points to k clusters based on centroids	Builds a hierarchy of clusters using distance metrics	Density based algorithm defining clusters and noise based on the data density
Approach	Iteratively minimizes variance within k clusters	Agglomerative (bottom-up) or divisive (top-down)	Groups density connected points using eps and min_samples parameters
Number of Clusters	Requires predefined k	Can be determined from dendrogram but subjective	Not determined, depending on the data and parameters
Cluster Shape	Prefers spherical clusters	Works well with various shapes but can be unstable	Irregular clusters welcome
Initialization	Randomly selects k initial centroids	No initialization needed	Starts with an arbitrary point p and retrieves all points density-reachable from that point.
Result	Hard assignments—each point belongs to a single cluster	Hierarchical structure (tree/dendrogram)	Hard assessment with marked noise as outliers
Interpretability	Moderate—cluster assignments but no hierarchy	High—dendrogram can be analyzed	High? Cluster structure visualised
Strengths	Simple, fast and efficient on large datasets	Can capture hierarchical relationships	Irregularly shaped clusters, identifies noise/outliers, no predefined number of clusters

Limitations	Sensitive to initial centroids and k choice	Computationally expensive for large datasets	Memory expensive, inefficient for larger datasets, struggles with varying density, sensitive to parameters

4. Code Documentation & Submission Quality (20%)

https://github.com/orzesia/BINF5507/tree/d3c37f1f78f8d0ff388c8a943c9546184f783b29/Assignments/Assignment%203

References:

Ester, M., H. P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise". In: Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining, Portland, OR, AAAI Press, pp. 226-231. 1996

Khater, I. M., Nabi, I. R., & Hamarneh, G. (2020). A review of Super-Resolution Single-Molecule localization microscopy cluster analysis and quantification methods. *Patterns*, *1*(3), 100038. https://doi.org/10.1016/j.patter.2020.100038

Websites (all accessed Jun 20 – Jun 22, 2025):

https://www.geeksforgeeks.org/machine-learning/dbscan-clustering-in-ml-density-based-clustering/

https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html

https://scikit-learn.org/stable/auto_examples/cluster/plot_cluster_comparison.html