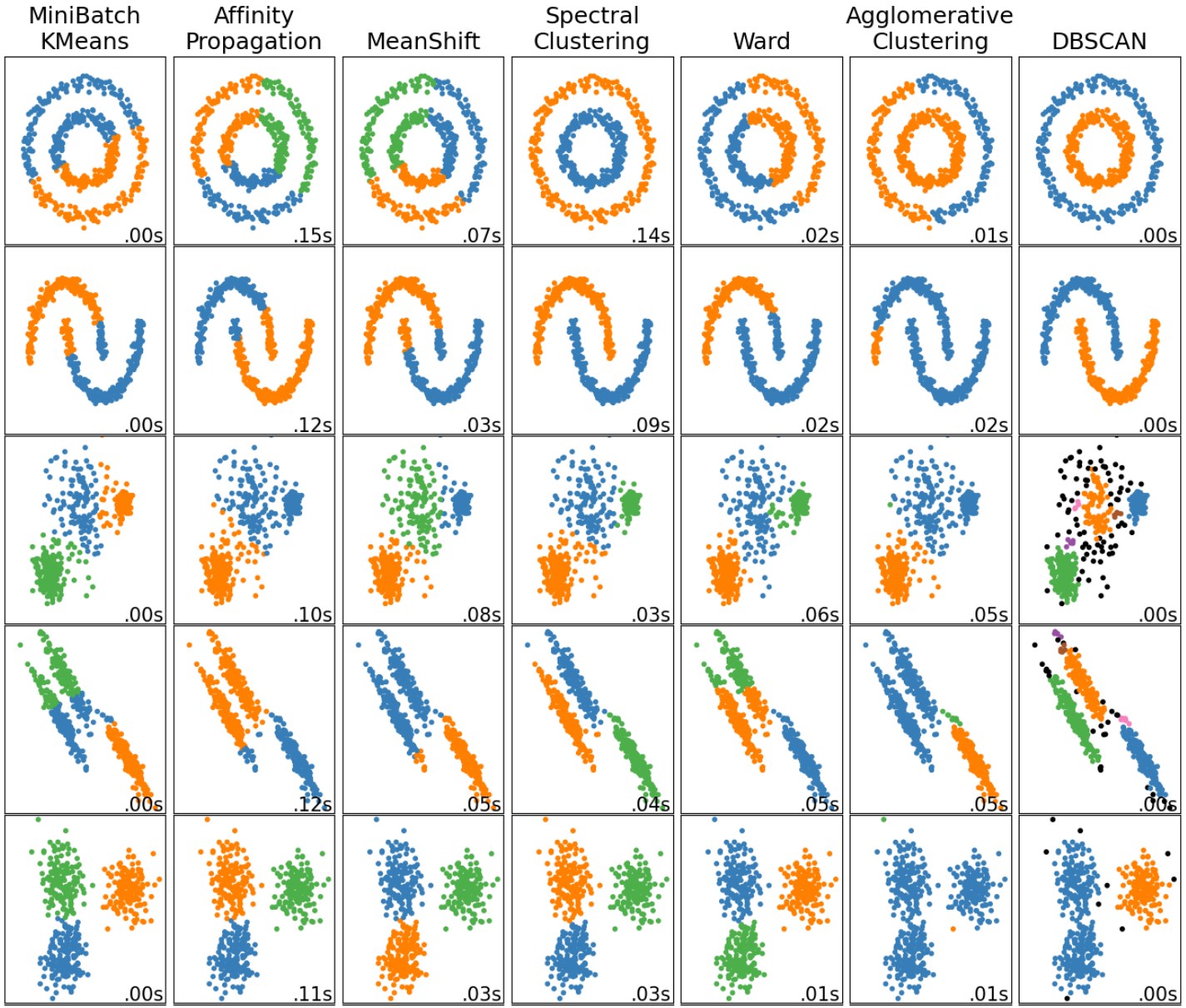
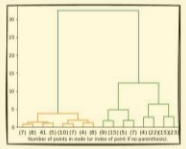
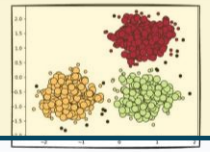
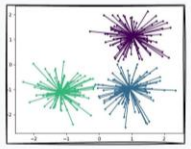
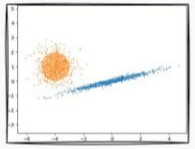
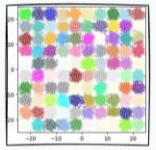
UNSUPERVISED LEARNING PART 3: DENSITY-BASED CLUSTERING



OUTLINE

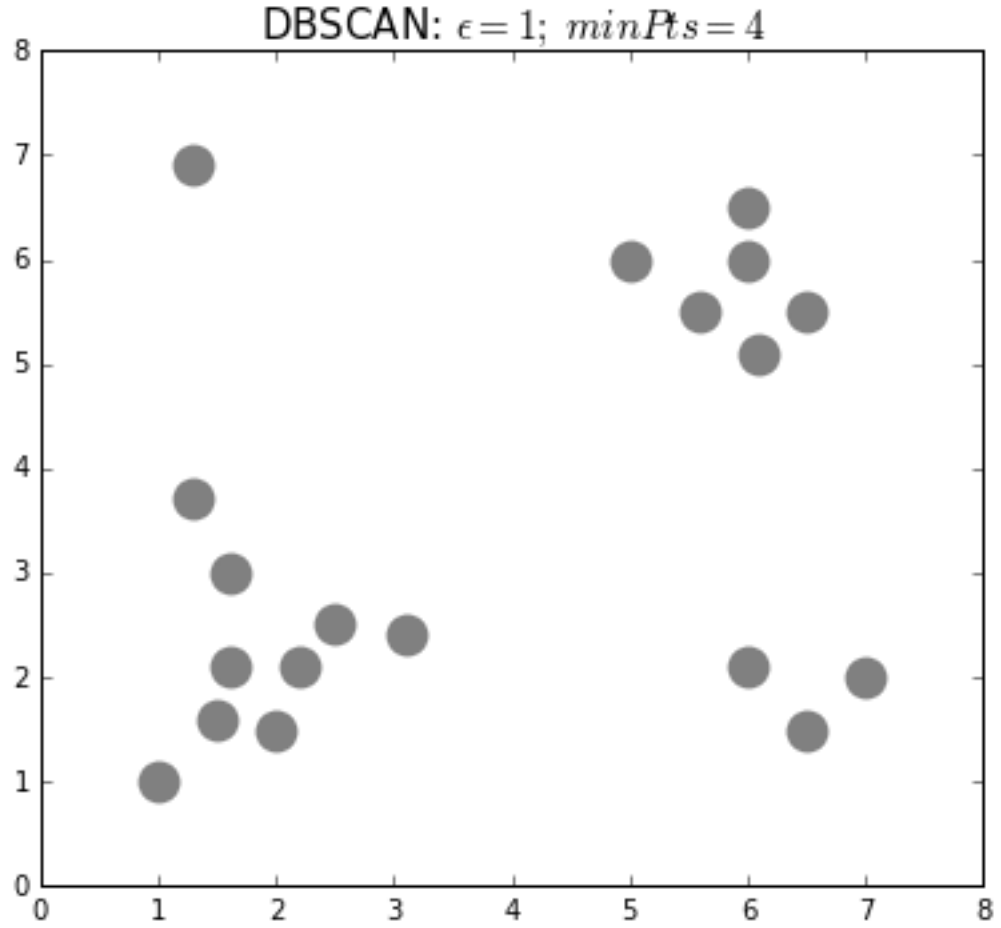
- Hierarchical Clustering
  - Agglomerative
  - Divisive
- Density Based Clustering
  - DBSCAN

Clustering Algorithm Type		Clustering Methodology	Algorithm(s)
	Centroid-based	Cluster points based on proximity to centroid	KMeans KMeans++ KMedoids
	Connectivity-based	Cluster points based on proximity between clusters	Hierarchical Clustering (Agglomerative and Divisive)
	Density-based	Cluster points based on their density instead of proximity	DBSCAN OPTICS HDBSCAN
	Graph-based	Cluster points based on graph distance	Affinitiy Propagation Spectral Clustering
	Distribution-based	Cluster points based on their likelihood of belonging to the same distribution.	Gaussian Mixture Models (GMMs)
	Compression-based	Transform data to a lower dimensional space and then perform clustering	BIRCH

DBSCAN (DENSITY-BASED SPATIAL CLUSTERING OF APPS W NOISE)

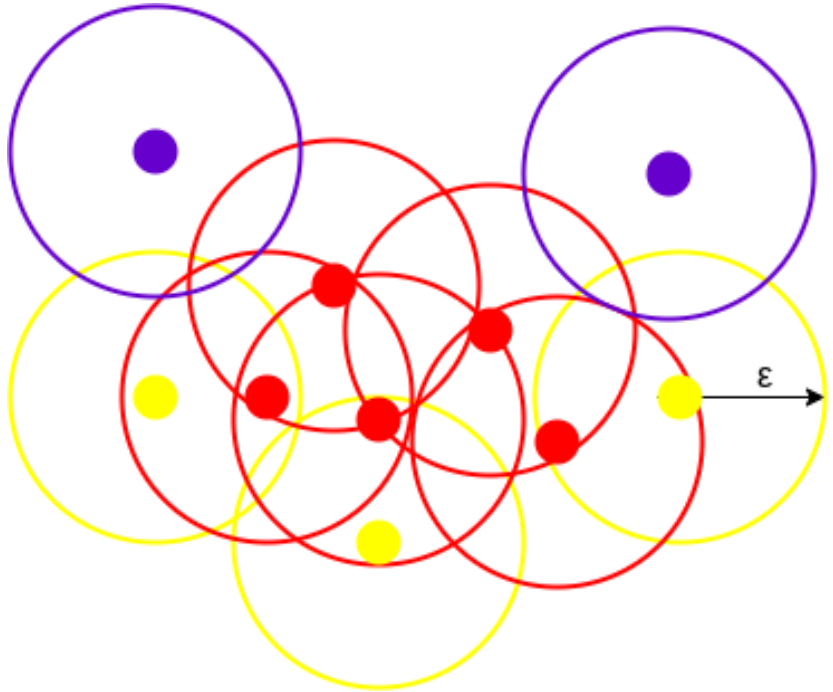
DBSCAN ALGORITHM

1. Pick hyperparamters: ε and min. points.
2. Pick any unvisited point. Find its neighbors within a specified distance (ε).
3. If there are enough neighbors (at least *minPts*), start a new cluster. Otherwise, mark it as noise.
4. Add all neighbors to the cluster. Expand the cluster by checking each new point's neighbors and adding them if they qualify.
5. Repeat until no more points can be added to the cluster.
6. Move to the next unvisited point and repeat the process until all points are either clustered or marked as noise



DENSITY BASED CLUSTERING - DBSCAN

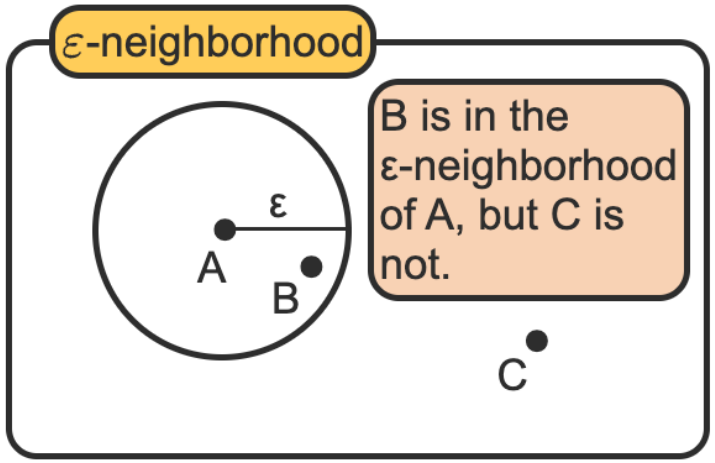
- **Density**: The number of points within a specified radius (Eps).
- **Core Point**: Has at least MinPts within Eps (interior of a cluster).
- **Border Point**: Has fewer than *MinPts* within Eps but is near a core point.
- **Noise Point**: Neither a core point nor a border point.

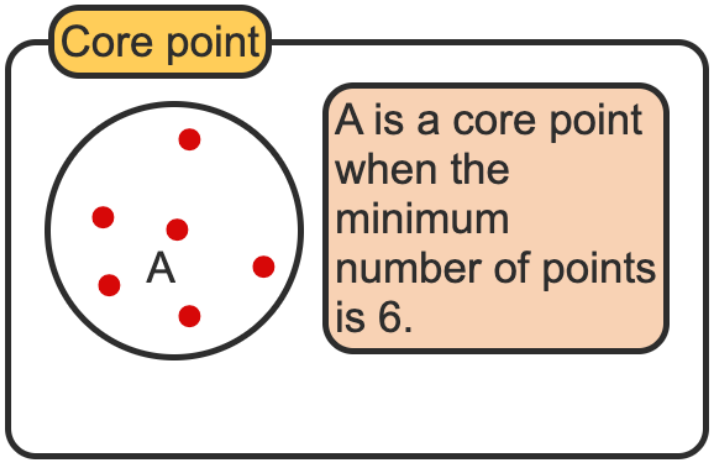


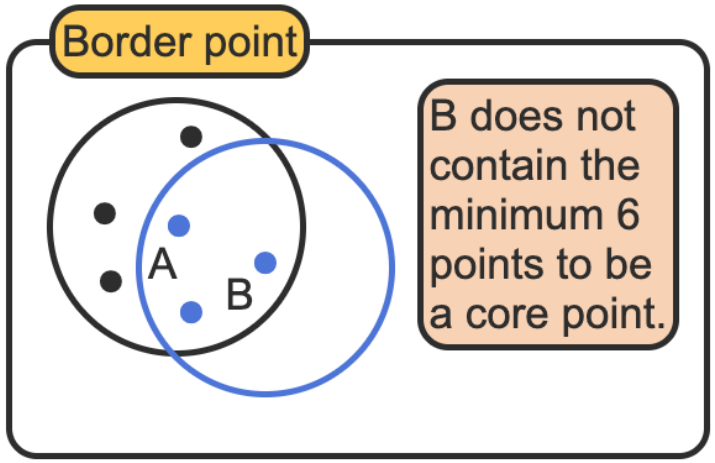
The above figure shows us a cluster created by DBCAN with *minPoints = 3*. Here, we draw a circle of equal radius *epsilon* around every data point.

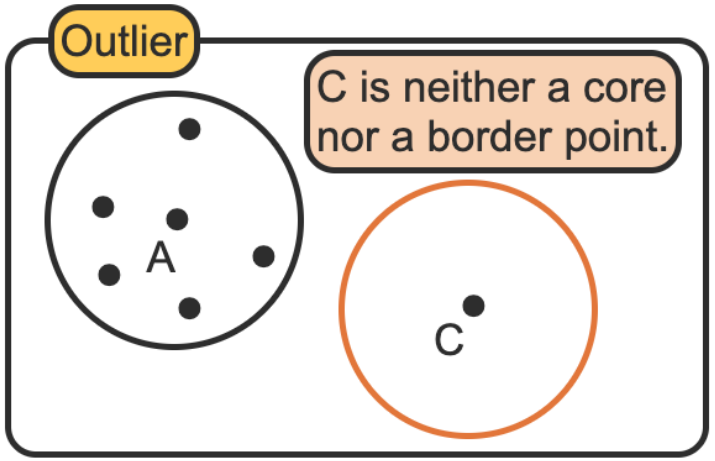
BYU

DBSCAN DEFINITIONS





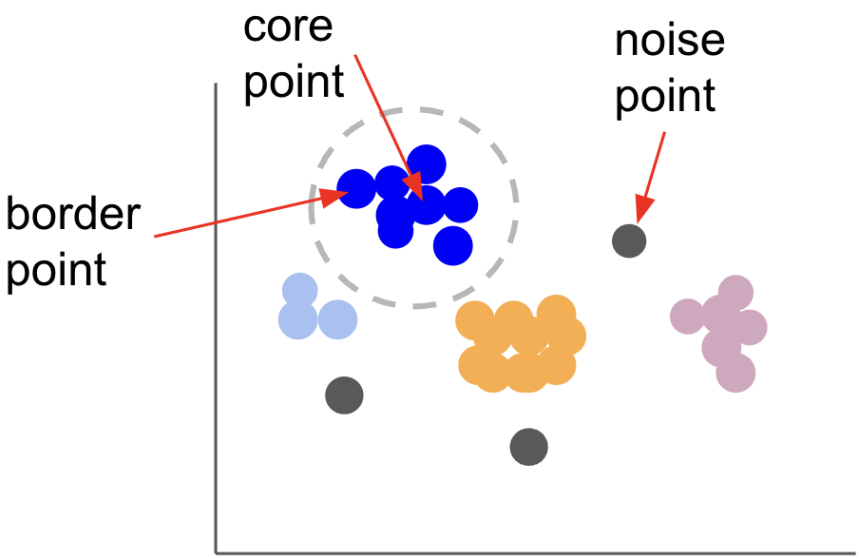




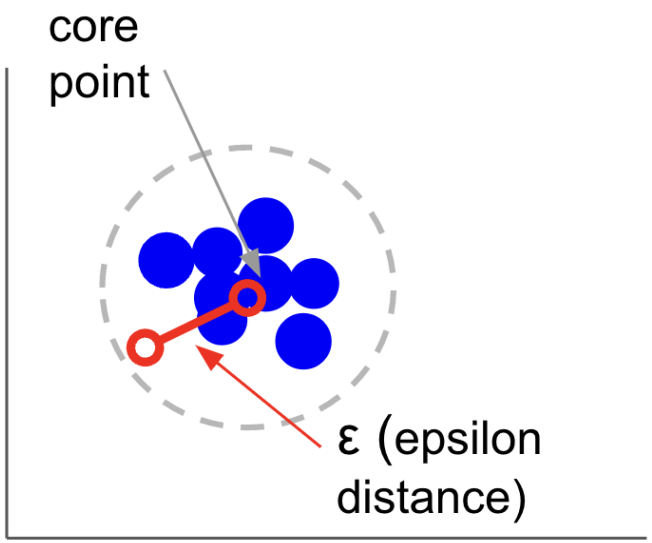
BYU

DBSCAN: CORE, BORDER, AND NOISE POINTS

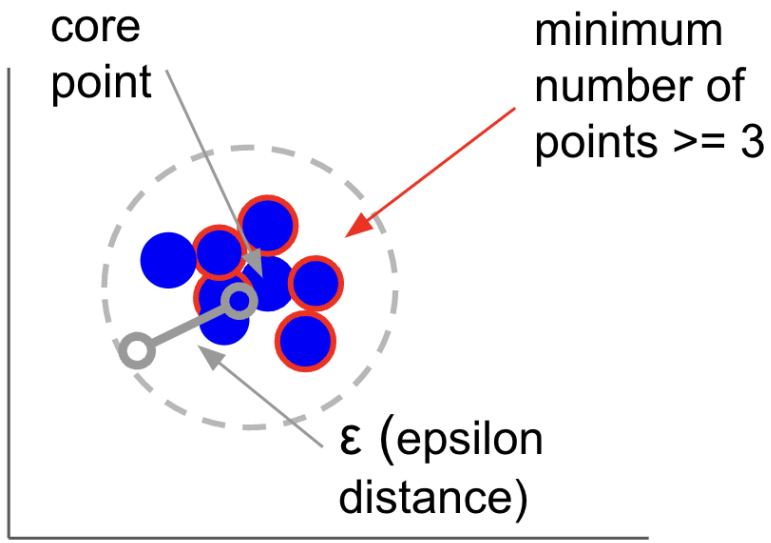
DBSCAN point types



DBSCAN ε (epsilon)



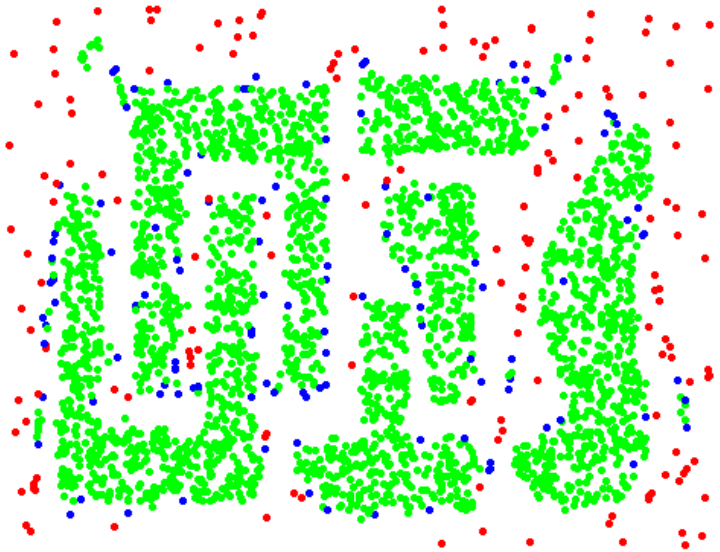
DBSCAN min. points



BYU

DBSCAN: CORE, BORDER AND NOISE POINTS

Original Points

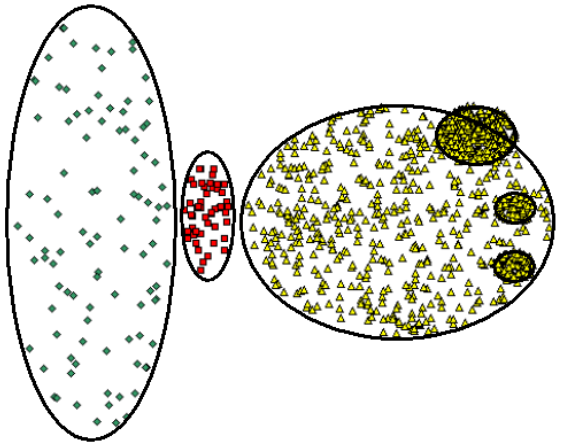


Point types: core, border and noise

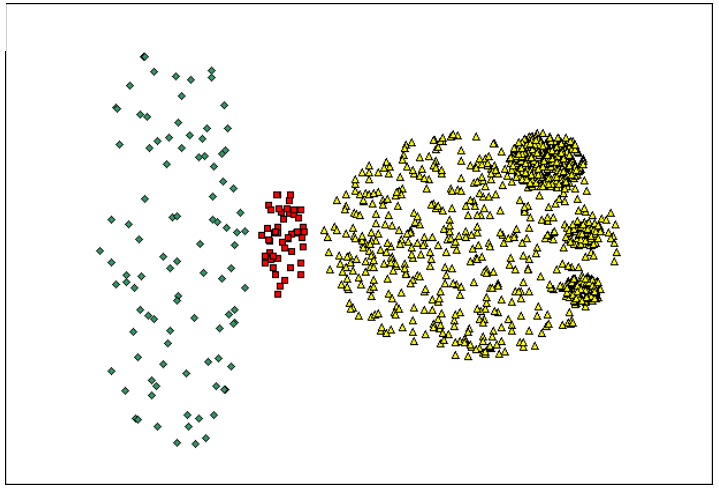
Eps = 10, MinPts = 4

BYU

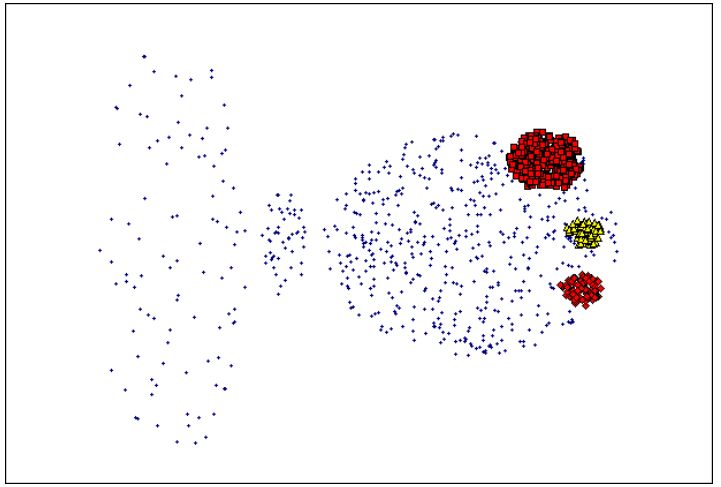
WHEN DBSCAN DOES NOT WORK WELL



Original Points



(MinPts=4, Eps=9.75).

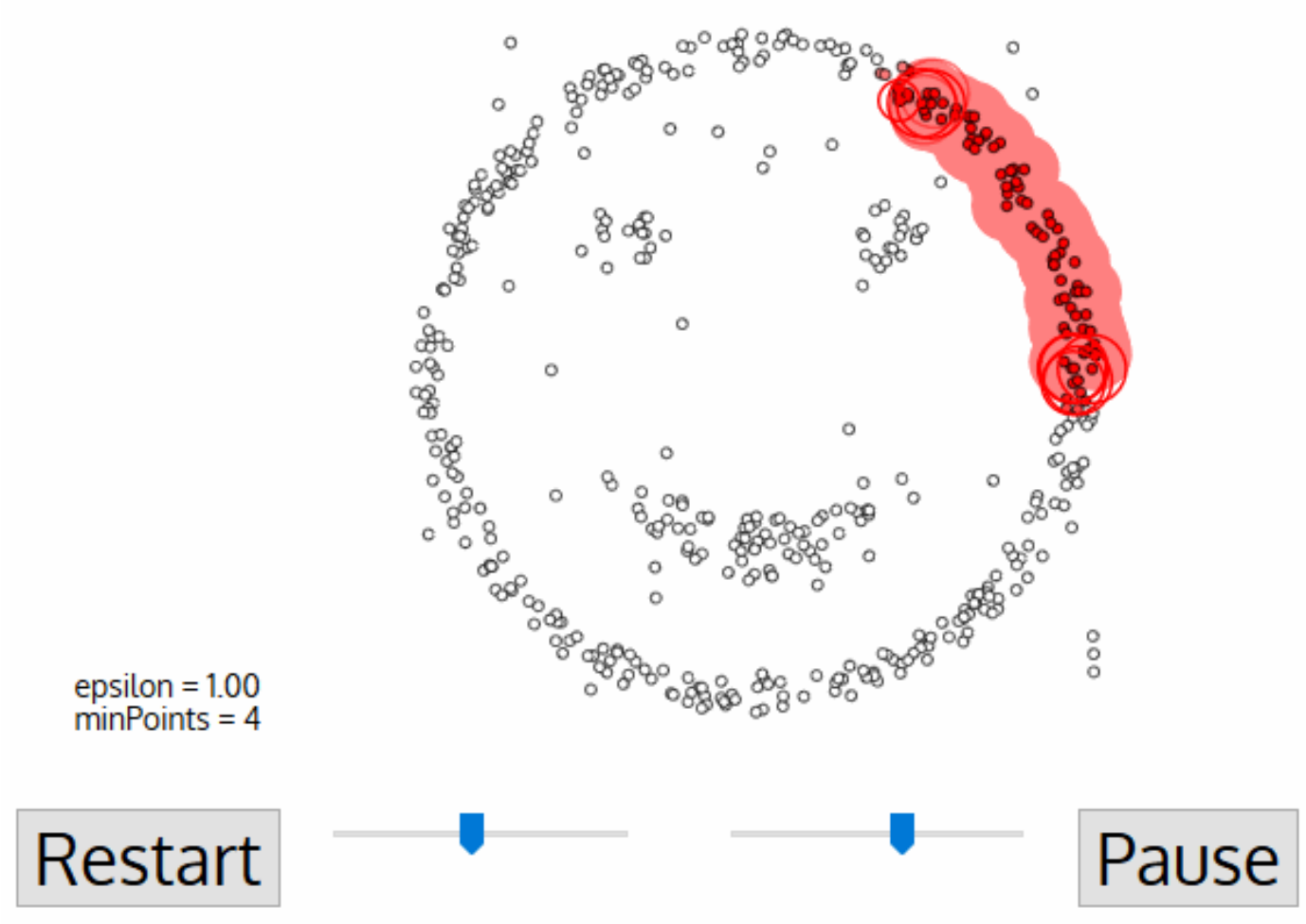


(MinPts=4, Eps=9.92)

- Varying densities
- High-dimensional data

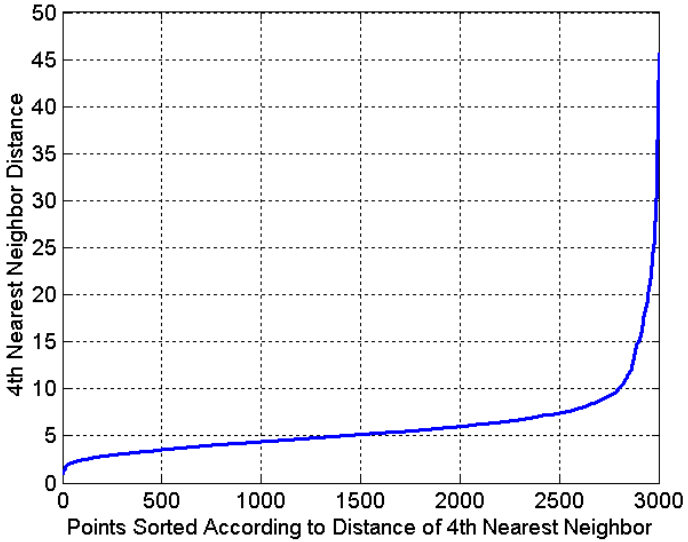
BYU

DBSCAN VISUALIZED (HTTPS://WWW.NAFTALIHARRIS.COM/BLOG/VISUALIZING-DBSCAN-CLUSTERING/)



DBSCAN: DETERMINING EPS AND MINPTS

- **Choosing Epsilon (ε):** Use the K-distance graph; the "elbow" point indicates the optimal value.
- **Nearest Neighbor Concept:** In a cluster, the k-th nearest neighbors are at similar distances.
- **Noise Points:** Their k-th nearest neighbor is farther away.
- **Finding ε:** Plot the sorted distances to the k-th nearest neighbor.
- **Choosing minPoints:** Set it at least one greater than the dataset's dimensions: ***minPoints*** ≥ ***Dimensions*** + ***1***



BYU

DBSCAN ADVANTAGES AND DISADVANTAGES

Advantages:

- If we look at its advantages, it is very good at picking up dense areas in data and points that are far from others. This means that the data doesn't have to have a specific shape and can be surrounded by other points if they are also densely connected.
- It has two hyperparameters, minimum points and ε, but there is no need to specify the number of clusters, as in K-Means.
- It can be used with large databases since it was designed for high-dimensional data.

Disadvantages:

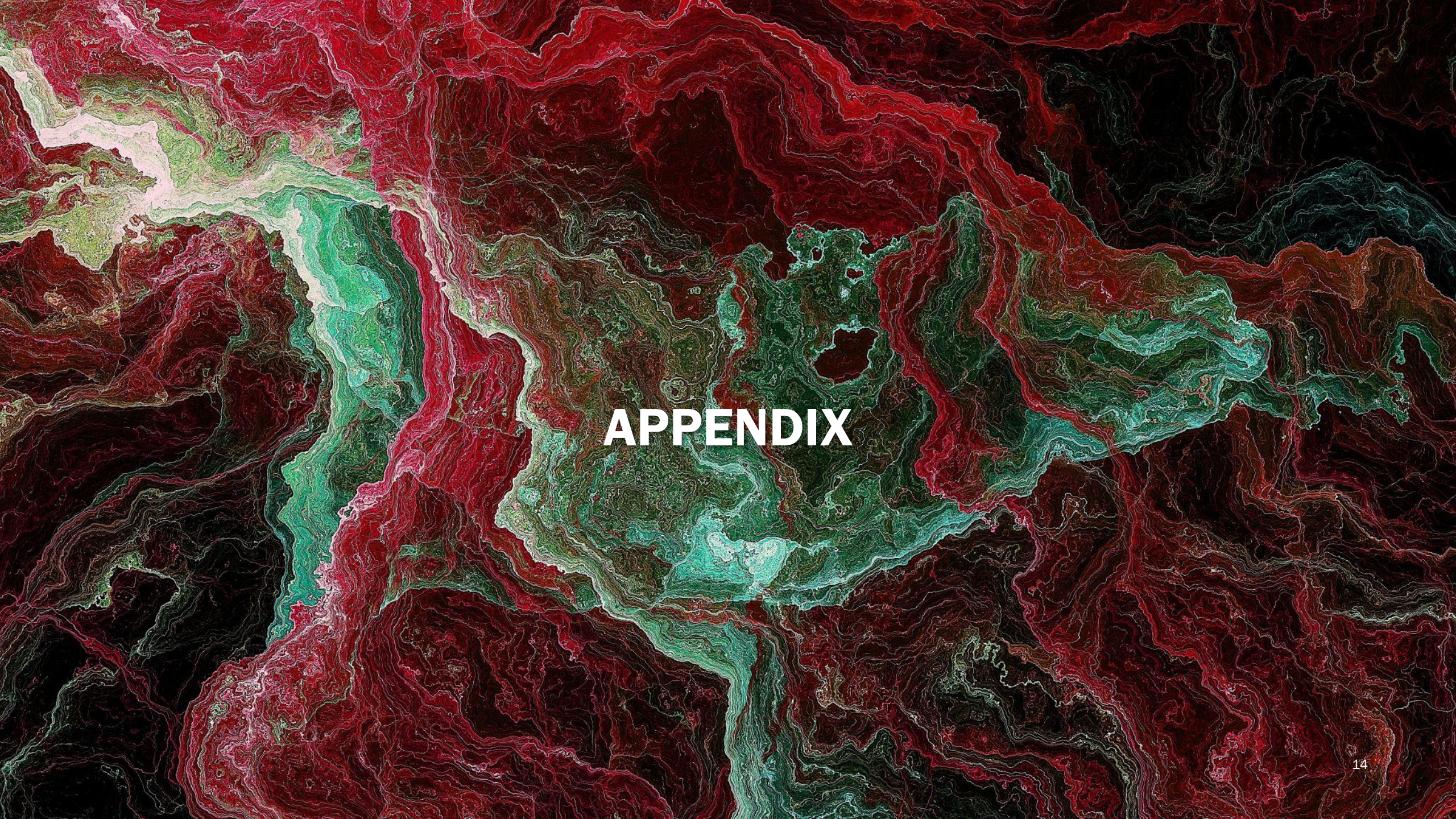
- Can’t address different densities in the same cluster
- It is also dependent on the distance metric and scaling of the points.
- O(nlog(n)) time complexity

DBSCAN Extensions

- Other algorithms, such as Hierarchical DBSCAN (HDBSCAN) and Ordering points to identify the clustering structure (OPTICS), are considered extensions of DBSCAN.
- HDBSCAN and OPTICS can usually perform better when there are clusters of varying densities in the data and are less sensitive to the choice or initial min points and ε parameters.

BYU


DBSCAN PYTHON EXAMPLE


APPENDIX

SUMMARY OF SELECTED CLUSTERING ALGORITHMS IN SCIKIT-LEARN

Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large n_samples, medium n_clusters with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters, inductive	Distances between points
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry, inductive	Graph distance (e.g., nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry, inductive	Distances between points
Spectral clustering	number of clusters	Medium n_samples, small n_clusters	Few clusters, even cluster size, non-flat geometry, transductive	Graph distance (e.g., nearest-neighbor graph)
Ward hierarchical clustering	number of clusters or distance threshold	Large n_samples and n_clusters	Many clusters, possibly connectivity constraints, transductive	Distances between points
Agglomerative clustering	number of clusters or distance threshold, linkage type, distance	Large n_samples and n_clusters	Many clusters, possibly connectivity constraints, non-Euclidean distances, transductive	Any pairwise distance
DBSCAN	neighborhood size	Very large n_samples, medium n_clusters	Non-flat geometry, uneven cluster sizes, outlier removal, transductive	Distances between nearest points
HDBSCAN	minimum cluster membership, minimum point neighbors	large n_samples, medium n_clusters	Non-flat geometry, uneven cluster sizes, outlier removal, transductive, hierarchical, variable cluster density	Distances between nearest points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation, inductive	Mahalanobis distances to centers
BIRCH	branching factor, threshold, optional global clusterer.	Large n_clusters and n_samples	Large dataset, outlier removal, data reduction, inductive	Euclidean distance between points

DISTANCE MEASURES

- Minkowski Distance (http://en.wikipedia.org/wiki/Minkowski_distance)

  For

  $$\mathbf{x} = (x_1\ x_2 \cdots x_n) \text{ and } \mathbf{y} = (y_1\ y_2 \cdots y_n)$$

  $$d(\mathbf{x},\mathbf{y}) = \left(|\ x_1 - y_1\ |^p + |\ x_2 - y_2\ |^p \cdots + |\ x_n - y_n\ |^p\right)^{\frac{1}{p}}, \quad p > 0$$

  - $p = 1$: Manhattan (city block) distance

    $$d(\mathbf{x},\mathbf{y}) = |\ x_1 - y_1\ | + |\ x_2 - y_2\ | \cdots + |\ x_n - y_n\ |$$

  - $p = 2$: Euclidean distance

    $$d(\mathbf{x},\mathbf{y}) = \sqrt{|\ x_1 - y_1\ |^2 + |\ x_2 - y_2\ |^2 \cdots + |\ x_n - y_n\ |^2}$$

  - Do not confuse p with n, i.e., all these distances are defined based on all numbers of features (dimensions).
  - A generic measure: use appropriate p in different applications

REFERENCES

- https://towardsdatascience.com/the-5-clustering-algorithms-data-scientists-need-to-know-a36d136ef68
- https://www.geeksforgeeks.org/ml-k-medoids-clustering-with-example/
- https://www.analyticsvidhya.com/blog/2021/06/kmodes-clustering-algorithm-for-categorical-data/#h-what-is-kmodes
- https://towardsdatascience.com/the-k-prototype-as-clustering-algorithm-for-mixed-data-type-categorical-and-numerical-fe7c50538ebb
- http://scikit-learn.org/stable/modules/clustering.html
- https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.15.4028&rep=rep1&type=pdf
- https://github.com/nicodv/kmodes/blob/master/kmodes/kprototypes.py
- https://dashee87.github.io/data%20science/general/Clustering-with-Scikit-with-GIFs/
- https://www.geeksforgeeks.org/difference-between-agglomerative-clustering-and-divisive-clustering/
- https://stackabuse.com/dbscan-with-scikit-learn-in-python/

BYU