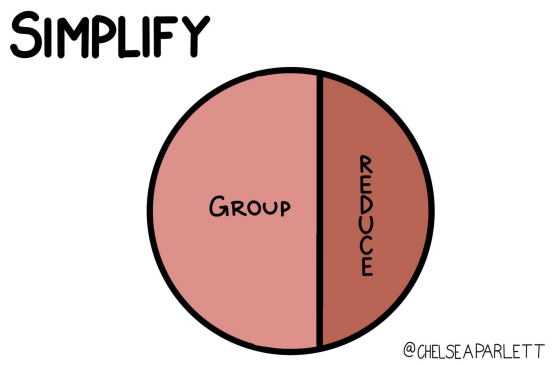


DBSCAN

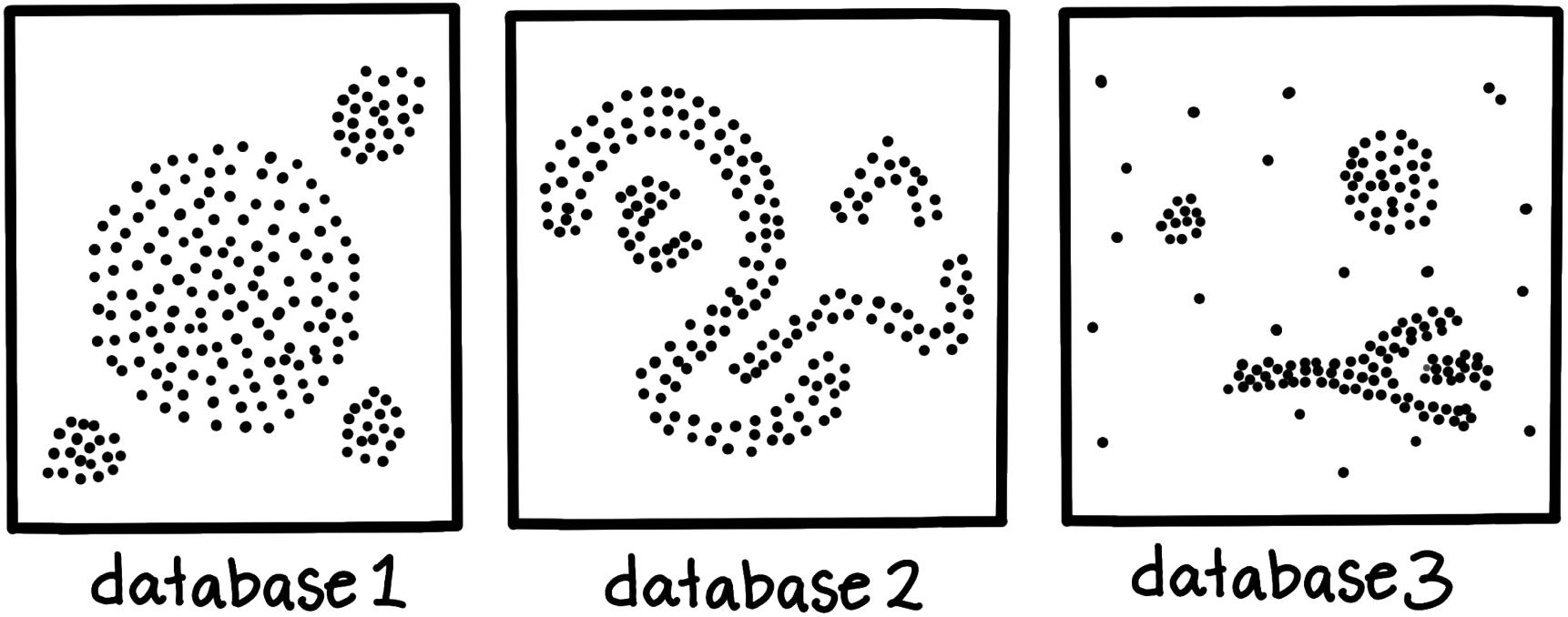
Chelsea Parlett-Pelleriti

DBSCAN

Density **B**ased **S**patial **C**lustering of **A**pplications with **N**oise

- Distance Metric
- Epsilon (*eps*)
- Minimum Points (*minpts*)

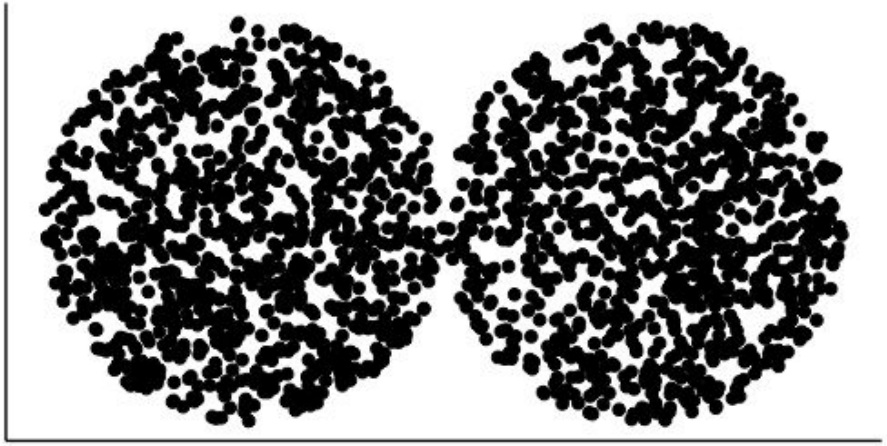
Benefits of DBSCAN



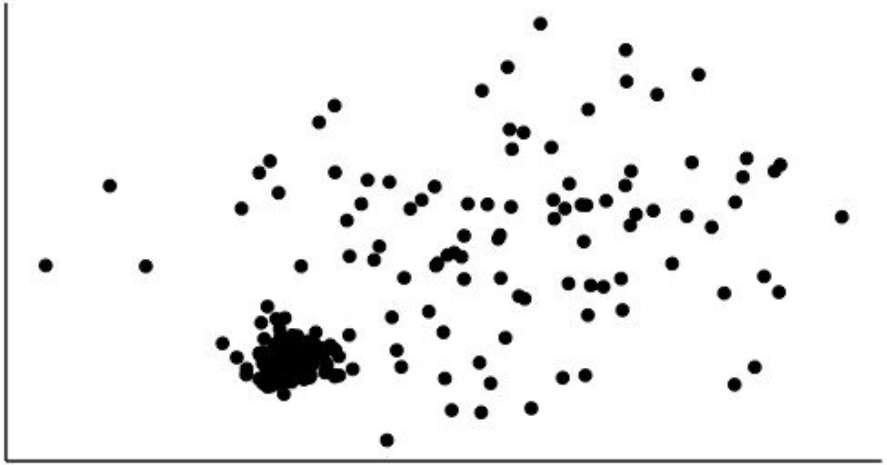
Disadvantages of DBSCAN

- Can be less effective in High Dimensional Data
- Not great with overlapping/touching clusters
- Suboptimal when clusters have different densities

Touching/Overlapping Clusters



Different Density Clusters

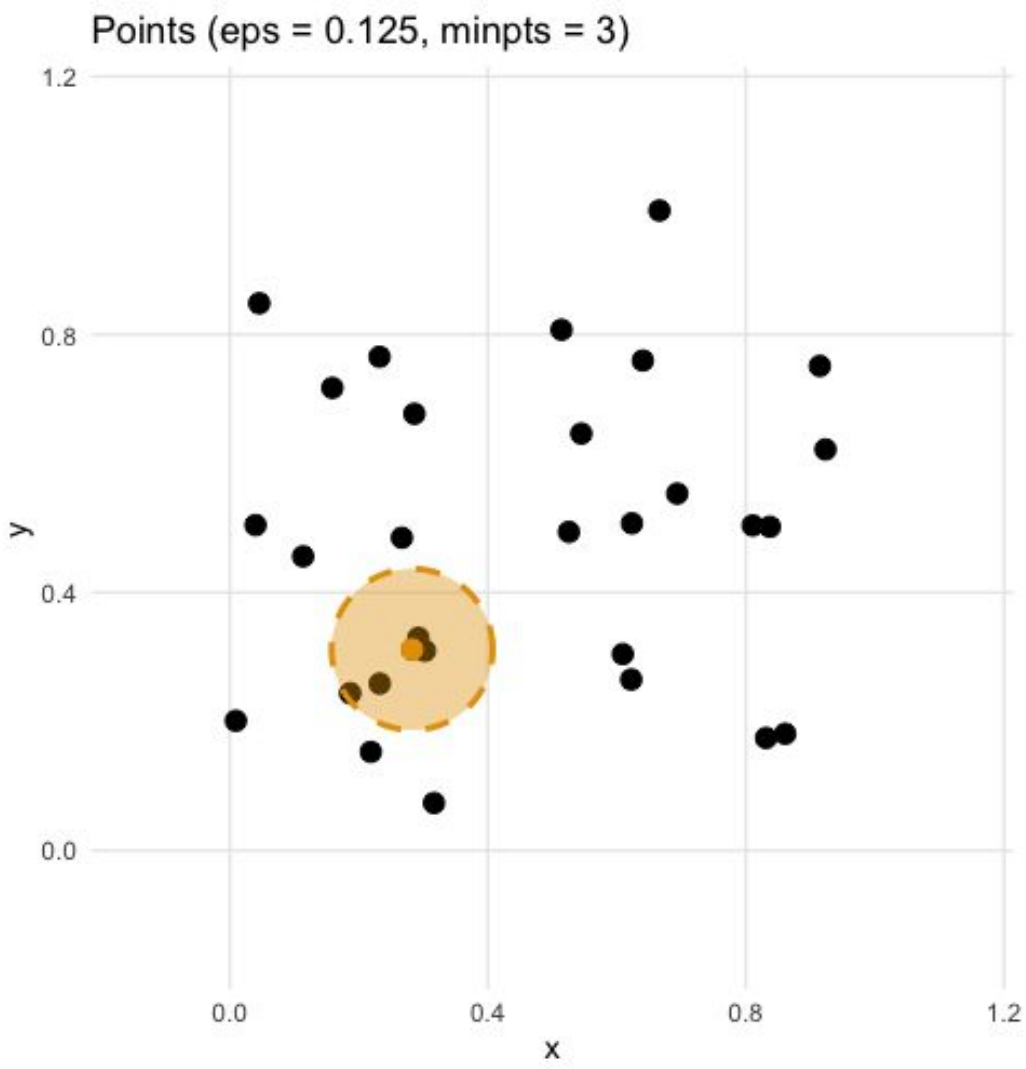


DBSCAN

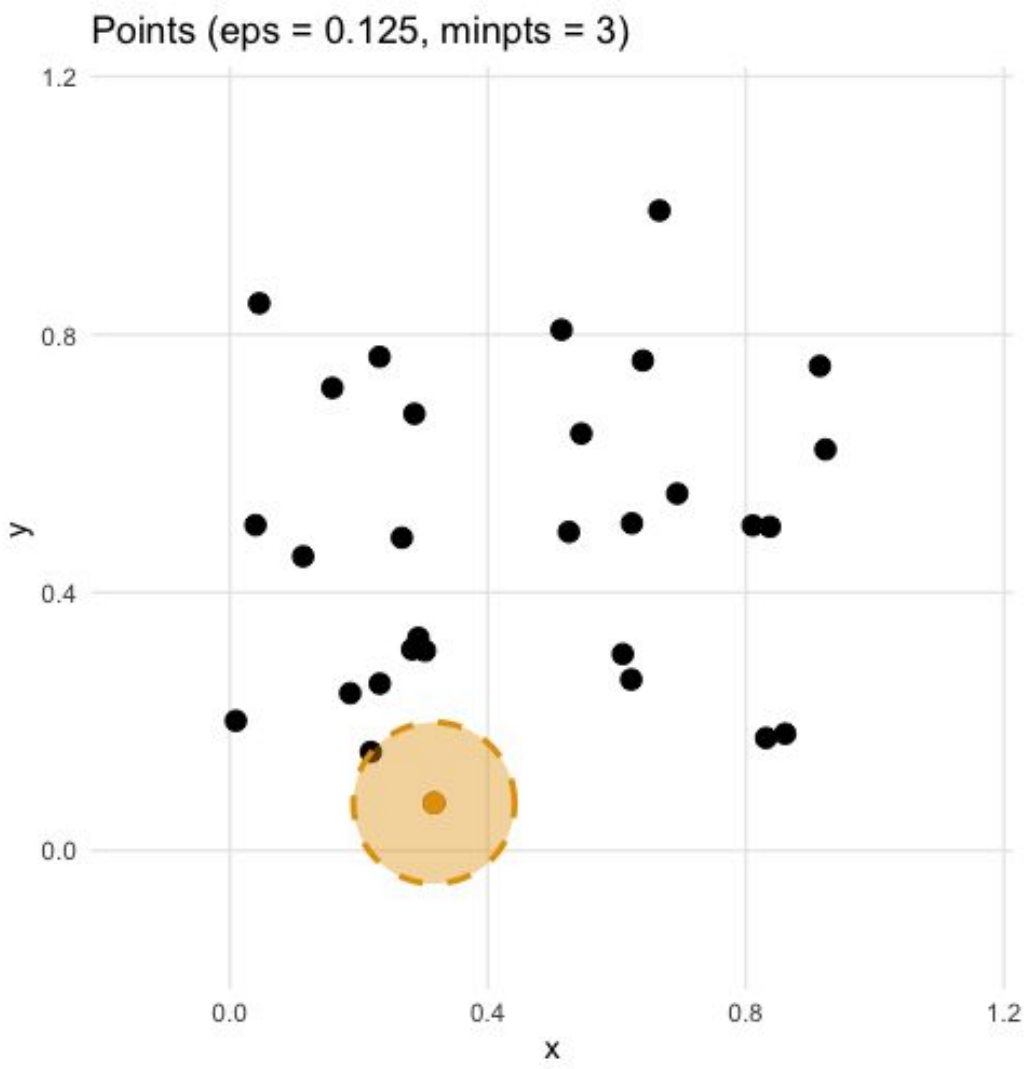
Density **B**ased **S**patial **C**lustering of **A**pplications with **N**oise

- Distance Metric
- Epsilon (*eps*)
- Minimum Points (*minpts*)

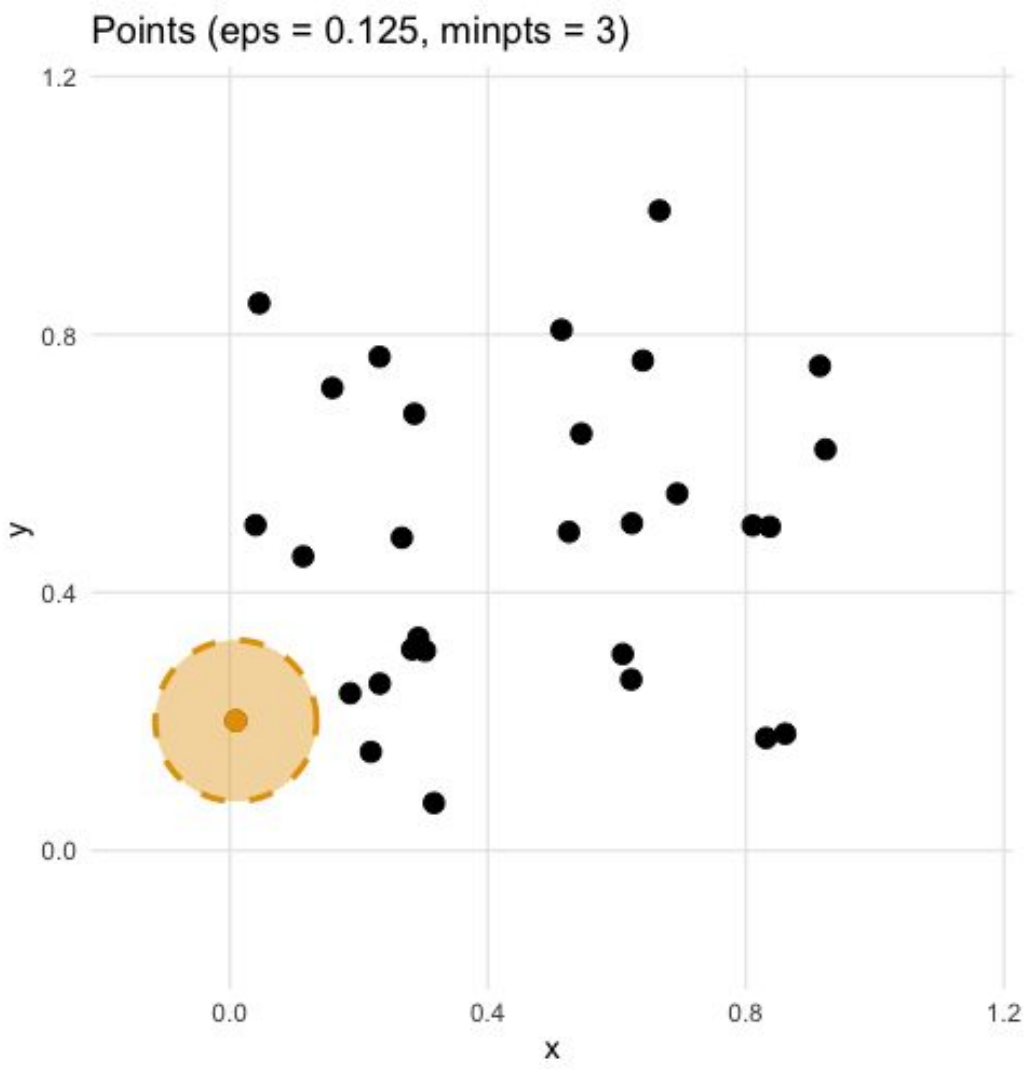
Core Point*:* **p** is a core point if it has at least *minpts* neighbors within *eps* distance of itself



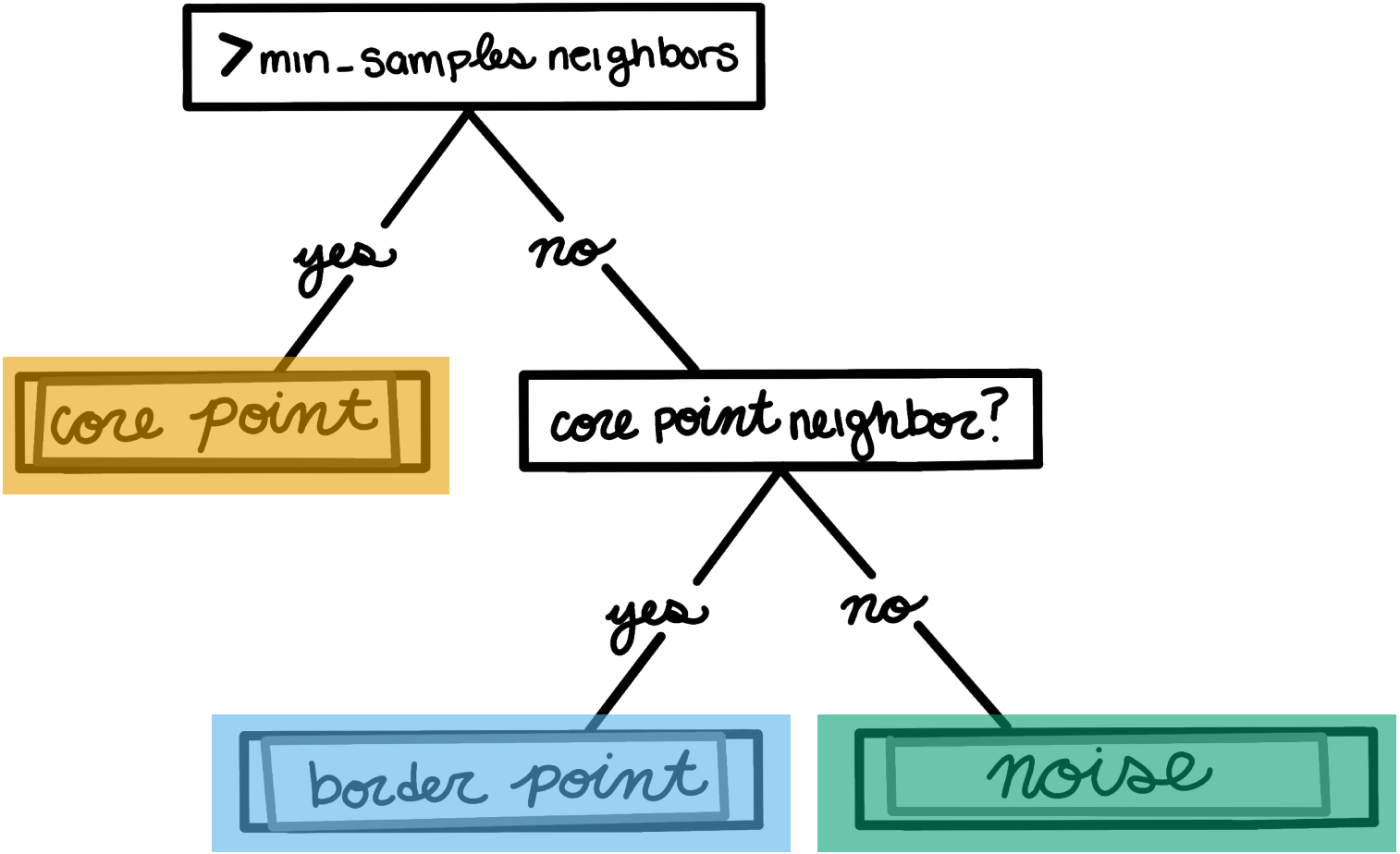
Border Point*: **p** is a border point if it DOES NOT have at least *minpts* neighbors within *eps* distance of itself, but is a neighbor of a core point



Noise*:* **p** is noise if it DOES NOT have at least *minpts* neighbors within *eps* distance of itself, and IS NOT a neighbor of a core point



DBSCAN

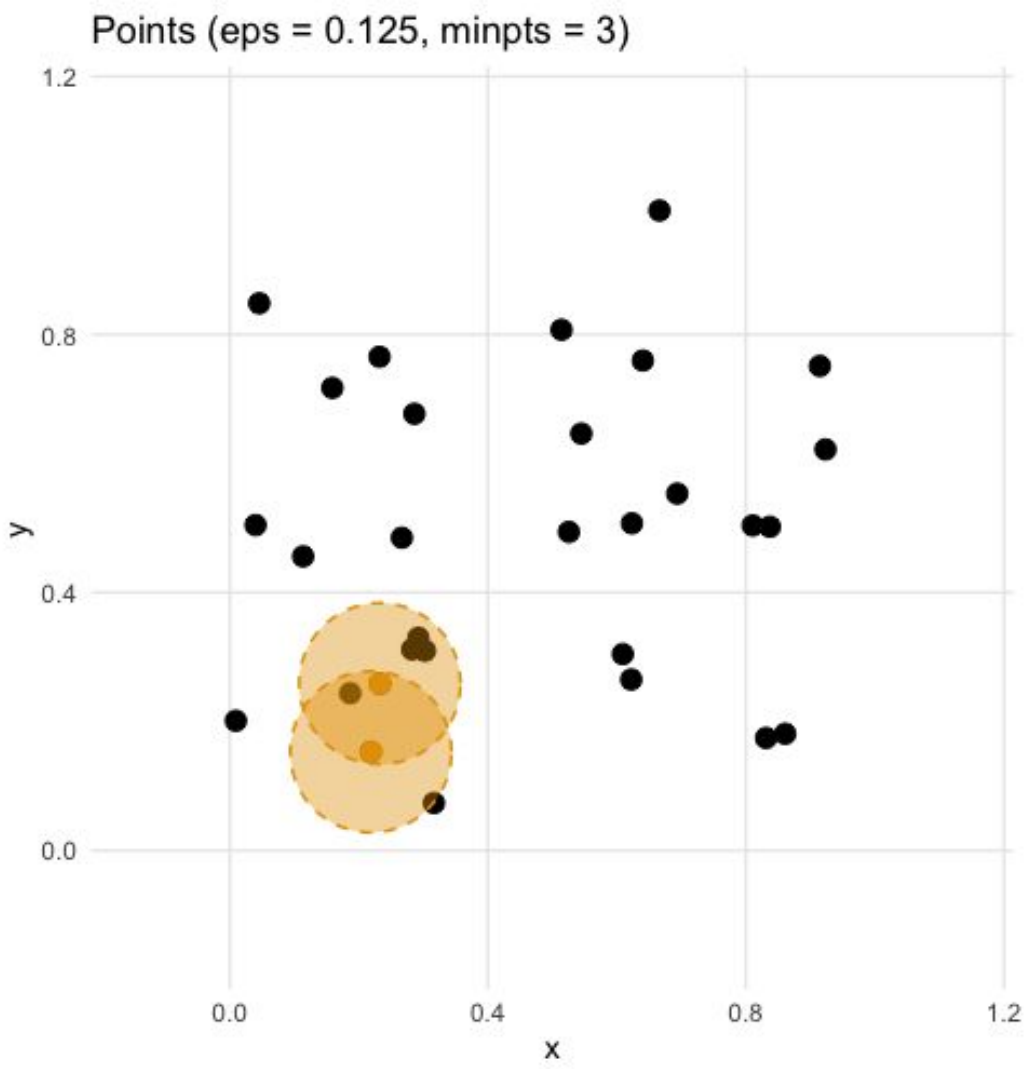

>min_samples neighbors
yes
no
core point
core point neighbor?
yes
no
border point
noise

Definitions

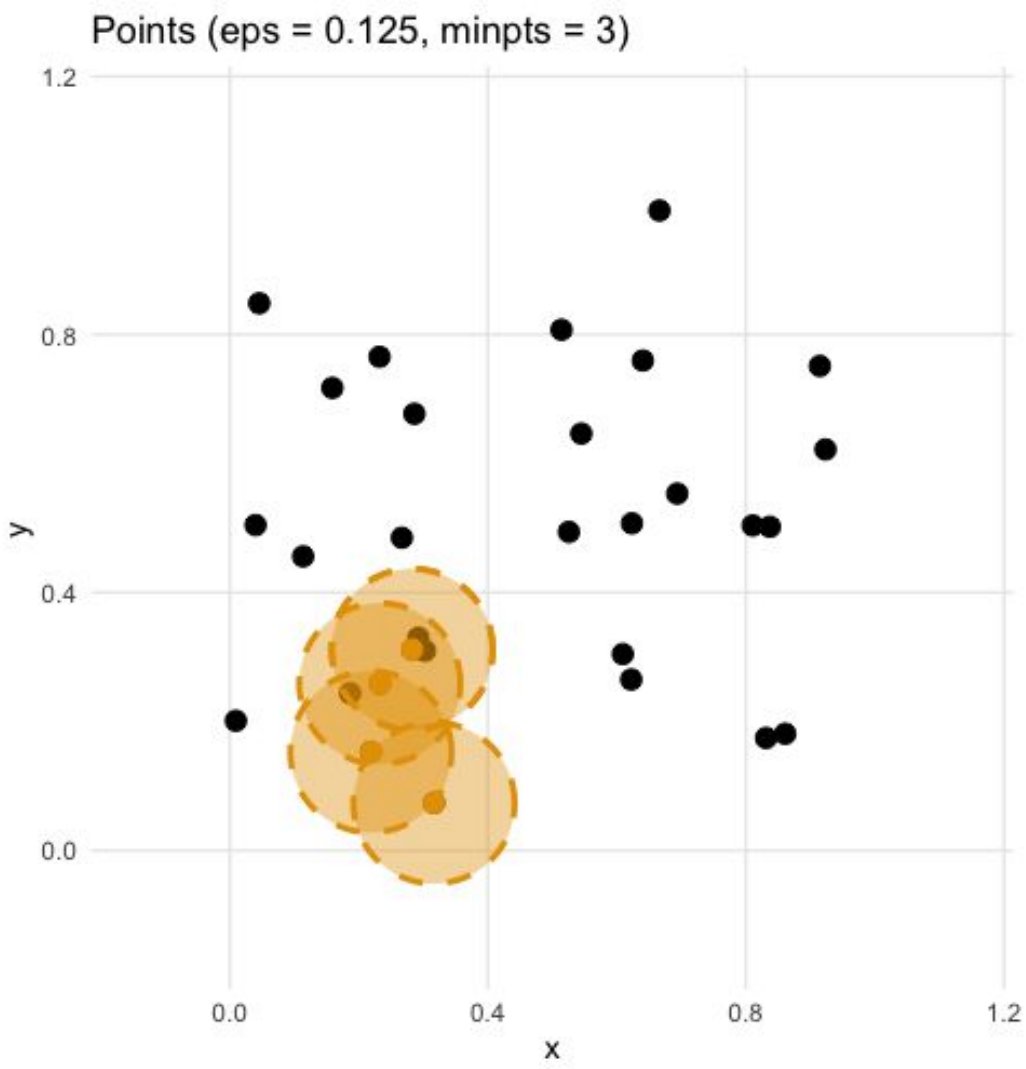
- Directly density reachable: **p** is directly density reachable from core-point **q** if it is in the neighborhood of **q**
- Density reachable*:* **p** is directly reachable from **q** if there are a chain of points that are directly density reachable from **q** to **p**
- Density connected: **p** and **q** are density connected if they are both density reachable from a third point, **o**

- Cluster: choose core point **q**, a cluster **C** contains all points density reachable by **q**
- Noise: any point not in a cluster

Directly density reachable: **p** is directly density reachable from core-point **q** if it is in the neighborhood of **q**

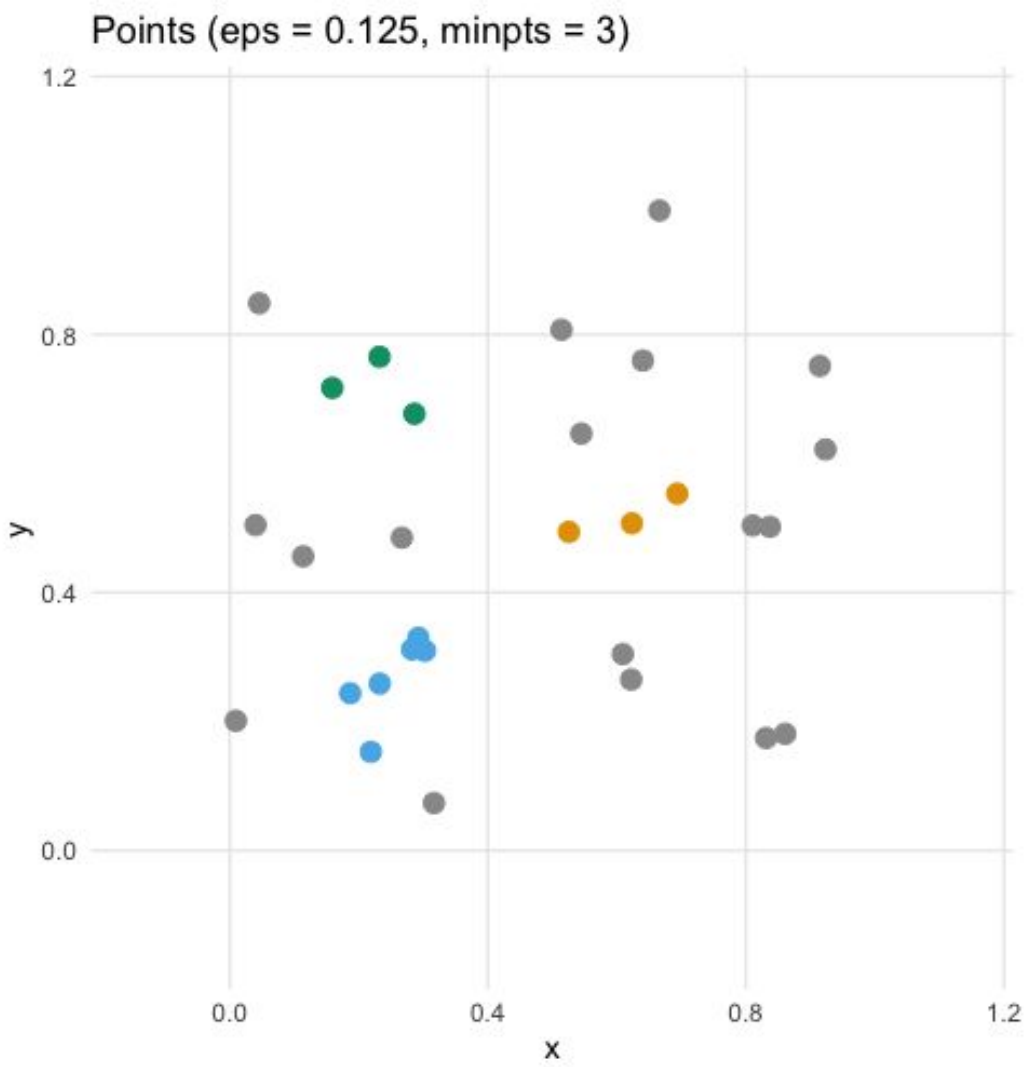


Density reachable*:* **p** is density reachable from **q** if there are a chain of points that are directly density reachable from **q** to **p**

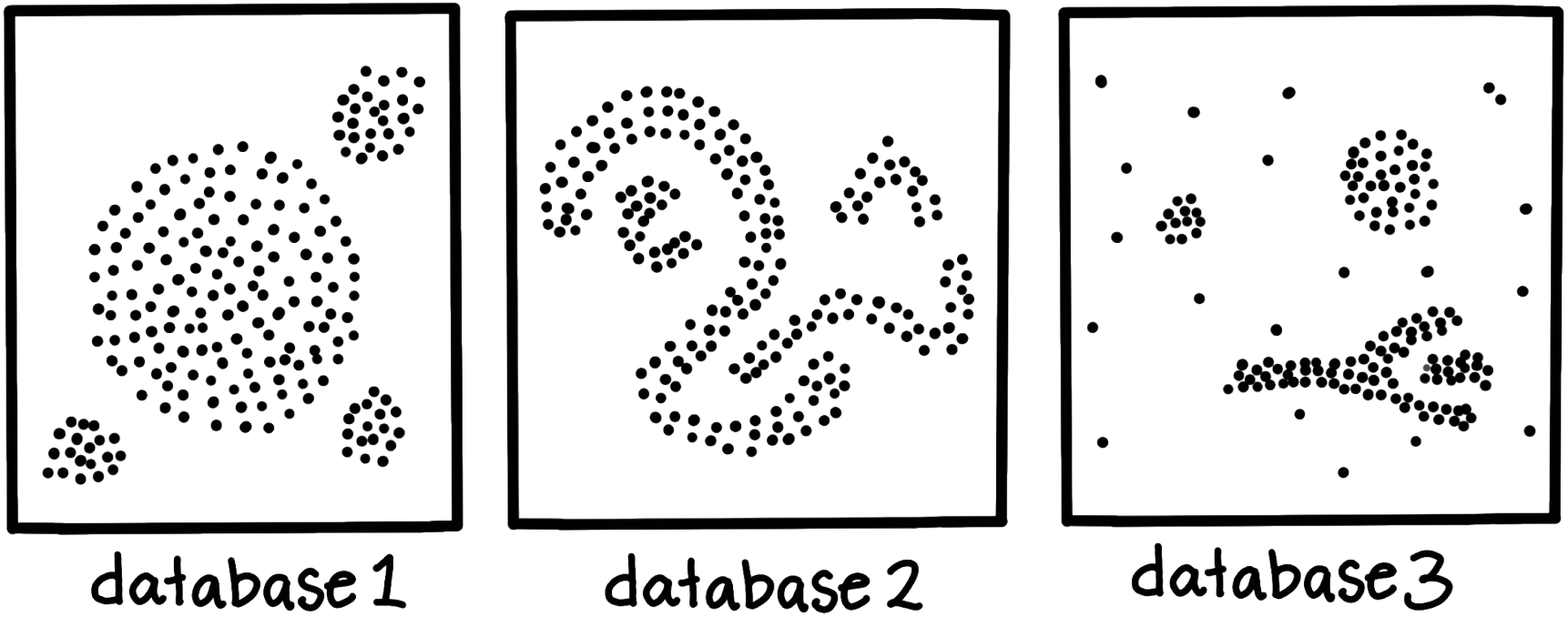


Cluster: choose core point **q**, a cluster **C** contains all points density reachable by **q**

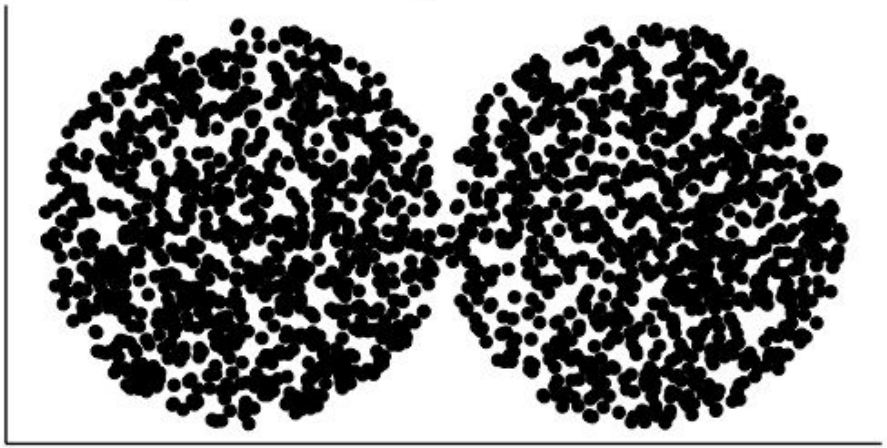
Noise: any point not in a cluster



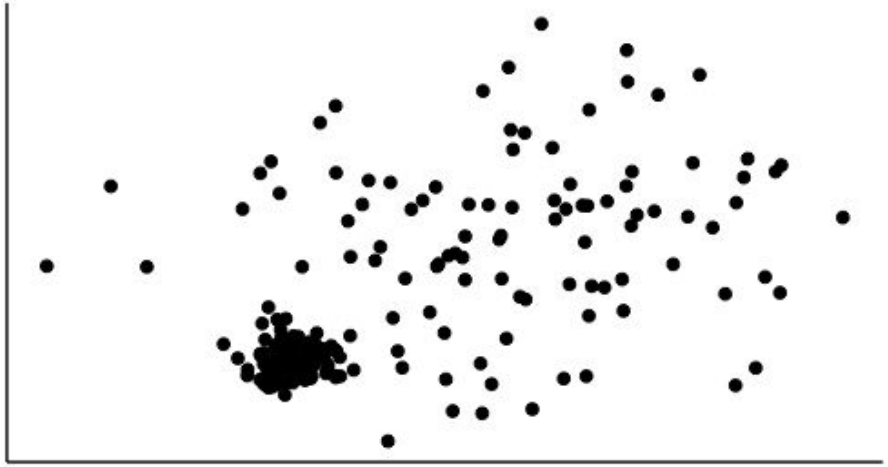
Benefits of DBSCAN



Touching/Overlapping Clusters



Different Density Clusters

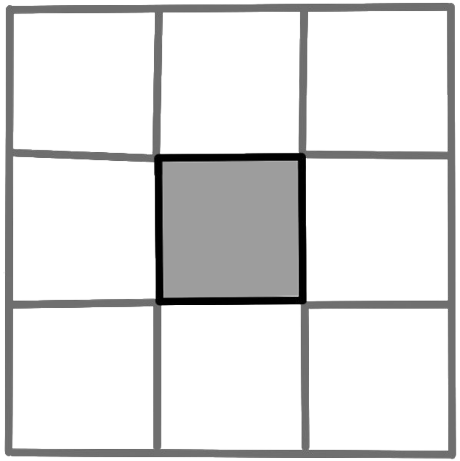


Choosing Minimum Points

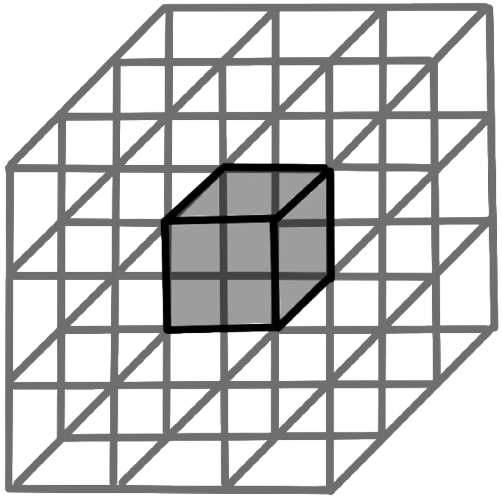
- Domain Knowledge + Distance Metrics
- More rows = larger min_pts
- More noise = larger min_pts
- More features = larger min_pts

Curse of Dimensionality

(a)



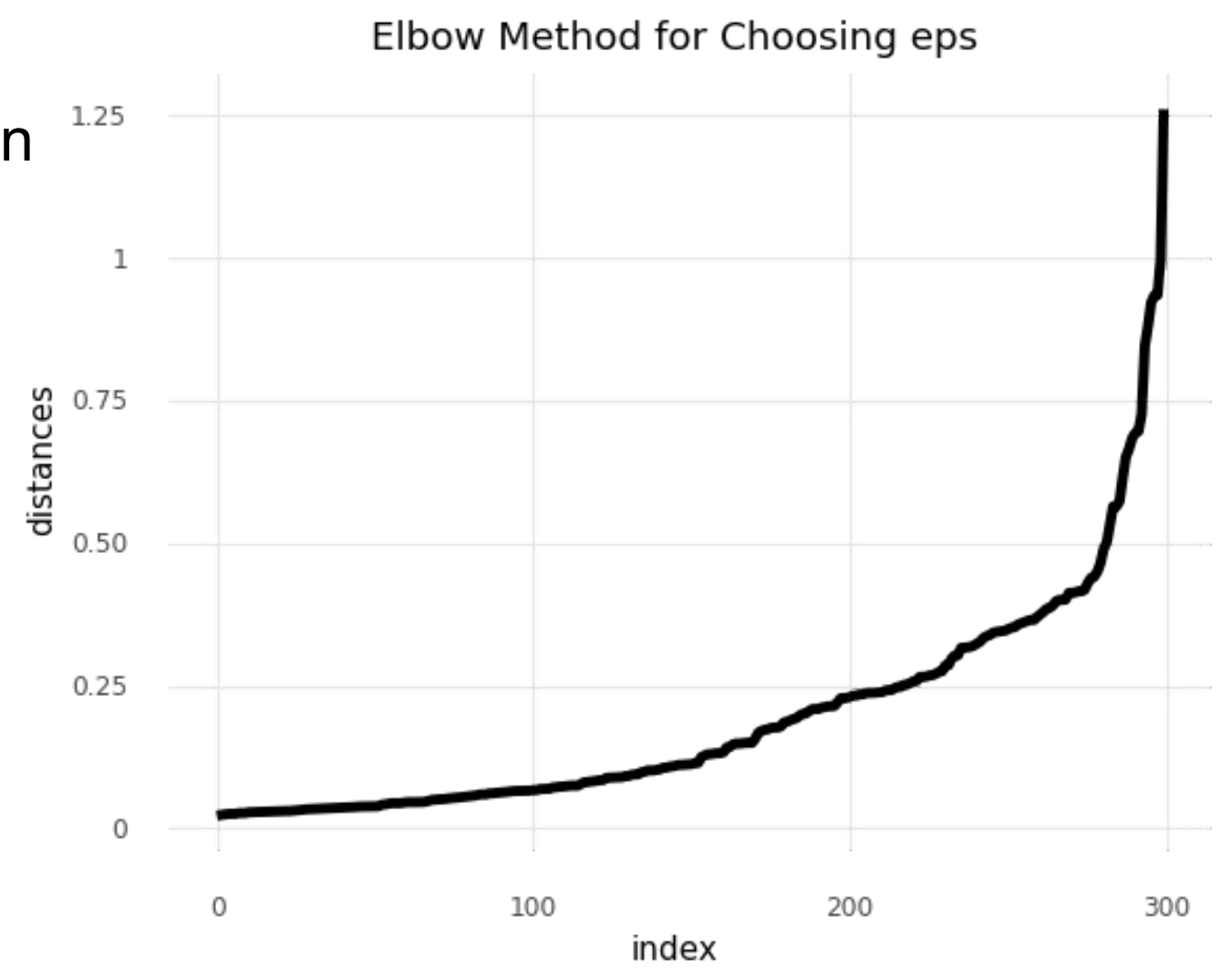
(b)



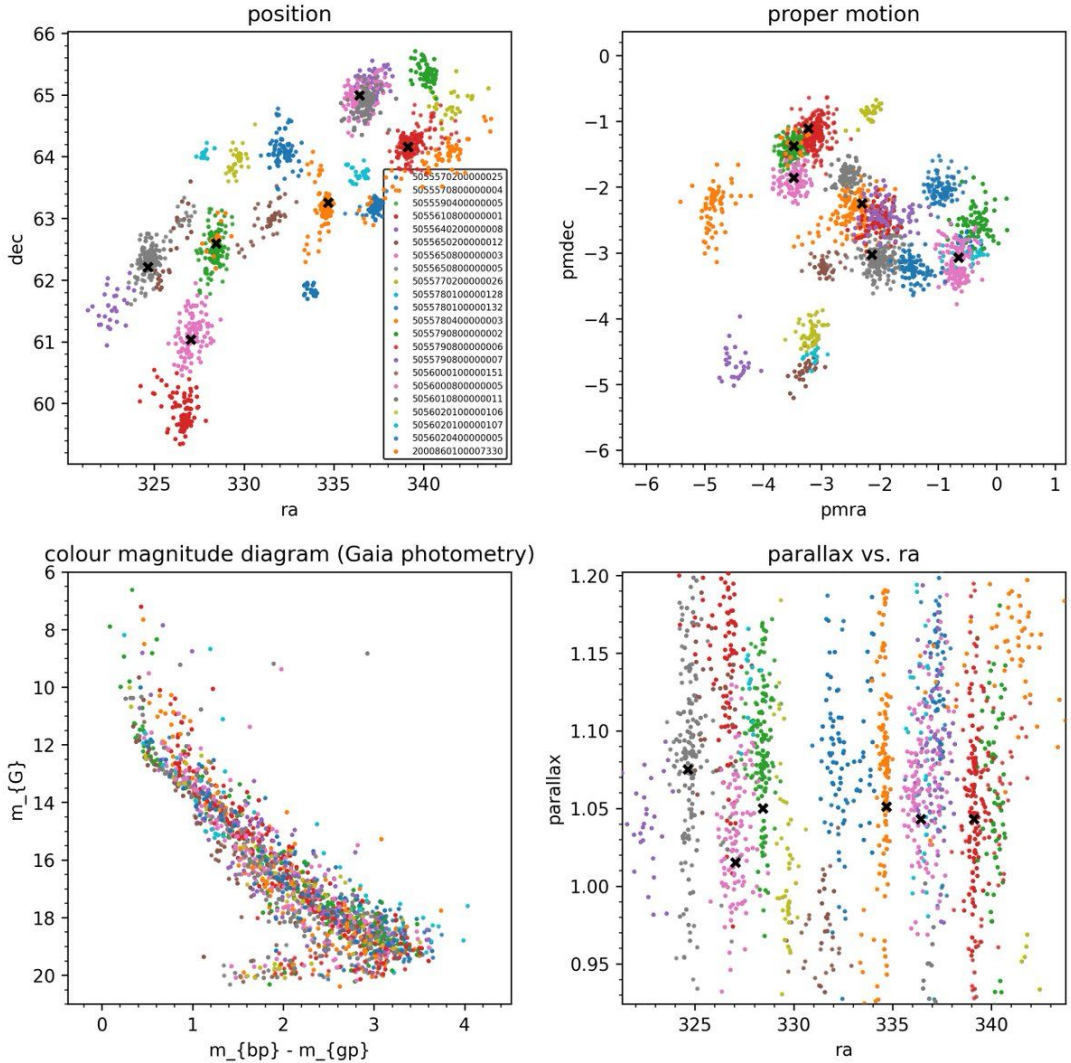
(c)

Choosing Epsilon

- Elbow method (k-dist)
- Domain Knowledge



Applications



https://twitter.com/emilydoesastro/status/1509156861957885953?s=20&t=RtQkzN2G_G7aZEH-en6WfA

Applications

Figure 9 presents a sample image that was segmented using DBSCAN. In the figure, the individual clusters are regrouped together forming close to the original image. It can be observed that the pixels of similar color are clustered together.

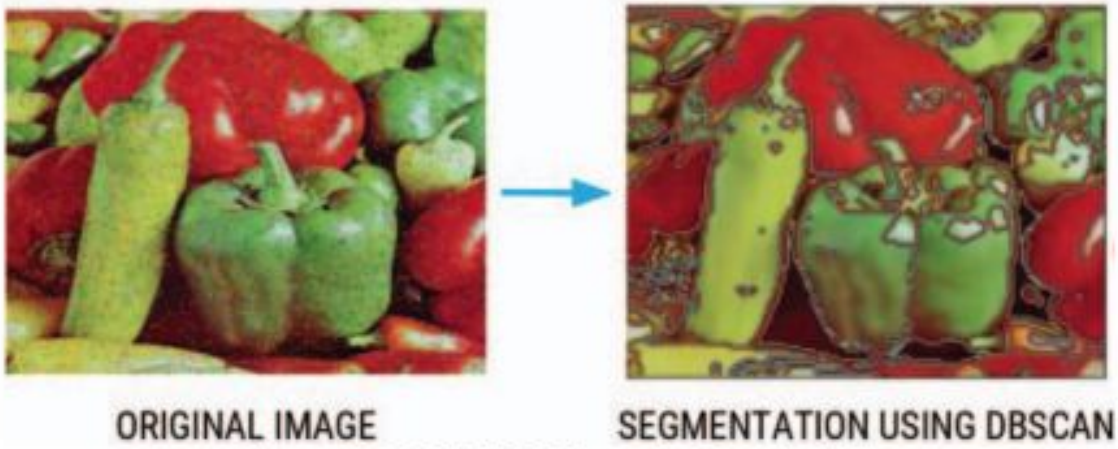


Figure 9: DBSCAN Example Result. [11]

https://ieeexplore.ieee.org/document/7848290

Applications

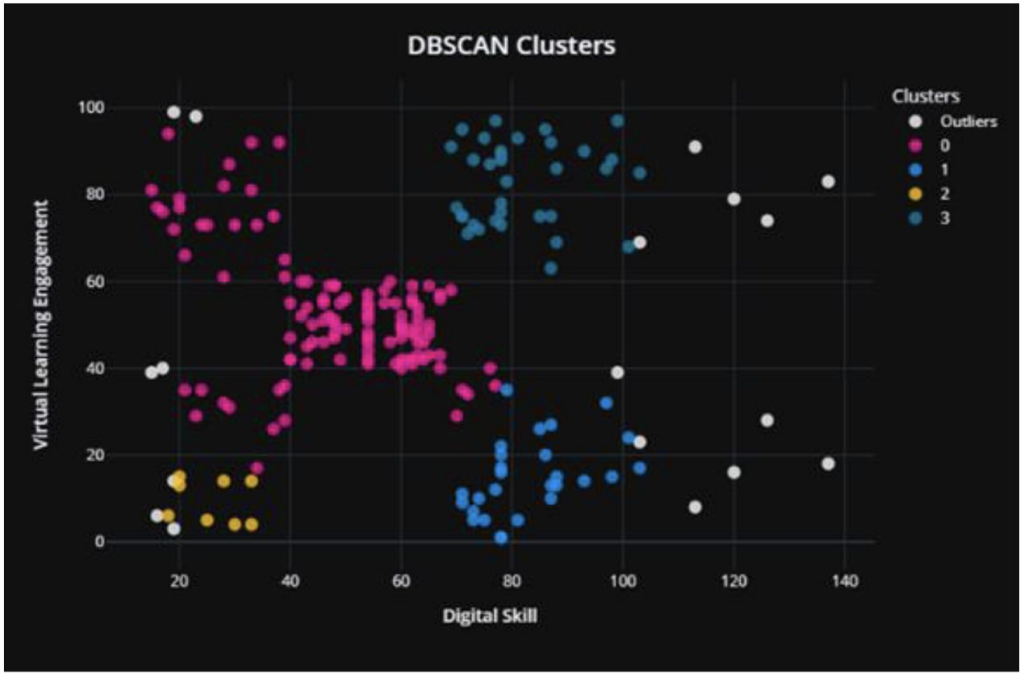


Figure 9.1 DBSCAN clusters

https://journals.sagescience.org/index.php/ssret/article/view/6

DBSCAN visits each point of the database, possibly multiple times (e.g., as candidates to different clusters). For practical considerations, however, the time complexity is mostly governed by the number of regionQuery invocations. DBSCAN executes exactly one such query for each point, and if an indexing structure is used that executes a neighborhood query in $O(\log n)$, an overall average runtime complexity of $O(n \log n)$ is obtained (if parameter ε is chosen in a meaningful way, i.e. such that on average only $O(\log n)$ points are returned). Without the use of an accelerating index structure, or on degenerated data (e.g. all points within a distance less than ε), the worst case run time complexity remains $O(n^2)$. The
(

2)
- $n = (n^2-n)/2$-sized upper triangle of the distance matrix can be materialized to avoid distance recomputations, but this needs $O(n^2)$ memory, whereas a non-matrix based implementation of DBSCAN only needs $O(n)$ memory.