Unsupervised Classification
Clustering

ML Crash Course – Informatics Department of UII

Friday, Februari 21 2020

Lizda.iswari@uii.ac.id

What is Clustering?

- Cluster: Collection of objects
  - Similar within cluster
  - Dissimilar between cluster
- Clustering: grouping objects in clusters
  - No labels: unsupervised classification
  - Plenty possible clustering
  - Exploratory data mining

Types of Clustering

- Hard Clustering: each data object or point either belongs to a cluster completely or not.

- Soft Clustering: a data point can belong to more than one cluster with some probability or likelihood value.

Clustering Methods

- Centroid-based clustering: clusters are represented by a central vector or a centroid.
- Density-based clustering: search the data space for areas of varied density of data points. Clusters are defined as areas of higher density within the data space compared to other regions.
- Connectivity-based clustering: data points that are closer in the data space are more related (similar) than to data points farther away.
- Distribution-based clustering: clustering is based on the notion of how probable is it for a data point to belong to a certain distribution, such as the Gaussian distribution.

Why clustering?

- Pattern Analysis
- Visualize Data
- Pre-processing Step
- Outlier Detection
- …..

- Targeted Marketing Programs
- Student Segmentations
- Data Mining
- …

How is clustering works?

- Measure of Similarity: d(…, …)
  - Numerical Variables → metrics: Euclidean, Manhattan, …
  - Categorical Variables → contruct your own distance
  - See how to measure the similarity of categorical variables: https://stackoverflow.com/questions/29771355/how-can-we-measure-the-similarity-distance-between-categorical-data

Compactness and Separation

- Within Cluster Sums of Squares (WSS):

$$\text{WSS} = \sum_{i=1}^{N_C} \sum_{x \in C_i} d(\mathbf{x}, \bar{\mathbf{x}}_{\mathrm{C_i}})^2$$

$\bar{\mathbf{x}}_{\mathrm{C_i}}$	Cluster Centroid
$\mathbf{x}$	Object
C_i	Cluster
N_C	#Clusters

Measure of **compactness** ← Minimise WSS

- Between Clusters Sums of Squares (BSS):

$$\text{BSS} = \sum_{i=1}^{N_C} |C_i| \cdot d(\bar{\mathbf{x}}_{\mathrm{C_i}}, \bar{\mathbf{x}})^2$$

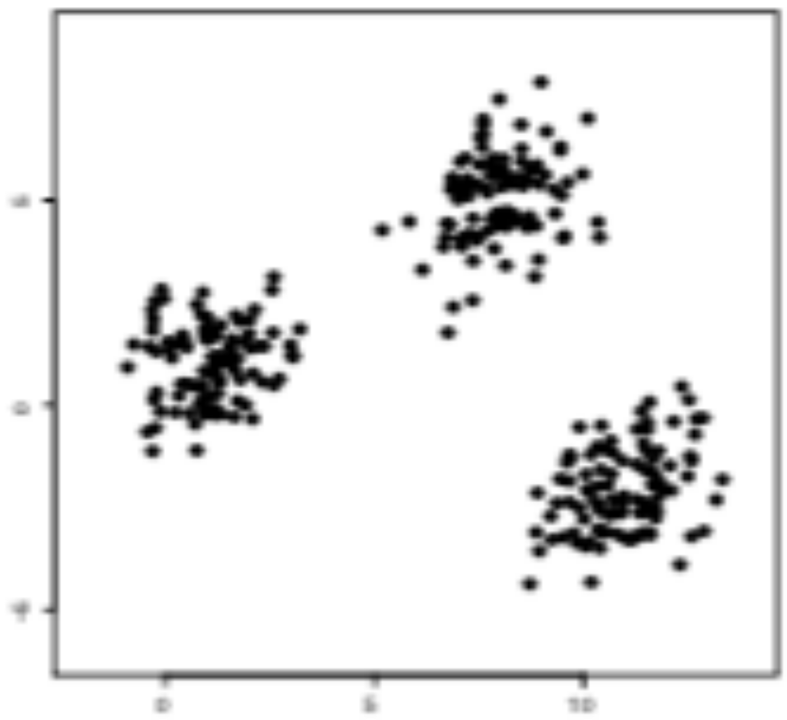
$\bar{\mathbf{x}}_{\mathrm{C_i}}$	Cluster Centroid
N_C	#Clusters
$\lvert C_i \rvert$	#Objects in Cluster
$\bar{\mathbf{x}}$	Sample Mean

Measure of **separation** ← Maximise BSS

K-Means Algorithm

- Goal: Partition data in k disjoint subsets

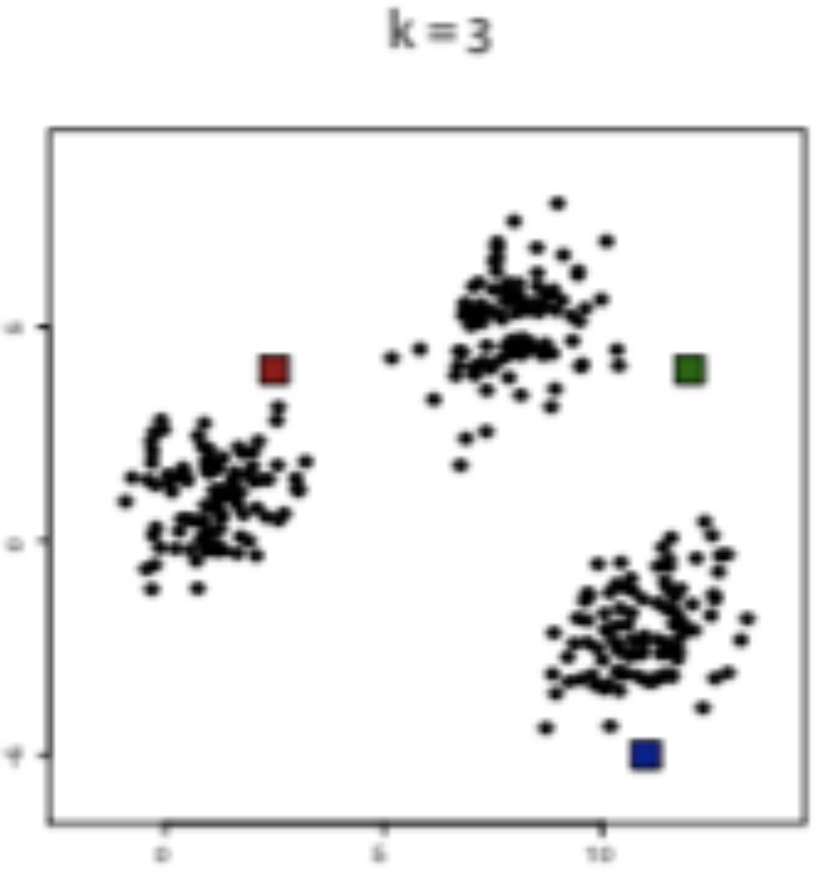
Let's take k = 3



K-Means Algorithm

- Goal: Partition data in k disjoint subsets

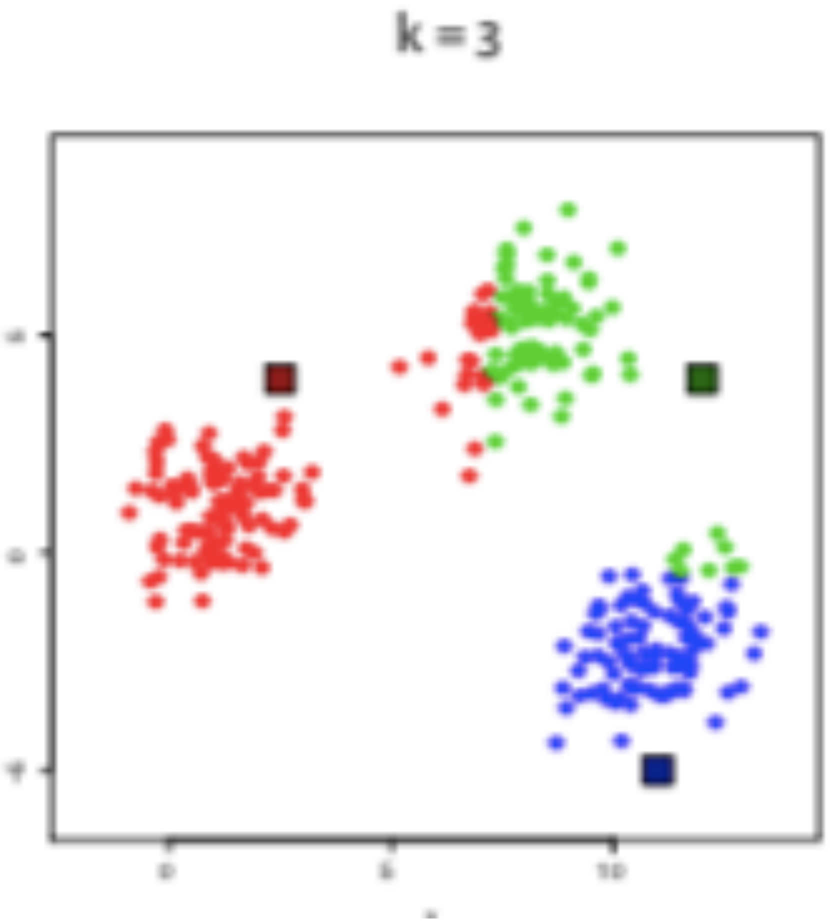
- 1. Randomly assign k centroids



Algoritma K-Means

- Goal: Partition data in k disjoint subsets

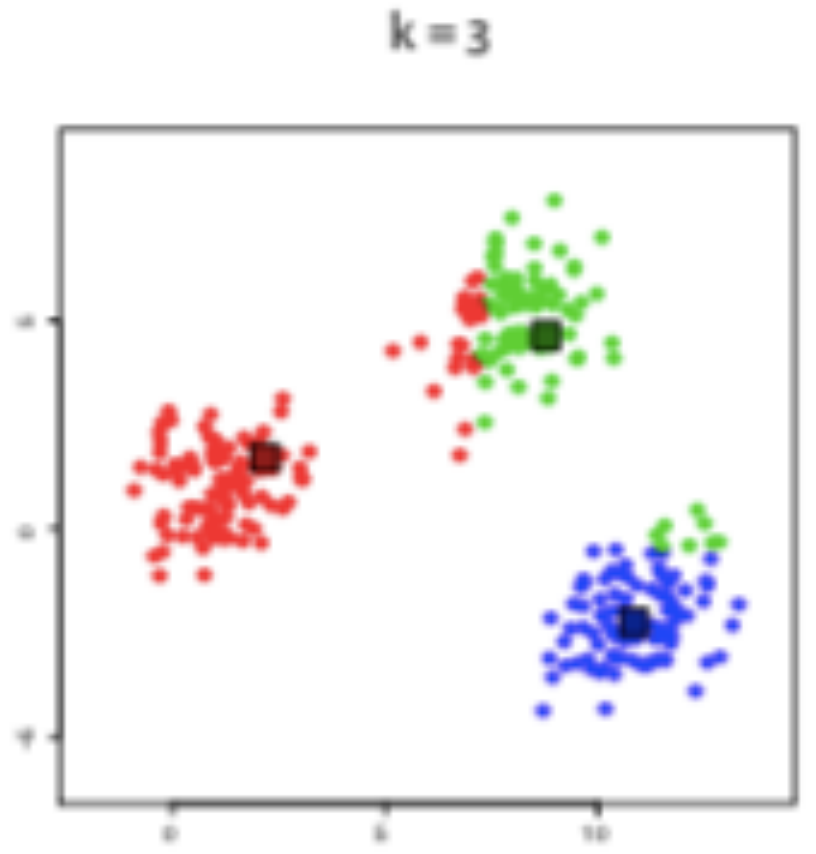
- 1. Randomly assign k centroids
- 2. Assign data to closest centroid



Algoritma K-Means

- Goal: Partition data in k disjoint subsets

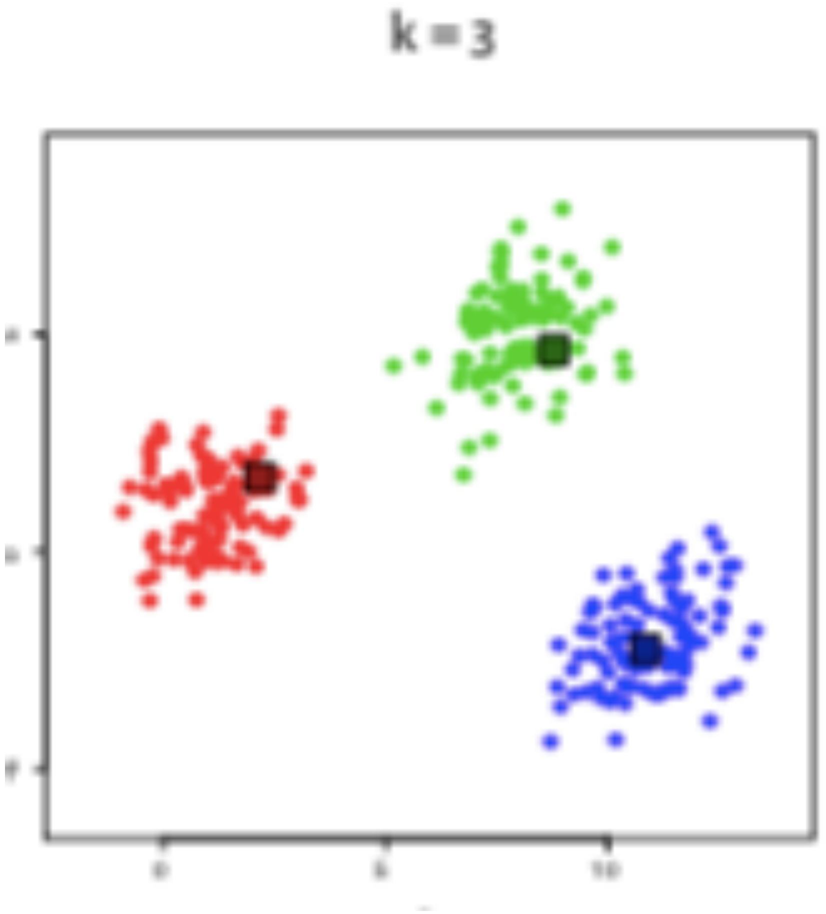
- 1. Randomly assign k centroids
- 2. Assign data to closest centroid
- 3. Moves centroid to average location



Algoritma K-Means

- Goal: Partition data in k disjoint subsets

- 1. Randomly assign k centroids
- 2. Assign data to closest centroid
- 3. Moves centroid to average location
- 4. Repeat step 2 and 3



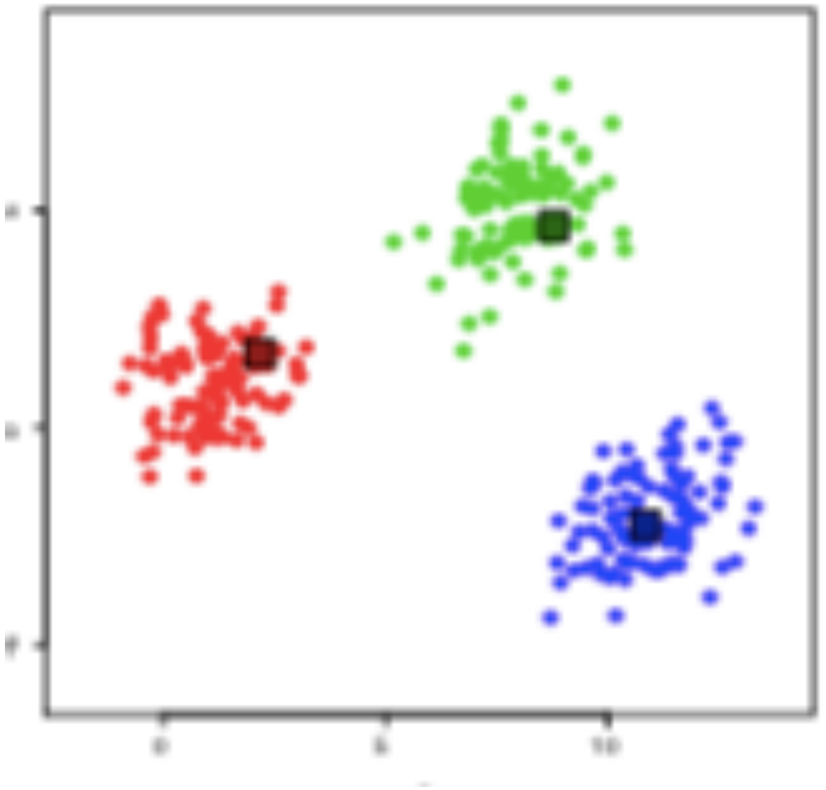
Algoritma K-Means

- Goal: Partition data in k disjoint subsets

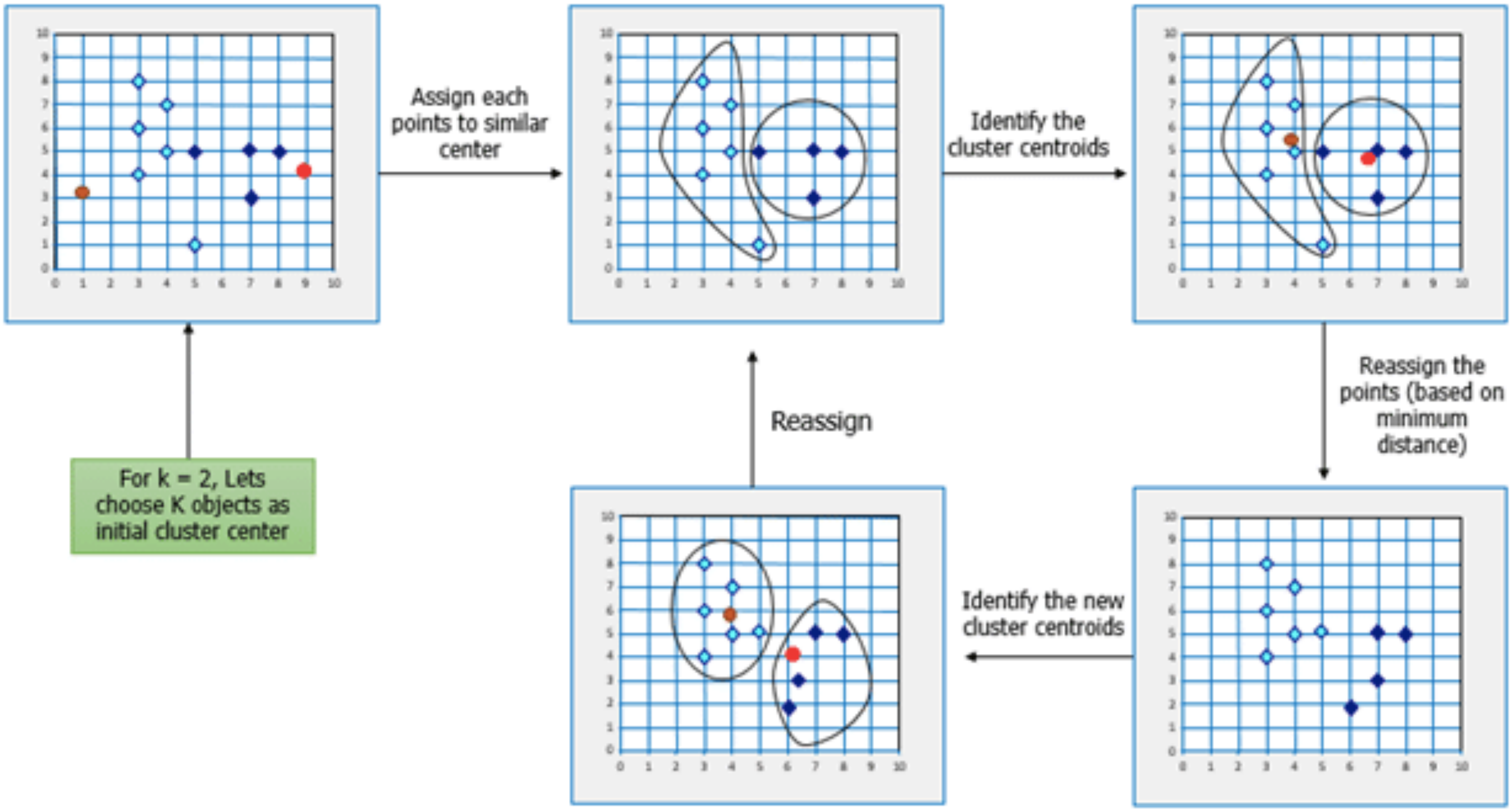
- 1. Randomly assign k centroids
- 2. Assign data to closest centroid
- 3. Moves centroid to average location
- 4. Repeat step 2 and 3

The algorithm has converged!

k = 3



The step by step process:



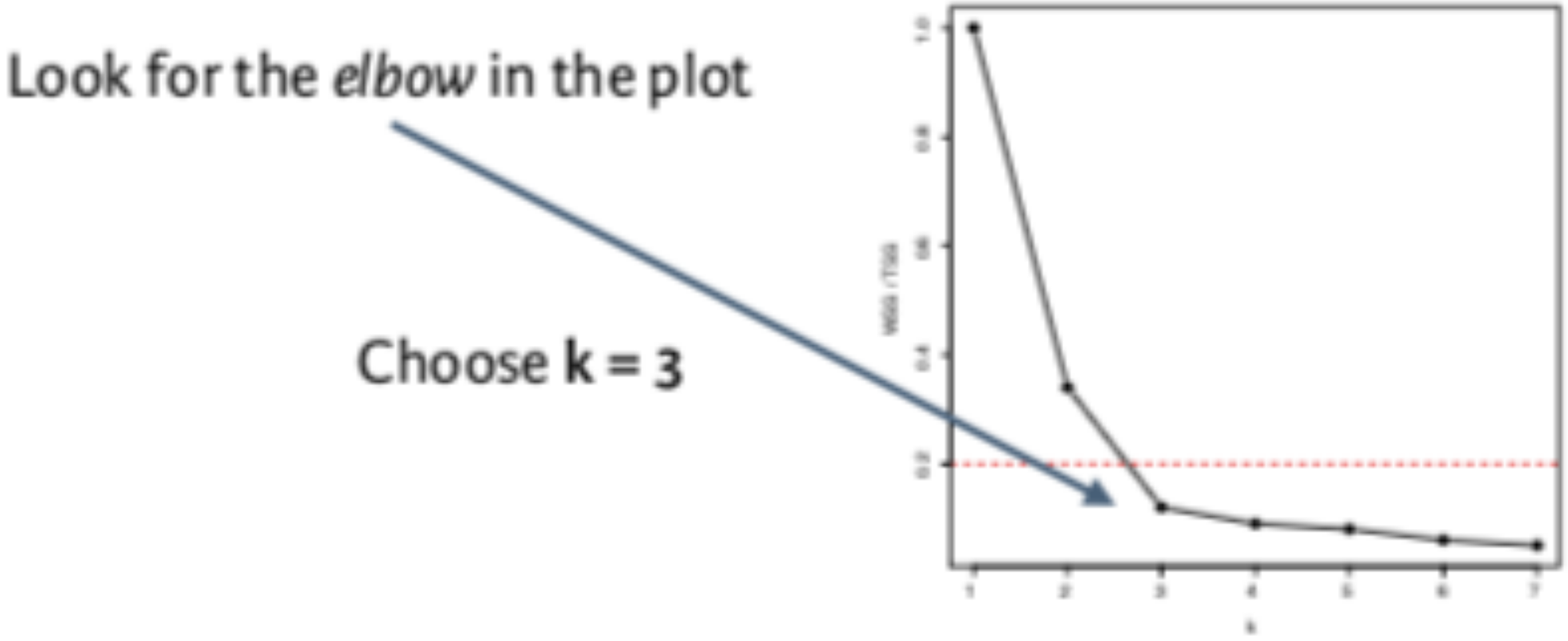
Choosing k

- **Goal**: Find k that minimizes WSS
- **Problem**: WSS keeps decreasing as k increase!
- **Solution**:
  - WSS starts decreasing slowly
  - WSS/TSS < 0.2

TSS = WSS + BSS

Choosing k

- **Scree Plot**: Visualizing the ratio WSS/TSS as function of k



K-Means in R

```
> my_km <- kmeans(data, centers, nstart)
```

- **Centers**: starting centroid or #clusters
- **Nstart**: #time R restart with different centroids

- *Distance*: Euclidean metric

```
> my_km$tot.withinss    ←  WSS

> my_km$betweenss       ←  BSS
```

Performance and Scaling

Evaluasi Cluster

- **Not trivial!! There is no truth.**

- No true labels
- No true response

- Evaluation methods? Depends on the goal.

- Goal: Compact and Separated ← Measurable

Cluster Measures

- WSS and BSS: Good indication

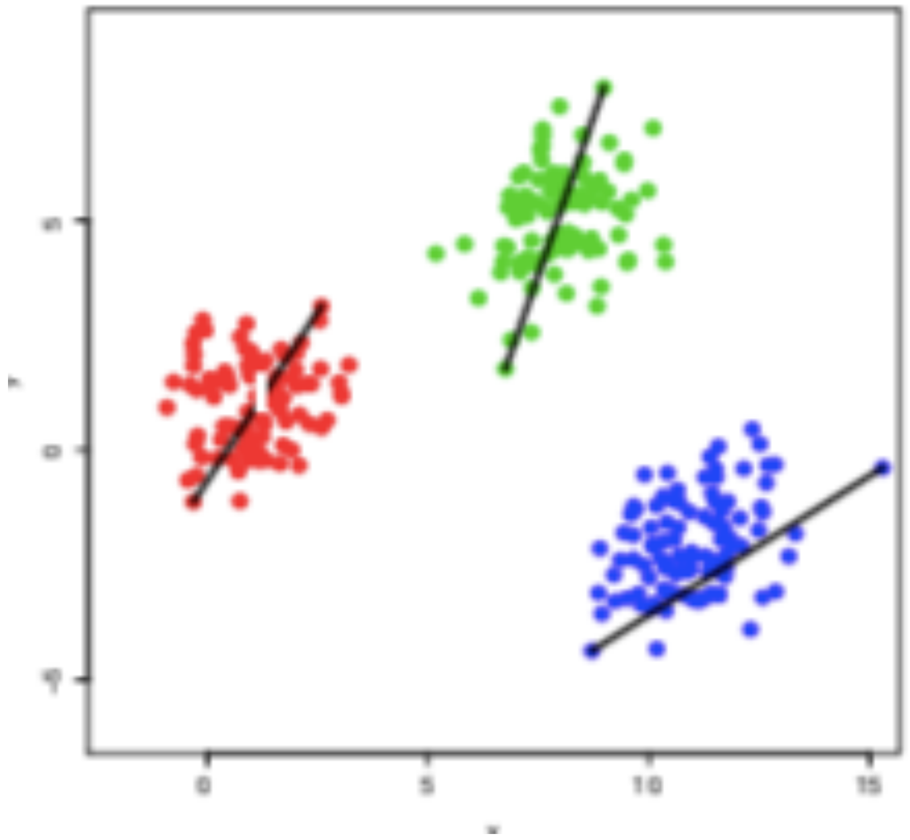
- Underlying idea, compare these measurements:
  - Variance within clusters
  - Separation between clusters

- Alternative:
  - Diameter.
  - Intercluster Distance.

Diameter

- **Measure of Compactness**

$$\mathrm{Dia}_i = \max_{x,y \in C_i} d(x, y)$$



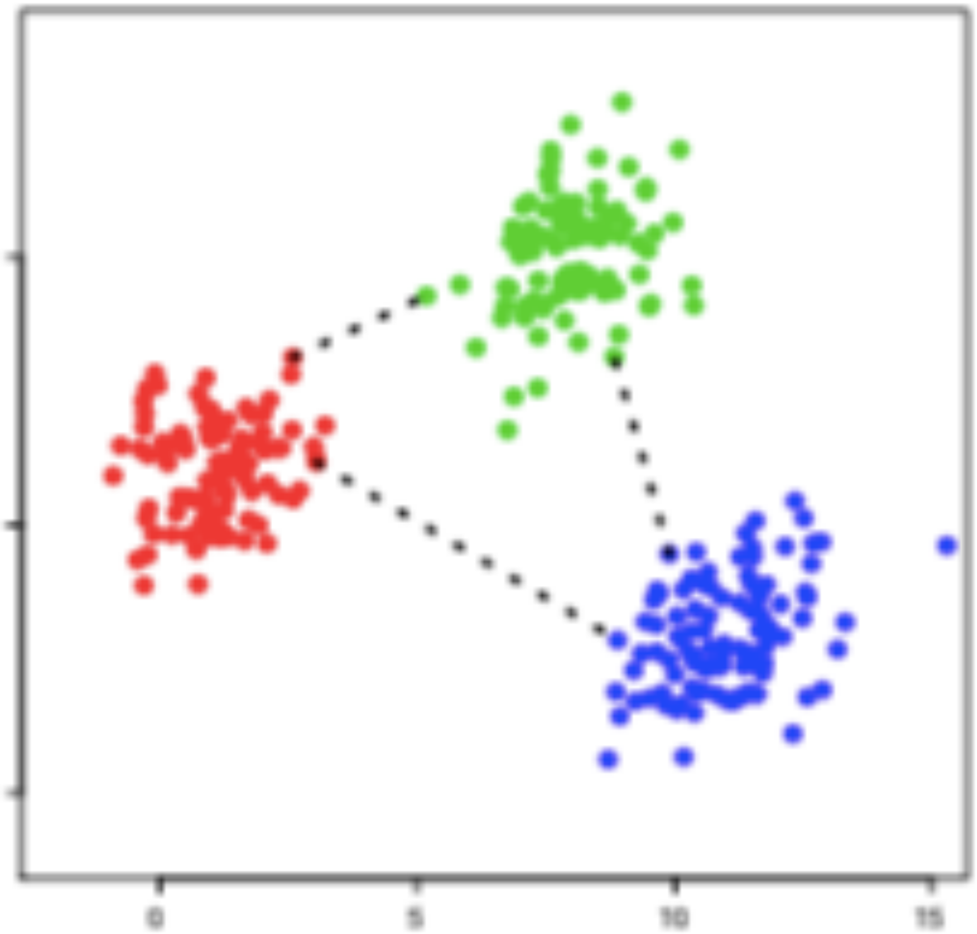
- X, Y : objects
- Ci : cluster
- D : distance (objects)

Intercluster Distance

- **Measure of Separation**

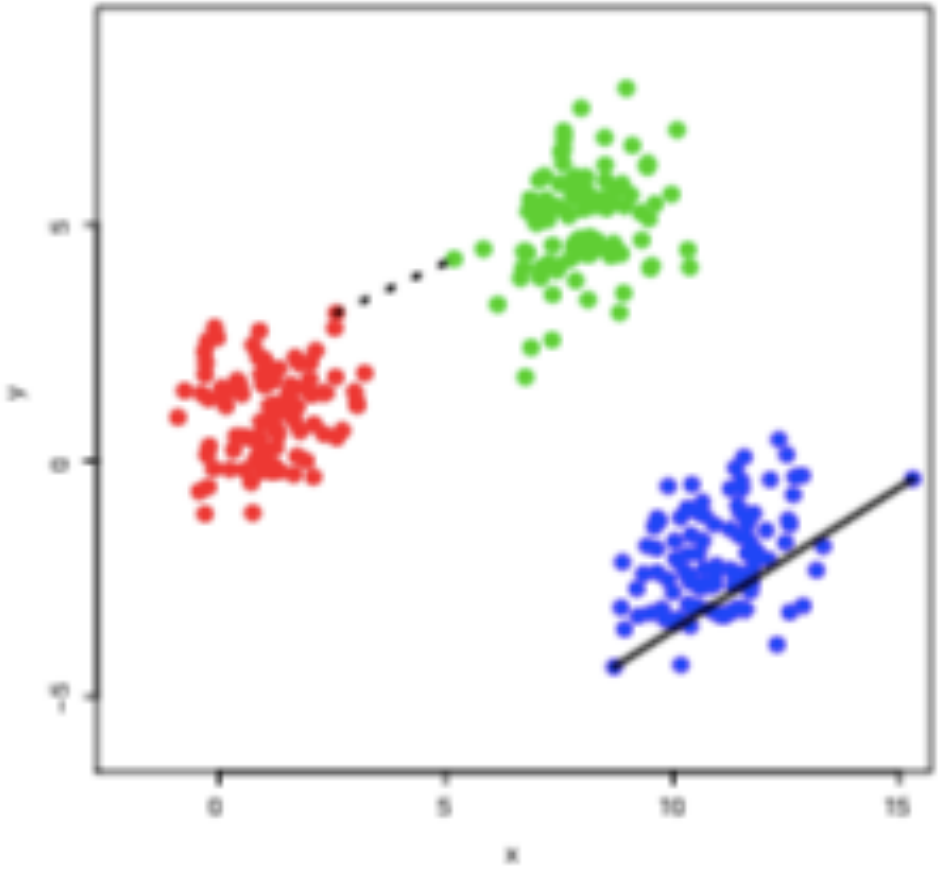
$$\delta(C_i, C_j) = \min_{x \in C_i, y \in C_j} d(x, y)$$

- X, Y : objects
- Ci, Cj : cluster
- D : distance (objects)



Dunn's Index

$$\frac{\min\limits_{1 \leq i < j \leq k} \delta(C_i, C_j)}{\max\limits_{1 \leq m \leq k} \mathrm{Dia}_m}$$



Dunn’s Index

- Higher Dunn → Better separated / more compact

$$\frac{\min\limits_{1\le i<j\le k} \delta(C_i, C_j)}{\max\limits_{1\le m\le k} \mathrm{Dia}_m}$$

- Notes:
  - High computational cost
  - Worst case indicator

Alternative Measures

- Internal Validation: based on intrinsic knowledge
  - BIC Index
  - Silhouette's Index
- External Validation: based on previous knowledge
  - Hulbert's Correlation
  - Jaccard's Coefficient

Scale Issues

Metrics are often scale dependent!

Which pair is most similar? (Age, Income, IQ)

- X1 = (28, 72000, 120)
- X2 = (56, 73000, 80)
- X3 = (29, 74500, 118)

Intuition: (X_1, X_3)

Euclidean: (X_1, X_2)

Solution: **Rescale** income / $1000

Standardizing

- Problem: Multiple variable on different scales
- Solution: Standardize yourdata

  - 1. Subtract the mean
  - 2. Divide by the standard deviation

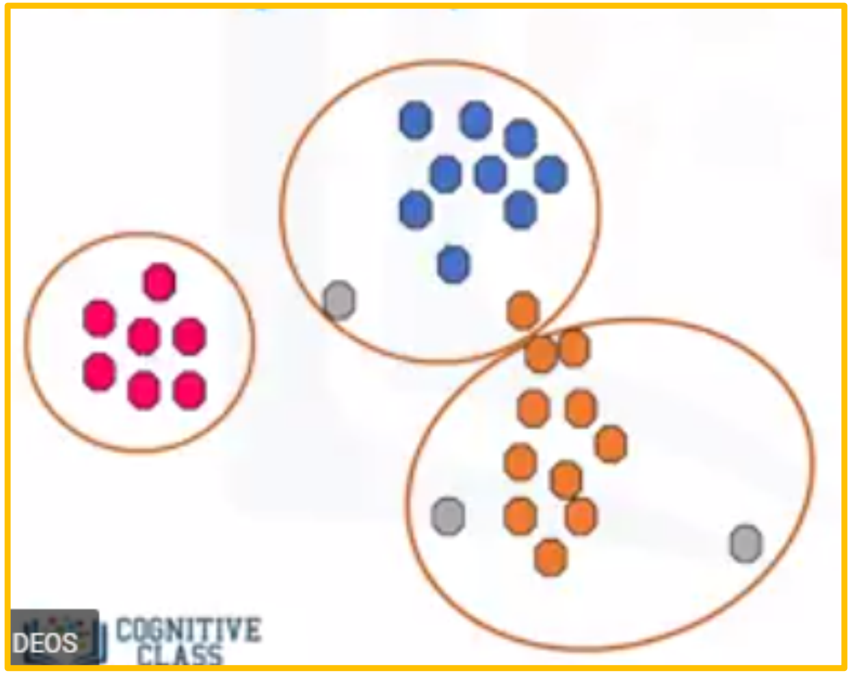
```
> scale(data)
```

  - Note: Standardizing → different interpretation

Disadvantages of Centroid-based Clustering

- Sometimes a dataset can contain extreme values that are outside the range of what is expected and unlike the other data → Outliers.

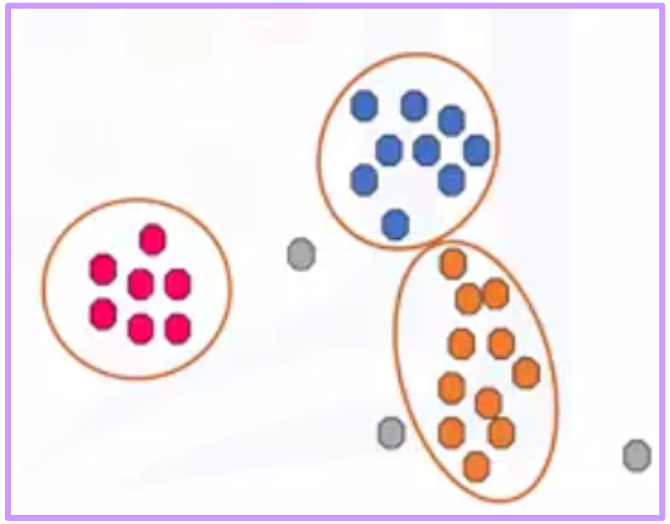
- Generally centroid-based fail to identify the data points that deviate from the normal distribution of the data to a great extent.



Density-based Clustering

Density-based Clustering

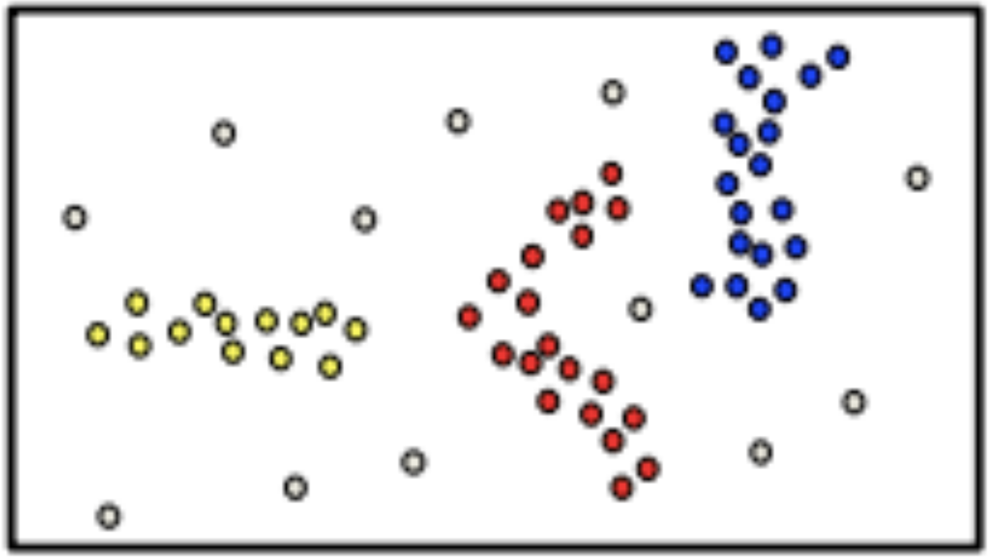
- Relies on a density-based notion of cluster.
- Discovers clusters of arbitrary shape in spatial databases with noise .
- Basic Idea
  - Group together points in high-density
  - Mark as outliers → points that lie alone in low-density regions



- The algorithm: **Density-based spatial clustering of applications with noise (DBSCAN)**

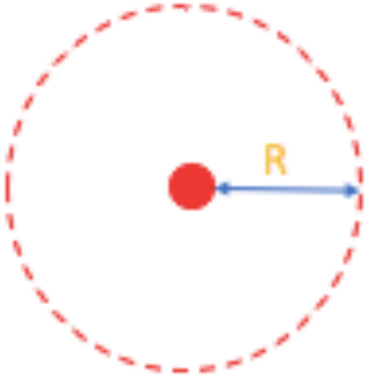
DBSCAN

- **Density-based spatial clustering of applications with noise (DBSCAN)**: groups together points that are close to each other based on a distance measurement (usually Euclidean distance) and a minimum number of points.

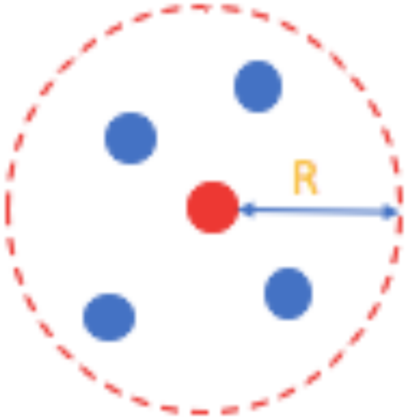


DBSCAN

- The DBSCAN algorithm basically requires 2 parameters:
  - ***eps***: *specifies how close points should be to each other to be considered a part of a cluster. It means that if the distance between two points is lower or equal to this value (eps), these points are considered neighbors.*



  - ***minPoints***: *the minimum number of points to form a dense region. For example, if we set the minPoints parameter as 5, then we need at least 5 points to form a dense region.*

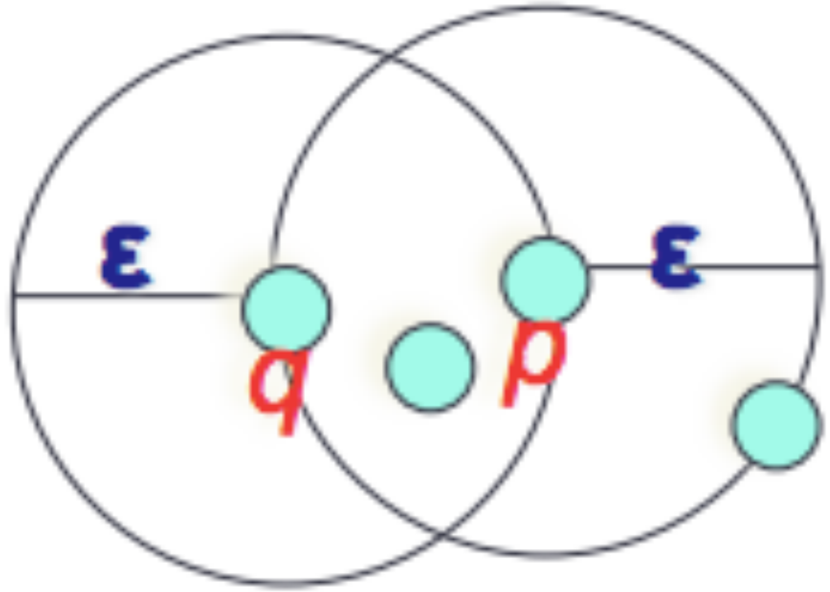


High Density?

- ε-Neighborhood of an point contains at least MinPts.

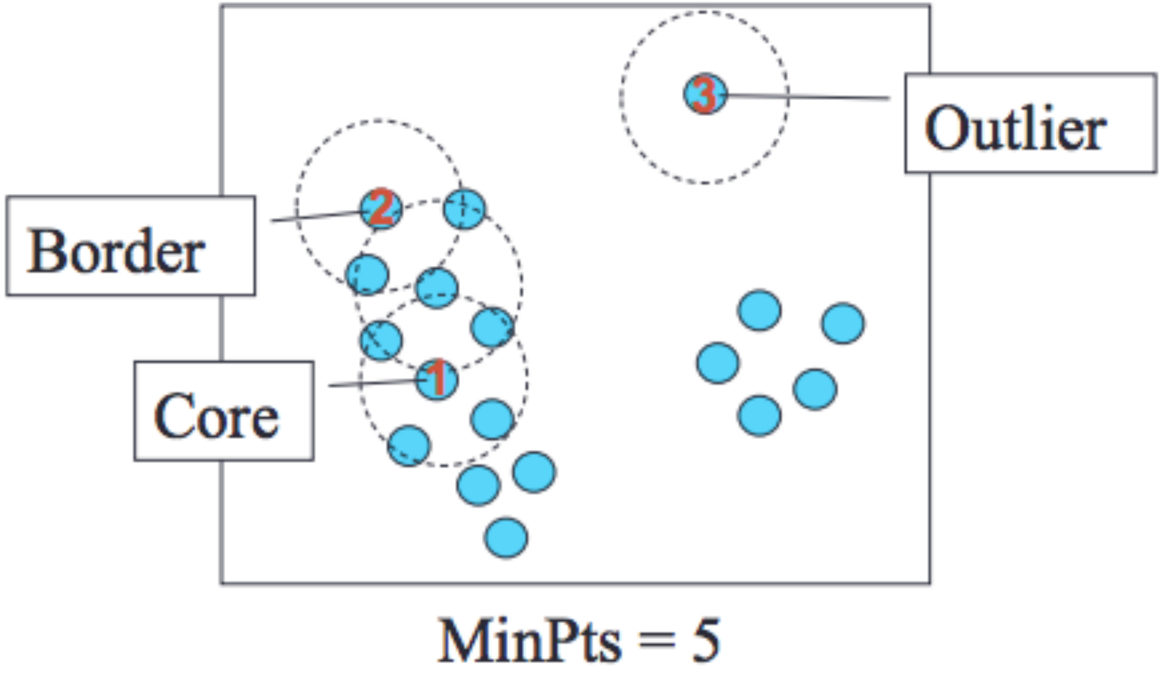
- ε-Neighborhood of p
- ε-Neighborhood of q

- Q. When MinPts = 4?
- Density of p is “high”
- Density of q is “low”



Core, Border & Outlier

- Three category for each point :
  - Core point: if its density is high
  - Border point: density is low (but in the neighborhood of a core point)
  - Noise point: any point that is not a core point nor a border point



DBSCAN Algorithm

1. Choose any arbitrary point.
2. Categorize the point as: core, border, or outlier.
3. Repeat step 1 and 2 for all data objects.
4. Define the number of clusters based on Core Points:
   - If within a radius has one core point, the core points and all related border points are combined into one cluster.
   - If there is more than one core point that has a short distance (within radius range) then all core points and all related border points are combined into one cluster.

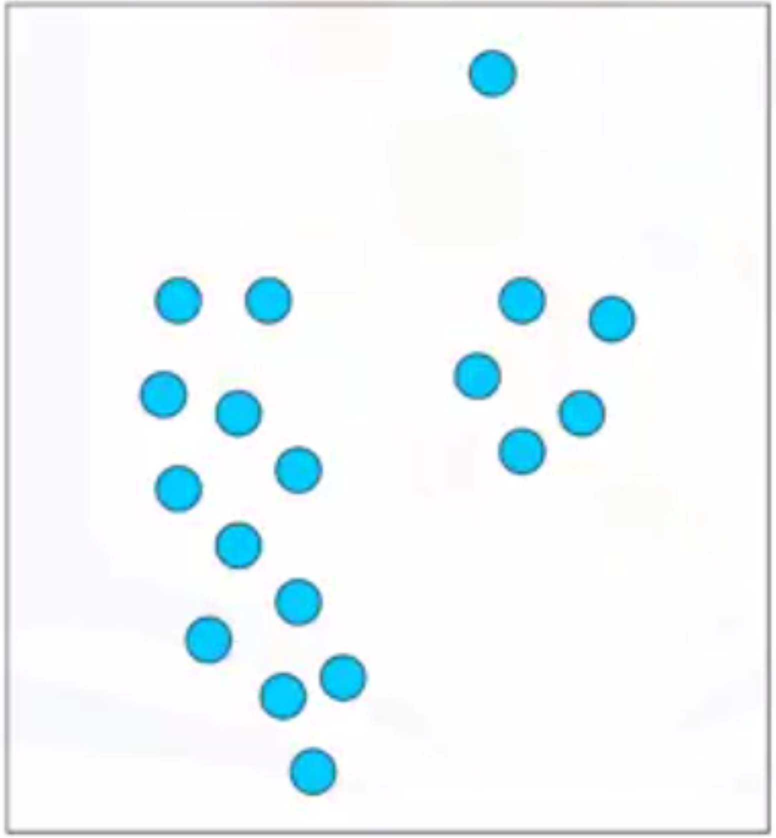
DBSCAN Algorithm

for each $o \in D$ **do**
  if o is not yet classified **then**
    if o is a core-object **then**
      collect all objects density-reachable from o
      and assign them to a new cluster.
    else
      assign o to NOISE

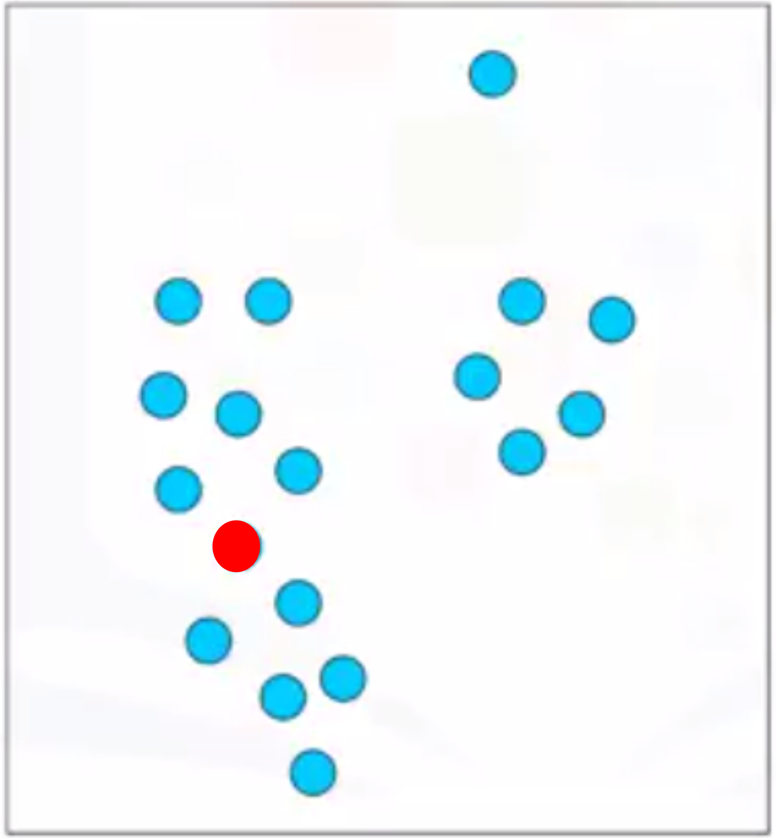
DBSCAN: Example

Parameter:

R = 2 unit dan M = 5



DBSCAN: Example



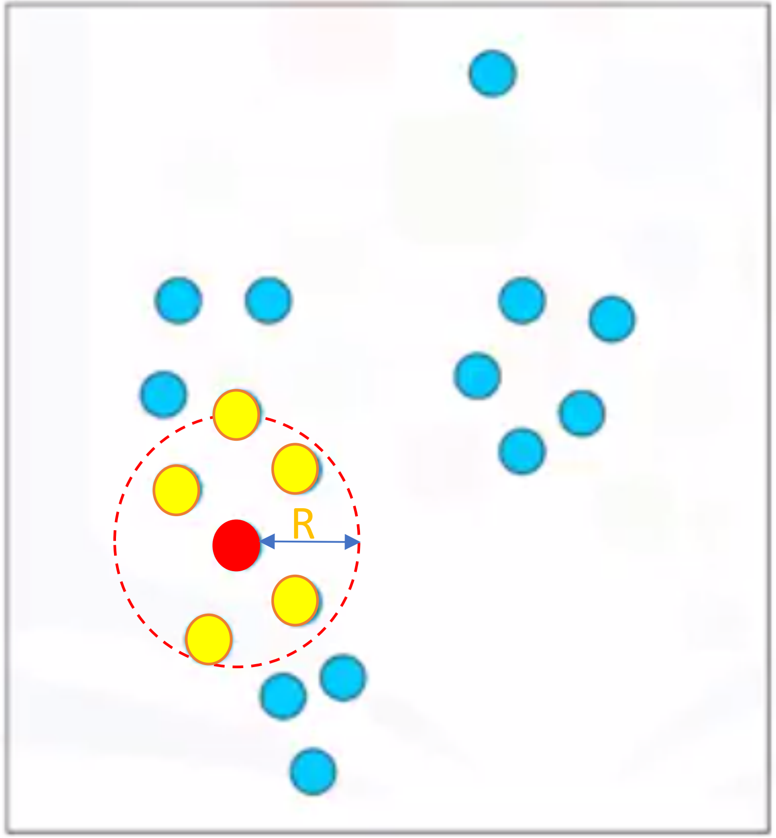
Step 1 :

Choose any arbitrary point

Parameter:

R = 2 unit dan M = 5

DBSCAN: Example



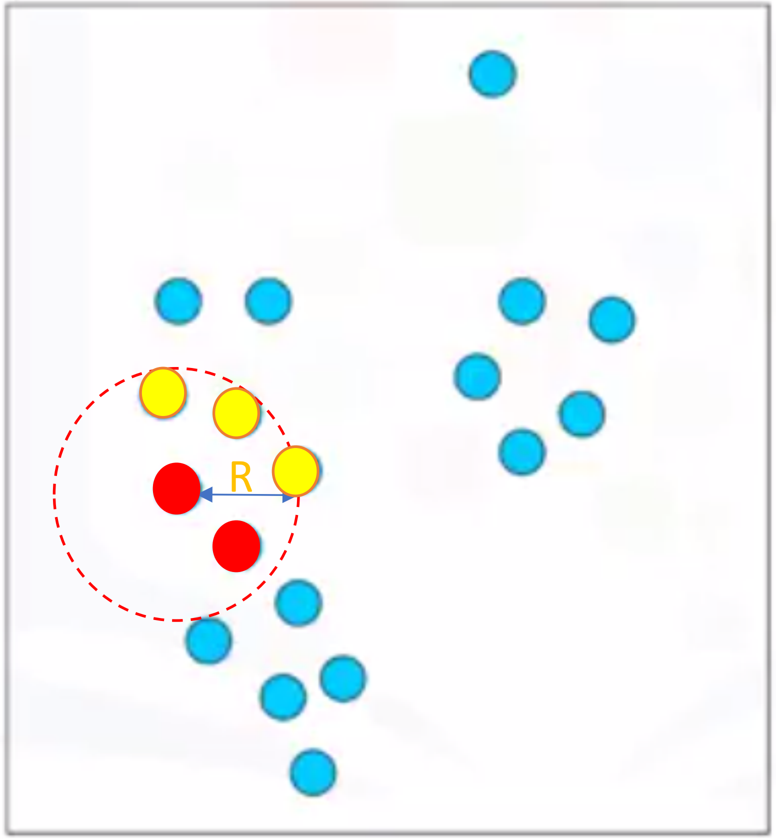
Step 2 :

Categorize the point

Parameter:

R = 2 unit dan M = 5

DBSCAN: Example



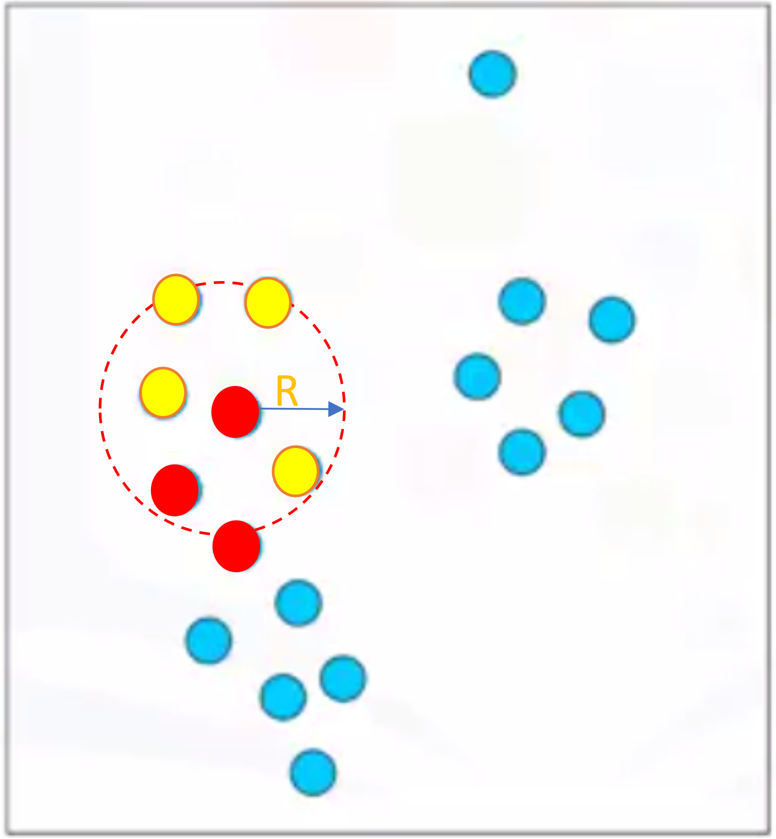
Step 3 :

Repeat step 1 and 2 for all data objects.

Parameter:

R = 2 unit dan M = 5

DBSCAN: Example



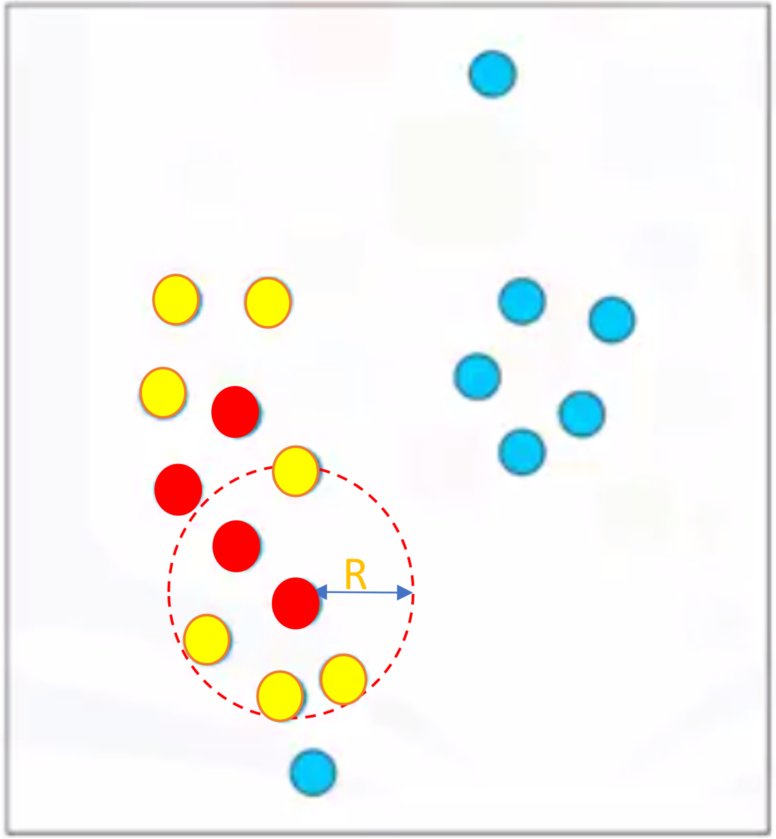
Step 3 :

Repeat step 1 and 2 for all data objects.

Parameter:

R = 2 unit dan M = 5

DBSCAN: Example



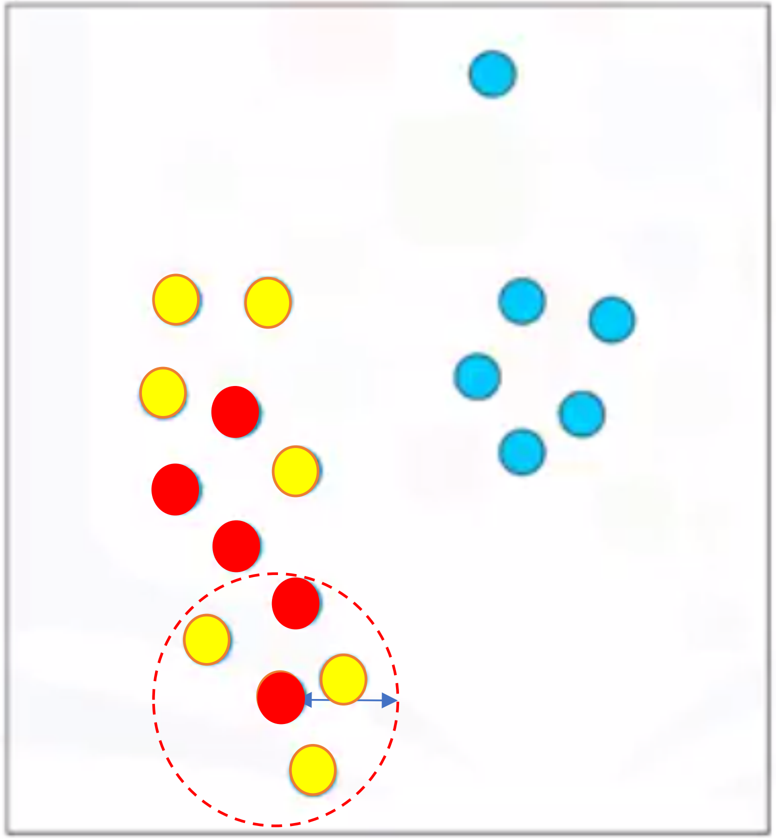
Step 3 :

Repeat step 1 and 2 for all data objects.

Parameter:

R = 2 unit dan M = 5

DBSCAN: Example



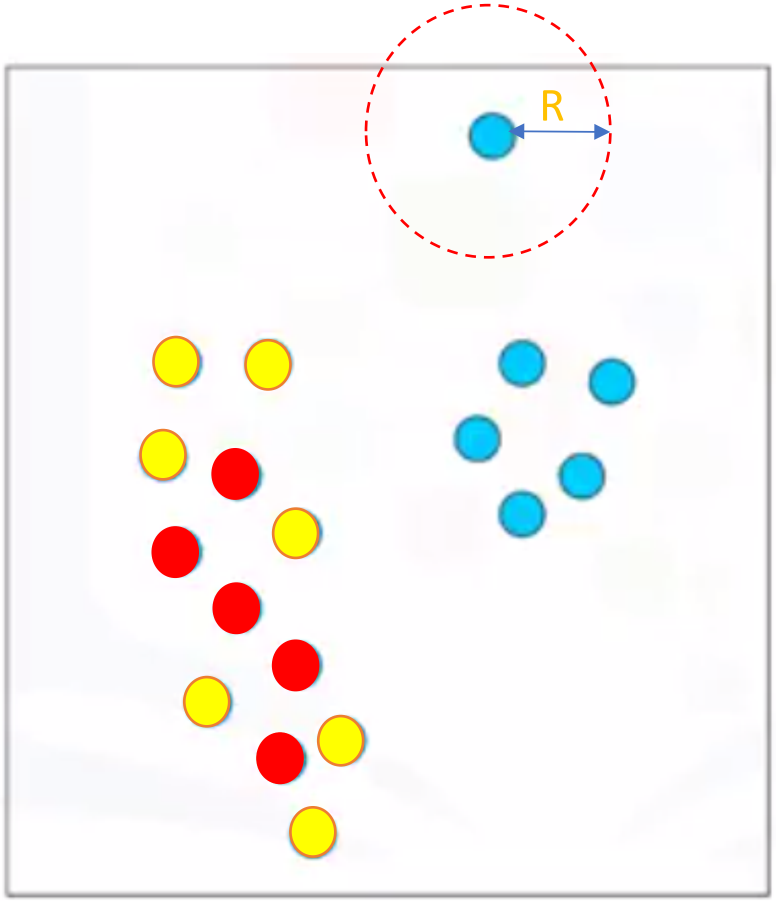
Step 3 :

Repeat step 1 and 2 for all data objects.

Parameter:

R = 2 unit dan M = 5

DBSCAN: Example



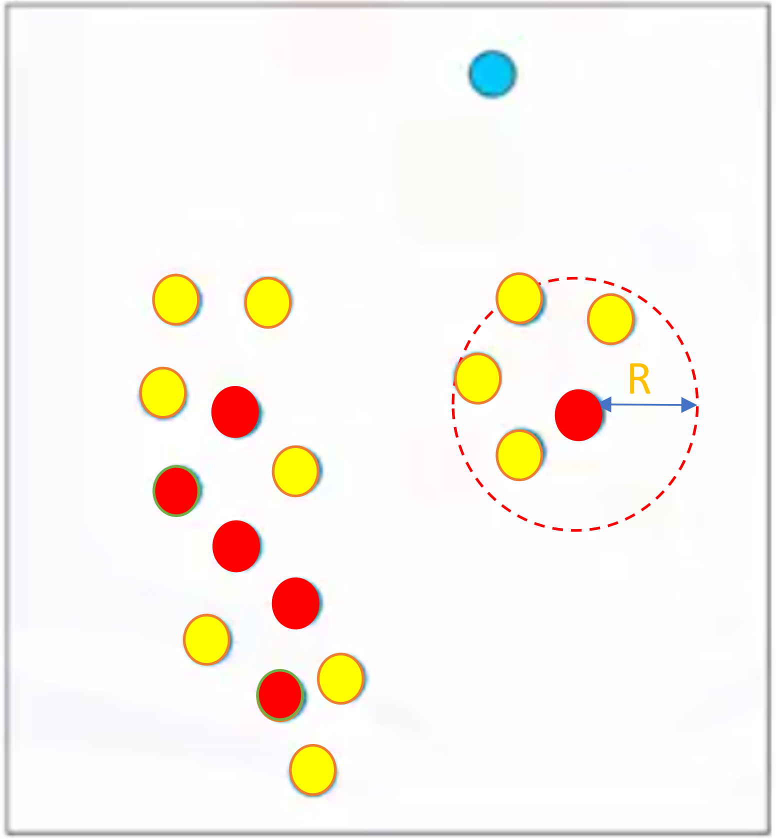
Step 3 :

Repeat step 1 and 2 for all data objects.

Parameter:

R = 2 unit dan M = 5

DBSCAN: Example



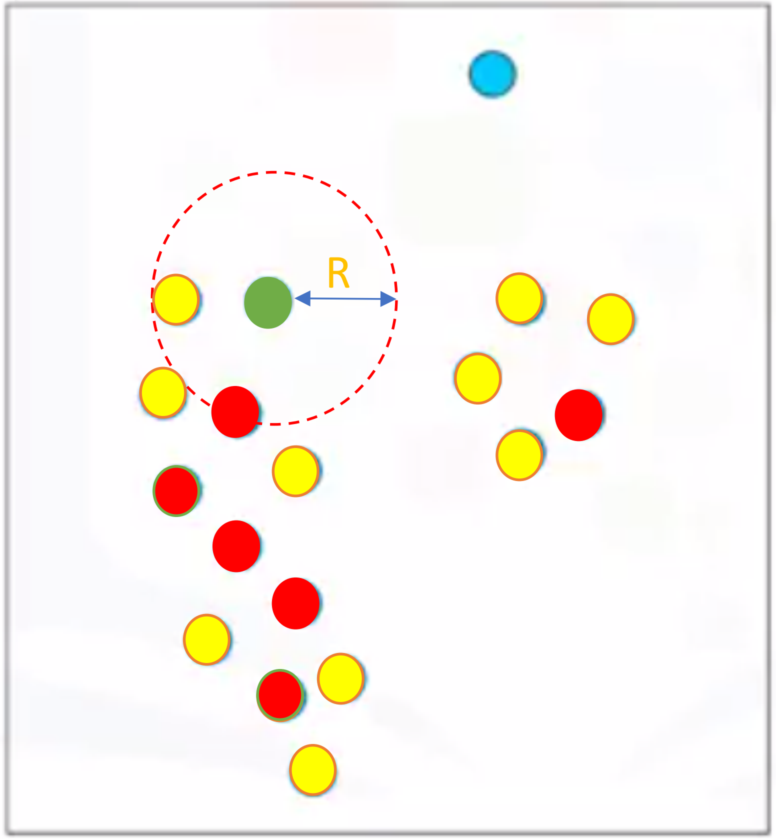
Step 3 :

Repeat step 1 and 2 for all data objects.

Parameter:

R = 2 unit dan M = 5

DBSCAN: Example



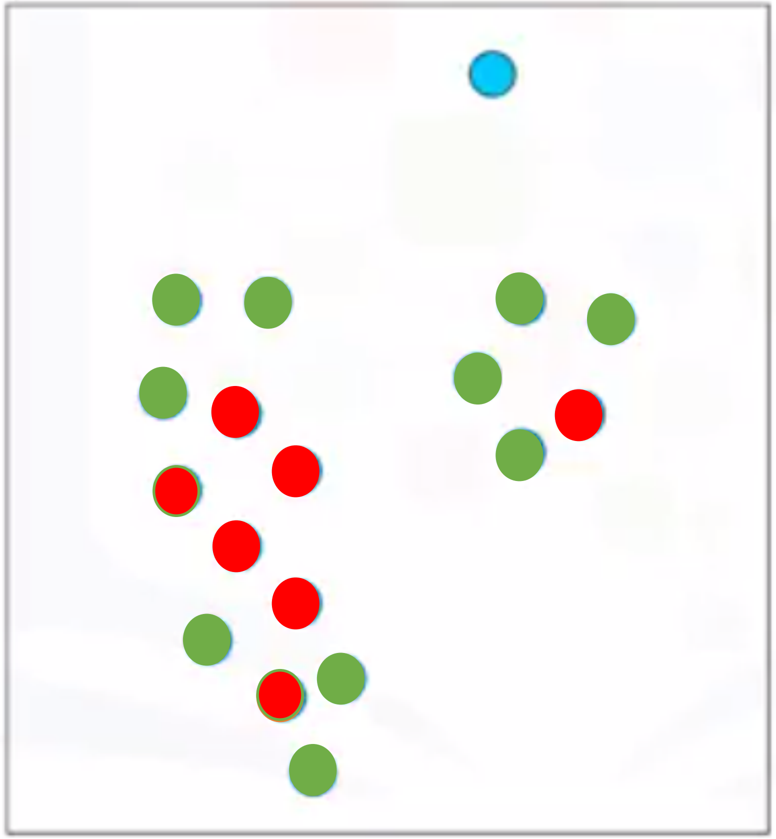
Step 3 :

Repeat step 1 and 2 for all data objects.

Parameter:

R = 2 unit dan M = 5

DBSCAN: Example



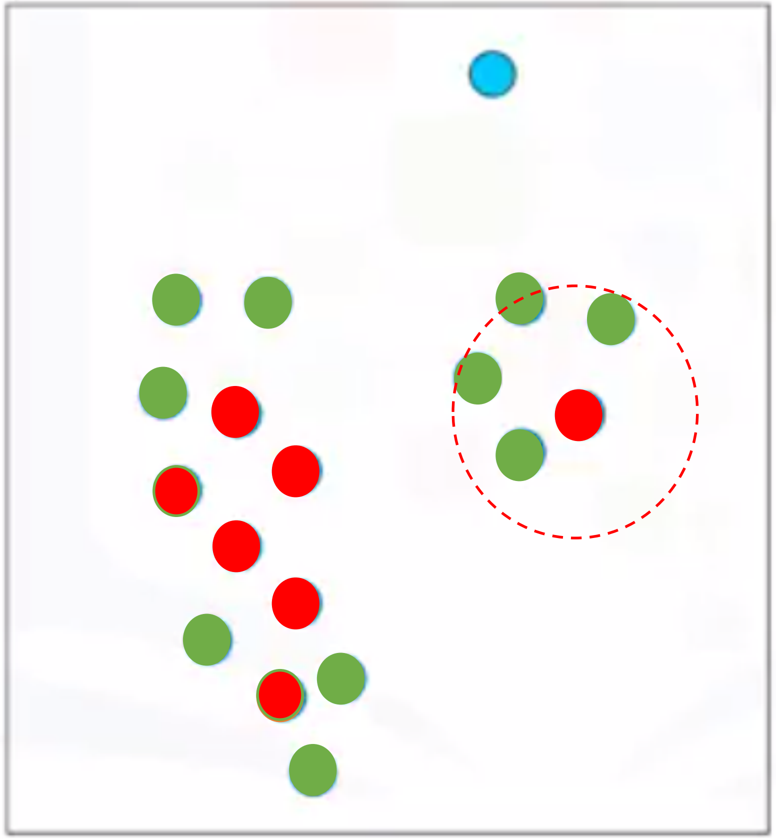
Step 3 :

Repeat step 1 and 2 for all data objects.

Parameter:

R = 2 unit dan M = 5

DBSCAN: Example



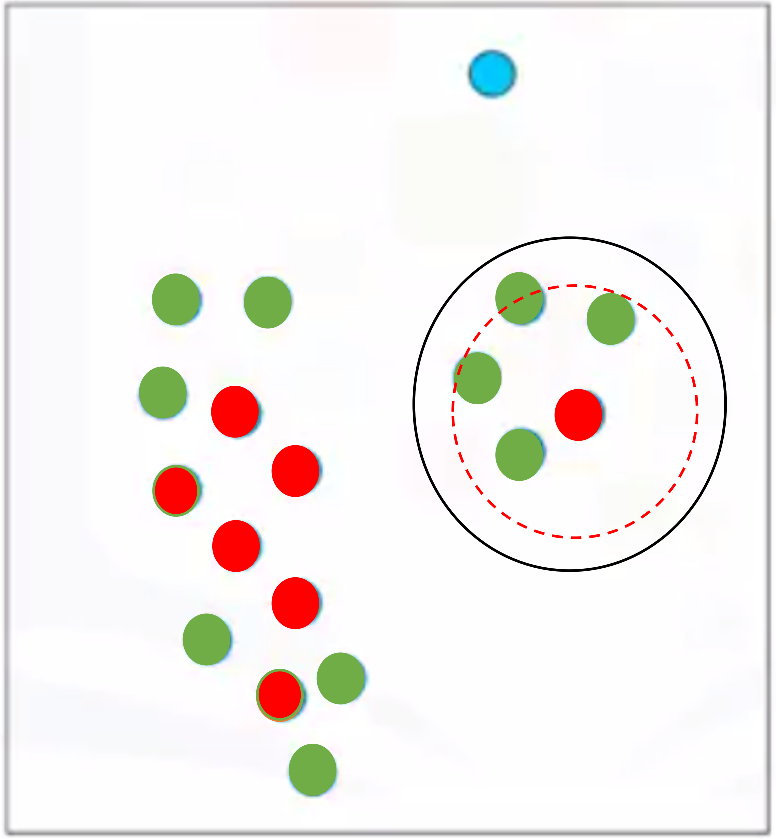
Step 4 :

Define the number of clusters

Parameter:

R = 2 unit dan M = 5

DBSCAN: Example



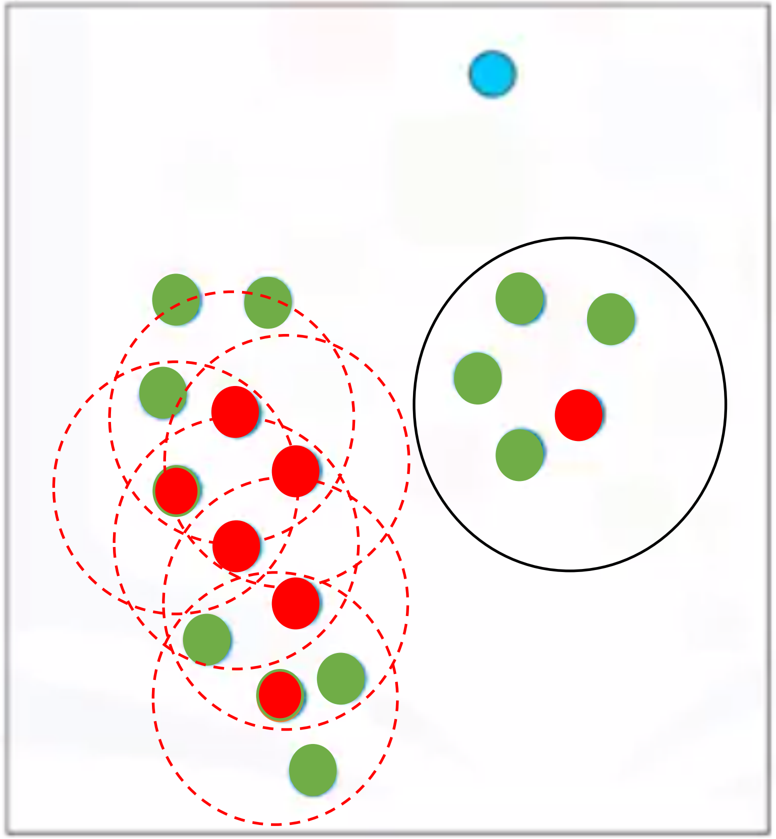
Step 4 :

Define the number of clusters

Parameter:

R = 2 unit dan M = 5

DBSCAN: Example



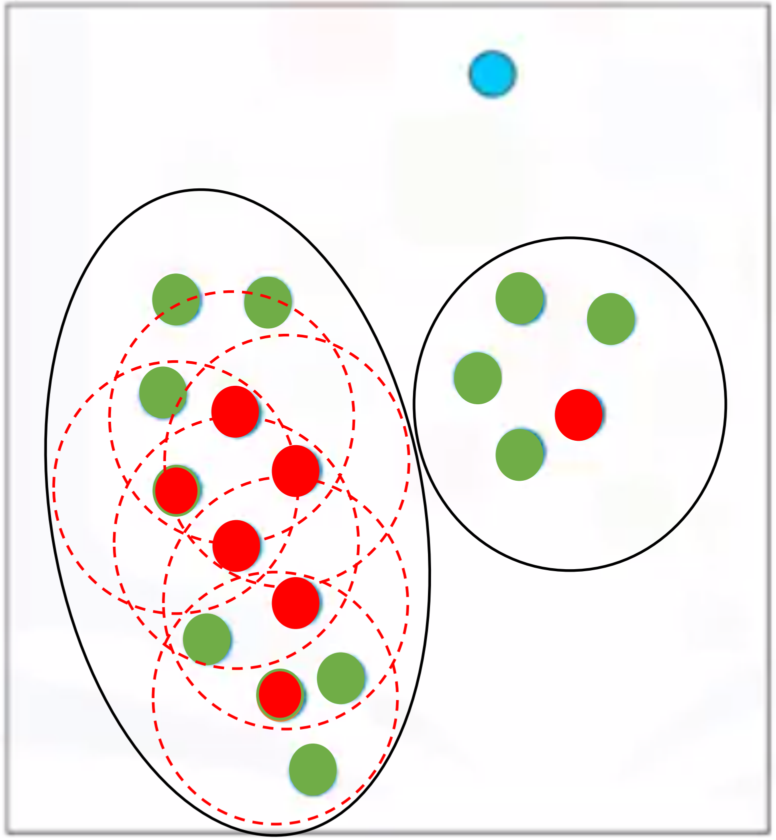
Step 4 :

Define the number of clusters

Parameter:

R = 2 unit dan M = 5

DBSCAN: Example



Step 4 :

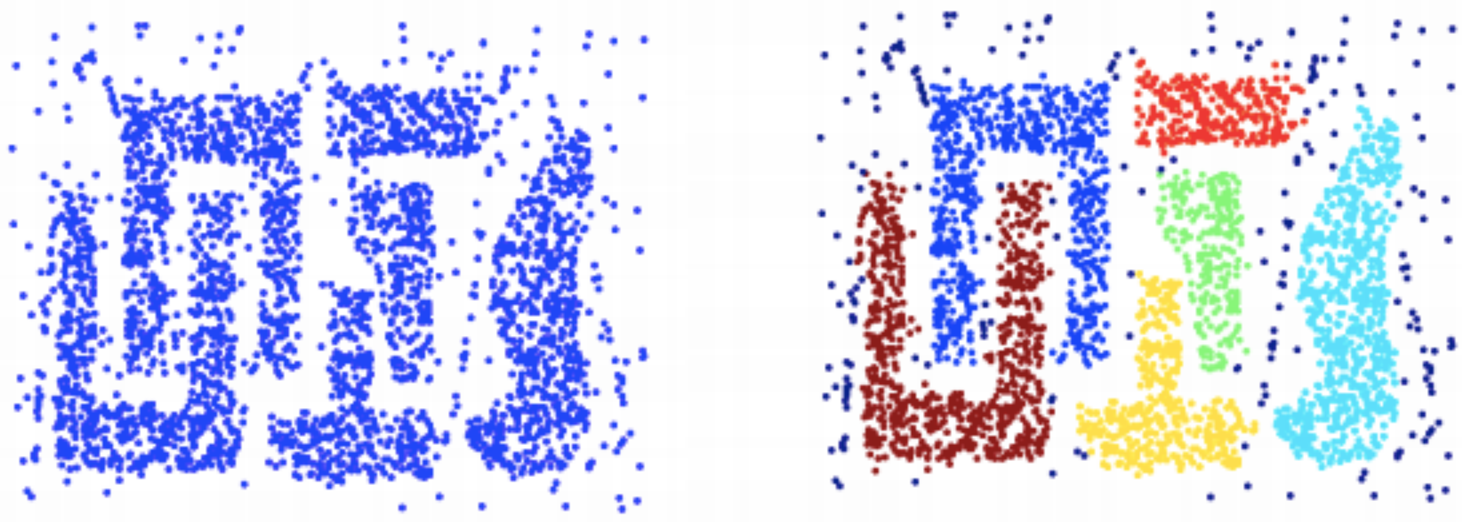
Define the number of clusters

Parameter:

R = 2 unit dan M = 5

When DBSCAN Works Well

- Resistant to Noise
- Can handle clusters of different shapes and sizes

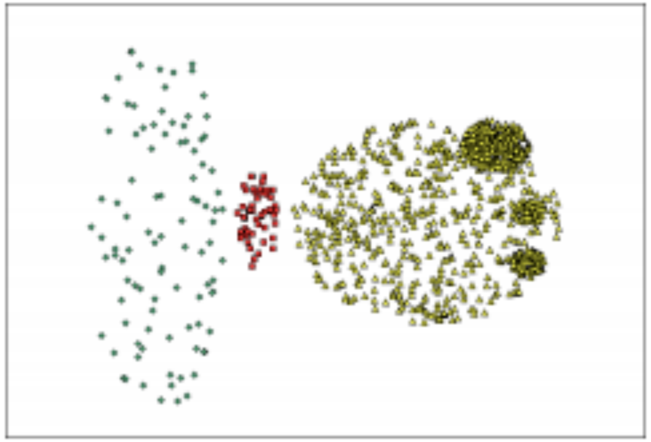


Original Points

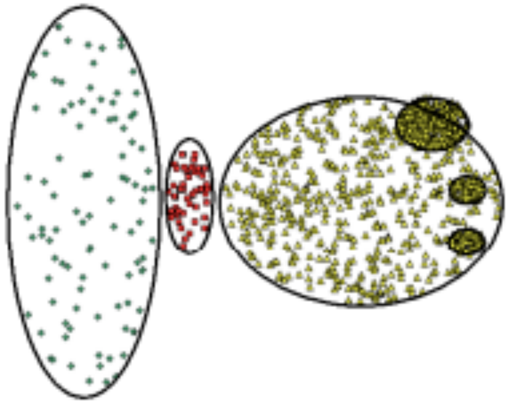
Clusters

When DBSCAN Does Not Work Well

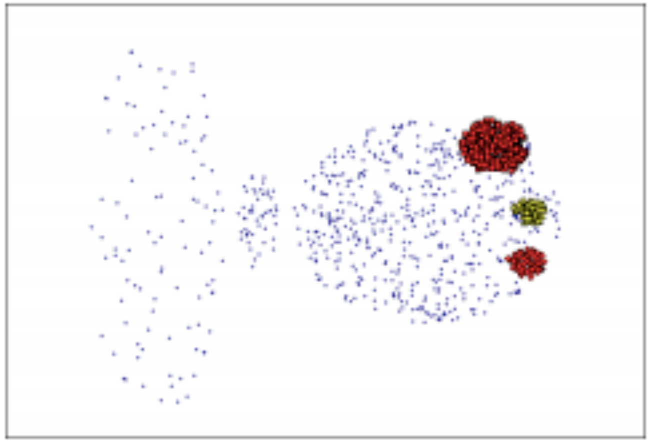
- Cannot handle varying densities.
- Sensitive to parameters—hard to determine the correct set of parameters



(MinPts=4, Eps=9.92).



Original Points



(MinPts=4, Eps=9.75)

Take-away Messages

- The basic idea of density-based clustering
- The two important parameters and the definitions of neighborhood and density in DBSCAN
- Core, border and outlier points
- DBSCAN algorithm
- DBSCAN's pros and cons

References

- https://www.datacamp.com/courses/introduction-to-machine-learning-with-r
- https://cse.buffalo.edu/~jing/cse601/fa12/materials/clustering_density.pdf