Chapter 06

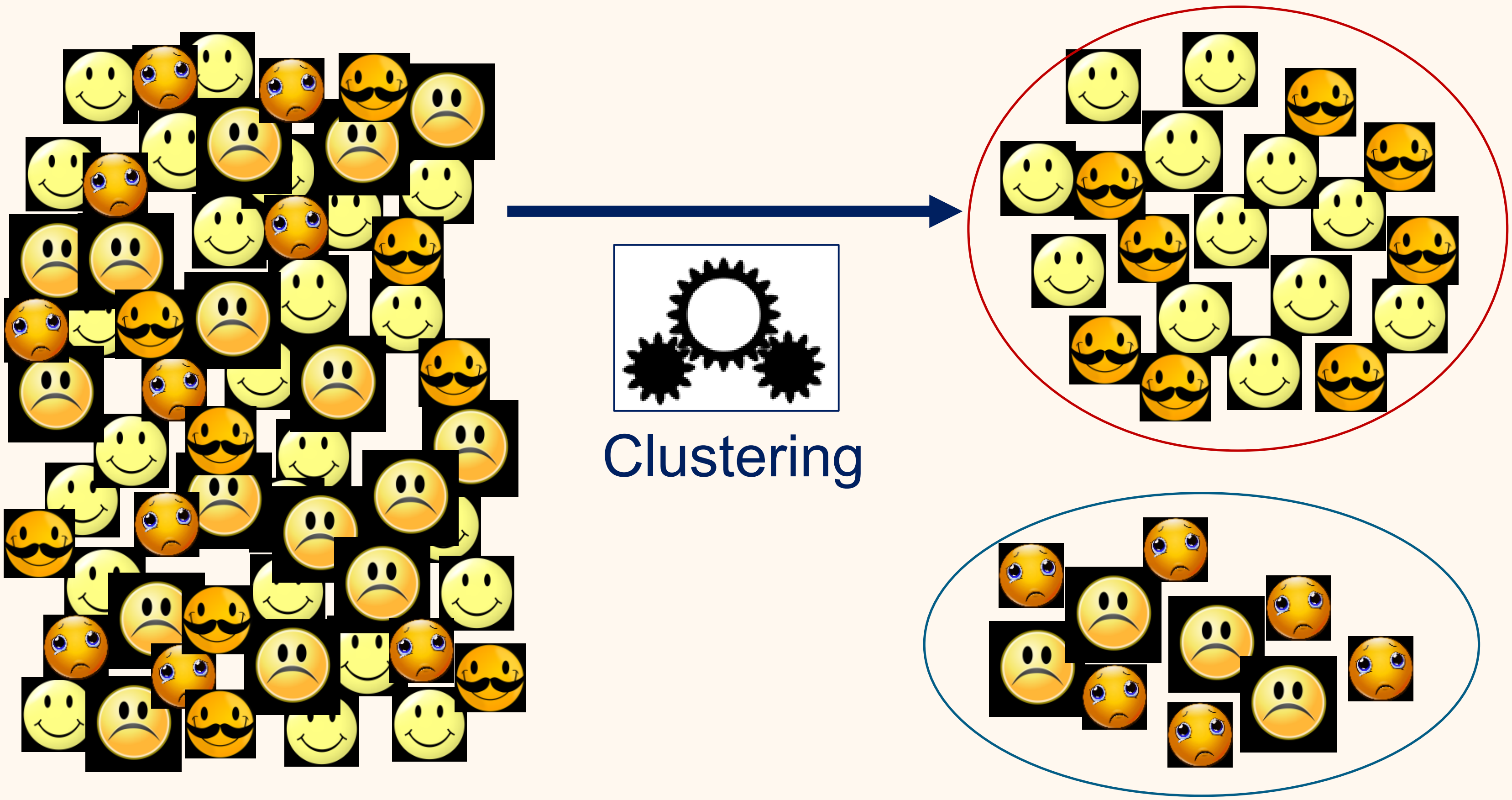
Clustering

Dr. Steffen Herbold
herbold@cs.uni-goettingen.de

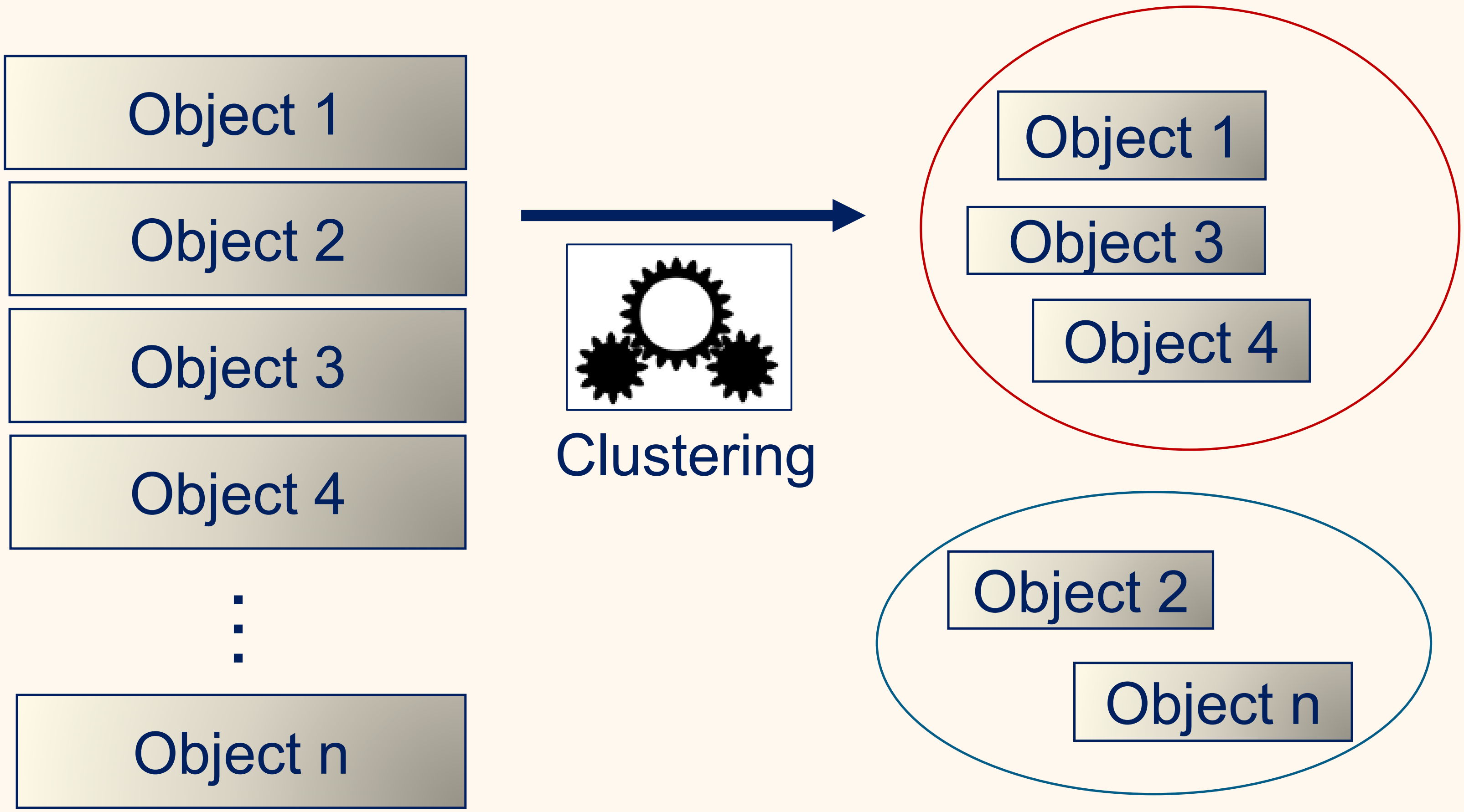
Outline

- Overview
- k-means Clustering
- DBSCAN Clustering
- Comparison of the Clustering Algorithms
- Summary

Example of Clustering



The General Problem



The Formal Problem

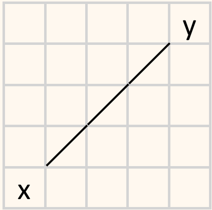
- Object space
  - $O = \{object_1, object_2, \dots\}$
  - Often infinite

How do you measure similarity?

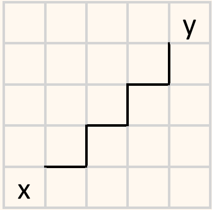
- Representations of the objects in a (numeric) feature spa
  - $\mathcal{F} = \{\phi(o), o \in O\}$
- Clustering
  - Grouping of the objects
  - Objects in the same group $g \in G$ should be similar
  - $c: \mathcal{F} \rightarrow G$

Measuring ~~Similarity~~ Distances

- Small distance = similar
- Eucledian Distance
  - Based the eucledian norm $||x||_2$
  - $d(x, y) = ||y - x||_2 = \sqrt{(y_1 - x_1)^2 + \cdots + (y_n - x_n)^2}$



- Manhatten Distance
  - Based on the Manhatten norm $||x||_1$
  - $d(x, y) = ||y - x||_1 = |y_1 - x_1| + \cdots + |y_n - x_n|$



- Chebyshev Distance
  - Based on the maximum norm $||x||_\infty$
  - $d(x, y) = ||y - x||_\infty = \max_{i=1..n} |y_i - x_i|$

2	2	2	2	2
2	1	1	1	2
2	1	0	1	2
2	1	1	1	2
2	2	2	2	2

Evaluation of Clustering Results

- No general metrics, depends on algorithms
  - Low variance for k-Means
  - High density for DBSCAN
  - Good fit in comparison to model variables for EM clustering
  - …
- Often manual checks
  - Do the clusters make sense?
  - Can be difficult
    - Very large data
    - Many clusters
    - High dimensional data

Outline

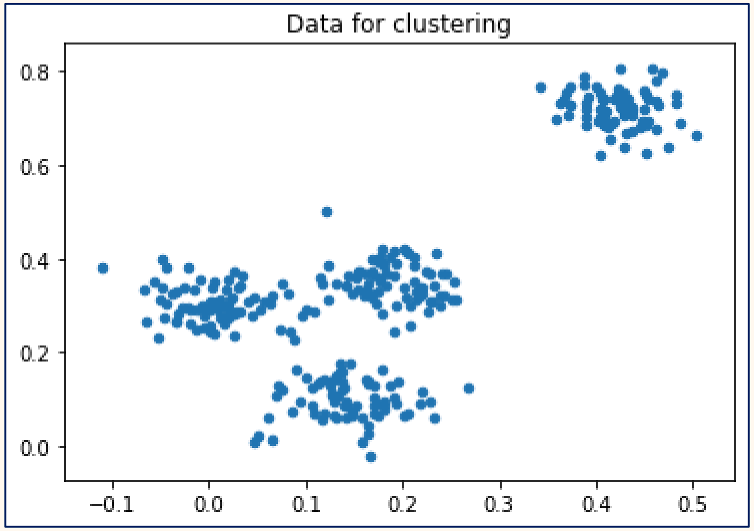
- Overview
- **k-means Clustering**
- DBSCAN Clustering
- Summary

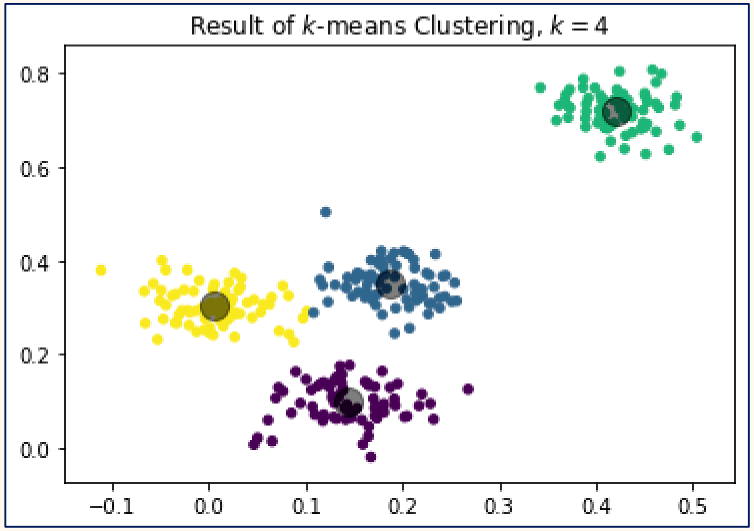
Idea Behind k–means Clustering

- Clusters are described by their center
  - The centers are called *centroid*
  - Centroid-based clustering

How do you get the centroids?

- Objects are assigned to the closests centroid





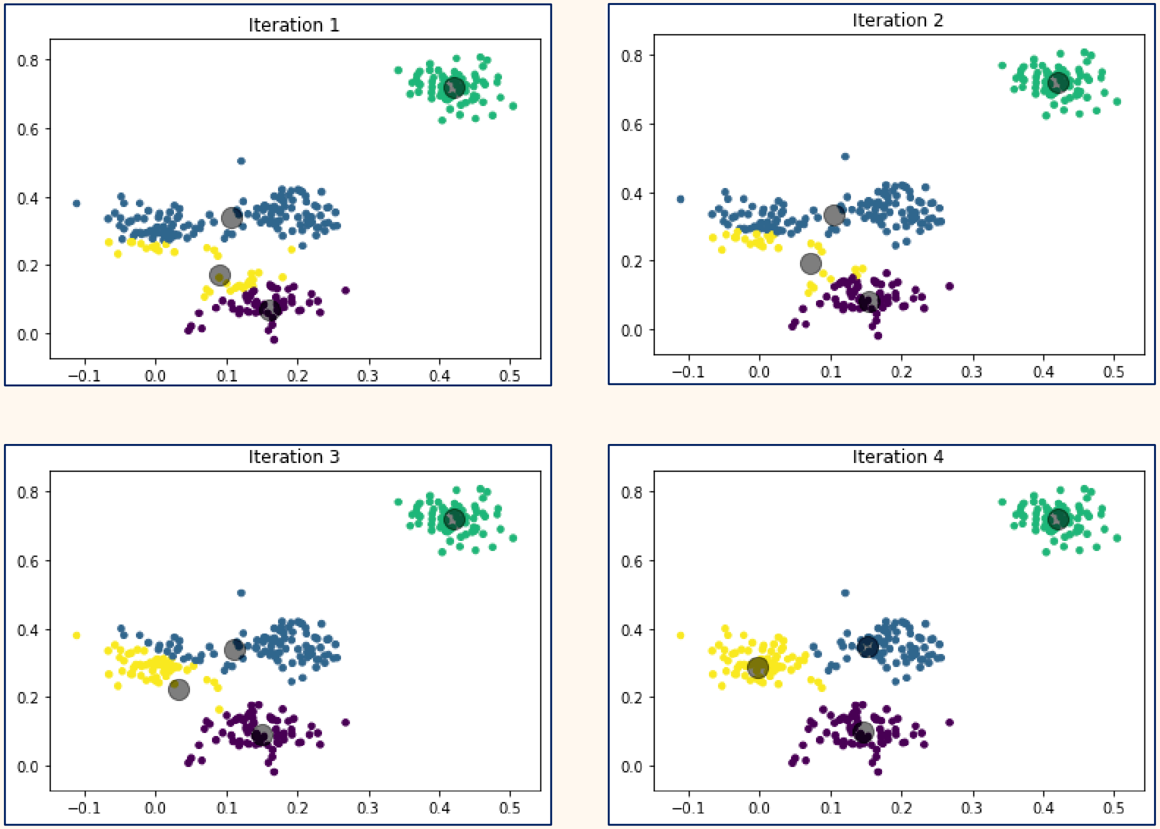
Simple Algorithm

- Select initial centroids $C_1, \ldots, C_k$
  - Randomized

- Assign each object to closest centroid
  - $c(x) = \operatorname{argmin}_{i=1..k} d(x, C_i)$
- Update centroid
  - Arithmetic mean of assigned objects
  - $C_i = \frac{1}{|\{x:c(x)=i\}|} \sum_{x:c(x)=i} x_i$

- Repeat update and assignment
  - Until convergence, or
  - Until maximum number of iterations

Visualization of the k–means Algorithm



Selecting k

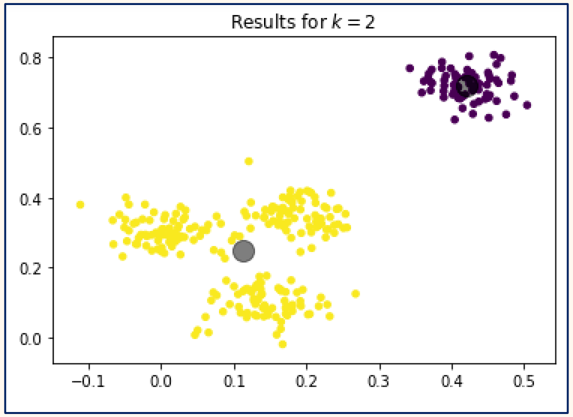
- Intuition and knowledge about data
  - Based on looking at plots
  - Based on domain knowledge

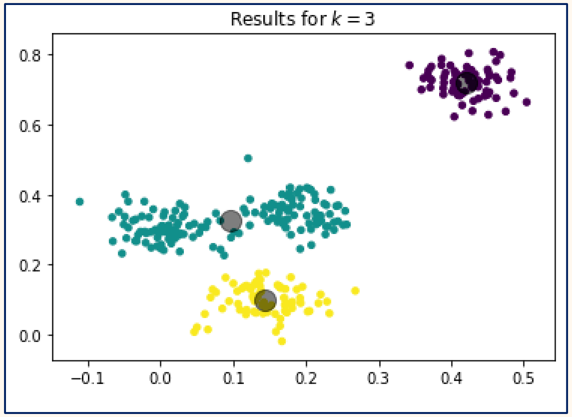
- Due to goal
  - Fixed number of groups desired

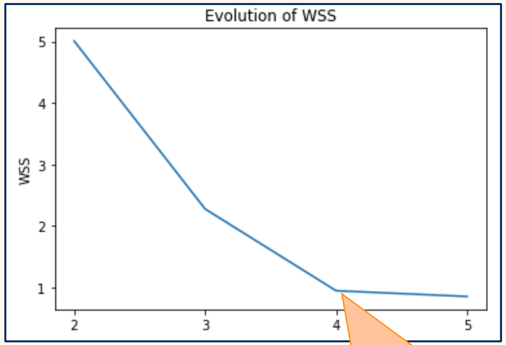
- Based on best fit
  - Within-sum-of-squares
  - $WSS = \sum_{i=1}^{k} \sum_{x:\, c(x)=i} d(x, C_i)^2$

Results for $k = 2, \ldots, 5$

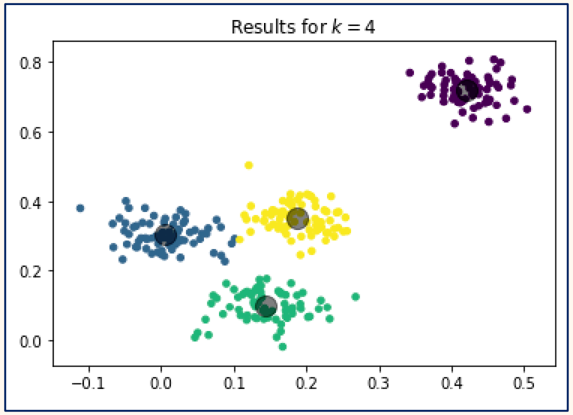
2, 3, and 4 all okay
→ use domain knowledge to decide

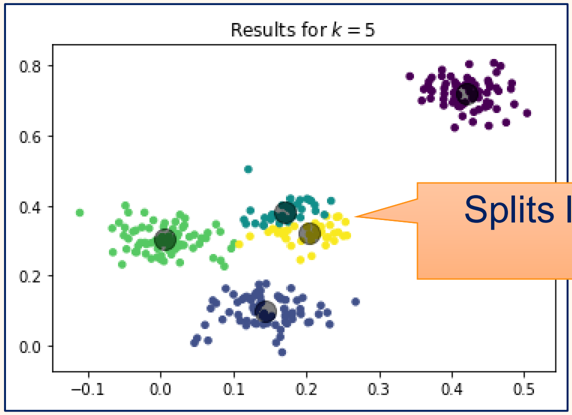






Big changes in slope (elbows) indicate potenially good values for k

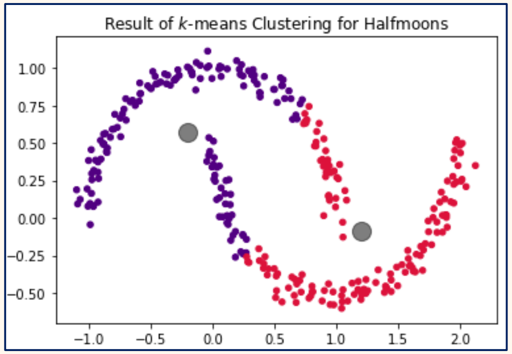




Splits like these indicate too many clusters

Problems of k-Means

- Depends on initial clusters
  - Results may be unstable
- Wrong k can lead to bad results
- All features must have a similar scale
  - Differences in scale introduce artificial weights between features
  - Large scales dominate small scales
- Only works well for “round“ clusters

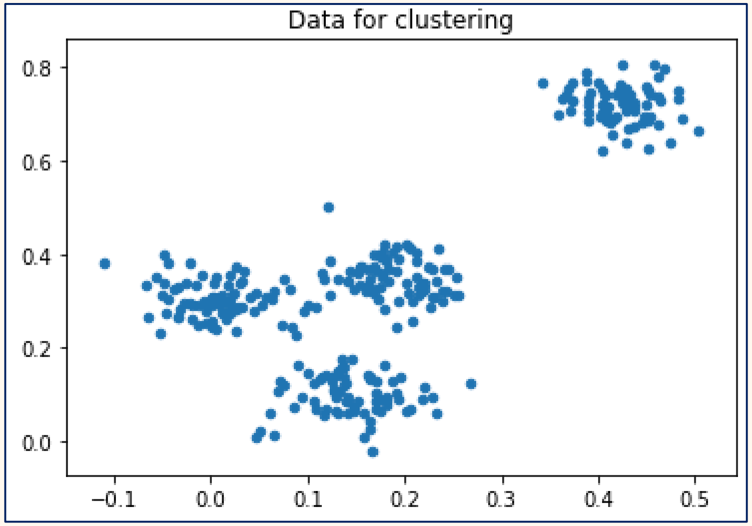


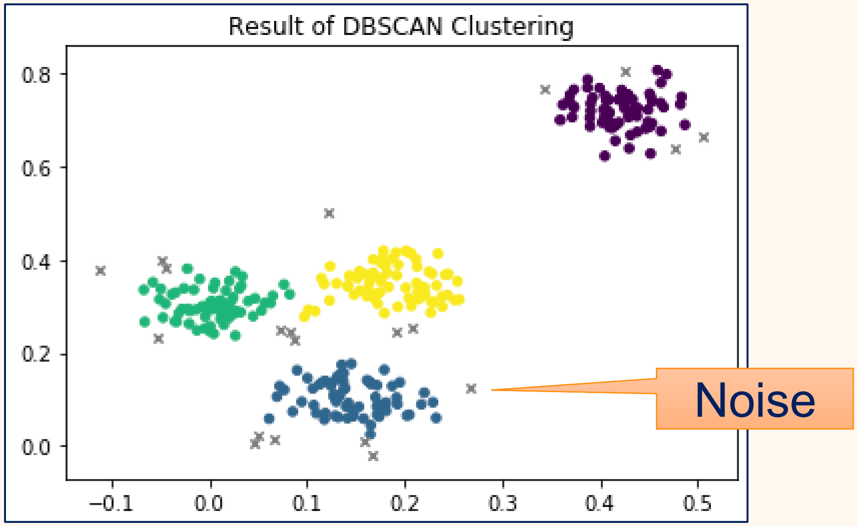
Outline

- Overview
- k-means Clustering
- **DBSCAN Clustering**
- Comparison of the Clustering Algorithms
- Summary

Idea behind DBSCAN

- Clusters are described by other objects close by
  - Density-based clustering
- Scan area around an object for other objects
  - If objects are found, they probably belong to the same group
  - If no objects are found, the object is probably noise

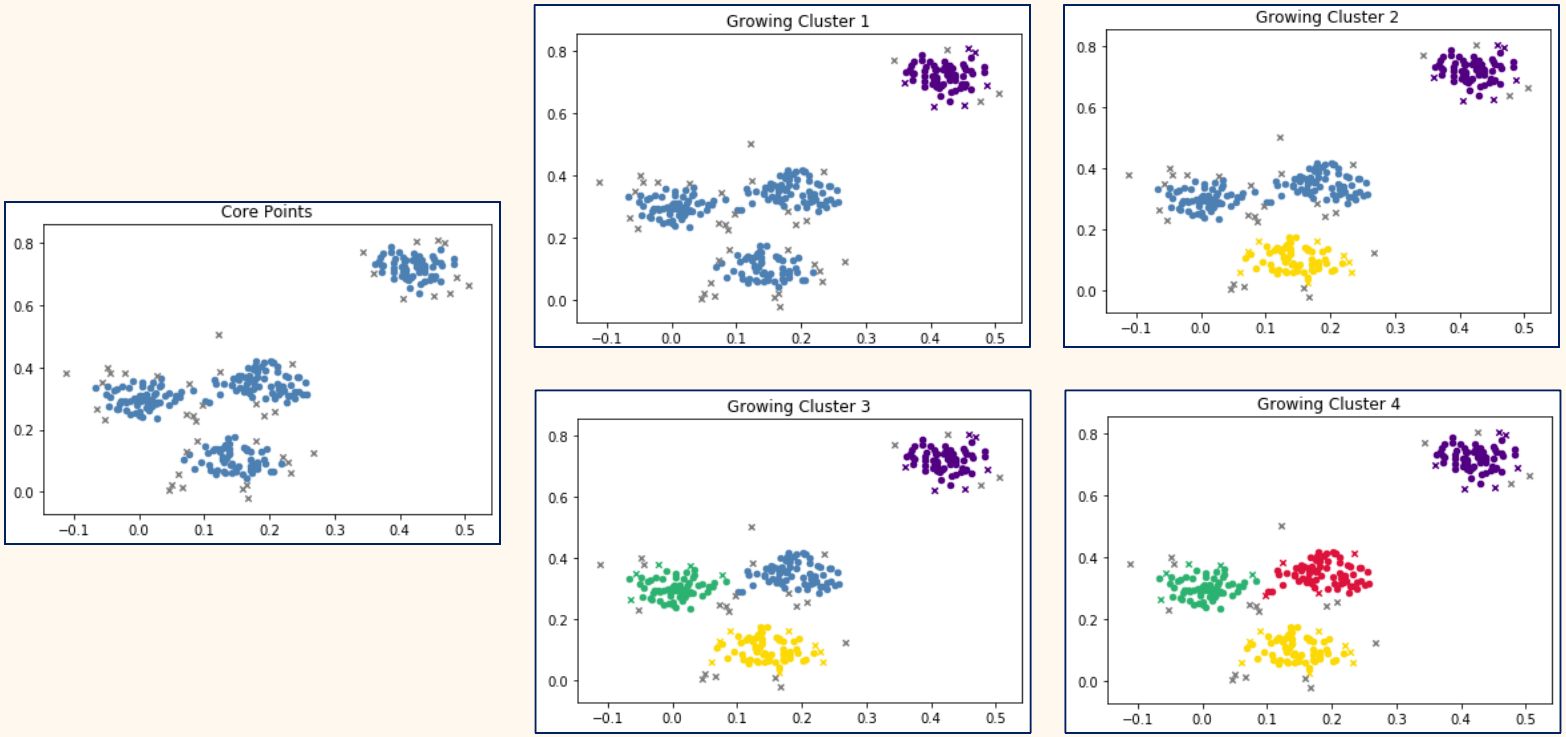




(Relatively) Simple Algorithm

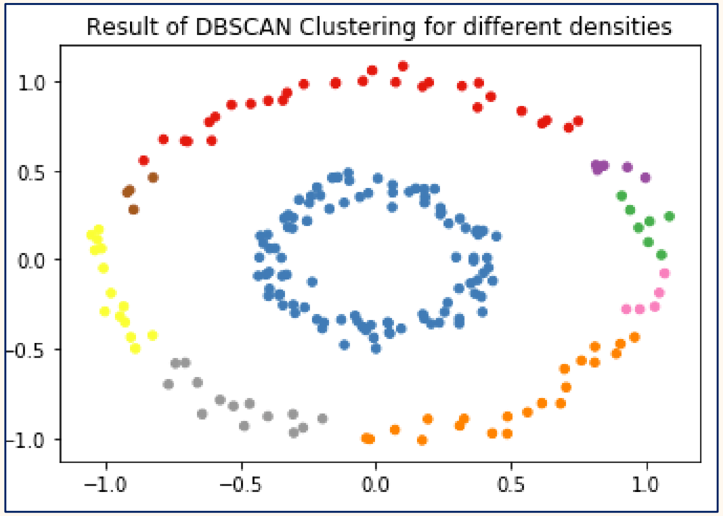
- Two parameters
  - Neighborhood size ϵ
  - Minimal number of points to be considered dense $minPts$
- Determine all objects with dense neighborhoods (core points)
  - $x \in X \textit{ such that } |\{x' \in X: d(x, x') \leq \epsilon\}| \geq minPts$
- Grow clusters by assigning all points that share a neighborhood to the same cluster
- All points that are neither core points nor in the neighborhood of a core point are noise

Visualization of the DBSCAN Algorithm



Problems of DBSCAN

- All features must be in the same range
- What if different clusters have different densities?
  → Main problem of DBSCAN!

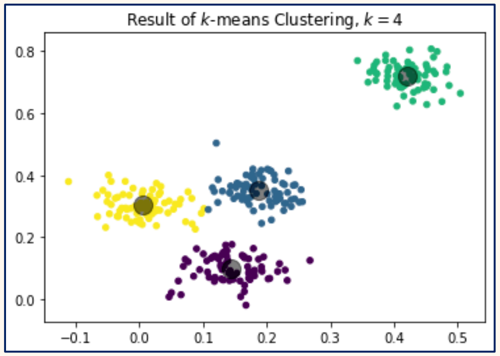


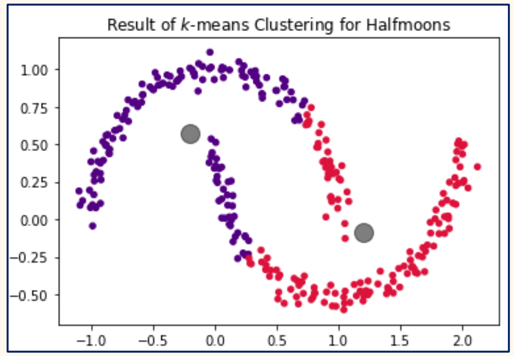
- This is also related to the size of the data
  → DBSCAN is very sensitive to sampling

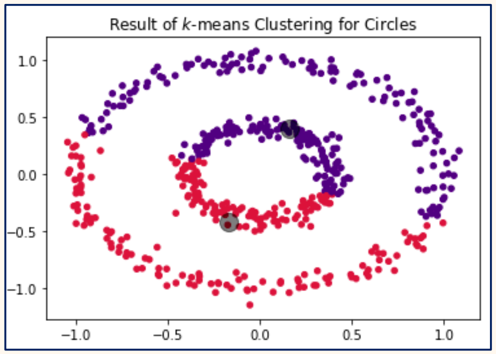
Outline

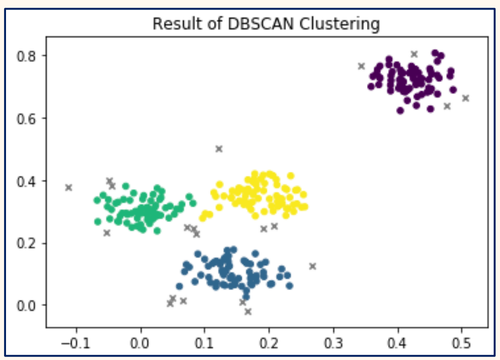
- Overview
- k-means Clustering
- DBSCAN Clustering
- **Comparison of the Clustering Algorithms**
- Summary

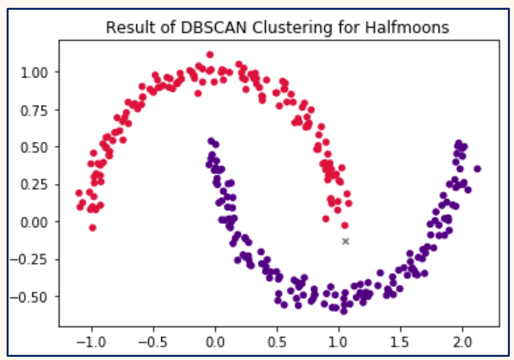
Comparison of Clusters

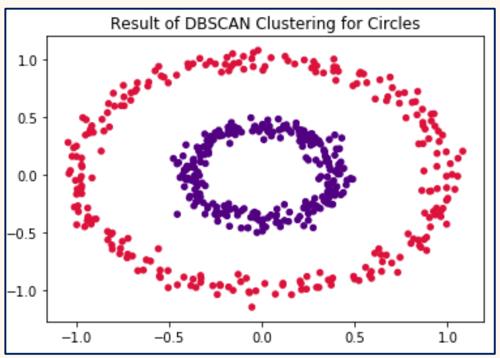






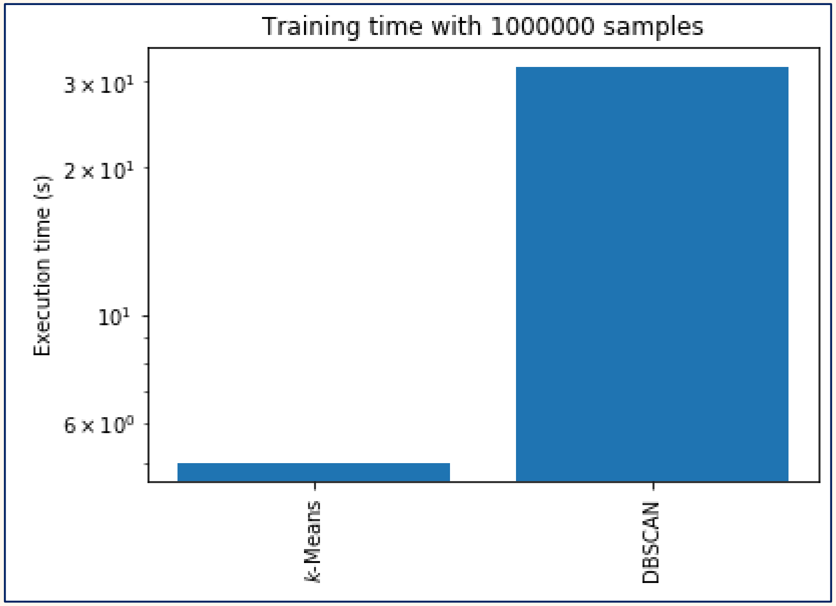






Comparison of Execution Times

- Both very fast for smaller data sets
- For larger data sets:



Strengths and Weaknesses

	Cluster number	Explanatory value	Consise representation	Categorical features	Missing features	Correlated features
k-means	-	+	+	-	-	0
DBSCAN	+	-	-	-	-	0

- Other important types of algorithms can resolve some problems
  - Connectivity-based clustering: creation of tree structures (single linkage clustering, …)
  - Distribution-based clustering: inference of normal distributions (EM clustering, …)
  - Clustering designed for categorical data (k-Modes, …)

Summary

- Clustering considers the inference of groups for objects
- Works well for numeric data but is often not well suited for categorical data
- Different types of clustering algorithms
  - Covered centroid-based and density-based clustering
- Scales are very important for most clustering algorithms
- Evaluation often difficult and requires manual intervention