Aalto University
School of Electrical Engineering

CS-C3240 – Machine Learning D

Clustering

Stephan Sigg

Department of Communications and Networking
Aalto University, School of Electrical Engineering
stephan.sigg@aalto.fi

Version 1.0, February 9, 2022

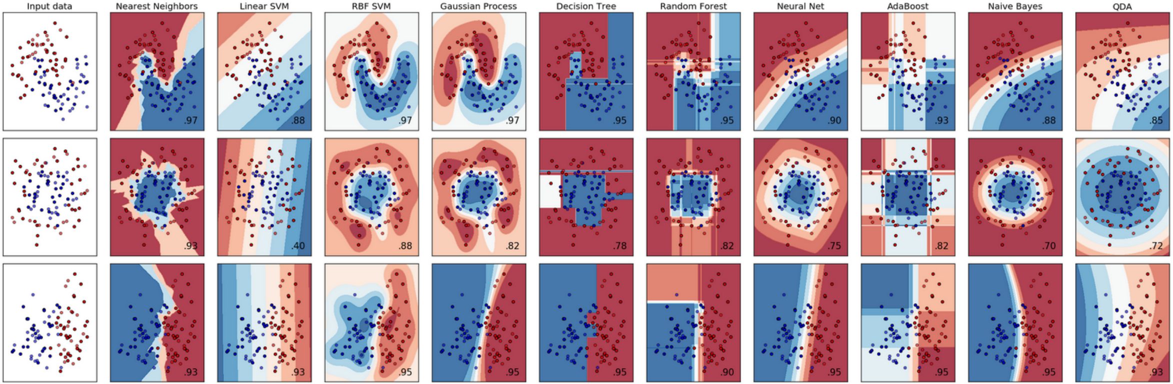
Learning goals

Understand the concepts of

- unsupervised learning
- clustering
- k-means
- DBSCAN

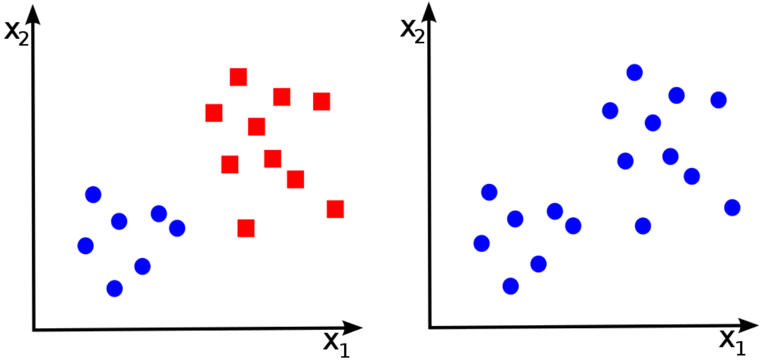
Outline

Summary supervised classification algorithms



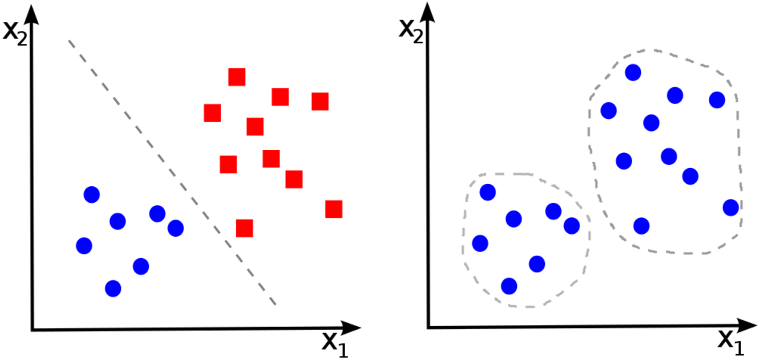
QDA: Quadratic Discriminant Analysis
AdaBoost: combine 'weak learners'; subsequent learners trained in favor of previous misclassified instances
RBF: Radial Basis Function

Unsupervised learning



Supervised: $\{(x_{1,1}, x_{1,2}) \rightarrow y_1, (x_{2,1}, x_{2,2}) \rightarrow y_2, \ldots, (x_{n,1}, x_{n,2}) \rightarrow y_n\}$
Unsupervised: $\{(x_{1,1}, x_{1,2}), (x_{2,1}, x_{2,2}), \ldots, (x_{n,1}, x_{n,2})\}$

Unsupervised learning



Supervised: $\{(x_{1,1}, x_{1,2}) \rightarrow y_1, (x_{2,1}, x_{2,2}) \rightarrow y_2, \ldots, (x_{n,1}, x_{n,2}) \rightarrow y_n\}$
Unsupervised: $\{(x_{1,1}, x_{1,2}), (x_{2,1}, x_{2,2}), \ldots, (x_{n,1}, x_{n,2})\}$

Outline

Introduction

k-means

DBSCAN

Gaussian Mixture Models

Unsupervised learning

k-means algorithm

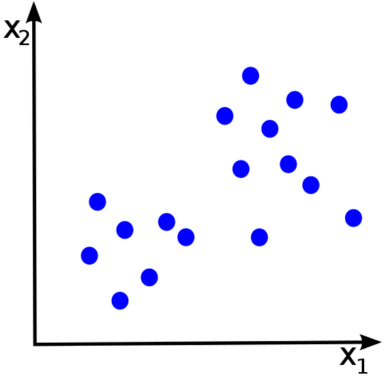
k-means algorithm

Iteratively find k clusters in the data

Init Randomly choose k points as initial cluster centroids

Repeat :

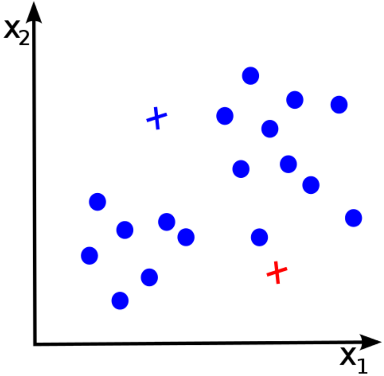
- $\rightarrow$ Assign data points $x_i, i \in \{1..n\}$ to these cluster centroids conditioned on distance: $\mathcal{C}_j = \{x_i | c_j \text{ is nearest centroid to } x_i\}$
- $\rightarrow$ Move cluster centroids to the center weight of the points associated to them



Unsupervised learning

k-means algorithm

Init: k cluster centroids c_i chosen randomly



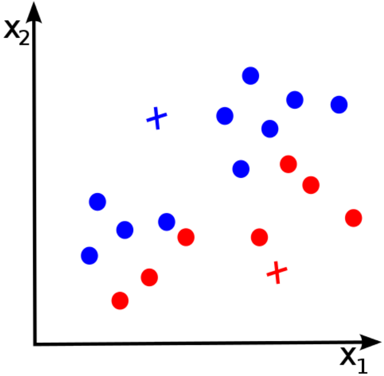
Unsupervised learning

k-means algorithm

Init: k cluster centroids c_i chosen randomly

Repeat:

1: assign data points x_i to centroids $\mathcal{C}_j$ conditioned on distance



Unsupervised learning

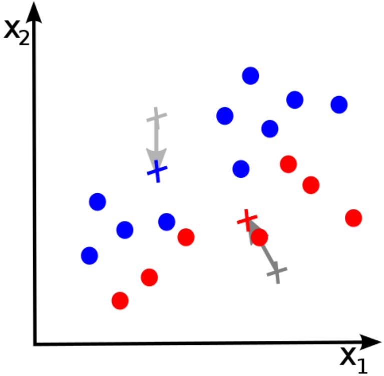
k-means algorithm

Init: k cluster centroids c_i chosen randomly

Repeat:

1: assign data points x_i to centroids $\mathcal{C}_j$ conditioned on distance

2: $c_j(t+1) = \frac{1}{|\mathcal{C}_j|} \sum_{i=1}^{|\mathcal{C}_j|} x_i$



Unsupervised learning

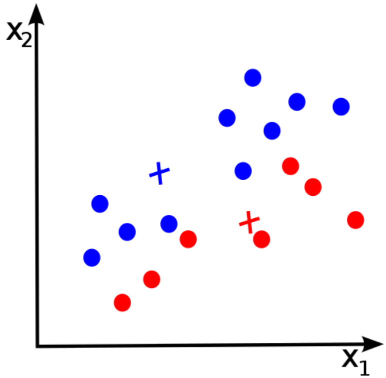
k-means algorithm

Init: k cluster centroids c_i chosen randomly

Repeat:

1: assign data points x_i to centroids $\mathcal{C}_j$ conditioned on distance

2: $c_j(t+1) = \frac{1}{|\mathcal{C}_j|}\sum_{i=1}^{|\mathcal{C}_j|} x_i$



Unsupervised learning

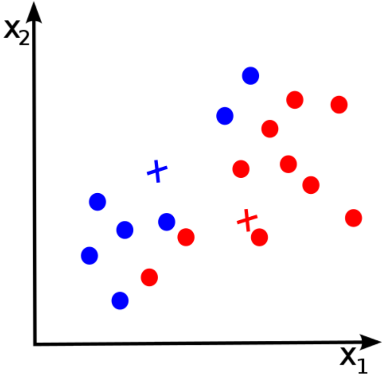
k-means algorithm

Init: k cluster centroids c_i chosen randomly

Repeat:

1: assign data points x_i to centroids $\mathcal{C}_j$ conditioned on distance

2: $c_j(t+1) = \frac{1}{|\mathcal{C}_j|} \sum_{i=1}^{|\mathcal{C}_j|} x_i$



Unsupervised learning

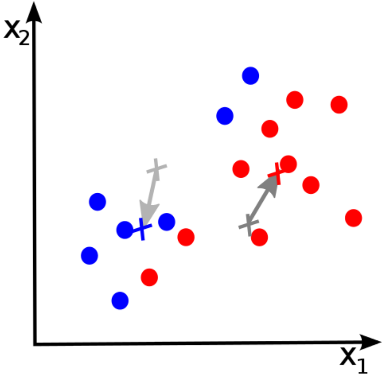
k-means algorithm

Init: k cluster centroids c_i chosen randomly

Repeat:

1: assign data points x_i to centroids $\mathcal{C}_j$ conditioned on distance

2: $c_j(t+1) = \frac{1}{|\mathcal{C}_j|} \sum_{i=1}^{|\mathcal{C}_j|} x_i$



Unsupervised learning

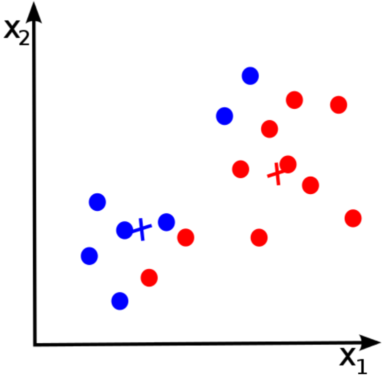
k-means algorithm

Init: k cluster centroids c_i chosen randomly

Repeat:

1: assign data points x_i to centroids $\mathcal{C}_j$ conditioned on distance

2: $c_j(t+1) = \frac{1}{|\mathcal{C}_j|} \sum_{i=1}^{|\mathcal{C}_j|} x_i$



Unsupervised learning

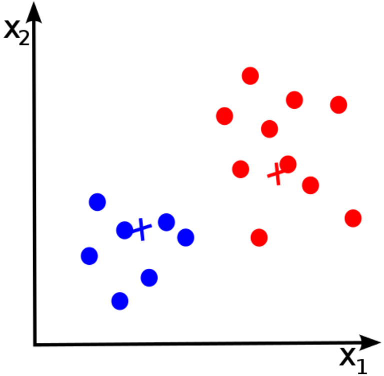
k-means algorithm

Init: k cluster centroids c_i chosen randomly

Repeat:

1: assign data points x_i to centroids $\mathcal{C}_j$ conditioned on distance

2: $c_j(t+1) = \frac{1}{|\mathcal{C}_j|} \sum_{i=1}^{|\mathcal{C}_j|} x_i$



How to randomly initialise the k-means algorithm

The k-means algorithm may compute different solutions for different initial choice of cluster centroids

With respect to the overall distance of the samples to their cluster centroids, k-means might run into local optima

Unsupervised learning

k-means algorithm

How to randomly initialise the k-means algorithm

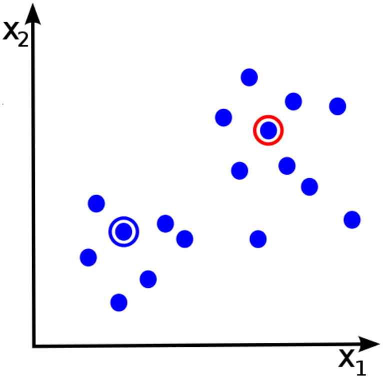
- The k-means algorithm may compute different solutions for different initial choice of cluster centroids
- With respect to the overall distance of the samples to their cluster centroids, k-means might run into local optima

Common choice of the initial k cluster centroids

Choose the initial k cluster centroids randomly from the set of training samples

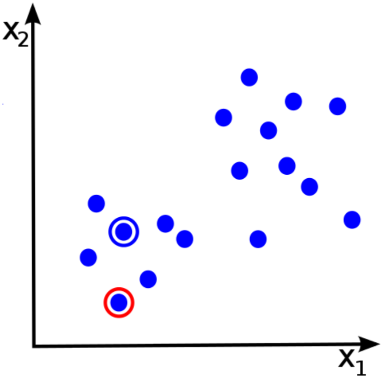
Unsupervised learning

k-means algorithm



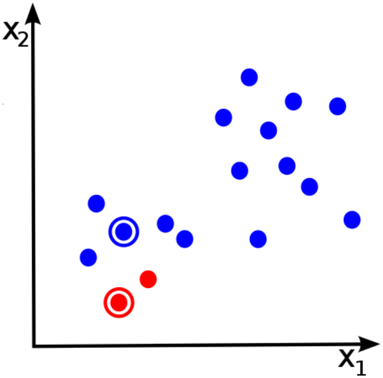
Unsupervised learning

k-means algorithm



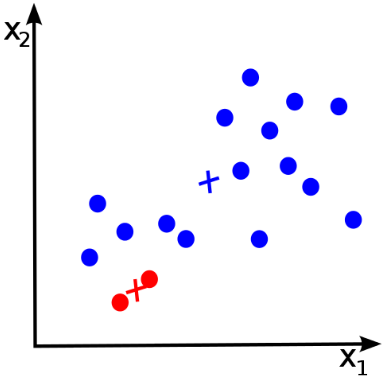
Unsupervised learning

k-means algorithm



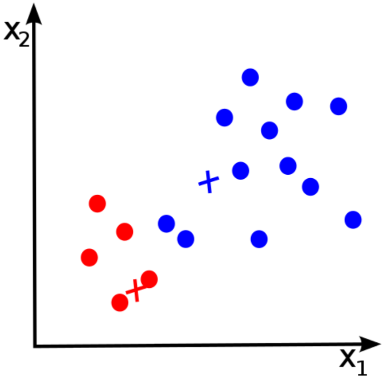
Unsupervised learning

k-means algorithm



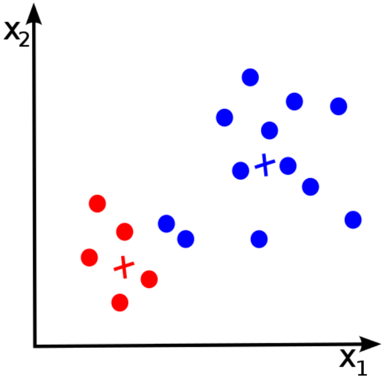
Unsupervised learning

k-means algorithm



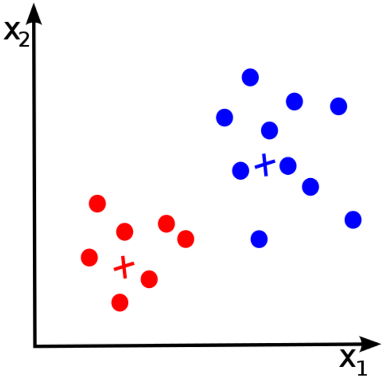
Unsupervised learning

k-means algorithm

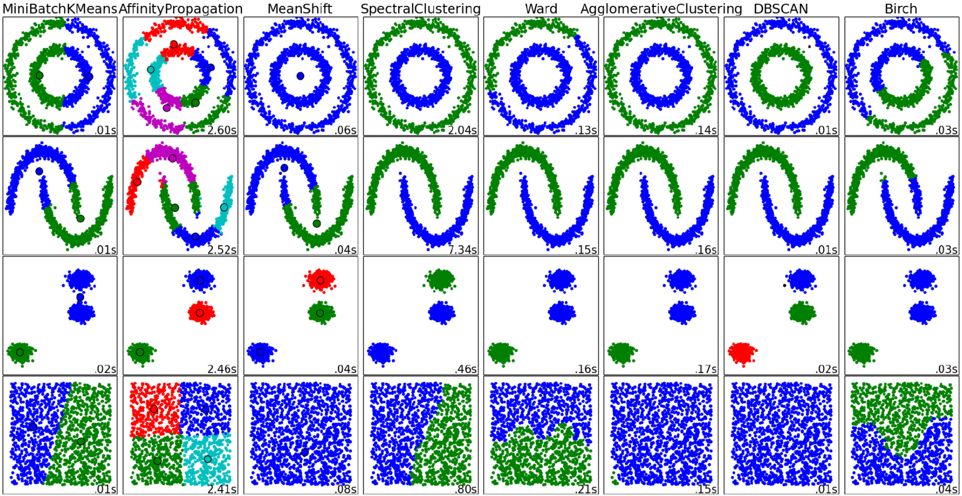


Unsupervised learning

k-means algorithm



Overview clustering algorithms



Outline

Introduction

k-means

DBSCAN

Gaussian Mixture Models

Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

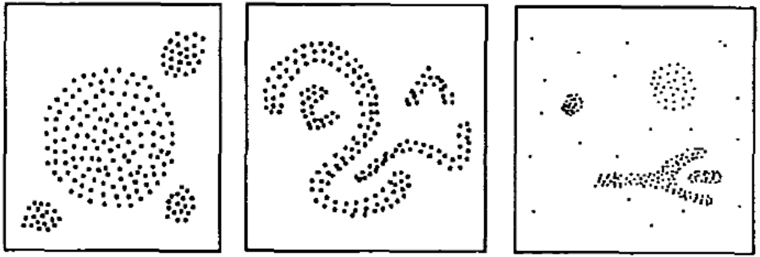
Requirements for a clustering algorithm

- Minimal required domain knowledge
- Discovery of clusters with arbitrary shape
- Good efficiency on large data sets

DBSCAN

Define cluster:

Each cluster has a typical density of points which is considerably higher than outside of the cluster[1]

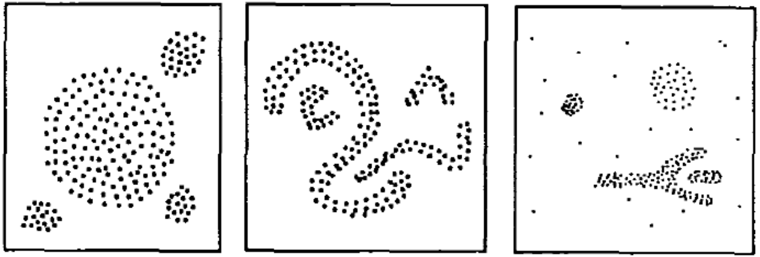


[1] Ester et al.: A Density-Based Algorithms for Discovering Clusters, AAAI

DBSCAN

Define cluster:

Each cluster has a typical density of points which is considerably higher than outside of the cluster[1]

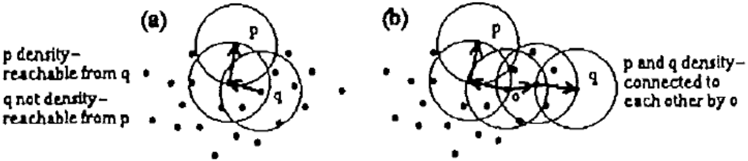


Concept: Density in the neighbourhood has to exceed some threshold such that a point is considered inside a cluster

[1] Ester et al.: A Density-Based Algorithms for Discovering Clusters, AAAI

Density-reachable

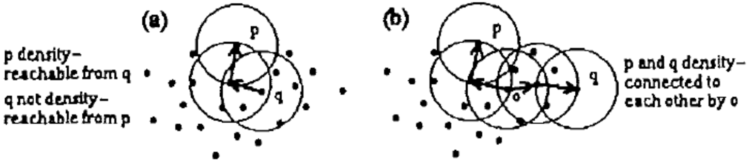
Given a Neighbourhood $\mathcal{N}(x)$ and a density function $\mathcal{D}(\mathcal{N}(x))$, a point p_1 is *density-reachable* from a point p_n if there is a chain of points $p_1, \ldots, p_n$ such that $p_{i+1} \in \mathcal{N}(p_i)$ and $\mathcal{D}(\mathcal{N}(p_i))$ exceeds a certain threshold τ [2]



[2] Ester et al.: A Density-Based Algorithms for Discovering Clusters, AAAI

Density-reachable

Given a Neighbourhood $\mathcal{N}(x)$ and a density function $\mathcal{D}(\mathcal{N}(x))$, a point p_1 is *density-reachable* from a point p_n if there is a chain of points $p_1, \dots, p_n$ such that $p_{i+1} \in \mathcal{N}(p_i)$ and $\mathcal{D}(\mathcal{N}(p_i))$ exceeds a certain threshold τ [2]



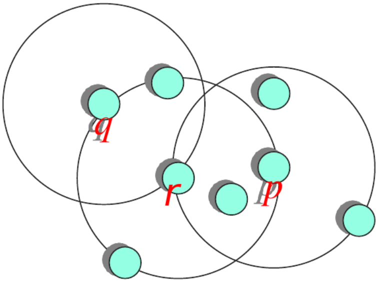
Density-connected

Two points p_1 and p_n are *density-connected*, if there is a point $r \in \mathcal{C}$ such that both p_1 and p_n are density-reachable from r

[2] Ester et al.: A Density-Based Algorithms for Discovering Clusters, AAAI

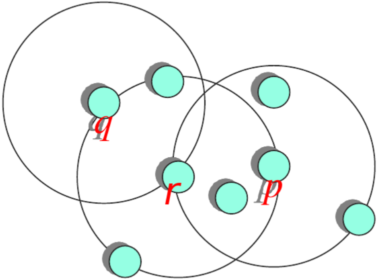
DBSCAN – examples

Density-reachable and density-connected



DBSCAN – examples

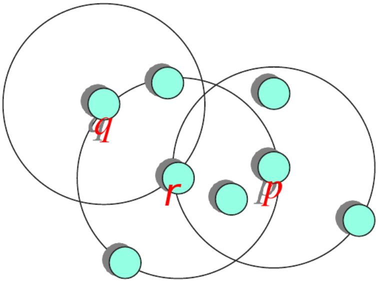
Density-reachable and density-connected



- q is density-reachable from p

DBSCAN – examples

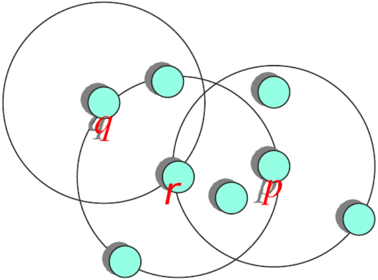
Density-reachable and density-connected



- q is density-reachable from p
- p is not density-reachable from q (low density around q)

DBSCAN – examples

Density-reachable and density-connected



- *q* is density-reachable from *p*
- *p* is not density-reachable from *q* (low density around *q*)
- *q* and *p* are density-connected via *r*

DBSCAN

DBSCAN algorithm

1. Start with an arbitrary point p
2. Retrieve all points density-reachable from p
   - If p is an inner point, this procedure yields a cluster
   - If p is a border point, no points are density-reachable from p - visit the next point in the data.

DBSCAN

DBSCAN algorithm

1. Start with an arbitrary point p
2. Retrieve all points density-reachable from p
   - If p is an inner point, this procedure yields a cluster
   - If p is a border point, no points are density-reachable from p - visit the next point in the data.

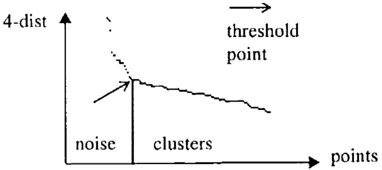
Need for recalculation with lower density for found clusters

Since the density Δ has to be chosen beforehand, it might happen that two clusters $\mathcal{C}_1$ and $\mathcal{C}_2$ with density higher than Δ are detected as one cluster (if for $c_1 \in \mathcal{C}_1$ and $c_2 \in \mathcal{C}_2$ it is $c_2 \in \mathcal{N}(c_1)$)

DBSCAN

Manually detect density of lowest density cluster:

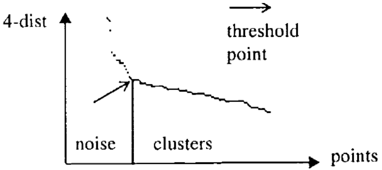
Plot the *k*-distance graphs for various values of *k*



DBSCAN

Manually detect density of lowest density cluster:

Plot the *k*-distance graphs for various values of *k*

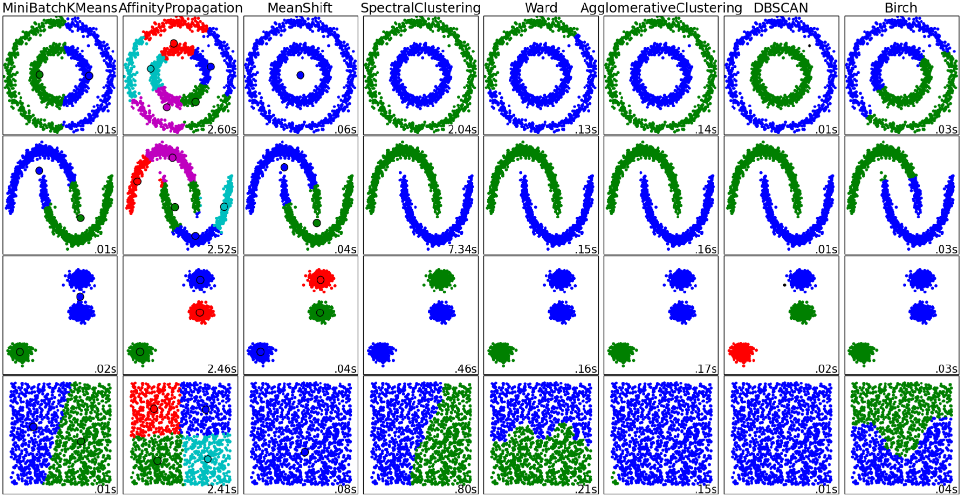


k-distance graph

A *k*-distance graph is an ordered mapping of each point to the distance from its *k*-th nearest neighbour.

Points in clusters will achieve similar values while there is a threshold point that indicates points outside all clusters.

Overview clustering algorithms



Outline

Introduction

k-means

DBSCAN

Gaussian Mixture Models

Gaussian Mixture Models

Questions?

Stephan Sigg
stephan.sigg@aalto.fi

Si Zuo
si.zuo@aalto.fi

Literature

- C.M. Bishop: Pattern recognition and machine learning, Springer, 2007.
- R.O. Duda, P.E. Hart, D.G. Stork: Pattern Classification, Wiley, 2001.