

Unsupervised Machine Learning

Janis Keuper



Basic Types of Machine Learning Algorithms

Supervised Learning

Unsupervised Learning

Reinforcement Learning

- NO Labeled data
- NO Direct and quantitative evaluation
- Explore structure of data

Unsupervised Learning

Data without “labels” (x_1, x_2, \dots, x_n)

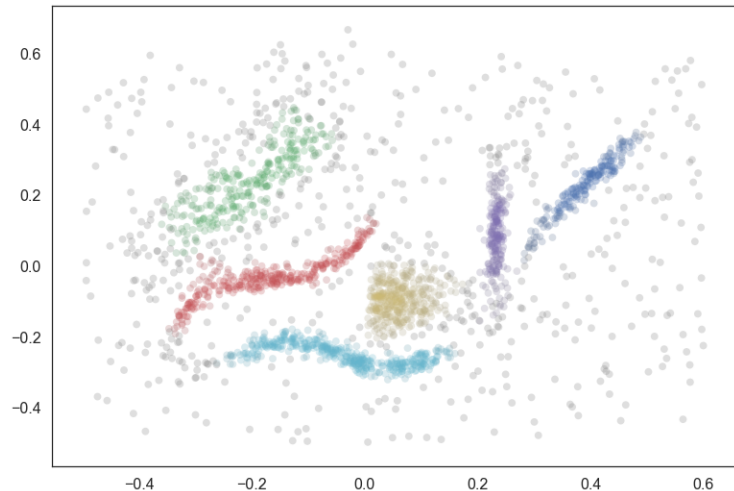
- **Clustering**
- **Outlier Detection (e.g. Defect or Intrusion detection)**

Definition

Cluster analysis or **clustering** is the task of **grouping a set of objects** in such a way that objects in the same group (called a cluster) are **more similar** (in some sense) to each other than to those in other groups (clusters). [Wikipedia]

Introduction

Cluster analysis or **clustering** is the task of **grouping a set of objects** in such a way that objects in the same group (called a cluster) are **more similar** (in some sense) to each other than to those in other groups (clusters). [Wikipedia]



Example 2d data set

Motivation

- Standard technique for data exploration and analysis
- Objective: find inherent structures in data
- Just like Multivariate Statistics
 - For high dimensional data
 - Geometric (manifold) and statistical motivations

Definition:

Given a set of observations (x_1, x_2, \dots, x_n)

where each observation is a d -dimensional real vector, **K-Means** clustering aims to partition the n observations into $k \leq n$

sets $S = \{S_1, S_2, \dots, S_k\}$

so as to minimize the within-cluster sum of squares:

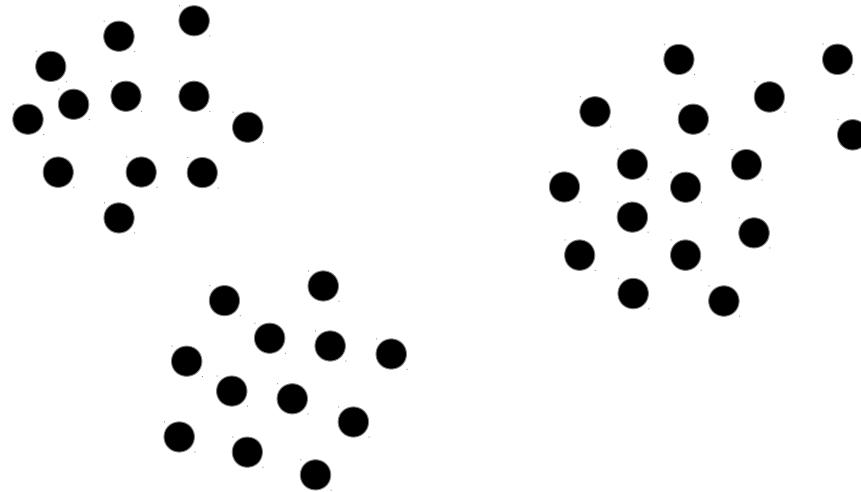
$$\arg \min[S] \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|^2$$

Clustering Algorithms: K-Means

Intuition:

$$\arg \min[S] \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|$$

Data:

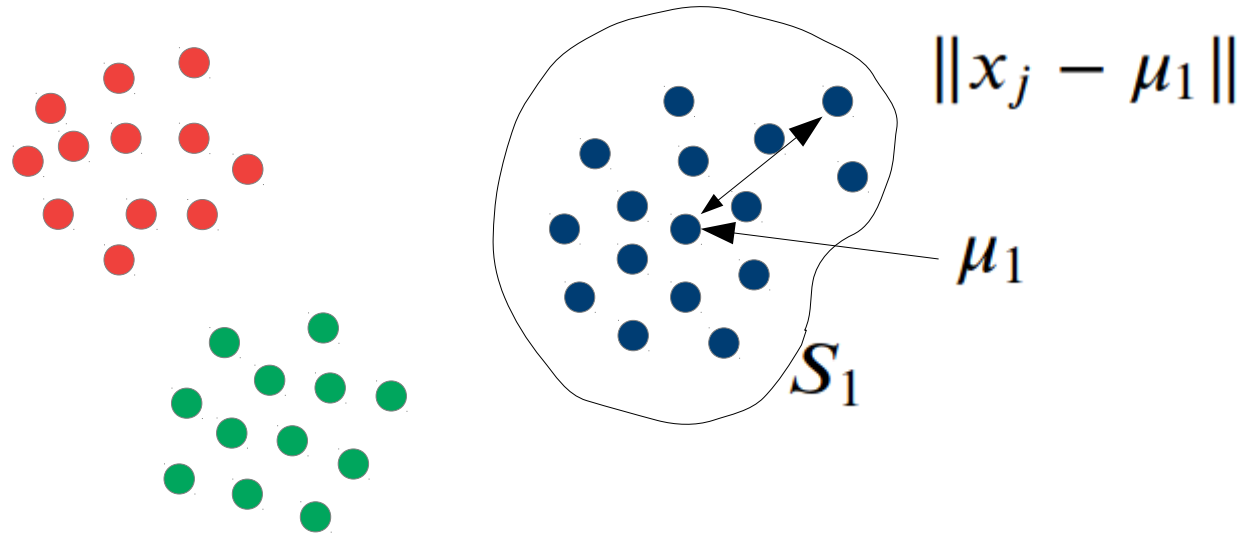


Clustering Algorithms: K-Means

Intuition:

$$\arg \min[S] \sum_{i=1}^k \sum_{x \in S_i} \|x - \mu_i\|$$

Clustering
for $k=3$



Basic Algorithm:

Init (t=1): Select k random cluster centers

$$\mu_1^{(1)} := x_{r1}, \mu_2^{(1)} := x_{r2}, \dots, \mu_k^{(1)} := x_{rk} \quad \text{for} \quad x_{rj} \in X$$

Repeat n times:

1. For step t : Assign all samples to “closest” center

$$\mathcal{S}_i^{(t)} = \{x_p : \|x_p - \mu_i^{(t)}\|^2 \leq \|x_p - \mu_j^{(t)}\|^2 \quad \forall j, 1 \leq j \leq k\}$$

2. Re-Compute cluster centers

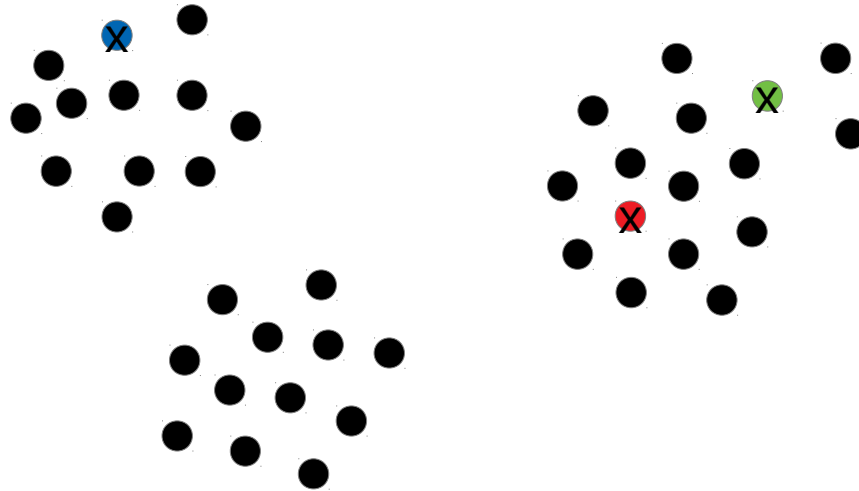
$$\mu_i^{(t+1)} = \frac{1}{|\mathcal{S}_i^{(t)}|} \sum_{x_j \in \mathcal{S}_i^{(t)}} x_j$$

Clustering Algorithms: K-Means

Intuition:

$$\mu_1^{(1)} := x_{r1}, \mu_2^{(1)} := x_{r2}, \dots, \mu_k^{(1)} := x_{rk} \quad \text{for } x_{rj} \in X$$

Init:



Random!

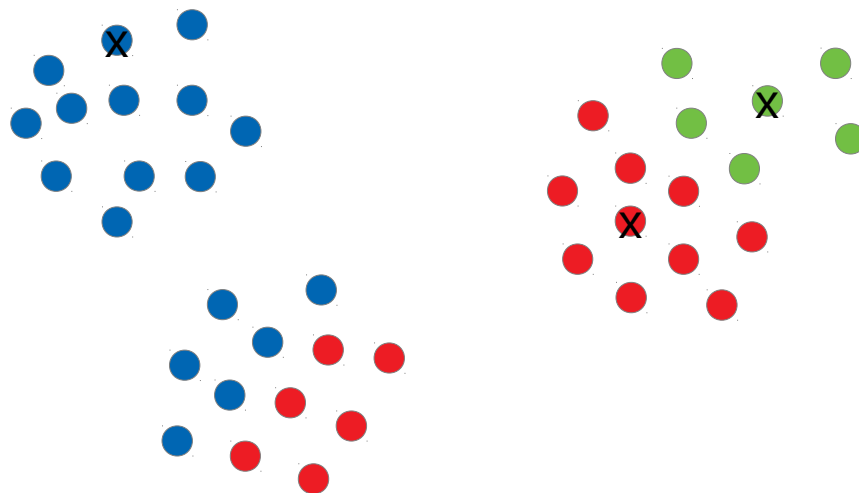
t=1

Clustering Algorithms: K-Means

Intuition:

$$\mathcal{S}_i^{(t)} = \{x_p : \|x_p - \mu_i^{(t)}\|^2 \leq \|x_p - \mu_j^{(t)}\|^2 \quad \forall j, 1 \leq j \leq k\}$$

Step 1:



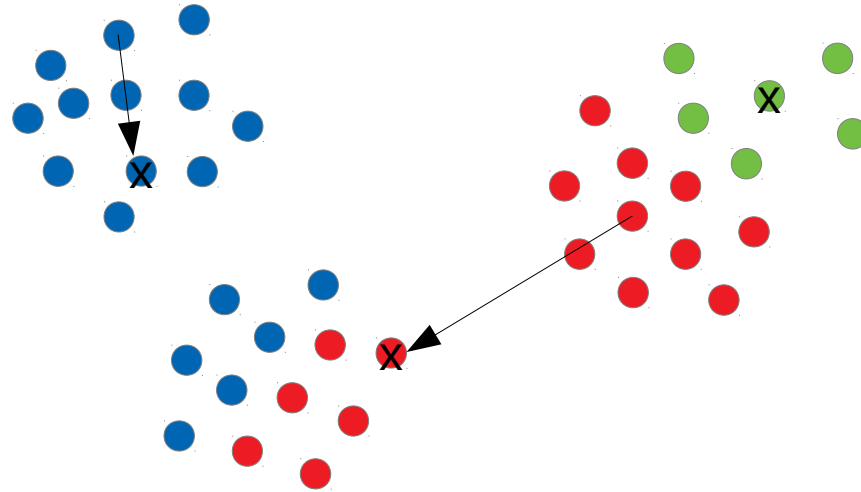
t=2

Clustering Algorithms: K-Means

Intuition:

$$\mu_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

Step 2:



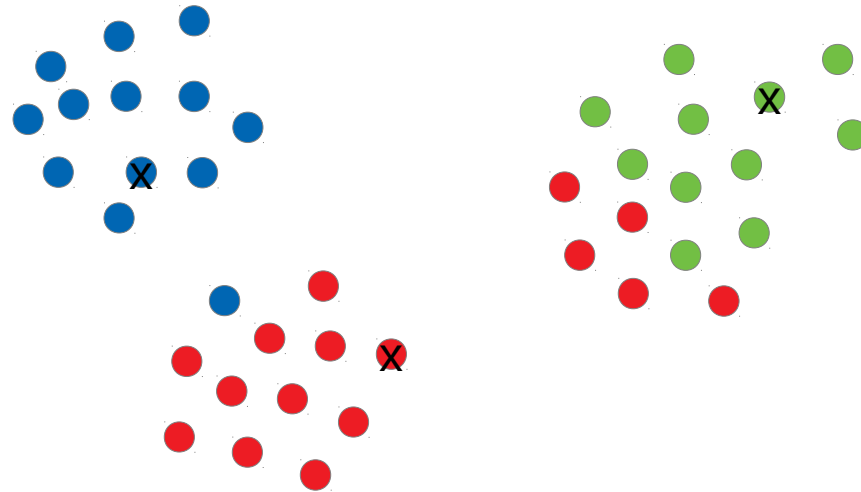
t=2

Clustering Algorithms: K-Means

Intuition:

$$S_i^{(t)} = \{x_p : \|x_p - \mu_i^{(t)}\|^2 \leq \|x_p - \mu_j^{(t)}\|^2 \quad \forall j, 1 \leq j \leq k\}$$

Step 1:



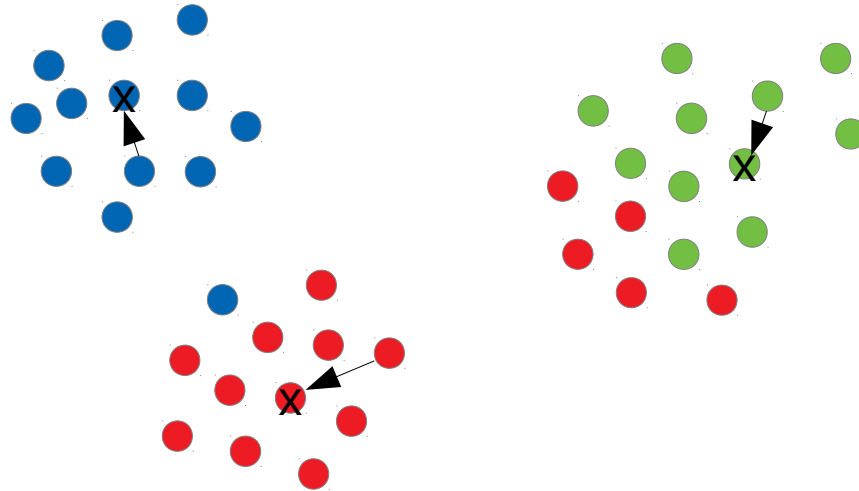
t=3

Clustering Algorithms: K-Means

Intuition:

$$\mu_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

Step 2:



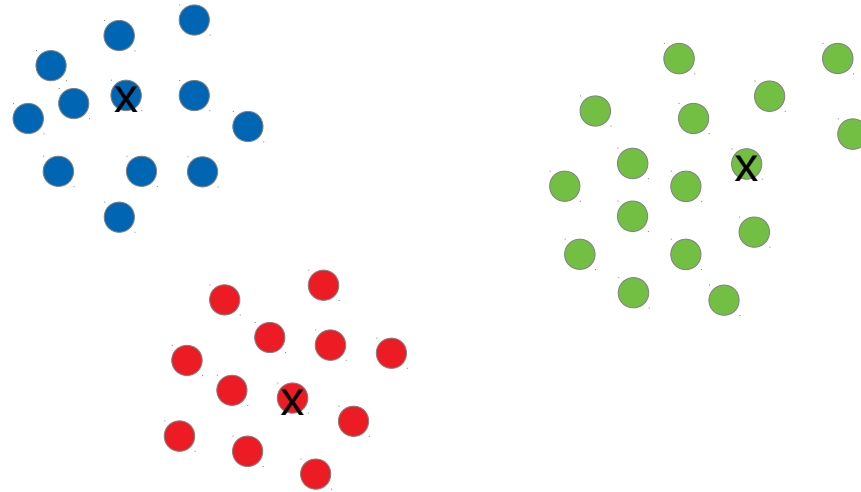
t=3

Clustering Algorithms: K-Means

Intuition:

$$\mathcal{S}_i^{(t)} = \{x_p : \|x_p - \mu_i^{(t)}\|^2 \leq \|x_p - \mu_j^{(t)}\|^2 \quad \forall j, 1 \leq j \leq k\}$$

Step 1:



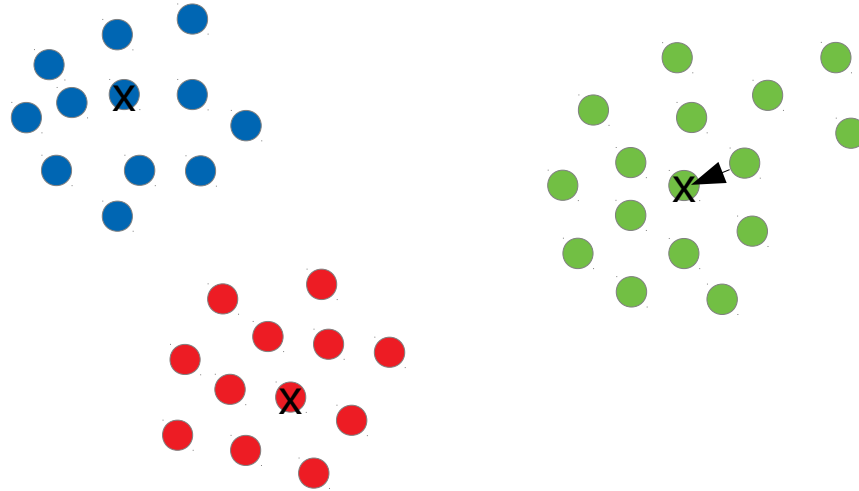
t=4

Clustering Algorithms: K-Means

Intuition:

$$\mu_i^{(t+1)} = \frac{1}{|S_i^{(t)}|} \sum_{x_j \in S_i^{(t)}} x_j$$

Step 2:

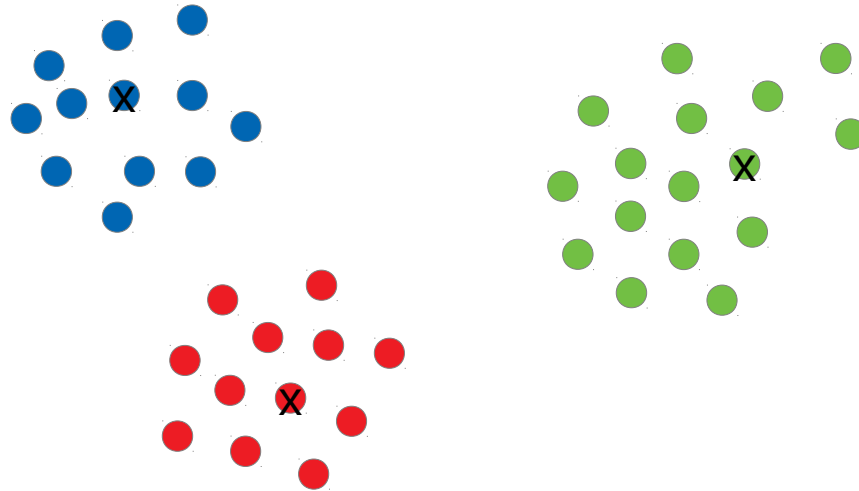


t=4

Clustering Algorithms: K-Means

Intuition: Convergence after fix t or fix means

Step 1:



$t=5,6,7,\dots$

Evaluation:

- Very simple but effective clustering algorithm
- Advantages:
 - Easy to implement
 - Easy to parallelize
- Disadvantages
 - Need to know k in advance (or search for best k)
 - High complexity: NP-hard (exponential in data dimension)
 - Problem with non-blob shaped (non-convex) clusters

Clustering Algorithms: K-Means

Evaluation:



More practical examples in the Lab session.... Now!

Definition

Density-based spatial clustering of applications with noise (DBSCAN) is a density-based clustering ***non-parametric*** algorithm:

Given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away).

Definition

Density-based spatial clustering of applications with noise (DBSCAN) is a density-based clustering ***non-parametric*** algorithm:

Given a set of points in some space, it groups together points that are closely packed together (points with many nearby neighbors), marking as outliers points that lie alone in low-density regions (whose nearest neighbors are too far away).

In contrast to K-Means, the number of clusters is not given

Definitions in DBSCAN

Given a set of observations (x_1, x_2, \dots, x_n) we need:

Some distance function on the data: $d(x_i, x_j)$

Parameter: radius ϵ

Set of Neighbors: $\mathbb{N}(x_i)$

Parameter: Minimum number of Neighbors n_ϵ

Definitions in DBSCAN

Density at a data sample:

number of neighbors in radius

$$\mathfrak{d}(x_i) := |\mathbb{N}(x_i)| = |\{x_p : d(x_p, x_i) \leq \varepsilon\}|$$

Core samples:

all samples with a density higher than a threshold

$$\{x_p : \mathfrak{d}(x_p) \geq n_\varepsilon\}$$

Reachable samples:

all samples with at least one neighbor

$$\{x_p : \mathfrak{d}(x_p) \geq 1\}$$

Definitions in DBSCAN

Outlier samples: All samples without neighbors

$$\{x_p : \mathfrak{d}(x_p) = 0\}$$

Basic Algorithm

Init: mark all samples as core, reachable or outlier
 Remove outlier

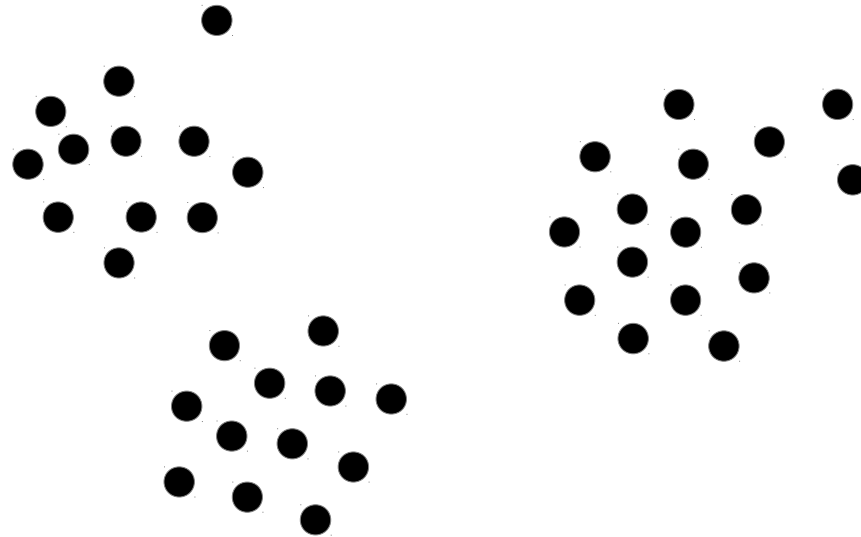
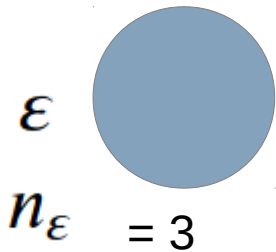
For all core samples: choose next random core sample and recursively merge it's neighborhood with all neighbors that are also core samples.
 Increment Cluster ID.

For all reachable samples: assign to closest cluster

Clustering Algorithms: DBSCAN

Intuition:

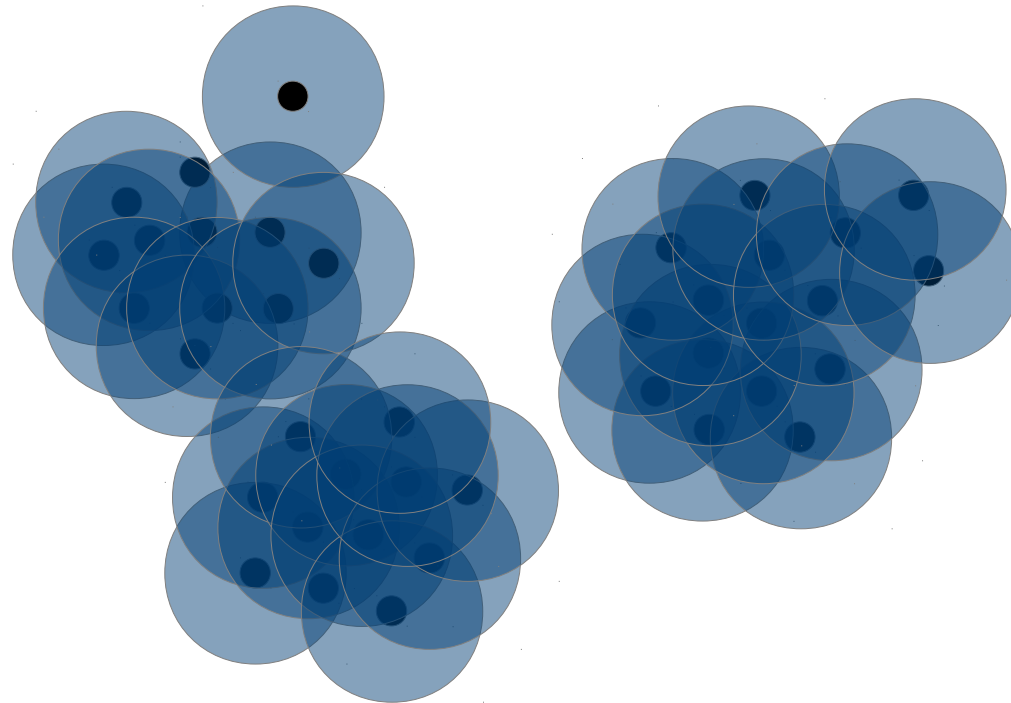
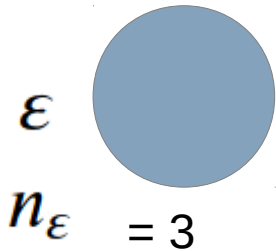
Init:



Clustering Algorithms: DBSCAN

Density:

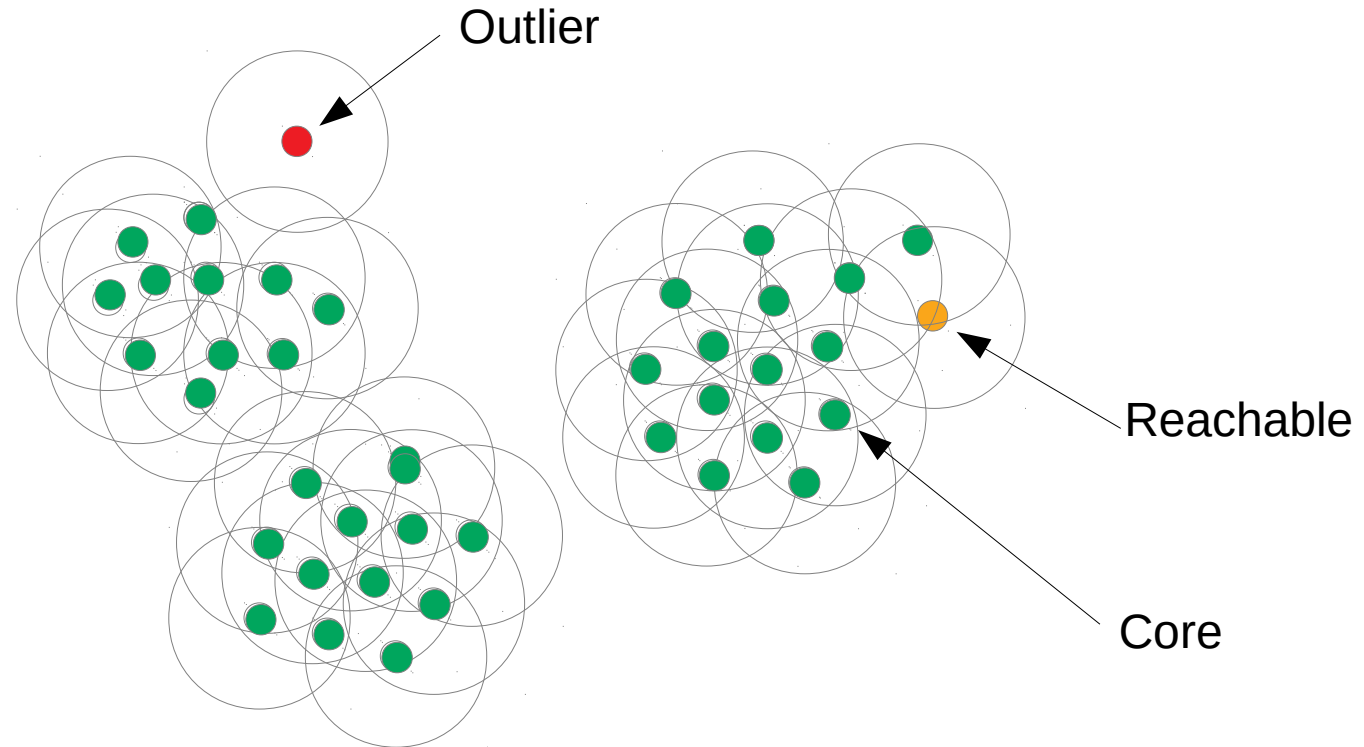
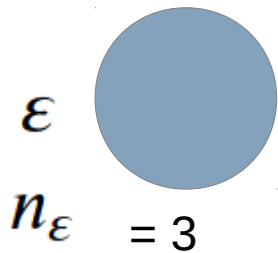
Init:



Clustering Algorithms: DBSCAN

Core – Reachable - Outlier:

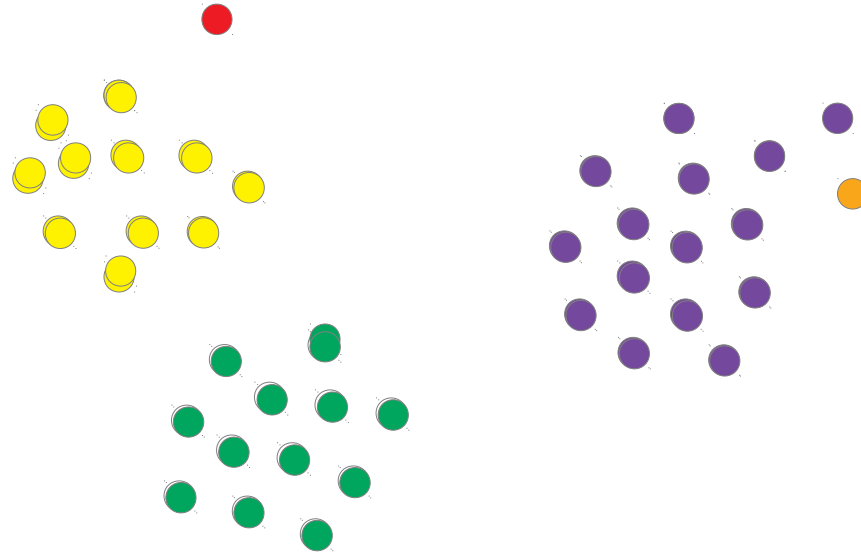
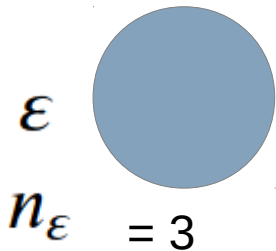
Init:



Clustering Algorithms: DBSCAN

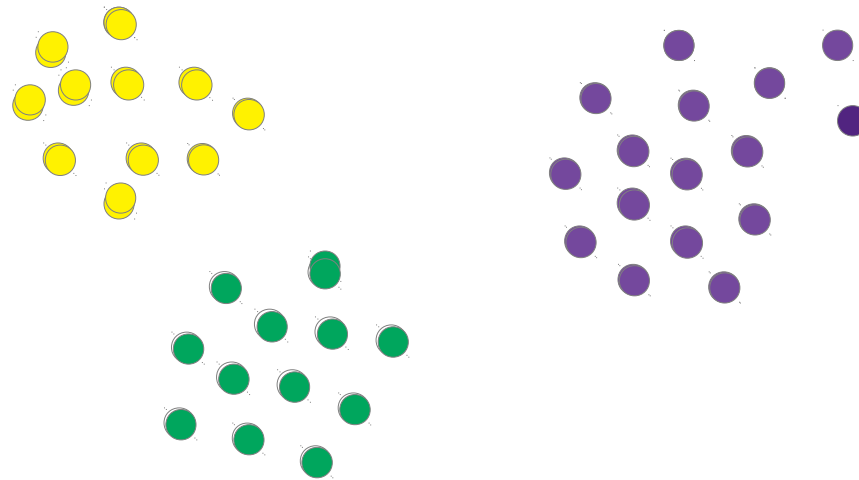
Merge:

Init:



Clustering Algorithms: DBSCAN

Final:



Evaluation:

- Very common Clustering Algorithm
- Advantages:
 - Does not need number of clusters
 - Works well for non convex clusters
 - Fast implementation possible (R-Trees)
- Disadvantages
 - Has two hyper-parameters to optimize
 - Fails on data with high variance in density
 - Not deterministic

Clustering Algorithms: DBSCAN

Evaluation:



More practical examples in the Lab session.... Now!

How to evaluate clustering:

- Visually → use dimension reduction techniques to visualize 2d or 3d

How to evaluate clustering:

- Visually → use dimension reduction techniques to visualize 2d or 3d
- Quantitative quality measures (what is a good cluster?)

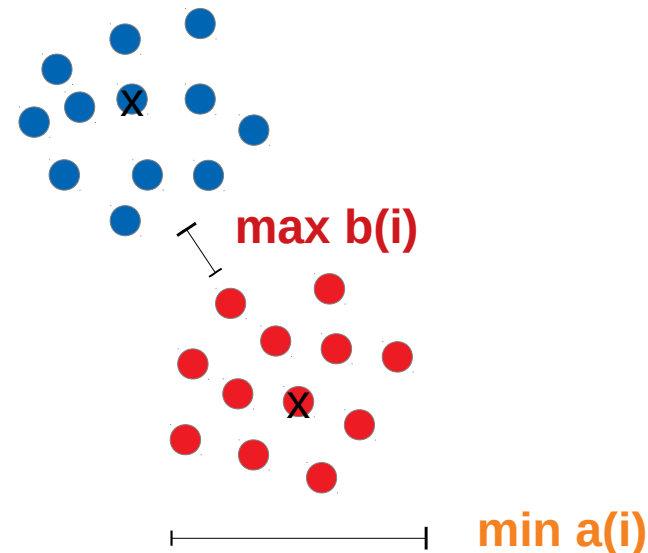
- **Low intra cluster variance**

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} d(i, j)$$

- **High extra cluster variance**

$$b(i) = \min_{i \neq j} \frac{1}{|C_j|} \sum_{j \in C_j} d(i, j)$$

For each data point $i \in C_i$ (data point i in the cluster C_i)

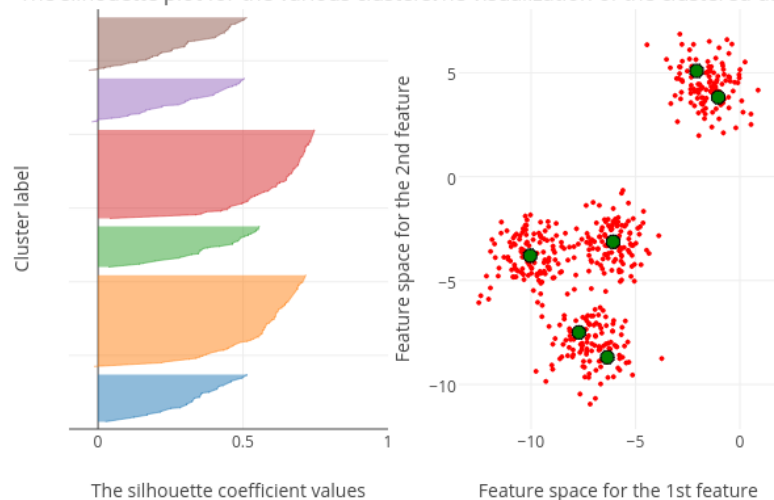


Clustering Algorithms: Evaluation

Silhouette Diagrams: finding the best number of clusters

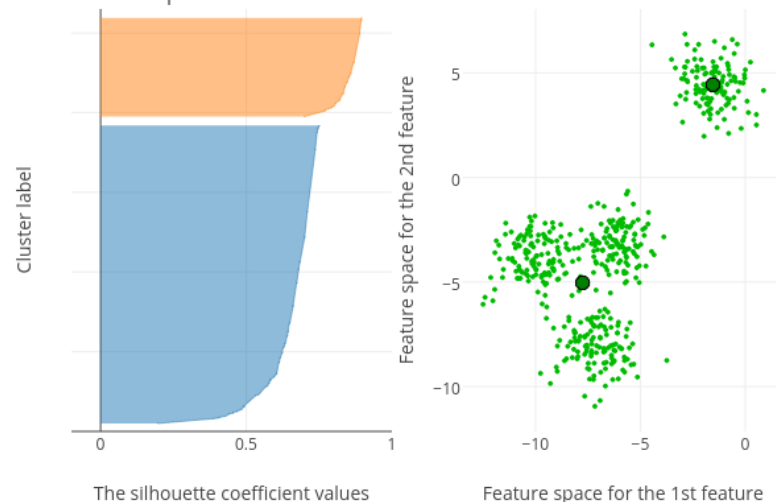
Silhouette analysis for KMeans clustering on sample data with $n_clusters = 6$

The silhouette plot for the various clustersThe visualization of the clustered data.



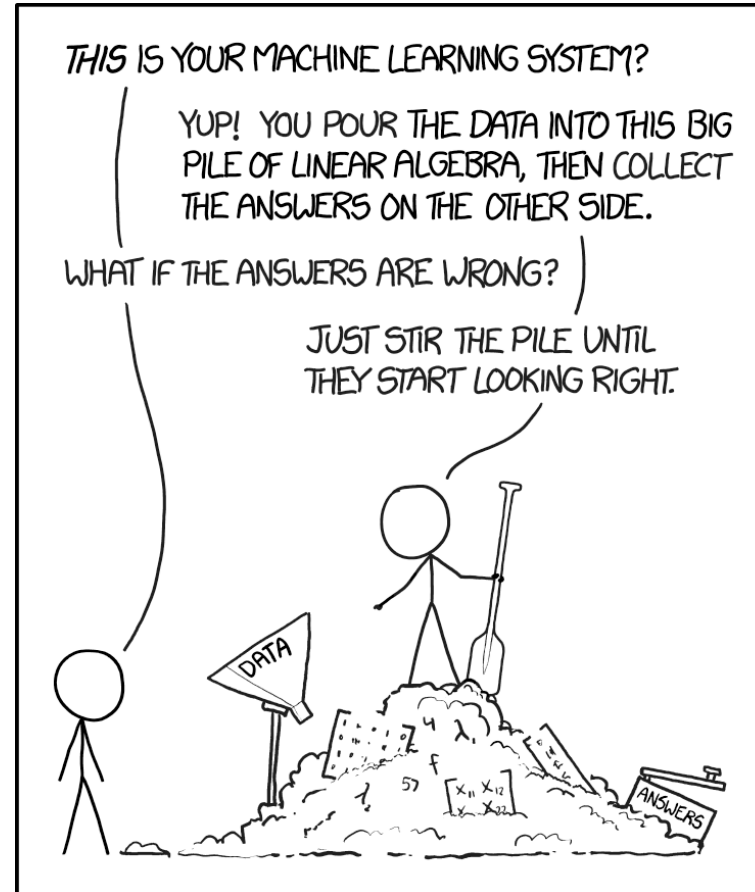
Silhouette analysis for KMeans clustering on sample data with $n_clusters = 2$

The silhouette plot for the various clustersThe visualization of the clustered data.



[plots: <https://plot.ly/scikit-learn/plot-kmeans-silhouette-analysis/>]

Discussion



<https://xkcd.com/1838/>