Unit 6. Unsupervised Learning

Artificial Intelligence and Learning

felipe.alonso@urjc.es

Bibliography

- Chapter 3. *Introduction to Machine Learning with Python*. AC Müller and S Guido. O'Reilly
- Chapters 5 and 6. *Hands-on Unsupervised Learning using Python*. AA Patel. O'Reilly
- Scikit-learn and Scipy documentation

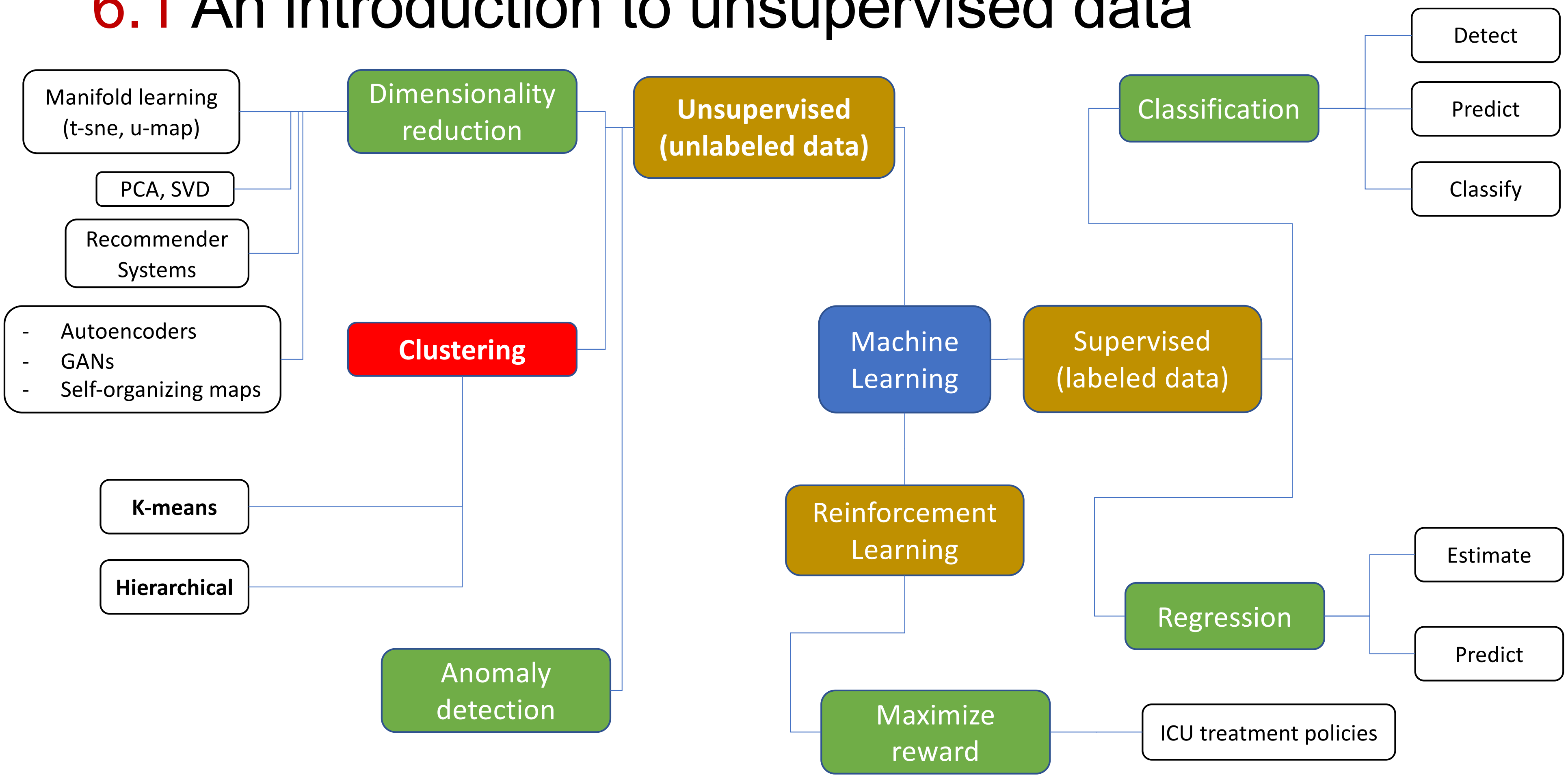
Contents

Contents

6.1 An introduction to unsupervised data

- **Data does not contain (or we do not use) any label information**
- **The learning algorithms just shown the input data and asked to extract knowledge/patterns from this data.**
- **A major challenge is evaluating whether the algorithm learned something useful**
- **Thus, unsupervised algorithms are used often in an exploratory setting, when a data scientist wants to understand the data better**
- **Another common application is as a preprocessing step for supervised algorithms**
  - Learning a new representation of the data can sometimes improve the accuracy of supervised algorithms, or can lead to reduced memory and time consumption

6.1 An introduction to unsupervised data



Contents

6.2 What's clustering

- **Grouping unlabeled examples**
  - It is the task of partitioning the dataset into groups, called **clusters**
  - Points within a single cluster are very similar, and points in different clusters are different
  - Clustering algorithms assign (or predict) a number to each data point, indicating which cluster a particular point belongs to

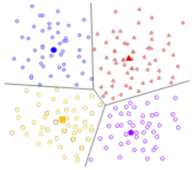
6.2 Uses of clustering

- **Market segmentation**
- **Group items by some features: docs, pictures, music, search results**
- **Anomaly detection**
- **Infer missing data**
- **Data compression**
- **Privacy: userID vs clusterID**

6.2 Types of clustering

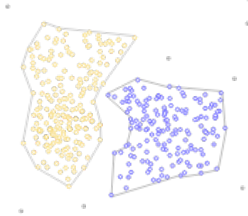
- **For an exhaustive list, see this [paper](#).**

1. Centroid-based: k-means

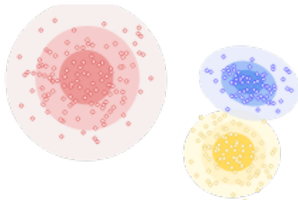


2. Connectivity-based: hierarchical

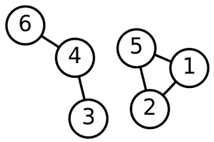
3. Density-based: DBSCAN



4. Distribution-based: mixture of gaussians



5. Graph-based: spectral clustering



Contents

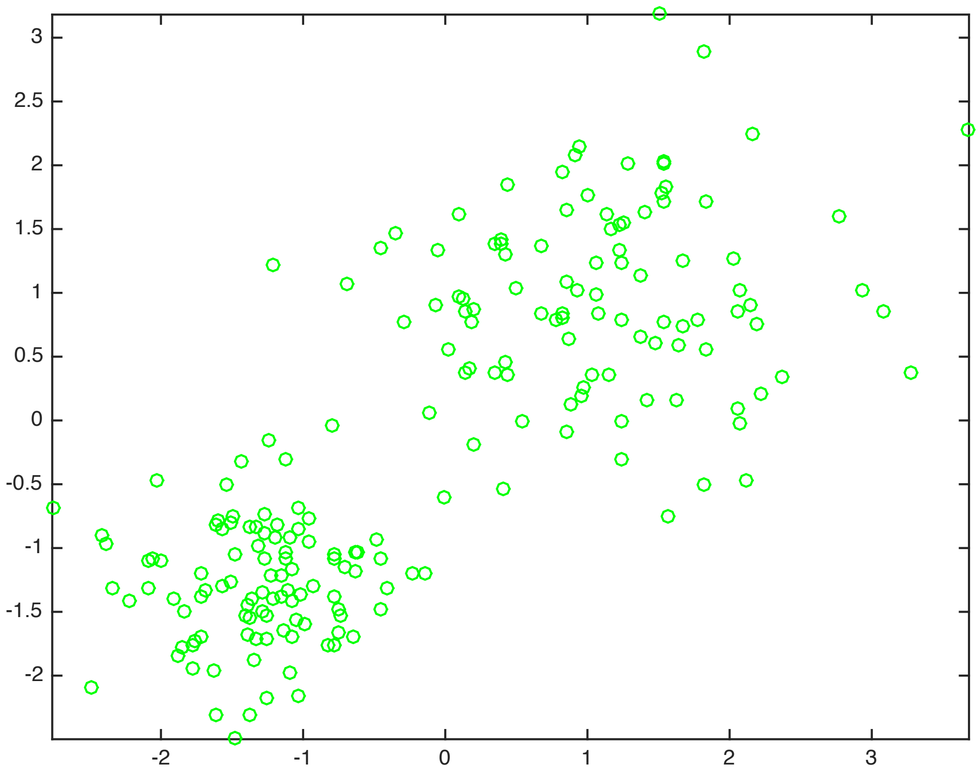
6.3 K-means

- **One of the simplest and most commonly used clustering algorithms**

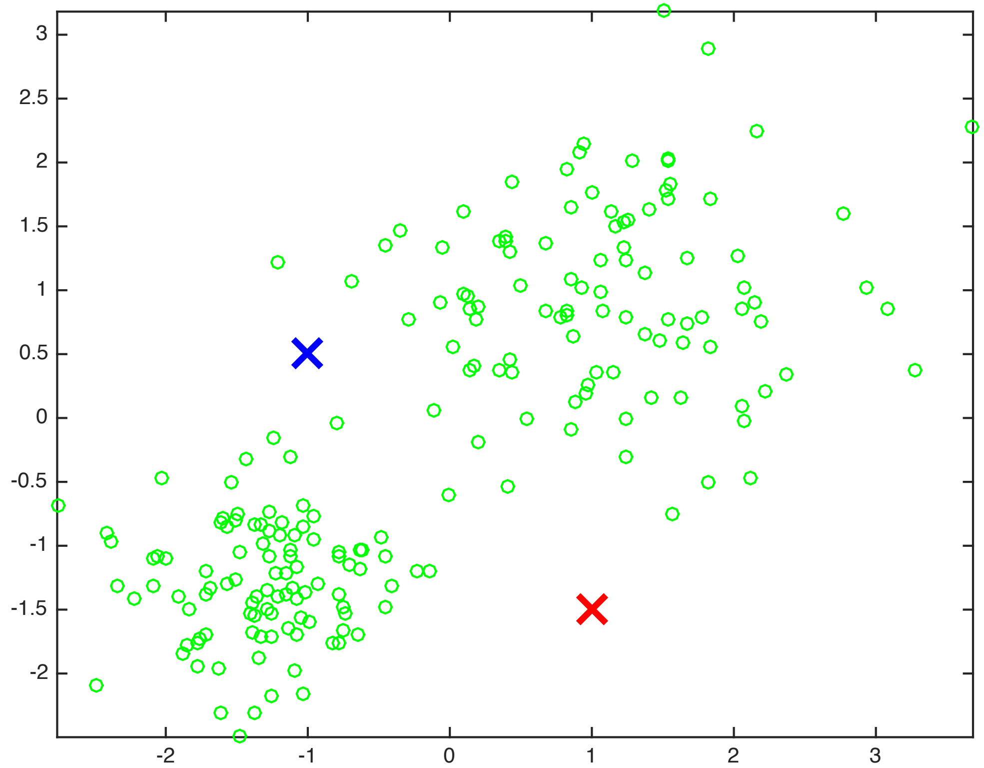
- **The algorithm alternates between two steps:**
  1. Assigning each data point to the closest cluster center, and then
  2. Setting each cluster center (a.k.a. **centroid**) as the mean of the data points that are assigned to it.

- **The algorithm is finished when the assignment of instances to clusters no longer changes**

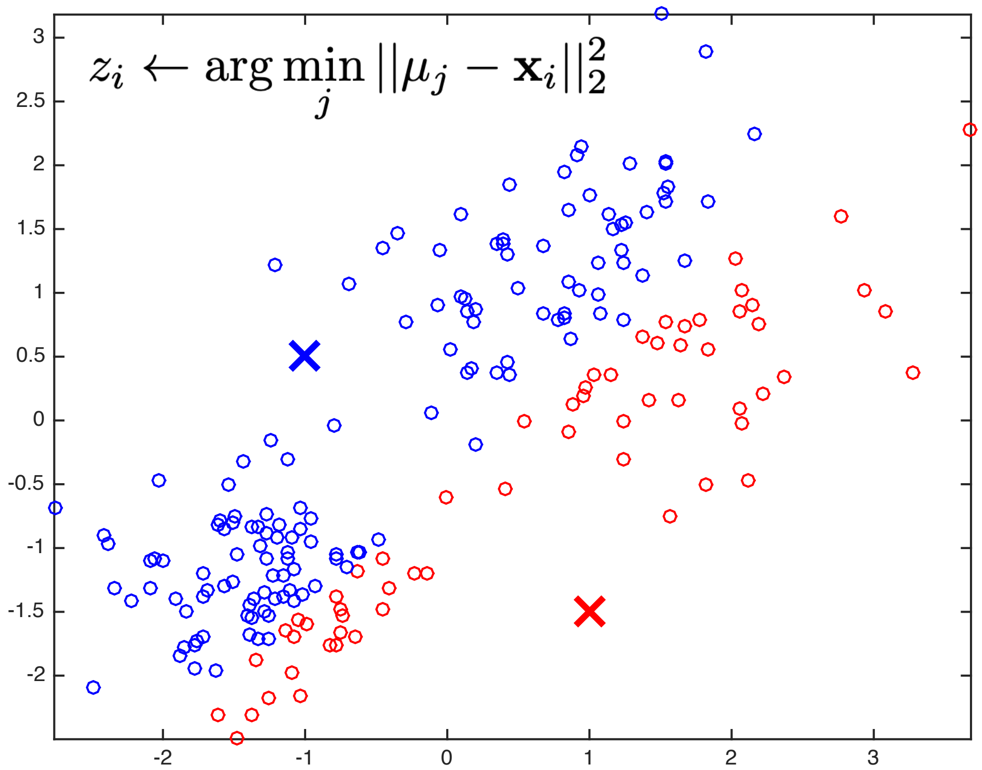
6.3 K-means



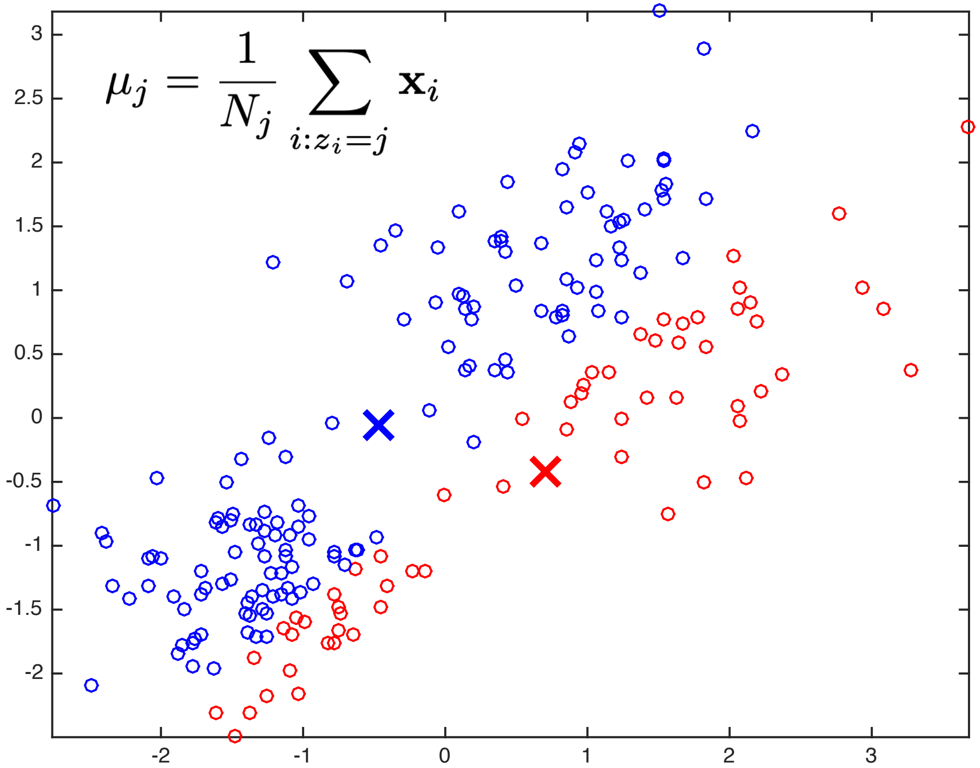
6.3 K-means: initialize cluster centers μ_k



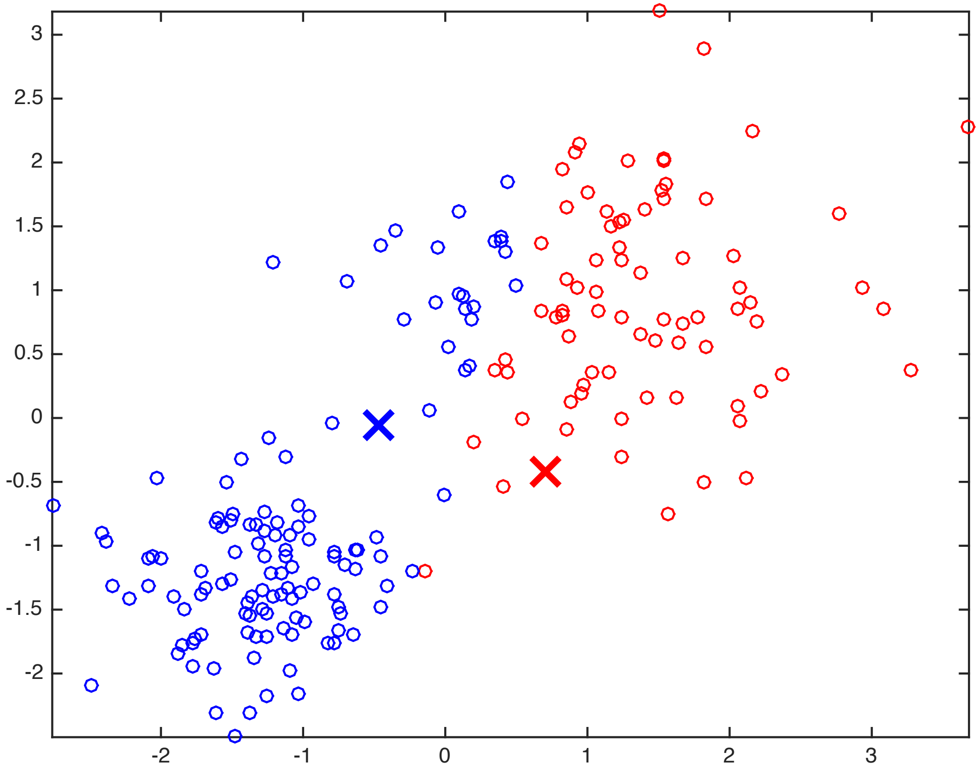
6.3 K-means: assign observations to cluster centers



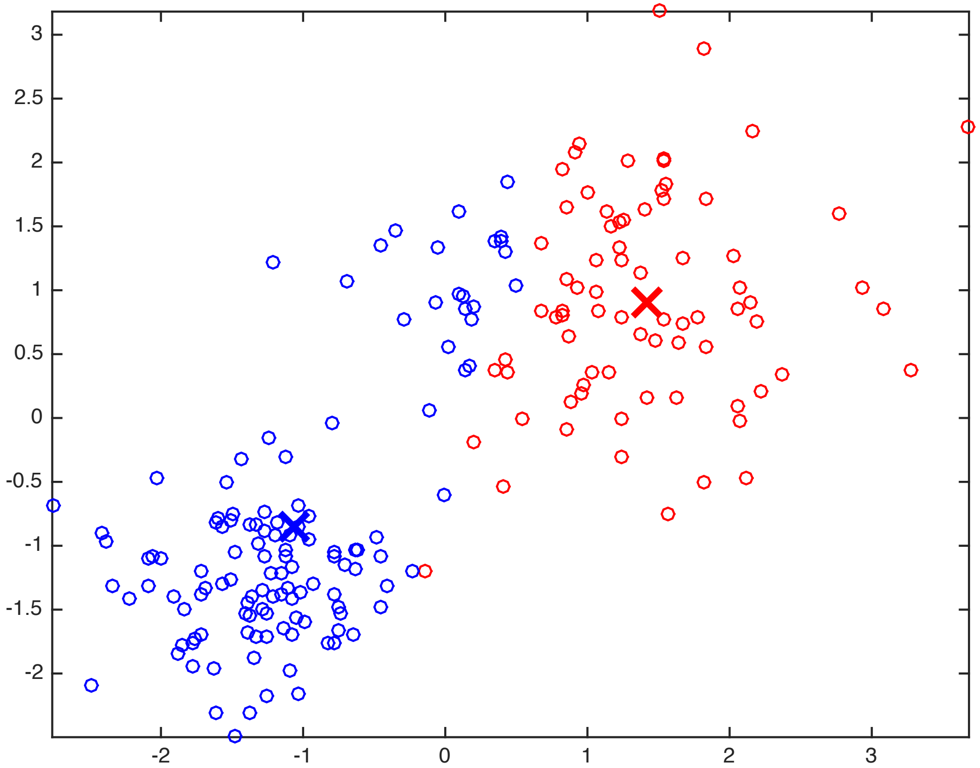
6.3 K-means: recompute centroids



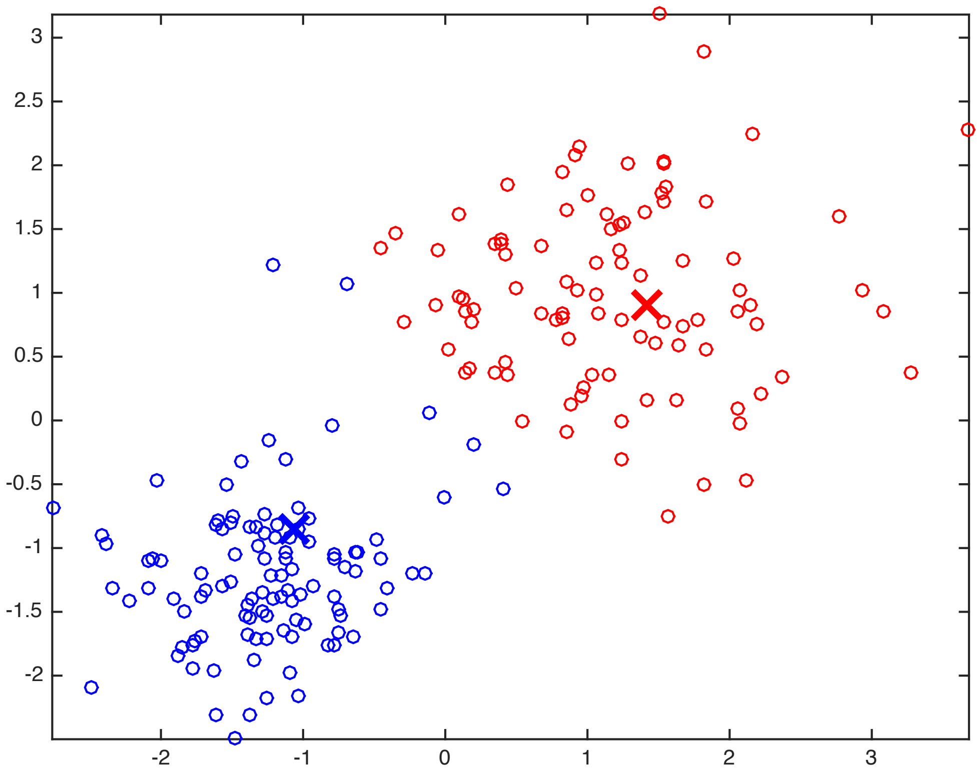
6.3 K-means: assign observations to cluster centers



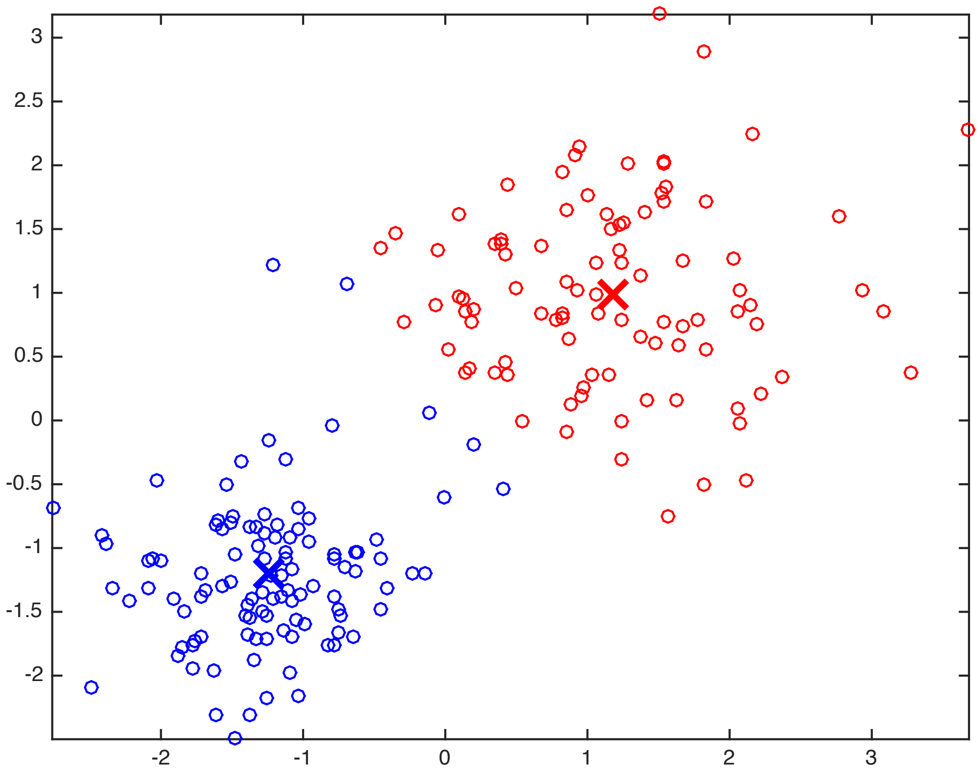
6.3 K-means: recompute centroids



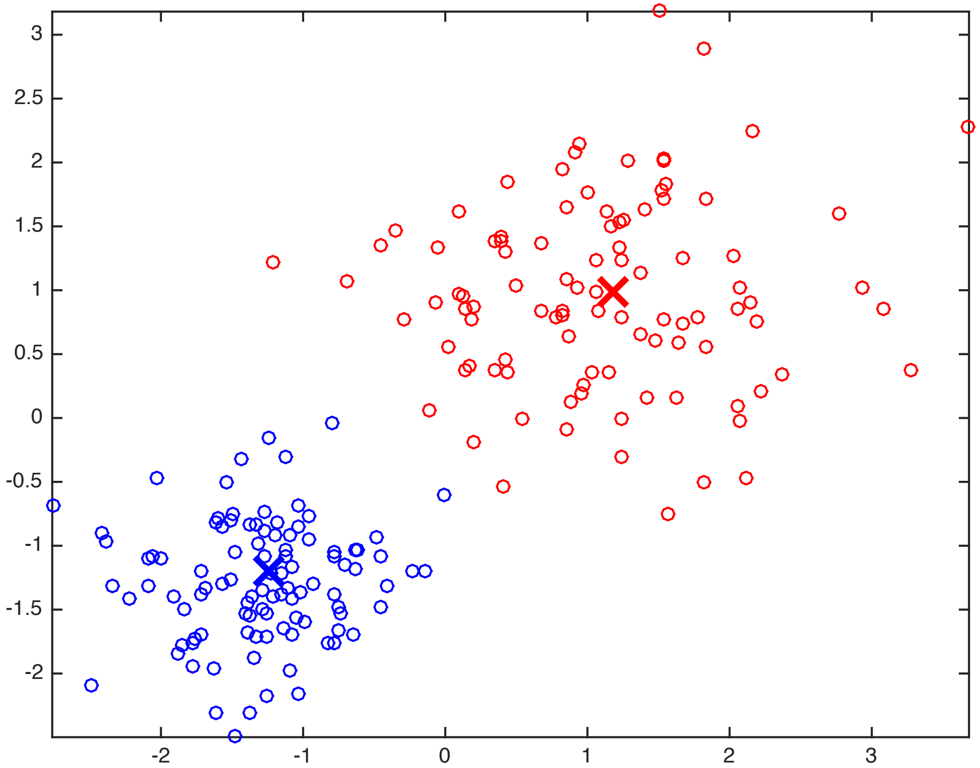
6.3 K-means: assign observations to cluster centers



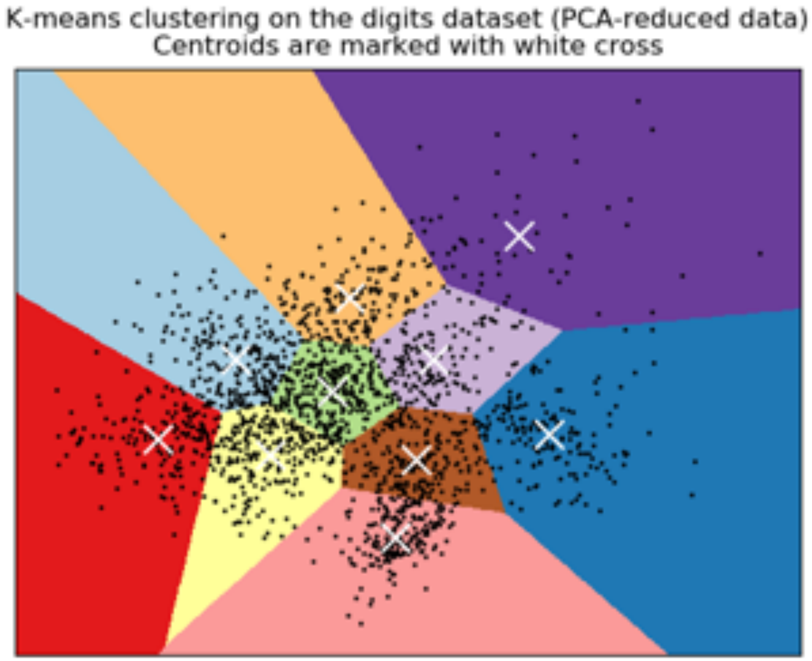
6.3 K-means: recompute centroids



6.3 K-means: assign observations to cluster centers



6.3 K-means in action



6.3 K-means recap

- **OKs**
  - Simple and scalable (large data sets)
  - Adapts to new examples (predict)

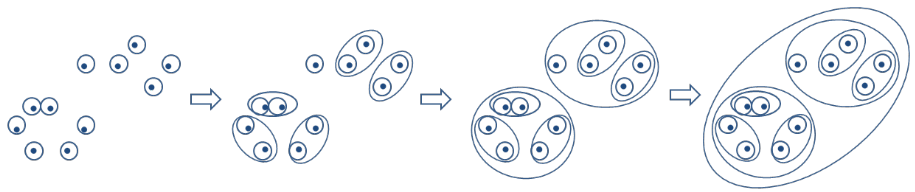
- **KOs:**
  - Choosing k manually.
  - Highly dependent on the initialization of the centroids
    - Computation is often done several times, with different initializations
    - K-means++
  - Scaling with number of dimensions: as the number of dimensions increases, a distance-based similarity measure converges to a constant value between any given examples.
  - Centroids can be dragged by outliers

Contents

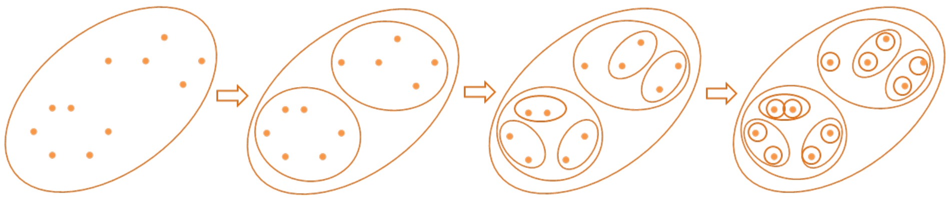
6.4 Hirerarchical clustering

- **General family of clustering algorithms that build nested clusters by merging or splitting them successively**:

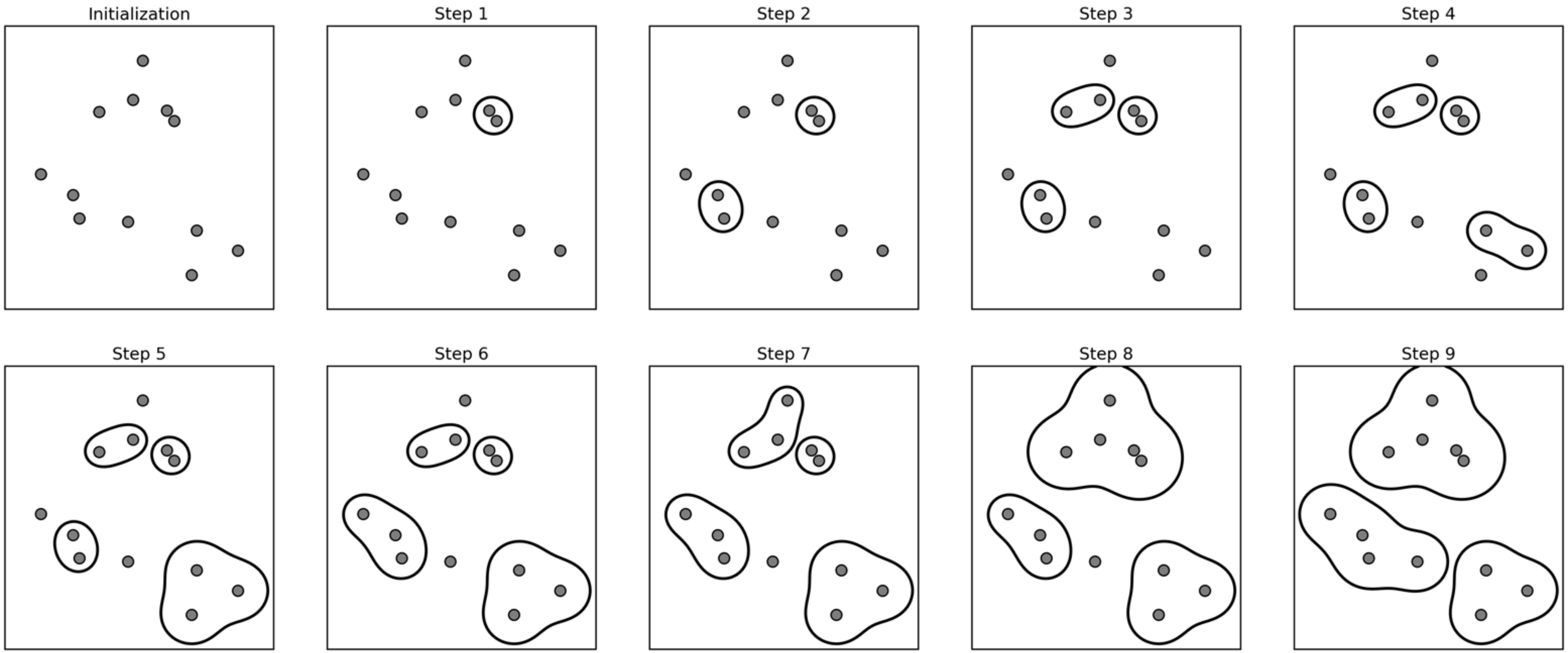
  - **Agglomerative** (bottom-up): each observation starts in its own cluster, and pairs of ***most similar*** **clusters are merged** as one moves up the hierarchy.



  - **Divisive** (top-down) all observations start in one cluster, and splits are performed recursively as one moves down the hierarchy (not in sklearn)



6.4 Agglomerative clustering



https://github.com/amueller/introduction_to_ml_with_python/blob/master/03-unsupervised-learning.ipynb

6.4 Agglomerative clustering

- **There are several *linkage* criteria that specify how exactly the "most similar cluster" is measured**

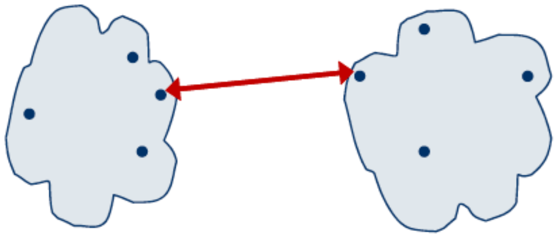
- **This measure is always defined between two existing clusters**

Ward linkage minimizes the sum of squared differences within all clusters.

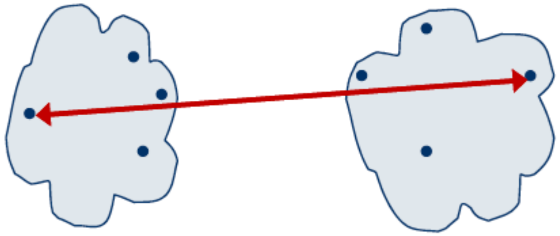
$$\Delta(A,B) = \sum_{i \in A \bigcup B} ||\vec{x_i} - \vec{m}_{A \bigcup B}||^2 - \sum_{i \in A} ||\vec{x_i} - \vec{m}_A||^2 - \sum_{i \in B} ||\vec{x_i} - \vec{m}_B||^2 = \frac{n_A n_B}{n_A + n_B} ||\vec{m}_A - \vec{m}_B||^2$$

6.4 Linkage criteria

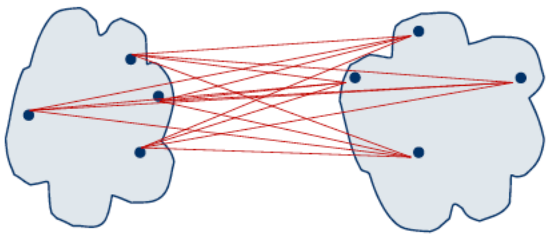
Single linkage minimizes the distance between the closest observations of pairs of clusters.



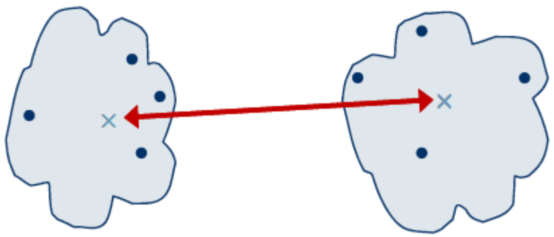
Maximum or **complete** linkage minimizes the maximum distance between observations of pairs of clusters.



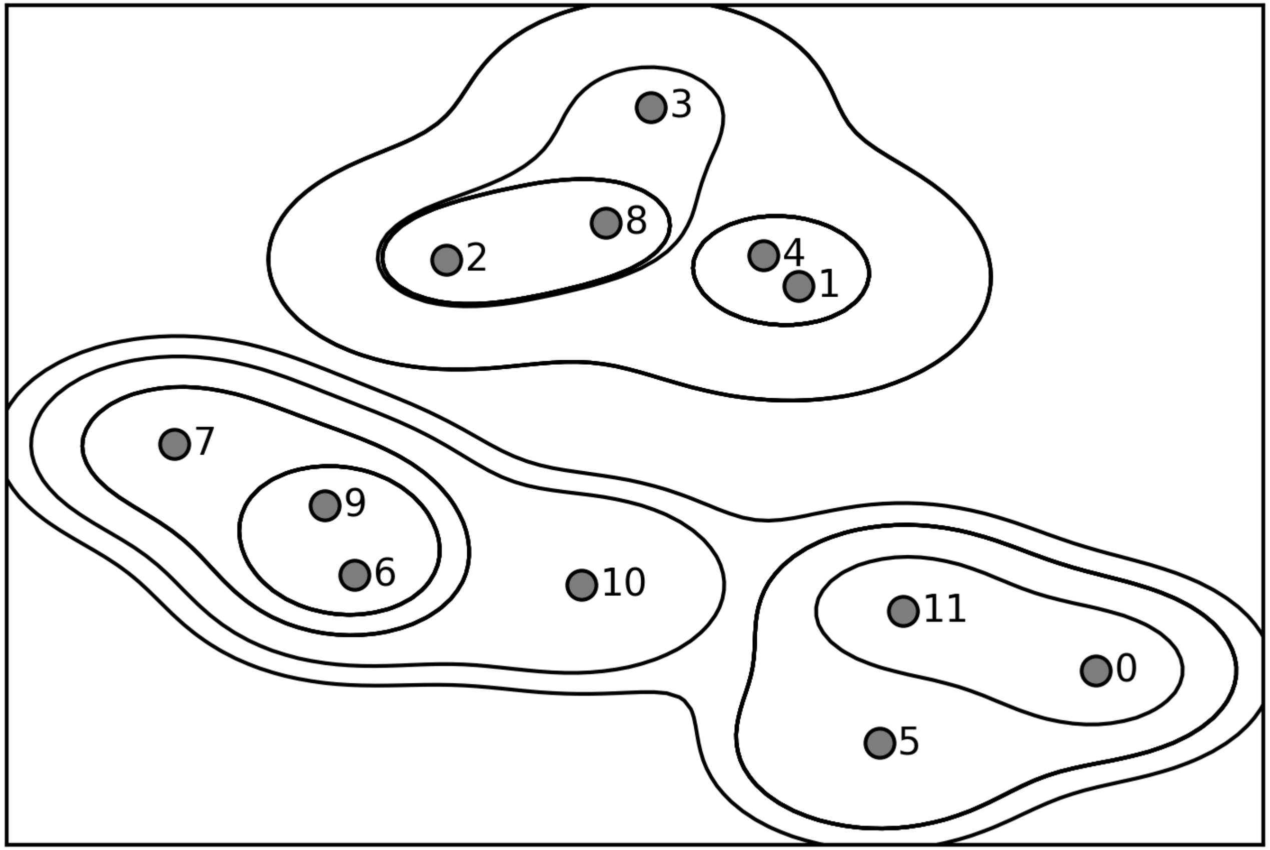
Average linkage minimizes the average of the distances between all observations of pairs of clusters.



Centroid linkage minimizes the distances between centroids

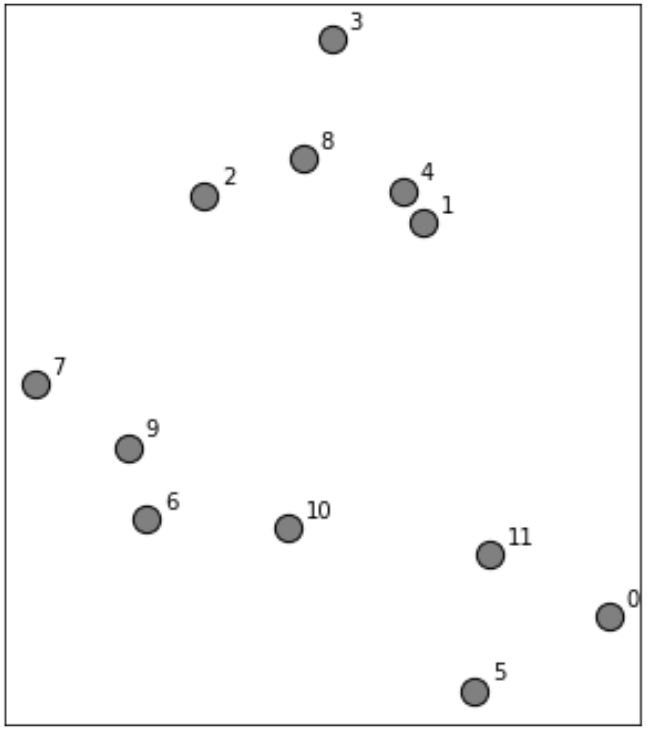


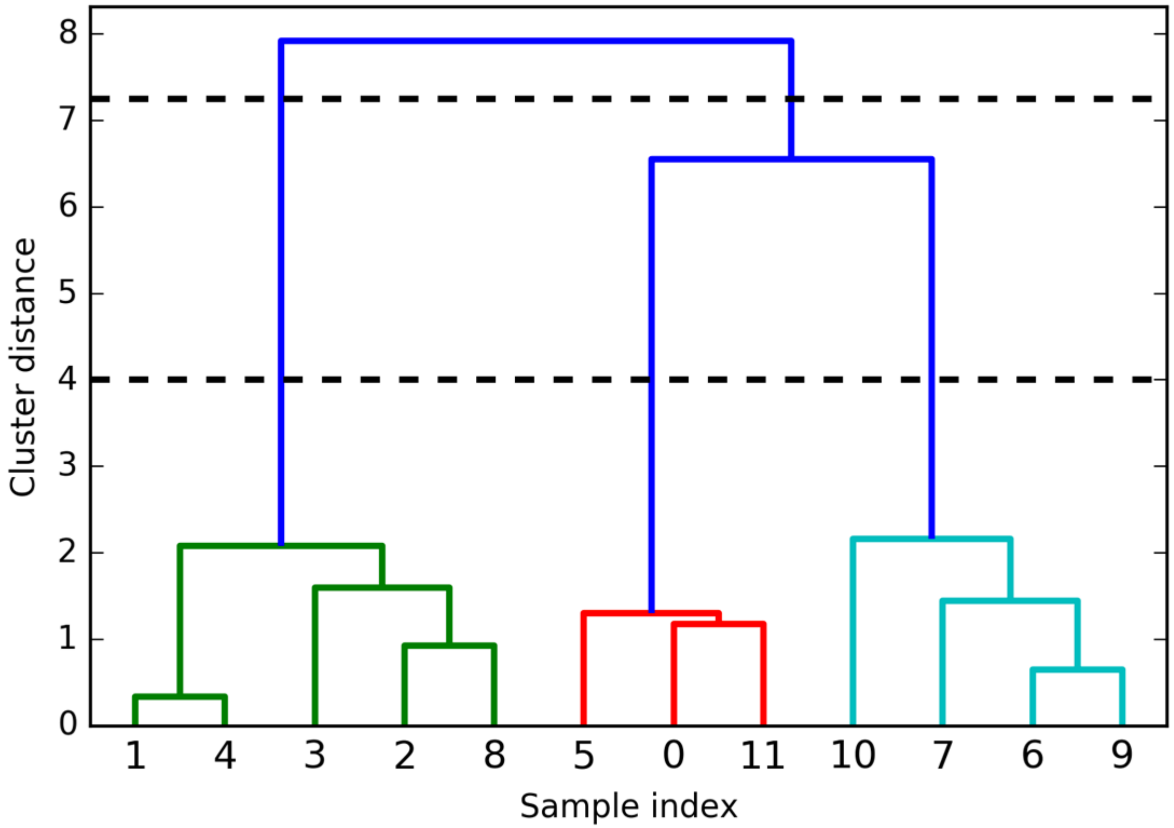
6.4 Visualizing agglomerative clustering



https://github.com/amueller/introduction_to_ml_with_python/blob/master/03-unsupervised-learning.ipynb

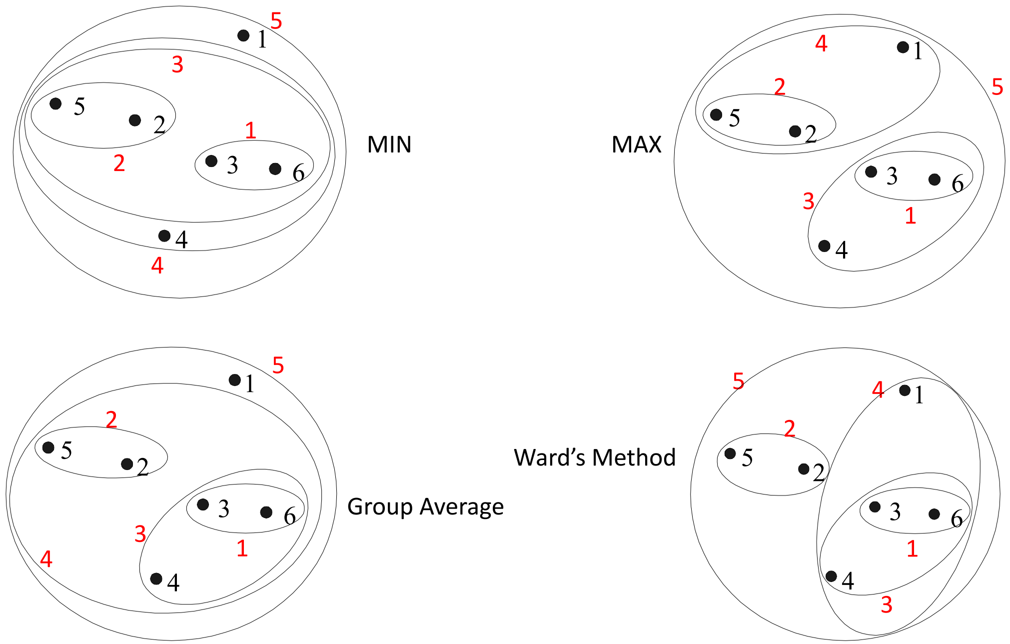
6.4 Dendogram





https://github.com/amueller/introduction_to_ml_with_python/blob/master/03-unsupervised-learning.ipynb

6.4 Agglomerative clustering in action



6.4 Agglomerative clustering recap

- **OKs**
  - Do not have to assume any particular number of clusters
  - They may correspond to meaningful taxonomies (e.g. shopping websites)

- **KOs:**
  - Does not adapt to new examples (predict)
  - Partitioning rather than clustering the data, and depending on the linkage the following problems might occur:
    - Sensitivity to noise and outliers
    - Difficulty handling different sized clusters and irregular/complex shapes
    - Breaking large clusters

Contents

Unit 6. Unsupervised learning

6.5 DBSCAN

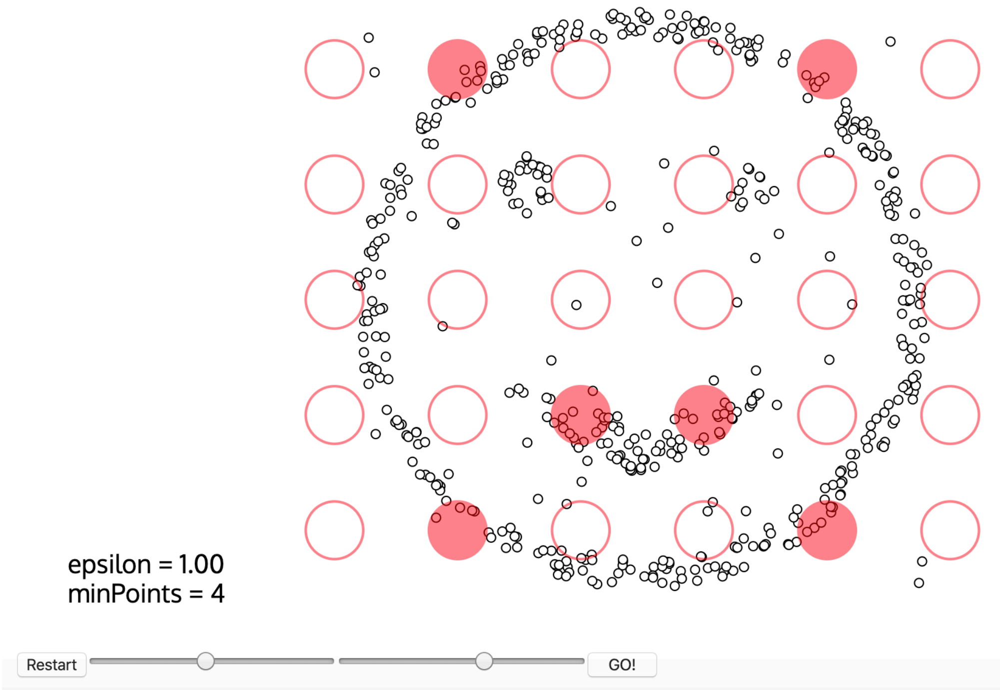
- **Stands for “density-based spatial clustering of applications with noise”**

- **Main features:**
  - It does not require the user to set the number of clusters a priori
  - It can capture clusters of complex shapes
  - It can identify points that are not part of any cluster
  - It is somewhat slower than agglomerative clustering and k-means, but still scales to relatively large datasets

- **The idea behind DBSCAN is that clusters form dense regions of data, separated by regions that are relatively empty**

6.5 DBSCAN

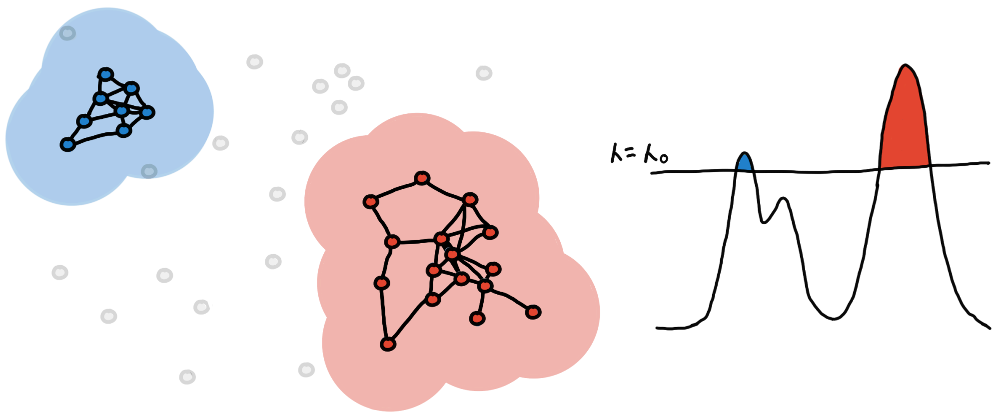
- **Better if we visualize how it works**



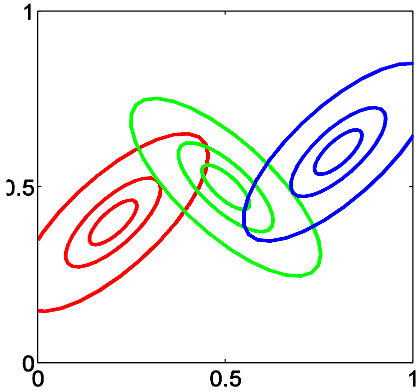
6.5 HDBSCAN

- **It stands for Hierarchical DBSCAN**

- **It looks for regions of the data that are denser than the surrounding space**



https://github.com/scikit-learn-contrib/hdbscan



6.5 Mixture of Gaussians

- **It is a probabilistic model**

- **Hard vs. soft clustering**
  - Hard clustering: every point belongs to exactly one cluster
  - Soft clustering: every point belongs to several clusters with certain degrees
    - e.g. 54% chance document is world news, 45% science, 1% sports, and 0% entertainment

- **Probabilistic clustering**
  - Each cluster is mathematically represented by a parametric distribution
  - The entire data set is modeled by a mixture of these distributions

6.5 Spectral clustering

- Graph-based method that clusters data using eigenvectors (**spectral decomposition**) of matrices derived from the distance between points

- **Similarity graphs:** given a set of points $X = \{x_1, x_2, \ldots, x_n\}$ with pairwise similarities s_{ij} or pairwise distances d_{ij}, transform them into a graph:

  - **ε-neighborhood graph**: we connect all points whose pairwise distances are smaller than ε

  - **K-nn graph**: connect x_i with x_j if x_j is among k-nn of x_i

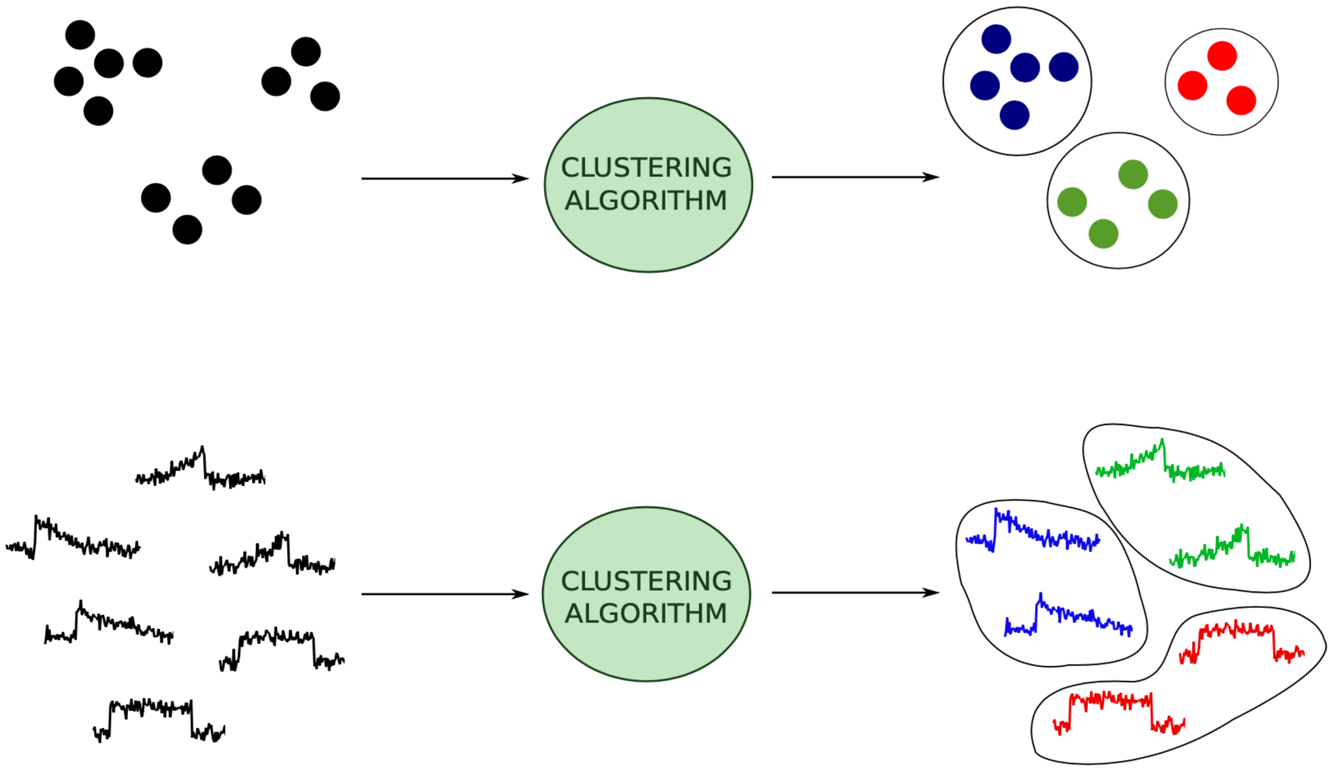
  - **Fully connected** (or affinity matrix A)

$$s_{ij} = s(x_i, x_j) = A_{ij} = \begin{cases} \exp\left(-||x_i - x_j||^2 / 2\sigma^2\right), & \text{if } i \neq j \\ 0, & \text{if } i = j \end{cases}$$

Contents

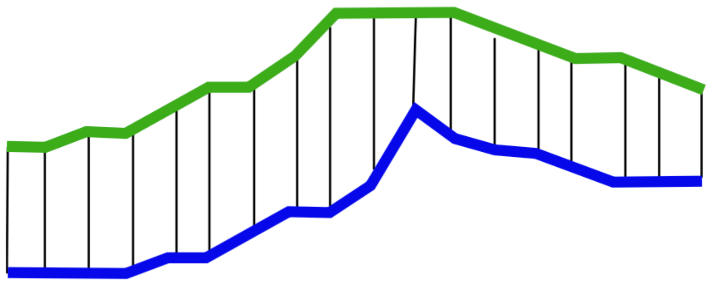
6.6 Time series clustering

- ECG, EEG
- Gait analysis
- Financial stocks
- Electric consumption
- Account balance

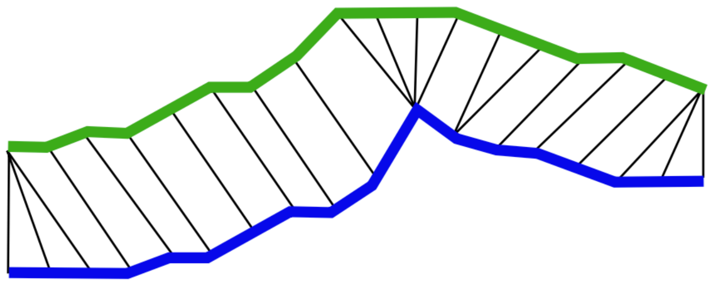


A distance metric is needed

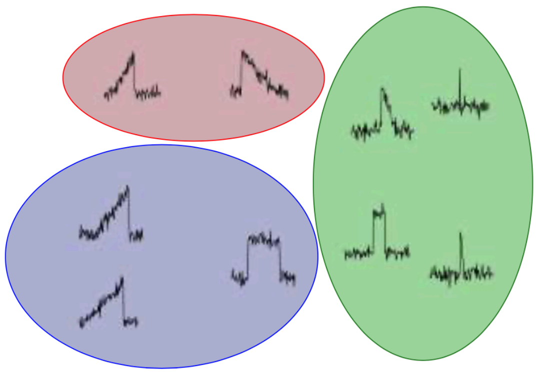
6.6 Time series clustering: distance metrics

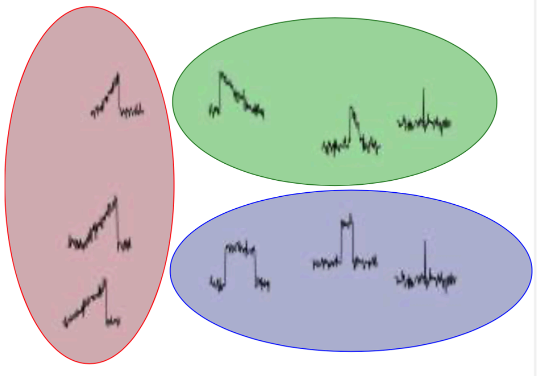


Rigid distance: Euclidean



Flexible distance: Dynamic Time Warping
- Others: Edit distance





6.6 Time series clustering: alternatives

- Represent each series by means of a set of **features** and calculate the distance between the features

- Learn a **parametric model** for each series and calculate the distance between the parameters

A nice review on time series clustering can be found here

Contents

Unit 6. Unsupervised learning

6.7 Biomedical examples

PLOS | ONE

RESEARCH ARTICLE

A clustering-based method for single-channel fetal heart rate monitoring

Encarnación Castillo[1]*, Diego P. Morales[1], Antonio García[1], Luis Parrilla[1], Víctor U. Ruiz[1], José A. Álvarez-Bermejo[2]

1 Department of Electronics and Computer Technology, Campus Universitario Fuentenueva, University of Granada, Granada, Spain, **2** Department of Informatics, University of Almería, Almería, Spain

ensure fetal well-being during pregnancy. This paper introduces a procedure for fetal heart rate extraction from a single-channel abdominal ECG signal. The procedure is composed of three main stages: a method based on wavelet for signal denoising, a new clustering-based methodology for detecting fetal QRS complexes, and a final stage to correct false positives and false negatives. The novelty of the procedure thus relies on using clustering techniques to classify singularities from the abdominal ECG into three types: maternal QRS complexes, fetal QRS complexes, and noise. The amplitude and time distance of all the local maxima followed by a local minimum were selected as features for the clustering classification. A

6.7 Biomedical examples

frontiers in Computational Neuroscience

SYSTEMATIC REVIEW
published: 24 May 2019
doi: 10.3389/fncom.2019.00031

Check for updates

The Application of Unsupervised Clustering Methods to Alzheimer's Disease

Hany Alashwal[1†], Mohamed El Halaby[2†], Jacob J. Crouse[3], Areeg Abdalla[2] and Ahmed A. Moustafa[4]*

have been applied to datasets of neurological diseases, especially Alzheimer's disease (AD). The aim is to provide insights into which clustering technique is more suitable for partitioning patients of AD based on their similarity. This is important as clustering algorithms can find patterns across patients that are difficult for medical practitioners to find. We further discuss the implications of the use of clustering algorithms in the treatment of AD. We found that clustering analysis can point to several features that underlie the conversion from early-stage AD to advanced AD. Furthermore, future work

6.7 Biomedical examples

Estiri *et al. BMC Medical Informatics and Decision Making* (2019) 19:142
https://doi.org/10.1186/s12911-019-0852-6

BMC Medical Informatics and Decision Making

TECHNICAL ADVANCE Open Access

A clustering approach for detecting implausible observation values in electronic health records data

Hossein Estiri[1,2*], Jeffrey G. Klann[1,2] and Shawn N. Murphy[1,2,3]

Abstract

Background: Identifying implausible clinical observations (e.g., laboratory test and vital sign values) in Electronic Health Record (EHR) data using rule-based procedures is challenging. Anomaly/outlier detection methods can be applied as an alternative algorithmic approach to flagging such implausible values in EHRs.

Methods: The primary objectives of this research were to develop and test an unsupervised clustering-based anomaly/outlier detection approach for detecting implausible observations in EHR data as an alternative algorithmic solution to the existing procedures. Our approach is built upon two underlying hypotheses that, (i) when there are large number of observations, implausible records should be sparse, and therefore (ii) if these data are clustered

6.7 Biomedical examples

SCIENTIFIC REPORTS

OPEN

Received: 28 January 2016

Accepted: 27 April 2016

Published: 17 May 2016

Deep Patient: An Unsupervised Representation to Predict the Future of Patients from the Electronic Health Records

Riccardo Miotto[1,2,3], Li Li[1,2,3], Brian A. Kidd[1,2,3], Joel T. Dudley[1,2,3]

Secondary use of electronic health records (EHRs) promises to advance clinical research and better inform clinical decision making. Challenges in summarizing and representing patient data prevent widespread practice of predictive modeling using EHRs. Here we present a novel unsupervised deep feature learning method to derive a general-purpose patient representation from EHR data that facilitates clinical predictive modeling. In particular, a three-layer stack of denoising autoencoders was used to capture hierarchical regularities and dependencies in the aggregated EHRs of about 700,000 patients from the Mount Sinai data warehouse. The result is a representation we name "deep patient". We evaluated this representation as broadly predictive of health states by assessing the