Machine Learning

Clustering

Fernando Rodríguez Sánchez

ferjorosa@gmail.com

Universidad Politécnica de Madrid

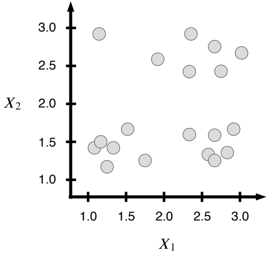
16/10/2020

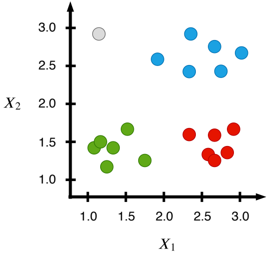
Table of contents

Table of contents

Introduction

Explore data by identifying groups of entities that are similar to each other





Homogeneity $\rightarrow$ "compact" groups

Heterogeneity $\rightarrow$ "separated" groups

Introduction

	X_1	$\dots$	X_m	C
$(\mathbf{x}^{(1)}, ?)$	$x_1^{(1)}$	$\dots$	$x_n^{(1)}$	$?$
$(\mathbf{x}^{(2)}, ?)$	$x_1^{(2)}$	$\dots$	$x_n^{(2)}$	$?$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$(\mathbf{x}^{(m)}, ?)$	$x_1^{(m)}$	$\dots$	$x_n^{(m)}$	$?$

Reminder: we have no labels

Types of clustering

Hierarchical

- **Agglomerative**
- **Divisive**

Non-hierarchical

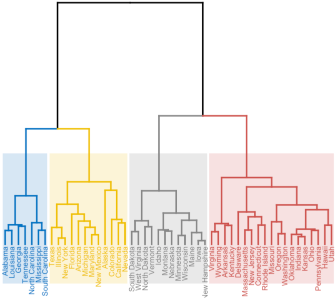
- **Partitional**
- **Probabilistic**
- **Density-based**

Table of contents

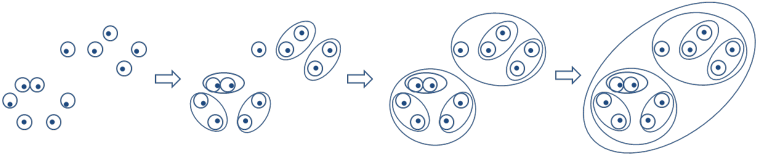
Hierarchical clustering

Hierarchical clustering assumes that data can be grouped in a tree-like manner

- Agglomerative
- Divisive

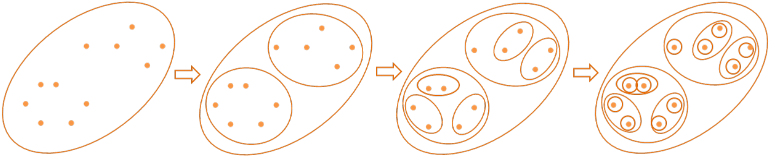


Agglomerative hierarchical clustering



1. Assign each entity to its own cluster
2. Compute similarity between each cluster
3. Join the two most similar clusters
4. Repeat steps 2 and 3 until there is only a single cluster

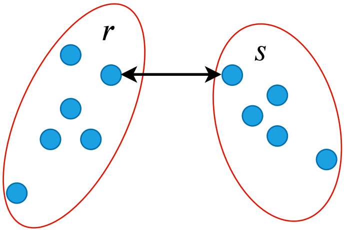
Divisive hierarchical clustering



1. Assign all entities to a single cluster
2. Partition the cluster into the two least similar clusters
3. Repeat step 2 until there is one cluster for each observation

Single linkage

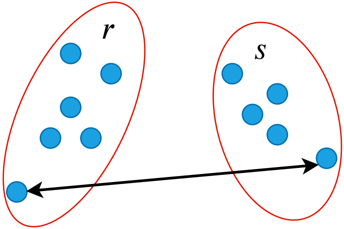
The distance between two clusters is the **shortest** distance



$$L(r, s) = min(D(x_{ri}, x_{sj}))$$

Complete linkage

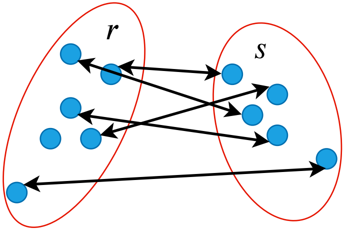
The distance between two clusters is the **longest** distance



$$L(r, s) = max(D(x_{ri}, x_{sj}))$$

Average linkage

The distance between two clusters is the **average** distance



$$L(r,s) = \frac{1}{n_r n_s} \sum_{i=1}^{n_r} \sum_{j=1}^{n_s} D(x_{ri}, x_{sj}))$$

Strengths and weaknesses

Strengths

- Easy to understand
- Fast
- Hierarchical (genomics)

Weaknesses

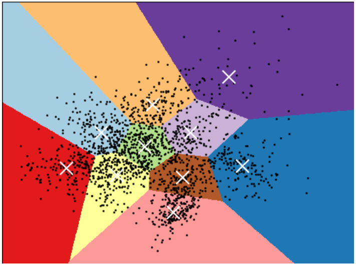
- Arbitrary number of clusters
- Lack of outlier detection
- Doesn't handle uncertainty

Table of contents

Partitional clustering

Partitional clustering generates K clusters where

- K must be known a priori
- Each entity belongs to a single cluster



General procedure

1. Select K initial centroids
2. Assign each entity to its closest cluster (centroid)
3. Update centroids ("center" of the cluster)
4. Repeat this process until centroids converge

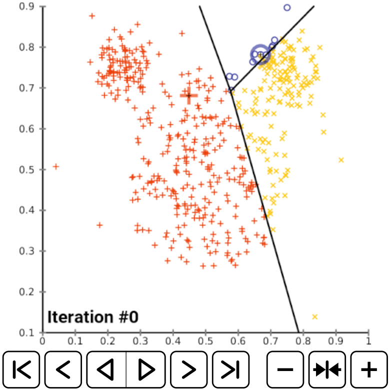


Figure 1: K-means algorithm

Multiple methods

K-means

- Centroid is a "new" point
- Euclidean distance: $\sum_{i=1}^{m} \min_{\mu_k \in C}(\| x_i - \mu_k \|_2)$

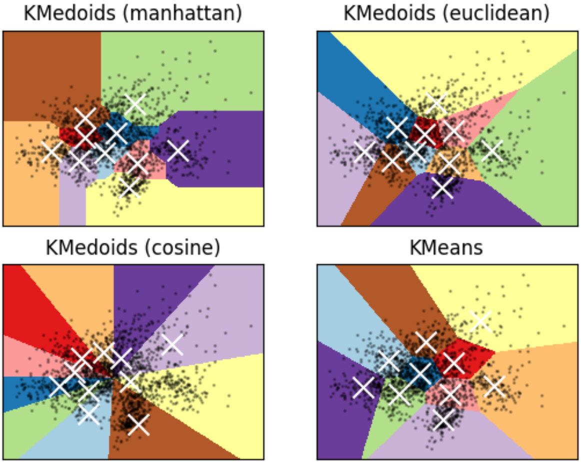
K-medians

- Centroid is a "new" point
- Manhattan distance: $\sum_{i=1}^{m} \min_{\mu_k \in C}(\| x_i - \mu_k \|)$

K-medoids

- Centroid is one of the points
- Any distance

Multiple methods



Strengths and weaknesses

Strengths

- Easy to understand
- Fast

Weaknesses

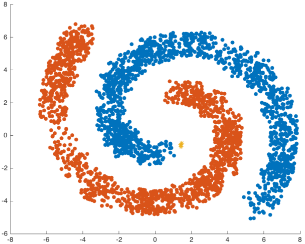
- Arbitrary number of clusters
- Lack of outlier detection
- Doesn't handle uncertainty

Table of contents

Density-based clustering

Density-based clustering is based on the idea that a cluster in a data space is a contiguous region of high point density

- DBSCAN
- OPTICS
- BIRCH

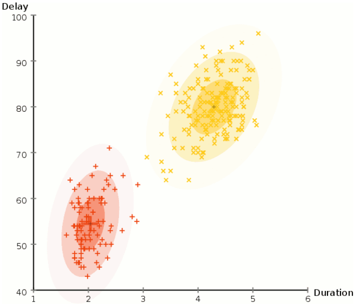


Link to graphical example

Table of contents

Probabilistic clustering

In probabilistic clustering each cluster has an associated **probability distribution**



Probabilistic clustering

	X_1	$\dots$	X_m	$p(c_1 \mid \mathbf{x}^{(i)})$	$\dots$	$p(c_K \mid \mathbf{x}^{(i)})$
$(\mathbf{x}^{(1)}, ?)$	$x_1^{(1)}$	$\dots$	$x_n^{(1)}$	?	$\dots$	?
$(\mathbf{x}^{(2)}, ?)$	$x_1^{(2)}$	$\dots$	$x_n^{(2)}$	?	$\dots$	?
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$(\mathbf{x}^{(m)}, ?)$	$x_1^{(m)}$	$\dots$	$x_n^{(m)}$	?	$\dots$	?

$$\hat{c} = \arg\max_c p(c \mid \mathbf{x}^{(i)}) \text{ where } c \in \{c_1, \dots, c_K\}$$

Data is assumed to be generated by a mixture of K conditional probability distributions (one for each cluster)

$$p(\boldsymbol{X}) = \sum_{k=1}^{K} p(c_k)\, p(\boldsymbol{X} \mid c_k)$$

Gaussian mixture model

Mixture of multivariate Gaussian distributions:

$$p(\boldsymbol{X}) = \sum_{k=1}^{K} \pi_k \, \mathcal{N}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$$

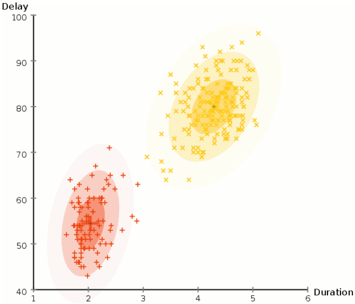
Parameters $\boldsymbol{\theta} = \{\boldsymbol{\Pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}\}$:

$\boldsymbol{\Pi} = \{\pi_1, \ldots, \pi_K\}$

$\pi_k \rightarrow$ mixture weight

$\boldsymbol{\mu}_k \rightarrow$ mean vector

$\boldsymbol{\Sigma}_k \rightarrow$ covariance matrix



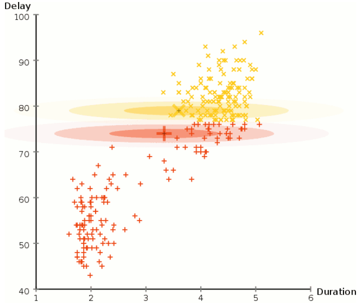
Gaussian mixture model

Learning process:

- EM algorithm

Determine number of clusters:

- BIC criterion
- AIC criterion



Strengths and weaknesses

Strengths

- Handles uncertainty
- Selects the number of clusters
- Allows statistical inference

Weaknesses

- Can be computationally expensive
- Lack of outlier detection

Table of contents

Evaluation measures

Metric Name	Knowledge of Groud Truth
Adjusted Rand index	Yes
Mutual Information based scores	Yes
Homogeneity, completeness and V-measure	Yes
Fowlkes-Mallows scores	Yes
Silhouette Coefficient	No
Calinski-Harabaz Index	No
Davies-Bouldin Index	No
Contingency Matrix	Yes

Machine Learning

Clustering

Fernando Rodríguez Sánchez

ferjorosa@gmail.com

Universidad Politécnica de Madrid

16/10/2020