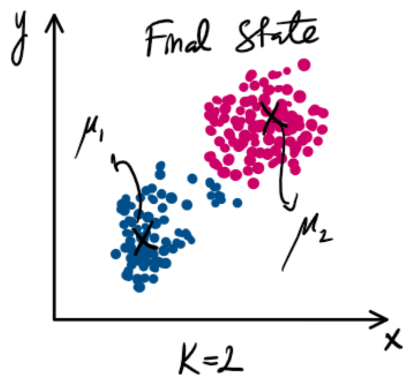
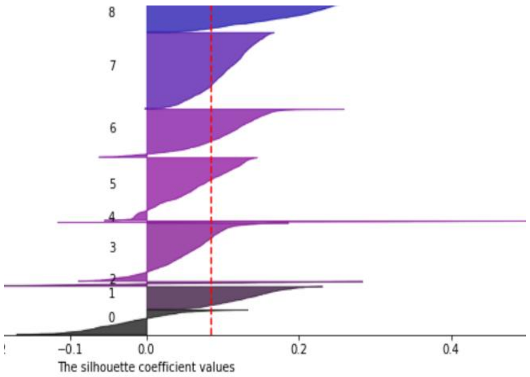
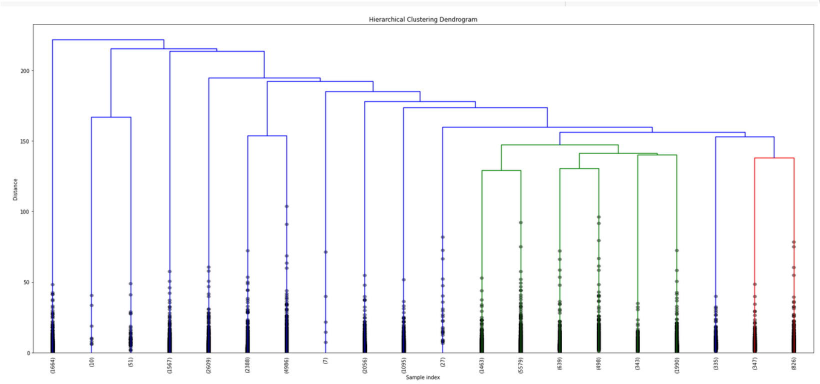
Data Driven Engineering I: Machine Learning for Dynamical Systems

Analysis of Static Datasets II: Clustering

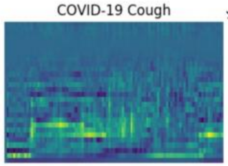
Institute of Thermal Turbomachinery
Prof. Dr.-Ing. Hans-Jörg Bauer

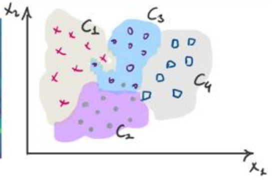






Learning landscape :





Machine Learning

<<<<< Nature of the feedback >>>>>

Predictive → Supervised

Descriptive → Unsupervised

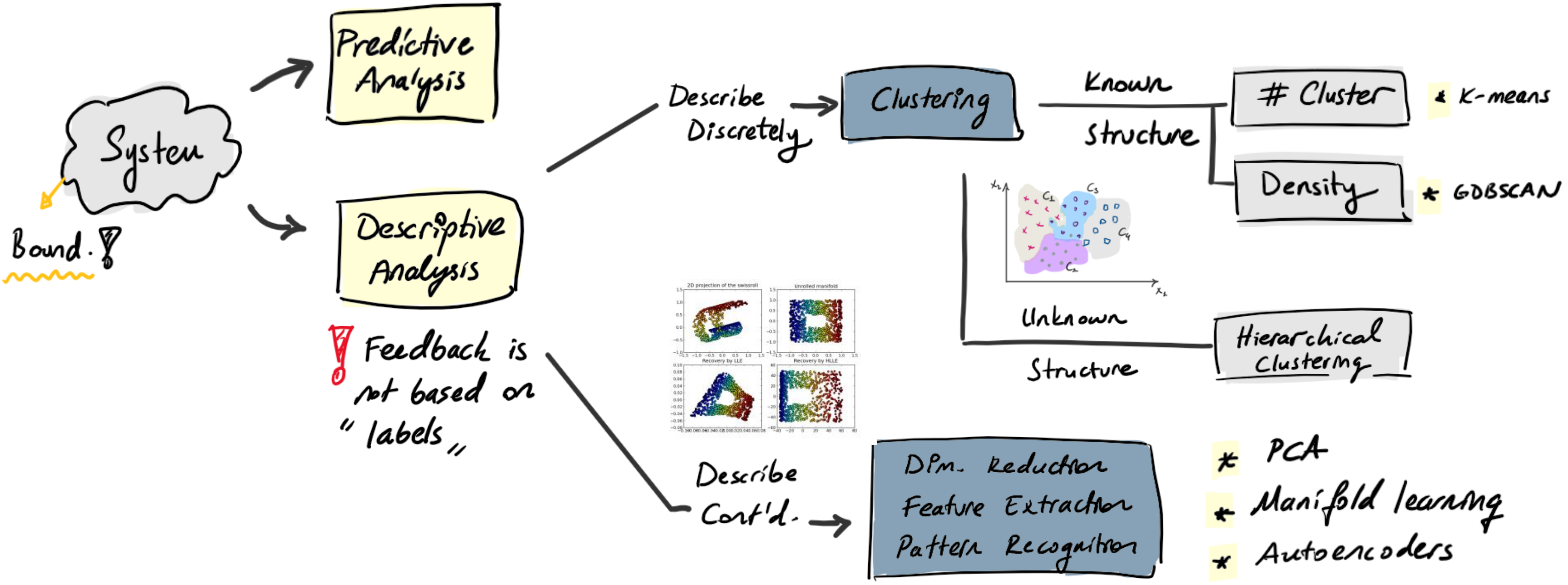
Reward / Punishment → Reinforcement

Learning types :

- ☐ Error-based learning
- ☐ Similarity-based learning
- ☐ Information-based learning
- ☐ Probability-based learning

How does it work?

"Different recipes (learning strategies) for different problems"



Today's Agenda

Basic Steps to Follow:

0.) Understand the business/task.
1.) Understand the data.
2.) Explore & prepare the data.
3.) Shortlist candidate models.
4.) ~~Training the model~~
5.) Evaluate the model predictions
6.) "Serve" the model!

Still valid

⇒ 4 major types

↪ Tricky!

#0 Understanding the task

- **Problem**: Manufacturing error in a production line
- **Modified sensory input:** 28 variables including sensory input
- 280,000 instances, where only a **small fraction** (~500) of products are **defective**.
- **Heuristic**: <0.5% is defective

A similar example for you:

"Bosch Production Line Performance
Reduce manufacturing failures"

#1 Understanding the data

Karlsruhe Institute of Technology

- ❑ Check the data source: understand what the data refers to
- ❑ Objective: understand the characteristics of the data
- ❑ Look at the feature columns:
  - ❑ Any missing values?
  - ❑ Any features with NaN values?
  - ❑ Uniqueness of the dataset? (“cardinality”)

=> Colab

```
 23   S23      284807 non-null  float64
 24   S24      284807 non-null  float64
 25   S25      284807 non-null  float64
 26   S26      284807 non-null  float64
 27   S27      284807 non-null  float64
 28   S28      284807 non-null  float64
 29   Class    284807 non-null  object
dtypes: float64(29), object(1)
memory usage: 65.2+ MB
time: 54.5 ms
```

	Time	S1	S2	S3	S4	S5	S6
count	284807.000000	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05	2.848070e+05
mean	94813.859575	1.758743e-12	-8.252298e-13	-9.636929e-13	8.316157e-13	1.591952e-13	4.247354e-13
std	47488.145955	1.958696e+00	1.651309e+00	1.516255e+00	1.415869e+00	1.380247e+00	1.332271e+00
min	0.000000	-5.640751e+01	-7.271573e+01	-4.832559e+01	-5.683171e+00	-1.137433e+02	-2.616051e+01
25%	54201.500000	-9.203734e-01	-5.985499e-01	-8.903648e-01	-8.486401e-01	-6.915971e-01	-7.682956e-01
50%	84692.000000	1.810880e-02	6.548556e-02	1.798463e-01	-1.984653e-02	-5.433583e-02	-2.741871e-01
75%	139320.500000	1.315642e+00	8.037239e-01	1.027196e+00	7.433413e-01	6.119264e-01	3.985649e-01
max	172792.000000	2.454930e+00	2.205773e+01	9.382558e+00	1.687534e+01	3.480167e+01	7.330163e+01

```
time: 447 ms
```

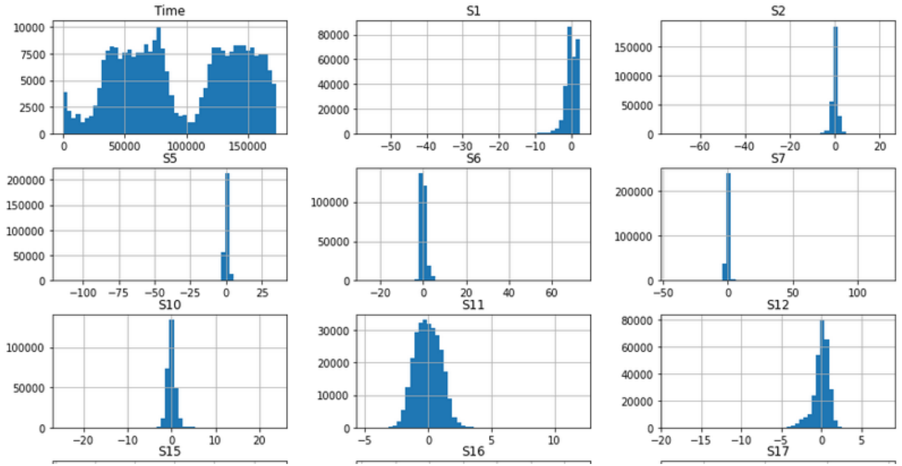
Ateliers & Saveurs in Montreal

colab

#2 Exploring the data

- **Objective**: generate a data quality report

- Using standard statistical measures of central tendency and variation
  - tabular data and visual plots
  - mean, mode, and median
  - standard deviation and percentiles
  - bars, histograms, box and violin plots



- ✓ Missing values,
- ✓ Irregular cardinality problems,
  - 1 or comparably small
- ✓ Outliers
  - invalid outliers and valid outliers

#2 Exporing the data: Correlation Matrix

- Shows the correlation between each pair of features

$$Cov(a,b) = \frac{1}{n-1}\sum_{i=1}^{n}\left[(a_i-\bar{a})\times(b_i-\bar{b})\right]$$

Features

instance

mean

mean

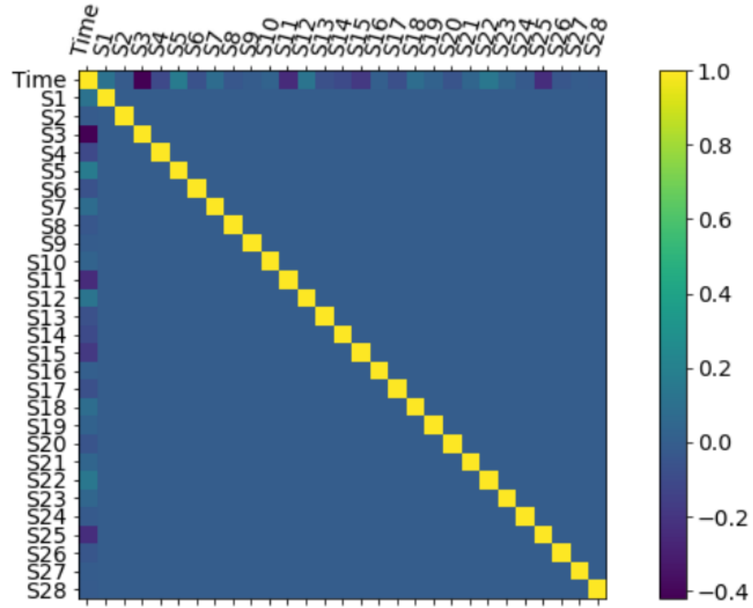
- Normalized form of “covariance”

$$Corr(a,b) = \frac{Cov(a,b)}{SD(a)\times SD(b)}$$

* Normalized
* Dimensionless

Easy to interpret

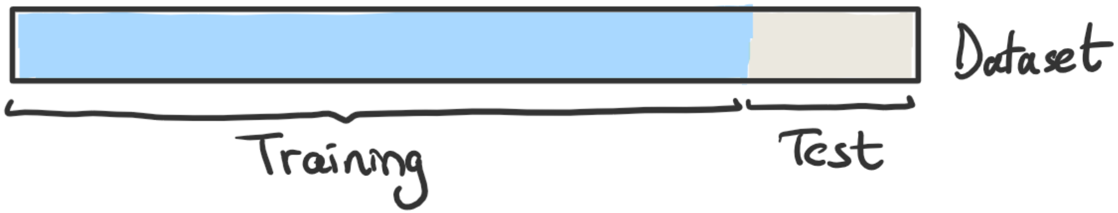
- Ranges between −1 and +1



time: 975 ms

#2 Preparing the Data

- Clustering >> unsupervised >> **training & test split not needed**



- We will use it to **reduce the volume of the data** when needed:

```
[ ]  X_train, X_test, y_train, y_test = train_test_split(dataX,
     dataY, test_size=0.9,
     random_state=2020, stratify=dataY)

     time: 188 ms
```

#3 Candidate models: k-Means

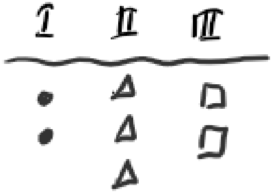
Clustering:

* Grouping similar objects together.

input → Model → Output

Partitional Clustering | Hierarchical Clustering

Partitional Clustering

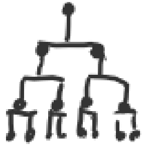


* $\tilde{O}(m \cdot n)$

(–) Need to know "k"

(–) Sensitive to initial cond.

Hierarchical Clustering

* 

"nested tree"

* $\tilde{O}(n^2 \log(n))$

(+) does not need "k" at the beginning

(–) Always work even for white noise!

#3 Candidate models: k-Means 2

k-means :

* partitioning n observation into 'k' clusters.

(-) 'k' is typically unknown ⇒ parametric analysis

* define a similarity distance.

* k-means is iterative & depends on its initialization

#3 Candidate models: k-Means 3

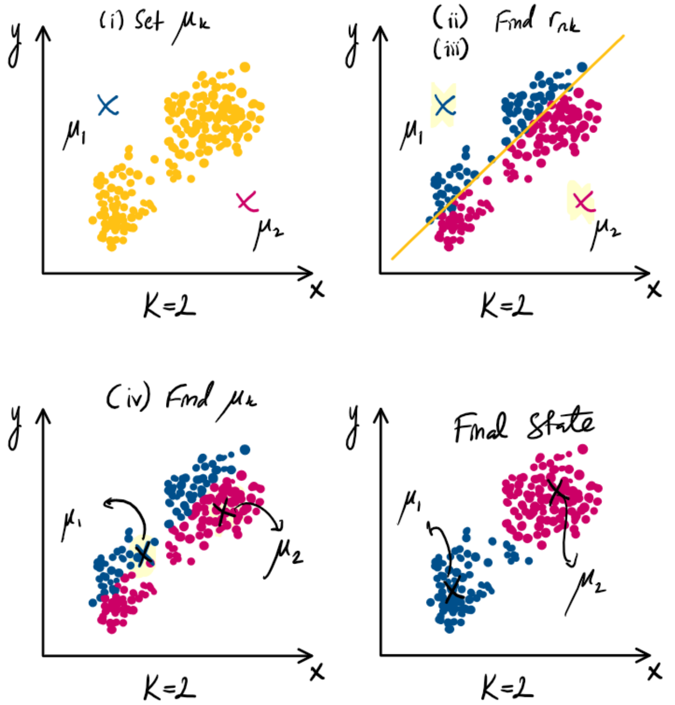
Algorithm:

(i) Assume a center of cluster for k cluster: μ_k.

(ii) Compute the distance between each observation X & μ

(iii) Label each observation as belonging to the nearest cluster.

(iv) Find the "center of mass" for each cluster $\longrightarrow \mu_k$



#3 Candidate models: k-Means 4

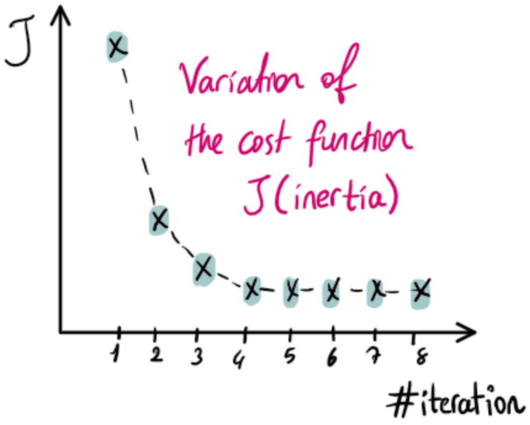
* Objective Function: $J = \sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \| x_n - \mu_k \|^2$ } find r_{nk} & μ_k minimizing J

$r_{nk} = 1$; if $n \rightarrow k$

$r_{nk} = 0$; if $n \not\rightarrow k$

① $r_{nk} = \begin{cases} 1, & \text{if } k = \arg\min_j \| x_n - \mu_j \|^2 \\ 0, & \text{otherwise} \end{cases}$ } given $\mu_j \rightarrow r_{nk}$

② $\frac{\partial J}{\partial \mu_k} = 0 \Rightarrow 2 \sum_{n=1}^{N} r_{nk} (x_n - \mu_k) = 0 \Rightarrow \mu_k = \frac{\sum_n r_{nk} x_n}{\sum_n r_{nk}}$ } "K means"



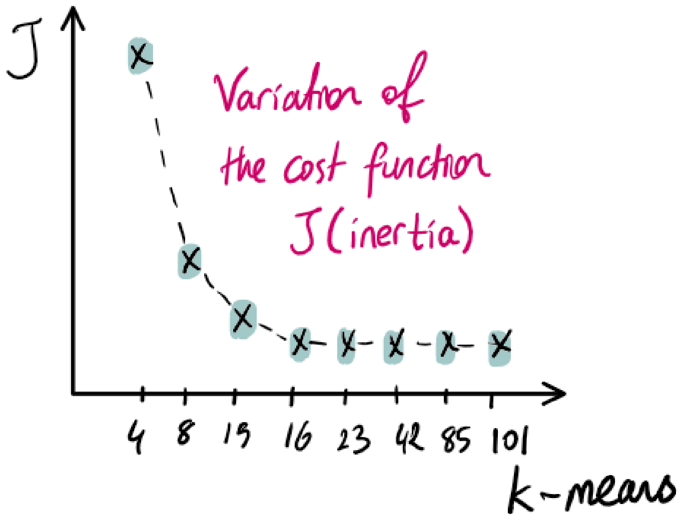
#3 Candidate models: k-Means 5

Deciding on the # Clusters

(i) Find J with increasing #k.

(ii) Look at the variation.

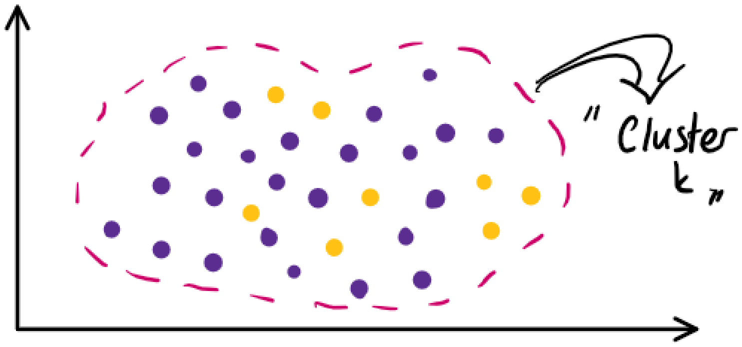
(iii) Pick a reasonable k value.



colab

#5 Evaluate model predictions

Predictive Accuracy



* Do you know a set of examples with labels?

(1) There is not any labelled data.

Silhouette Score:

* Relative distances between instances.

(2) There are some labelled data

Homogeneity (Purity):

#5 Evaluate model predictions 2

Silhouette Score:

* Relative distances between instances.

(1) $SC = b - a / \max(a, b)$

where;

a → mean intra-cluster distance

b → mean distance to the instances of the next closest cluster

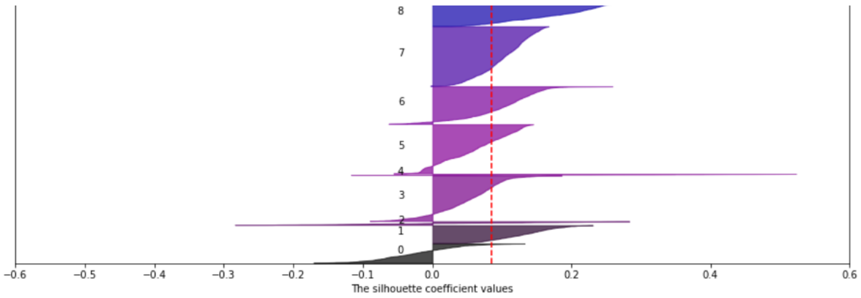
(2) Plot every instances Sil. coeff;

⇒ Silhouette Diagram

SC = +1 ⇒ Well inside in its own cluster
Away from others
$\sim b/b = 1.0$

SC = −1 ⇒ Wrong Cluster $(-a/a \sim -1)$

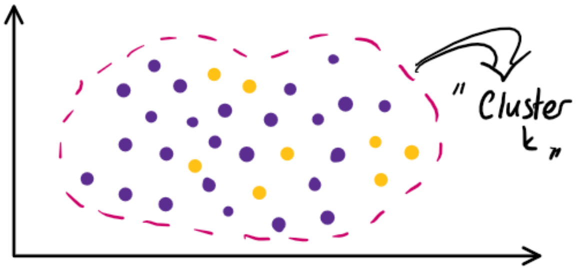
SC ≈ 0 ⇒ Near the cluster boundary $(a \sim b)$

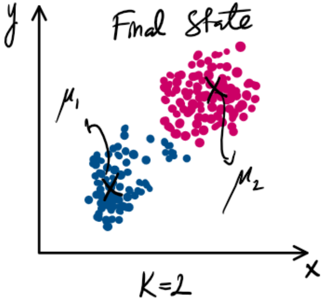


#5 Evaluate model predictions 3

Homogeneity (Purity):

* k: cluster index $(1, 2, \dots, K)$
* j: class index $(0, 1)$





(1) N_{kj} = # instances in cluster k belonging class j

(2) N_k = # instances in cluster k

(3) $H_{kj} = N_{kj} / N_k$ } homog. of cluster k for class j

(4) $H_k := \max(H_{kj})$ ⇒ homo. of cluster k

(5) $H = \sum_{k=1}^{K} N_k / N \, H_k$ ⇒ Overall homog.

colab

#3 Candidate Models: Gaussian Mixtures

Mixture Models :

* Idea : Obseration is constituted by P (Gaussian) processes

mean

$\underbrace{\text{variance}}_{\text{uncertainity}}$?

$$f_i = \sum_{p=1}^{k} \alpha_p f_p$$

PDF

weight

$$f_i = \sum_{p=1}^{k} \alpha_p \mathcal{N}(x_i, \mu_p, \sigma_p)$$

Gaussian MM

#3 Candidate Models: Gaussian Mixtures 2

Mixture Models:

* Uses expectation-maximization (EM) algorithm
* Similar to k-Means: Define "k".
* Bayesian Gaussian Mixture: probabilistic interpretation
  Cluster # Optimization

Hint: EM algorithm is much slower than k-Means. Therefore, you can use k-means to determine the better initial conditions for GMM.

colab

#3 Candidate Models: Hierarchical Clustering

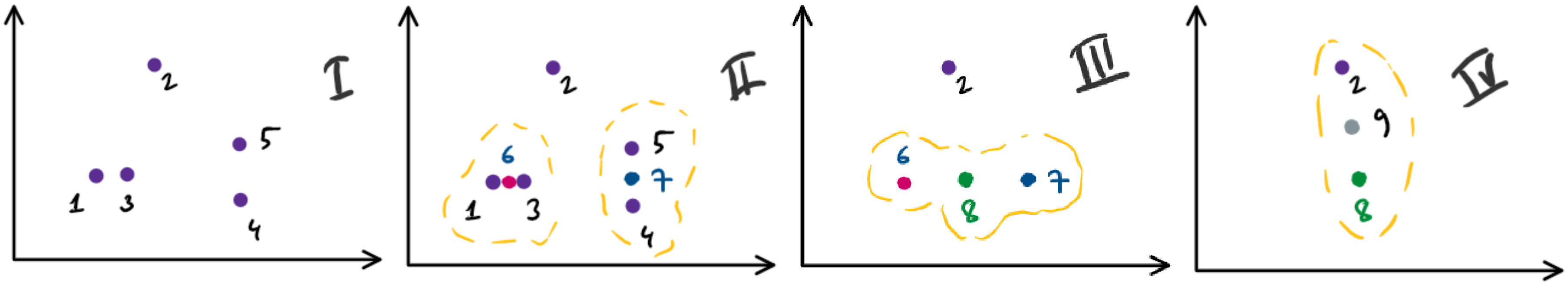
Hierarchical Clustering

* Clusters are nested:

(i) Bottom-up (agglomerative)

(ii) Top-down (divisive)

$$\begin{bmatrix}1\\2\\3\\4\\5\end{bmatrix} \rightarrow \begin{bmatrix}6\\4\\5\\2\end{bmatrix} \rightarrow \begin{bmatrix}6\\7\\2\end{bmatrix} \rightarrow \begin{bmatrix}8\\2\end{bmatrix} \rightarrow \begin{bmatrix}9\end{bmatrix}$$



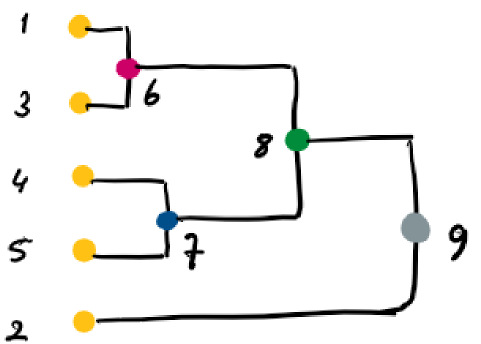
#3 Candidate Models: Hierarchical Clustering

Hierarchical Clustering

* Clusters are nested:

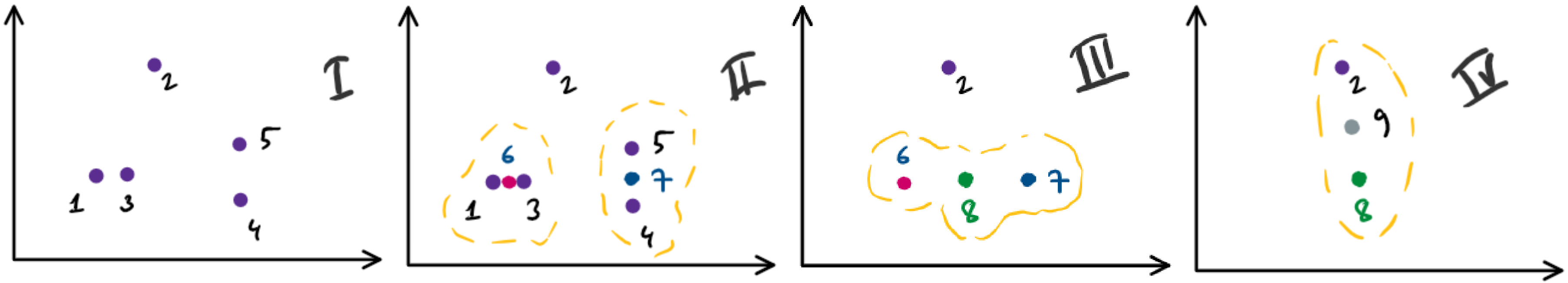
(i) Bottom-up (agglomerative)

(ii) Top-down (divisive)

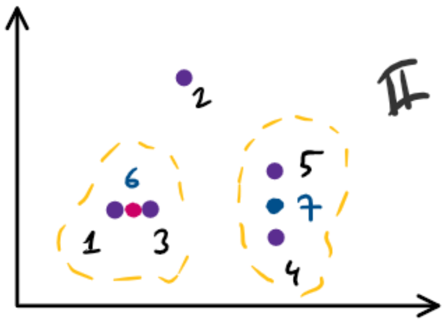


Binary Tree

"Dendrogram"



#3 Candidate Models: Hierarchical Clustering 2



Distance Calculation btw. Clusters : $\begin{bmatrix} 2 \\ 6 \\ 7 \end{bmatrix}$

* There are three options here for distance calculation:

① Single Link ⇒ nearest neighbour clustering

(✓) Distance $:= \min(d_{ij})$

} $\tilde{O}(n^2)$ time

② Complete Link ⇒ furthest neigh. clustering

(✓) Distance $:= \max(d_{ij})$

} $\tilde{O}(n^3)$ time

Compact Clusters

③ Average link ⇒ mean distance

(✓) Distance $:= \frac{1}{n_i n_j} \sum_i \sum_j d_{ij}$

} $\tilde{O}(n^3)$ time

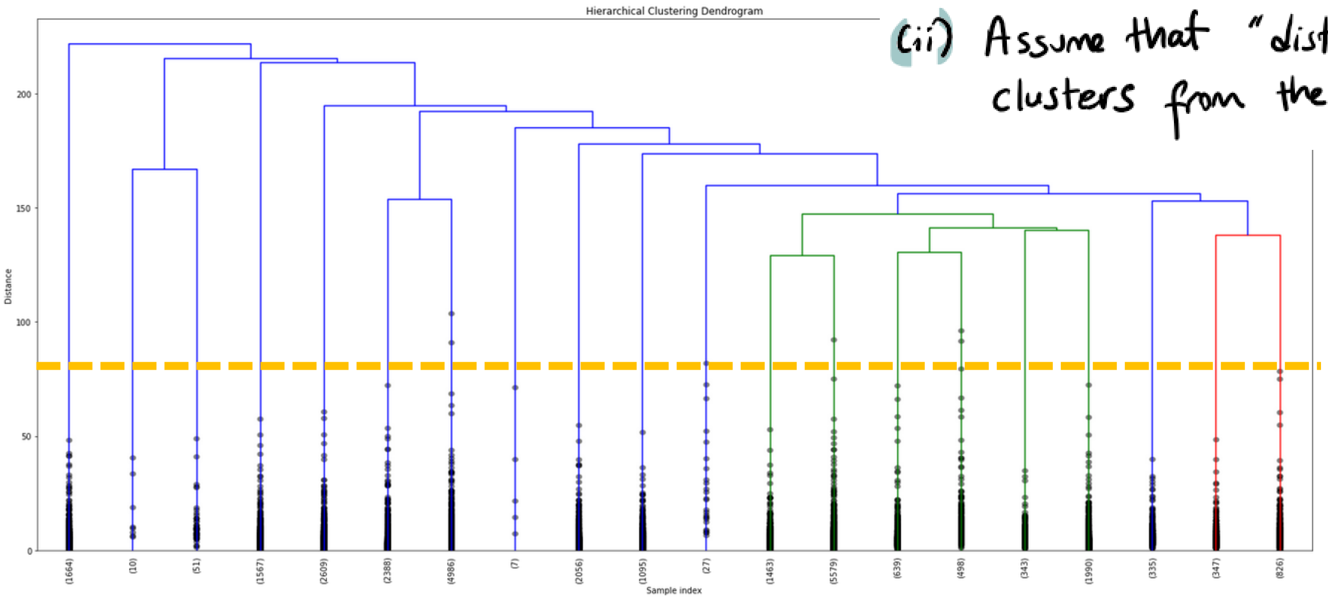
feature scaling is important

#3 Candidate Models: Hierarchical Clustering 3

* Deciding on # cluster:

(i) Select a threshold value that separates clusters in the dendrogram

(ii) Assume that "distance" segragates natural clusters from the unnatural ones.



* Alternative :

"Bayesian Hier. Clustering"


Ateliers & Saveurs in Montreal

#3 Candidate Models: DBSCAN

Density-based Clustering: DBSCAN

* Groups points that are closely packed together
  [points with many neighbours]

* Outliers ⇒ "Low density" regions

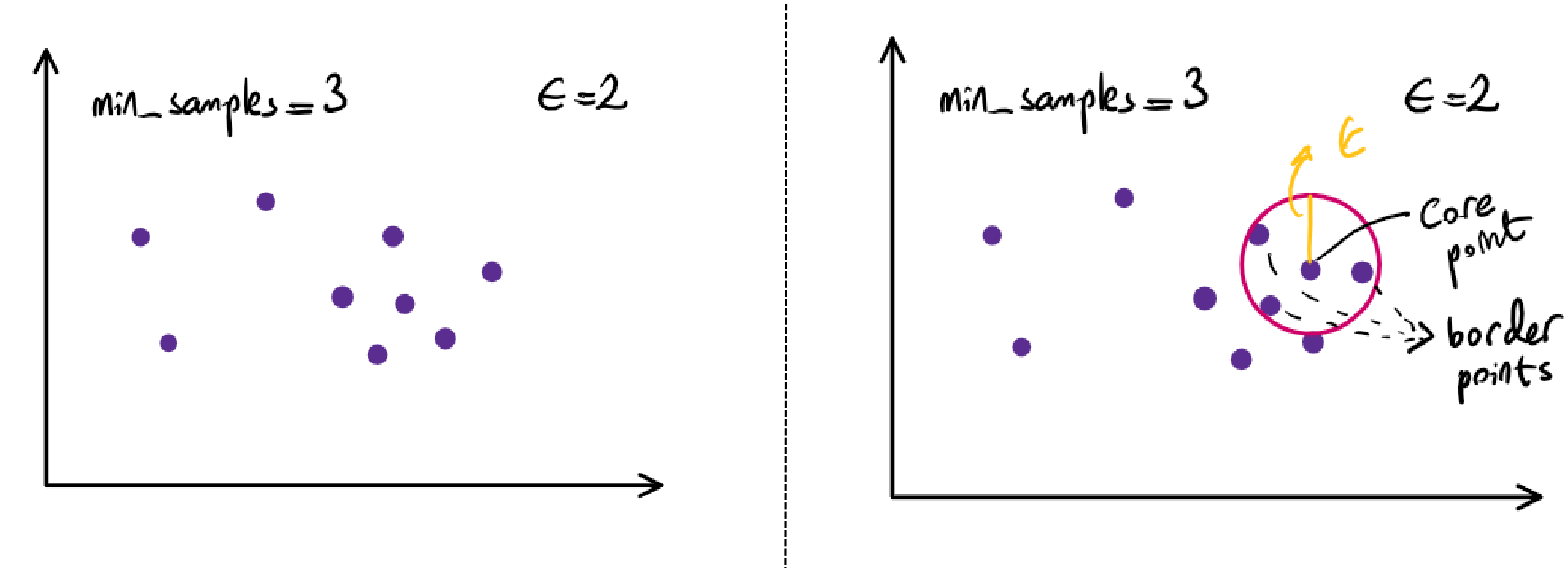
* In k-Means ⇒ all instances are assigned to a cluster k.

* In DBSCAN ⇒ • k is not needed ⇒ There is also noise class.

You need: { min # points to be considered as a dense cluster. / a distance measure to locate neighbours (ϵ)

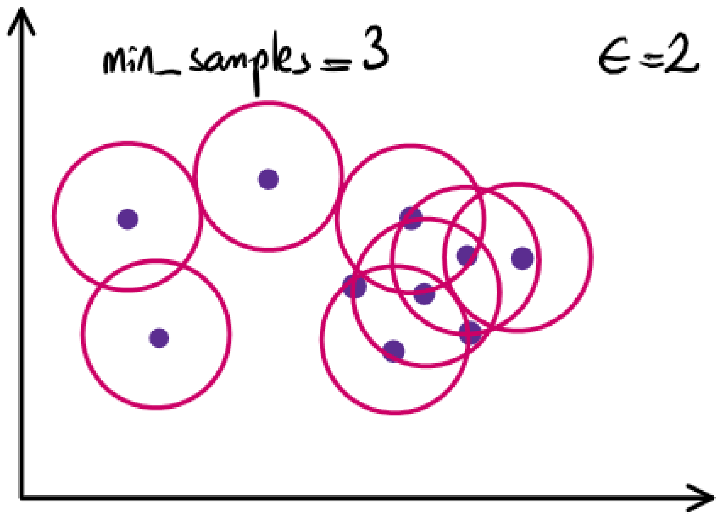
#3 Candidate Models: DBSCAN 2

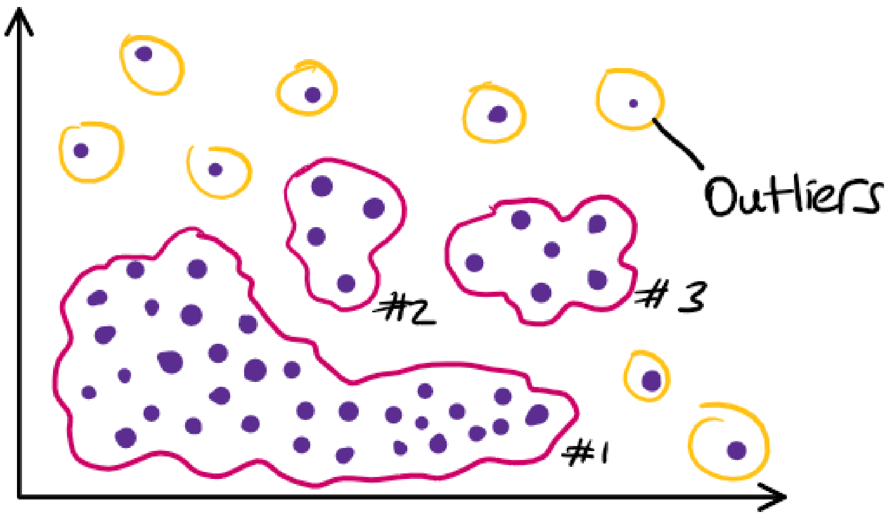
Density-based Clustering: DBSCAN



#3 Candidate Models: DBSCAN 2

Density-based Clustering: DBSCAN





colab

Additional Notes