Introduction to clustering

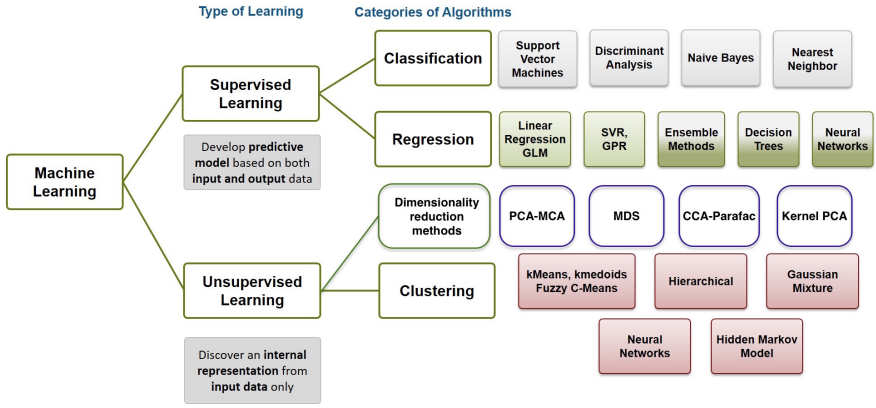
Ghislain Durif, Laurent Modolo, Franck Picard

Laboratoire Biologie et Modélisation de la Cellule, CNRS ENS-Lyon

franck.picard@ens-lyon.fr

Outline

Rough typology of ML methods



Introduction

- When data are heterogeneous, can we detect some clusters of homogeneous individuals ?
- Objective : reduce the number of individuals into cluster centers
- Clustering is a descriptive method, not explanatory

$$\underset{[n\times p]}{\mathbf{X}} = \begin{bmatrix} x_1^1 & \cdots & \cdots & x_1^p \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ x_n^1 & \cdots & \cdots & x_n^p \end{bmatrix} \rightarrow \begin{bmatrix} g_1^1 & \cdots & g_1^p \\ \vdots & \ddots & \vdots \\ g_K^1 & \cdots & g_K^p \end{bmatrix}$$

Clustering into K groups

$$\searrow \begin{bmatrix} z_1^1 & \cdots & z_1^K \\ \vdots & & \vdots \\ z_n^K & \cdots & z_n^K \end{bmatrix}$$

PCA with K axis

Underlying Hypotheses

- Some individuals are closer to others, there exists some clusters
- How to define a distance / dissimilarity between individuals ?
- Is this hypothesis realistic ? How many clusters ?
- A clustering method is defined by a number of groups, a distance between individuals, and an algorithm to define the groups

Clustering vs. Classification

Supervised Learning

- Observe $(y_1, x_1), \ldots, (y_n, x_n)$
- y_i is a label, x_i the associated data
- Construct a predictor $f : \mathcal{X} \to \mathcal{Y}$
- Define a loss function $\ell(y, f(x))$ to score predictions
- Minimize the generalization error (new y ?)

Non-Supervised Learning

- Observe $(x_1, \ldots, x_n)$
- Describe the structure of X without external information
- Group individuals ?
- Loss is more difficult to define
- How accurate is the result ?

A combinatorial nightmare

- Can we find the "best" partition ?
- If E is an ensemble with n points, partitioned into K clusters
  - $\rightarrow$ The number of partitions of E into K groupes (Stirling number)

$$p(n,k) \sim K^n/K!$$

  - $\rightarrow$ The total number of partitions of E (Bell number)

$$B_n = \sum_{k=1}^{n} p(n,k) = \frac{1}{e}\sum_{k\geq 1}\frac{k^n}{k!}$$

- The exploration of all partitions is not possible
- Algorithms will be iterative and approximate

Iterative Strategies

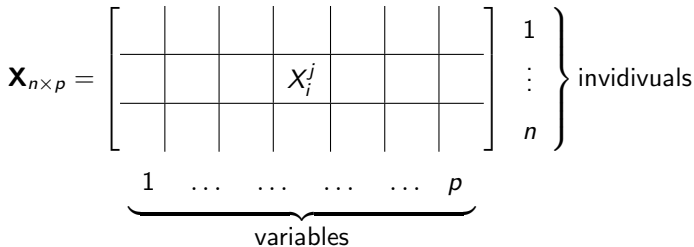
- Explore partitions and hopefully visit the best one !

Agglomerative	Hierarchical Clustering
Partitionning	K-means
Probabilistic	Mixture Models

- There exist linear and non linear clustering methods
- How to cluster unusual data like texts, networks, curves ?

High-dimensional data

$$X_i^j = \text{measurement for variable } j \text{ on individual } i$$



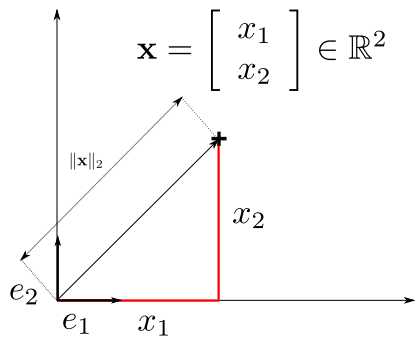
- **High dimension:** p grows and $\gg n$
- **Big Data:** n and p grow

Vectors of $\mathbb{R}^p$

- Focus on inviduals, with $\mathbf{x}_i$ a vector of $\mathbb{R}^p$ defined by a $p-$uplet $(x_1, \ldots, x_p)$ (coordinates)

$$\mathbf{x}_i \in \mathbb{R}^p, \quad \mathbf{x}_i = \sum_{j=1}^{p} x_i^j \mathbf{e}_j$$

- x_i^j is the j^{th} recording (variable) for individual i
- By default, $\mathbf{x}_i$ is a column vector, $\mathbf{x}_i'$ its transpose



Centering and scaling

- The empirical mean of $\mathbf{x}^j$:

$$\overline{\mathbf{x}}^j = \frac{1}{n}\sum_{i=1}^{n} x_i^j$$

- The vector of means is the barycenter of the data

$$\overline{\mathbf{x}} = \left[\overline{\mathbf{x}}^1, \ldots, \overline{\mathbf{x}}^p\right]$$

- To avoid scaling issues, consider the empirical variance of $\mathbf{x}^j$:

$$\text{var}(\mathbf{x}^j) = \frac{1}{n}\sum_{i=1}^{n}(x_i^j - \overline{\mathbf{x}}^j)^2$$

- Consider the scaled dataset:

$$\widetilde{\mathbf{X}}_c = \left[\frac{\mathbf{x}^1 - \overline{\mathbf{x}}^1}{\text{var}^{1/2}(\mathbf{x}^1)}, \ldots, \frac{\mathbf{x}^p - \overline{\mathbf{x}}^p}{\text{var}^{1/2}(\mathbf{x}^p)}\right]$$

Norm of a vector and basic properties

- The euclidean norm (length of a vector)

$$\|\widetilde{\mathbf{x}}_{c,i}\|_2^2 = \langle \widetilde{\mathbf{x}}_{c,i}, \widetilde{\mathbf{x}}_{c,i} \rangle = \widetilde{\mathbf{x}}_{c,i}' \widetilde{\mathbf{x}}_{c,i} = \sum_{j=1}^{p} \left(\tilde{x}_i^j \right)^2$$

- The norm of $\mathbf{x}_i$ quantifies the variability of individual i

$$\text{var}(\widetilde{\mathbf{x}}_{c,i}) = \frac{1}{n} \|\widetilde{\mathbf{x}}_{c,i}\|_2^2$$

Principal norms used in Machine Learning

- L^1 norm or Manhattan norm:

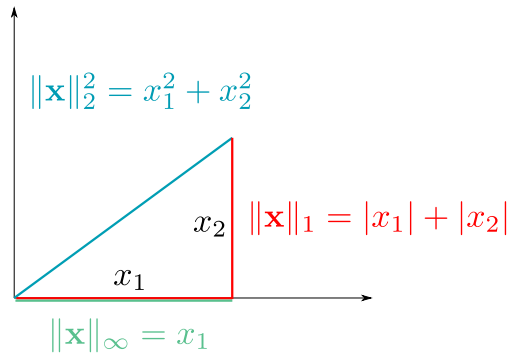
$$\|\mathbf{x}_i\|_1 = \sum_{j=1}^{p} |x_i^j|$$

- L^2 norm or Euclidian norm:

$$\|\mathbf{x}_i\|_2^2 = \sum_{j=1}^{p} (x_i^j)^2$$

- L^∞ norm or sup-norm:

$$\|\mathbf{x}_i\|_\infty = \max_{j=1,\dots,p} \left(|x_i^j|\right)$$



There are different ways to measure the norm of a vector

From norms to distances between individuals

- L^1 distance or Manhattan distance:

$$d_1(\mathbf{x}_i, \mathbf{x}_{i'}) = \|\mathbf{x}_i - \mathbf{x}_{i'}\|_1 = \sum_{j=1}^{p} |x_i^j - x_{i'}^j|$$

- L^2 distance or Euclidean distance:

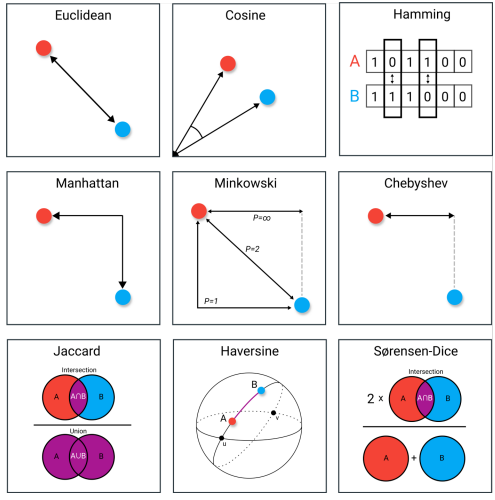
$$d_2(\mathbf{x}_i, \mathbf{x}_{i'}) = \|\mathbf{x}_i - \mathbf{x}_{i'}\|_2^2 = \sum_{j=1}^{p} (x_i^j - x_{i'}^j)^2$$

- L^∞ distance or sup-distance:

$$d_\infty(\mathbf{x}_i, \mathbf{x}_{i'}) = \|\mathbf{x}_i - \mathbf{x}_{i'}\|_\infty = \max_{j=1,\dots,p} \left(|x_i^j - x_{i'}^j| \right)$$

What drives the choice of a distance ?

- L^1 distance or Manhattan distance:
  - Adapted to discrete inputs
  - Robust to outliers
  - Non differentiable
- L^2 distance or Euclidean distance:
  - Most common, differentiable
  - Sensitive to dimension and outliers
  - Sensitive to the scale of the different inputs
- L^∞ distance or sup-distance:
  - Applied in logistical problems
  - More specific, less used



https://towardsdatascience.com/

Dissimilarities and Distances

- A dissimilarity d is defined by

$$\begin{aligned} d : E \times E &\rightarrow \mathbb{R}^+ \\ (i, i') &\rightarrow d(i, i') \end{aligned}$$

- Properties:
  - Non negativity for distinct elements $d(i, i') > 0$ if $i \neq i'$
  - Symmetry: $\forall (i, i'),\ d(i, i') = d(i', i)$
  - $d(i, i') = 0$ i.i.f $i = i'$
- Distance : additional triangular inequality

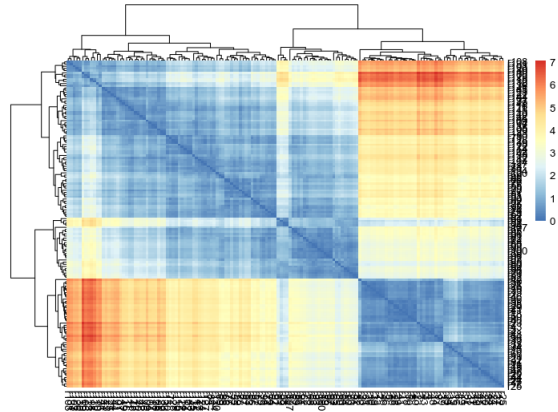
$$d(i, i') \leq d(i, i') + \leq d(i', i'')$$

Gram Matrix and distance between individuals

- Pairwise distance matrix between individuals

$$\mathbf{D} = \begin{bmatrix} \mathrm{d}(\mathbf{x}_1, \mathbf{x}_1) & \dots & \mathrm{d}(\mathbf{x}_{i'}, \mathbf{x}_i) \\ & \ddots & \\ \mathrm{d}(\mathbf{x}_i, \mathbf{x}_{i'}) & \dots & \mathrm{d}(\mathbf{x}_n, \mathbf{x}_n) \end{bmatrix}$$

- Symmetric, invertible (semi definite positive)



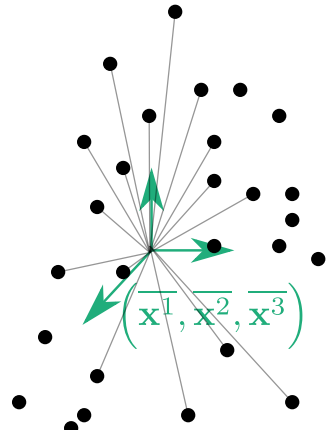
Total Inertia of a dataset

The global variance of a dataset for centered variables

$$
\begin{aligned}
I_T(\mathbf{X}) &= \frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{p}(x_i^j - \overline{\mathbf{x}}^j)^2 \\
&= \frac{1}{n}\sum_{i=1}^{n} d_2^2(\mathbf{x}_i, \overline{\mathbf{x}})
\end{aligned}
$$

For centered and scaled data

$$
I_T(\widetilde{\mathbf{X}}_c) = \frac{1}{n}\sum_{i=1}^{n} d_2^2(\widetilde{\mathbf{x}}_{c,i}, 0)
$$



Partitioning the data into clusters

- Suppose there exists K clusters
- Introduce indicator variables z_{ik}:

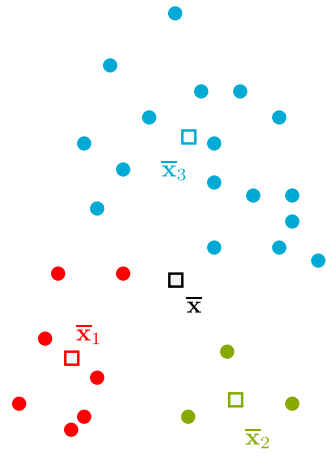
$$z_{ik} = \begin{cases} 1, & \text{if } i \in \text{cluster } k \\ 0, & \text{otherwise} \end{cases}$$

- Each cluster has size

$$n_k = \sum_{i=1}^{n} z_{ik}$$

- Each cluster has center

$$\bar{\mathbf{x}}_k = \left[\bar{\mathbf{x}}_k^1, \ldots, \bar{\mathbf{x}}_k^p\right], \quad \bar{\mathbf{x}}_k^j = \frac{1}{n_k} \sum_{i=1}^{n} z_{ik} x_i^j$$



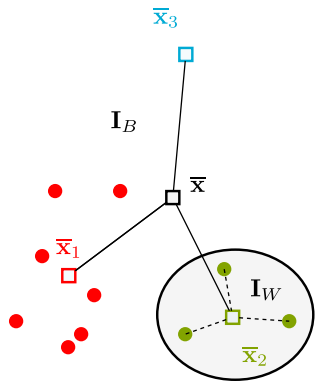
Partitioning the variance

- Between-class variance : distance of clusters barycenters to the global barycenter

$$\mathbf{I}_B = \sum_{k=1}^{K} n_k d^2(\overline{\mathbf{x}}_k, \overline{\mathbf{x}})$$

- Within-class variance: distance of points to their cluster center

$$\mathbf{I}_W = \sum_{k=1}^{K} \sum_{i=1}^{n} d^2(\mathbf{x}_i, \overline{\mathbf{x}}_k)$$

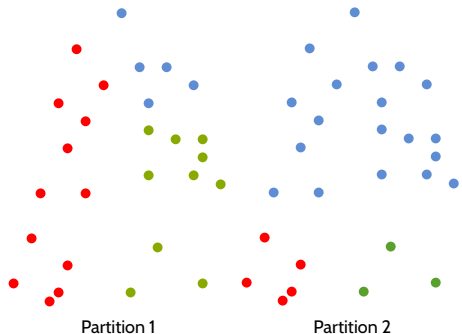


Decomposition of the total inertia

- The main theorem is that

$$\mathbf{I}_T = \mathbf{I}_W + \mathbf{I}_B$$

- $\mathbf{I}_T$ is constant for a given dataset
- $\mathbf{I}_W$ we want it to be minimal (homogeneous clusters)
- $\mathbf{I}_B$ we want it to be maximal (well separated clusters)



How to compare partitions

- Given two classifications $\mathcal{P}$ and $\mathcal{P}'$, how to compare the different results:
- Contingency tables:

$\mathcal{P}\backslash\mathcal{P}'$	cluster 1	...	cluster K'
cluster 1	n_{11}	...	$n_{1K'}$
$\vdots$			
cluster K	n_{K1}	...	$n_{KK'}$

- $n_{kk'}$ the number of individuals in cluster k of $\mathcal{P}$ and cluster k' of $\mathcal{P}'$
- Partitions are similar when the contingency table is diagonal

(Adjusted) Rand Index

- a the number of pairs in the same subset in $\mathcal{P}$ and $\mathcal{P}'$ (concordance)
- b the number of pairs in different subsets in $\mathcal{P}$ and $\mathcal{P}'$ (discordance)
- The Rand Index ($\in [0, 1]$) is

$$RI(\mathcal{P}, \mathcal{P}') = \frac{a + b}{\binom{n}{2}}$$

- The Rand Index is the percentage of correct decisions:

$$RI(\mathcal{P}, \mathcal{P}') = \frac{TP + TN}{\binom{n}{2}}$$

- The Adjusted Rand Index (preferred) is the RI adjusted for the chance grouping of elements (expected similarity of all pairwise comparison)

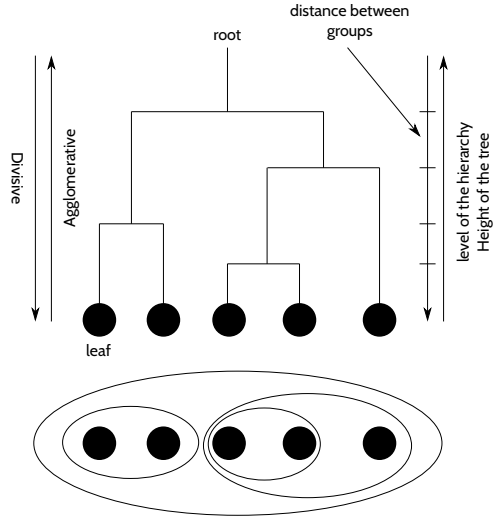
Outline

Intuitions et principles

- From a dissimilarity matrix, create groups step by step:
  - divide two groups (descending hierarchical clustering)
  - collapse two groups (ascending hierarchical clustering)
- Create hierarchies between groups (even if the nesting is not interpreted)
- Each level of the hierarchy represents a partition with disjoint groups
- The hierarchy can be represented by a tree called **dendrogram**

3 Ingredients for hierarchical clustering

- dissimilarity between individuals
- Merge the closest individuals and create groups
- dissimilarity between groups
- Merge most similar groups
- a fusion (or division) rule



Euclidian Distance and the Ward Method

- What is the best way to collapse groups ?
- Motivation: when groups are collapsed, the within inertia increases

$$\begin{aligned} \mathbf{I}_W(A,B) &= \mathbf{I}_W(A) + \mathbf{I}_W(B) = \sum_{i\in A} d^2(\mathbf{x}_i, \overline{\mathbf{x}}_A) + \sum_{i\in B} d^2(\mathbf{x}_i, \overline{\mathbf{x}}_B) \\ \mathbf{I}_W(A\cup B) &= \sum_{i\in A\cup B} d^2(\mathbf{x}_i, \overline{\mathbf{x}}_{AB}) \end{aligned}$$

- When two groups are collapsed we want to minimize this increase:

$$\mathbf{I}_W(A,B) - \mathbf{I}_W(A\cup B)$$

Ward Distance between groups

- The Ward distance between groups is defined by

$$d_{\text{Ward}}(A, B) = \frac{n_A n_B}{n_A + n_B} d^2(\bar{\mathbf{x}}_A, \bar{\mathbf{x}}_B)$$

- Using this distance minimizes the increase in the within group inertia at each step of the hierarchy
- This distance accounts for desequilibria of clusters size
- This is the default between-group distance implemented in software

Other between-groups distance

- Minimum Link (simple link)

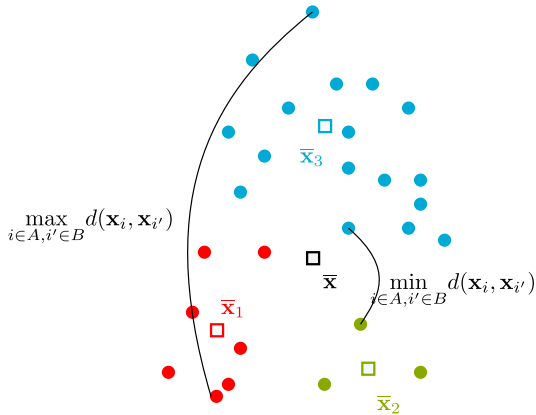
$$d(A, B) = \min_{i \in A, i' \in B} d(\mathbf{x}_i, \mathbf{x}_{i'})$$

- Maximal link :

$$d(A, B) = \max_{i \in A, i' \in B} d(\mathbf{x}_i, \mathbf{x}_{i'})$$

- Average Link

$$d(A, B) = \frac{1}{n_A \times n_B} \sum_{i \in A, i' \in B} d(\mathbf{x}_i, \mathbf{x}_{i'})$$



Other between-groups distance

- Minimum Link (simple link)

$$d(A, B) = \min_{i \in A, i' \in B} d(\mathbf{x}_i, \mathbf{x}_{i'})$$

- Maximal link :

$$d(A, B) = \max_{i \in A, i' \in B} d(\mathbf{x}_i, \mathbf{x}_{i'})$$

- Average Link

$$d(A, B) = \frac{1}{n_A \times n_B} \sum_{i \in A, i' \in B} d(\mathbf{x}_i, \mathbf{x}_{i'})$$

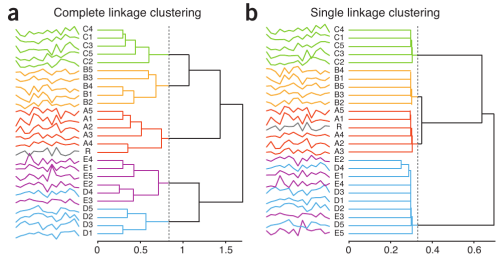


Figure 3 | Dendrograms of hierarchical clustering of gene expression profiles based on correlation distance. The data were generated by creating core profiles A1, B1, C1, D1, and E1 with correlation values of 0.7, 0.5, 0, -0.5, and -0.7 (respectively) with the reference profile R from **Figure 1**. For each core profile (e.g., A1), four additional highly correlated random profiles were generated (e.g., A2–A5). Profiles are colored by group and clusters formed by cutting at a fixed height (dashed line). (**a**) Complete linkage clustering tends to create balanced dendrograms by first clustering objects into small nodes and then clustering the nodes. (**b**) Single linkage clustering tends to create stringy dendrograms by first creating a few nodes and then adding objects to them one at a time.

Example of tree construction

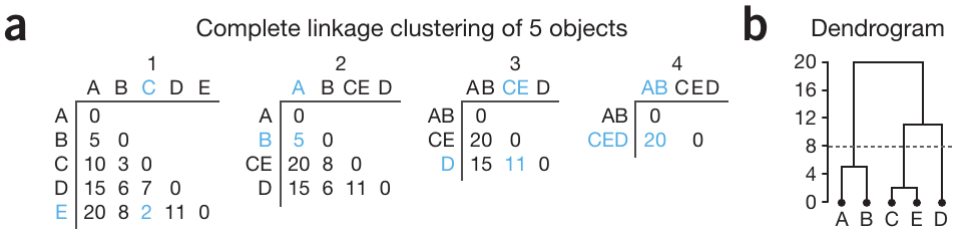


Figure 2 | Complete linkage clustering of five objects. (**a**) Pairwise distances (step 1) are used to merge objects (steps 2–4) where the maximum of all pairwise distances is used. At each merging step, the shortest distance is chosen (blue). (**b**) A dendrogram with a vertical axis showing the distance between merged nodes. To create clusters, one can cut the tree at a fixed height (dashed line).

Properties of different links

- If the cluster structure is strong, results will be comparable
- Minimum Link: only considers one observation per group: can create paquets (high within group variance)
- Maximum Link: two groups are close if all observations are close once collapsed: can create small homogeneous groups (high between-group variability)
- Average Link: trade-off between Minimum and Maximum Links

Outline

A simple idea

- One of the most used algorithm : quick and easy
- Implemented for the euclidean distance in most software
- Based on the decomposition of inertia

Le groupe le plus proche

- Consider the indicator variable z_{ik} that equals 1 if individual i is in cluster k

$$n_k = \sum_{i=1}^{n} z_{ik}$$

- Inertia boils down to

$$\begin{aligned} \mathbf{I}_W &= \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik} d^2(\mathbf{x}_i, \overline{\mathbf{x}}_k) \\ \mathbf{I}_B &= n \times \sum_{k=1} d^2(\overline{\mathbf{x}}_k, \overline{\mathbf{x}}) \end{aligned}$$

- To determine the nearest cluster for each individual

$$\widehat{z}_i = \arg\min_k \left\{ d^2(\mathbf{x}_i, \overline{\mathbf{x}}_k) \right\}$$

Two-Step algorithm

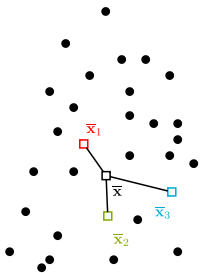
- Step $[h]$: update centers $\overline{\mathbf{x}}_1, \ldots, \overline{\mathbf{x}}_K$ when labels $\mathbf{z}^{[h]}$ are known

$$\overline{\mathbf{x}}_k^{[h+1]} = \frac{1}{n_k^{[h]}} \sum_{i=1}^{n} \widehat{z}_{ik}^{[h]} \mathbf{x}_i$$

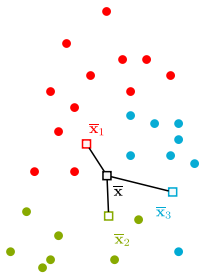
- Step $[h+1]$: update labels $\mathbf{z}$ when centers $\overline{\mathbf{x}}_1^{[h+1]}, \ldots, \overline{\mathbf{x}}_K^{[h+1]}$ are updated

$$\widehat{z}_i^{[h+1]} = \arg \min_{1,\ldots,K} \left\{ d^2(\mathbf{x}_i, \overline{\mathbf{x}}_k^{[h+1]}) \right\}$$

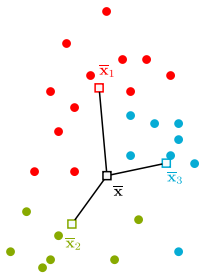
Illustration of kmeans



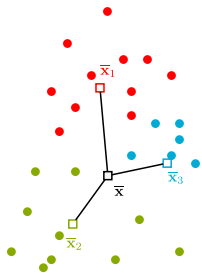
$\overline{\mathbf{x}}_k^{(0)}$



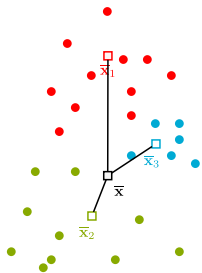
$\overline{\mathbf{z}}_k^{(1)} \mid \overline{\mathbf{x}}_k^{(0)}$



$\overline{\mathbf{x}}_k^{(1)} \mid \overline{\mathbf{z}}_k^{(1)}$



$\overline{\mathbf{z}}_k^{(2)} \mid \overline{\mathbf{x}}_k^{(1)}$



$\overline{\mathbf{x}}_k^{(2)} \mid \overline{\mathbf{z}}_k^{(2)}$

Decreasing inertia

- Denoting by $[h]$ the step h of the algorithm, $\mathbf{z}^{[h]}, \overline{\mathbf{x}}^{[h]}$
- The inertia depends on both quantities $\mathbf{I}_W(\mathbf{z}^{[h]}, \overline{\mathbf{x}}^{[h]})$
- Updating centers

$$\mathbf{I}_W(\mathbf{z}^{[h]}, \overline{\mathbf{x}}^{[h+1]}) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik}^{[h]} d^2(\mathbf{x}_i, \overline{\mathbf{x}}_k^{[h+1]})$$

- Updating labels

$$\mathbf{I}_W(\mathbf{z}^{[h+1]}, \overline{\mathbf{x}}^{[h+1]}) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_{ik}^{[h+1]} d^2(\mathbf{x}_i, \overline{\mathbf{x}}_k^{[h+1]})$$

- The criterion decreases at each step

$$\mathbf{I}_W(\mathbf{z}^{[h]}, \overline{\mathbf{x}}^{[h]}) \geq \mathbf{I}_W(\mathbf{z}^{[h]}, \overline{\mathbf{x}}^{[h+1]}) \geq \mathbf{I}_W(\mathbf{z}^{[h+1]}, \overline{\mathbf{x}}^{[h+1]})$$

Convergence & initialization

- The inertia is a bounded suite, so the algorithm converges within a finite number of steps
- The solution is only a local minimizer : depends on the initialization step
- Clustering algorithm are very sensitive to initialization (try different points)

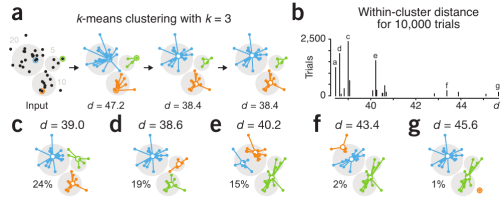
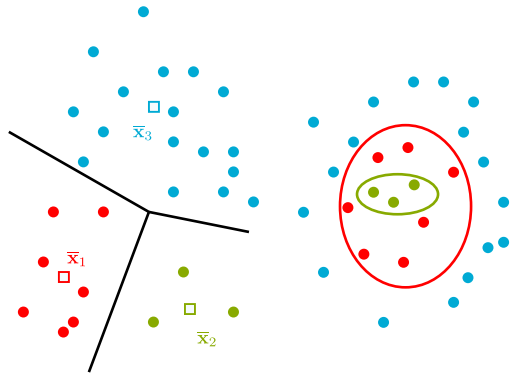


Figure 4 | Simulation of 10,000 trials of k-means clustering with $k = 3$ of 35 points (black), of which 20, 10, and 5 were centered on each of the gray circles, respectively, and spatially distributed normally within the circle with s.d. half of the circle radius. Centroids are indicated by colored hollow points; initial centroids were randomly selected points from the data set. (**a**) Evolution of a trial that results in the lowest total within-cluster distance, $d = 38.4$. With each iteration, d generally drops. Points are shown connected to and colored by their assigned centroid. (**b**) Histogram of the total within-cluster distance for 10,000 trials. The lowest $d = 38.4$ solution (**a**) was found in 1,236 (12%) of trials. Bar labels indicate figure panels in which the solution is shown. (**c**,**d**) Two most common solutions, their d and frequency observed. (**e**,**f**) Examples of solutions whose clusters do not follow the original grouping of points. (**g**) Solution with largest d.

Outline

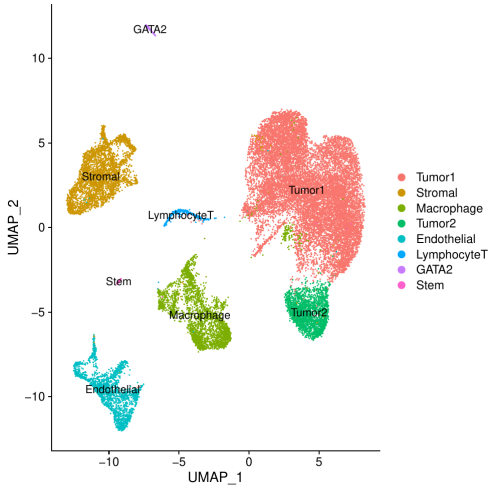
Linear vs non linear methods

- Linear methods provides clusters that can be separated by planes
- Recent developments propose to generalize clustering beyond linear methods
- Popular methods consist in constructing a proximity graph between points to represent interactions



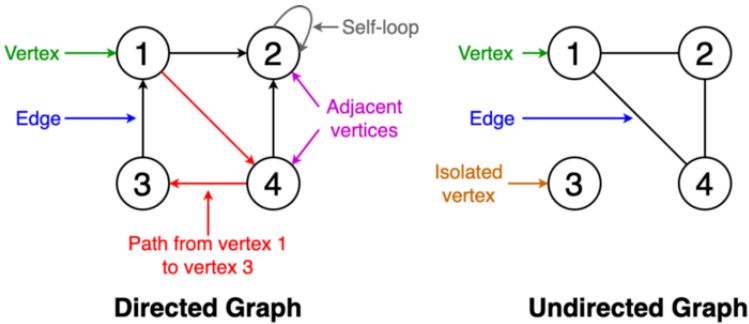
Graph-based clustering in single cell genomics

- Single Cell transcriptomic data: given cell expression, assign cells to cell-types
- Group cells according to their transcriptomic proximities
- The graph represents distances between cells



Definition of a graph

A graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is defined by a set of vertices $\mathcal{V} = \{1, \dots, n\}$ and a set of edges $\mathcal{E} = \{(i,j) \in \mathcal{V}^2, \quad i \sim j\}$, directed or non directed



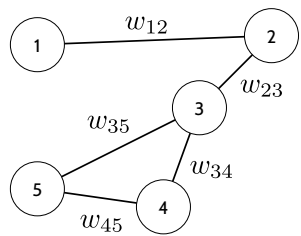
Basic Features of graphs

- Define the adjacency matrix of a graph

$$A = \begin{cases} w_{ij} \in \mathbb{R}, \text{ if } i \sim j \\ 0 \text{ otherwise} \end{cases}$$

- A graph can be binary: $w_{ij} \in \{0, 1\}$
- Or weighted : $w_{ij} \in \mathbb{R}$

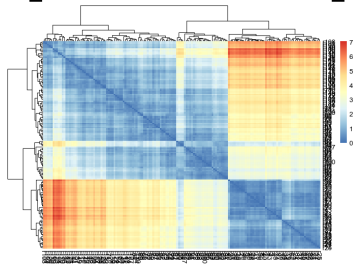
$$A = \begin{bmatrix} w_{11} & \dots & w_{1n} \\ \vdots & & \\ w_{n1} & \dots & w_{nn} \end{bmatrix}$$



From a dissimilarity matrix to a graph

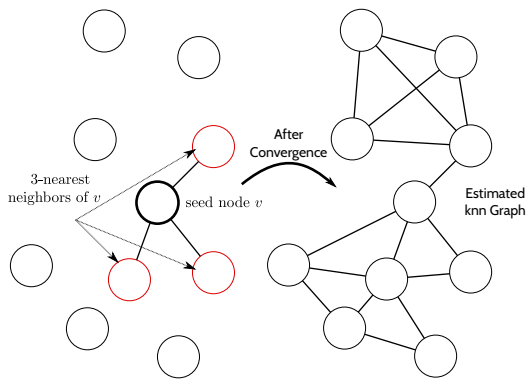
- How to construct the graph from the data **X**?
- From any dissimilarity matrix (Gram Matrix)
- Most popular method : neighborhood graph (kNN graph)
- Clustering can be restated as finding clusters of vertices

$$A = \begin{bmatrix} w_{11} & \dots & w_{1n} \\ \vdots & & \\ w_{n1} & \dots & w_{nn} \end{bmatrix}$$



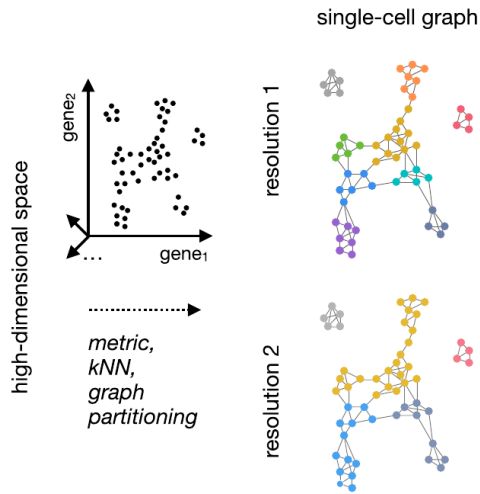
knn-Graphs

- Consider a dissimilarity matrix
- Choose a number of neighbors (resolution parameter)
- For a given vertex, consider the k nearest neighbors
- Construct the proximity graph iteratively
- Can also consider shared neighborhoods
- Sparsification of the original dense graph
- Need efficient methods on large datasets



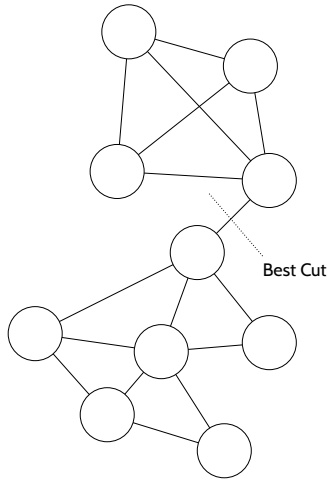
Clustering strategy based on modules/community

- In a network, a module is a densely connected subgraph
- It is a quantitative definition (maximum: clique)
- Modularity: what is the connectivity of nodes vs random connectivity
- Find clusters such that the modularity is maximal



The Graph-Cut problem

- How to partition a graph into subgraphs with a given objective ?
- The `size` of a cut is the number of cut edges
- Clustering by graph-cuts: smallest cut that make homogeneous subgraphs

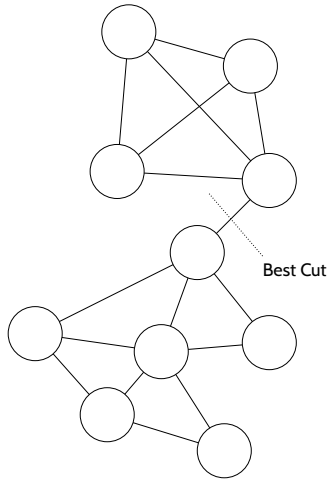


Finding the best cut

- $\text{Vol}(S)$ volume of subgraph S (nb of nodes)
- $\text{Cut}(S, S')$ number of edges that link two subgraphs S and S'
- The normalized cut value:

$$\text{NormCut}(S, S') = \frac{\text{Cut}(S, S')}{\text{Vol}(S)} + \frac{\text{Cut}(S, S')}{\text{Vol}(S')}$$

- Avoids cuts that generate too-small subgraphs
- The combinatorial complexity is too high, need heuristics

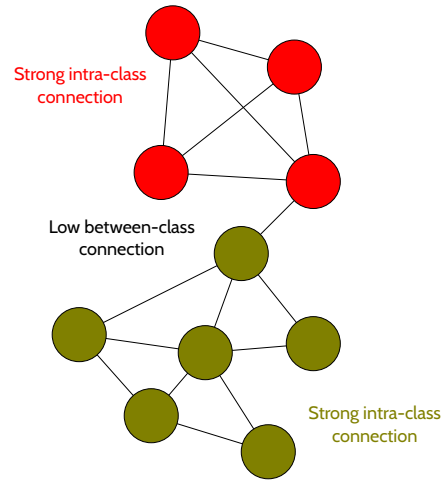


Modularity optimization using the Louvain Algorithm

- Approximation of a graph-cut problem
- A cluster is equivalent to a module
- If K clusters (module) with indicator variables z_{ik}, the modularity is

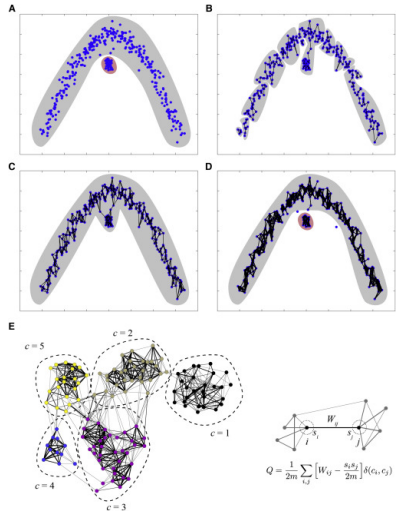
$$M_K(\mathbf{z}) = \frac{1}{2m} \sum_{k=1}^{K} \sum_{\ell=1}^{K} z_{ik} z_{j\ell} \left(A_{ij} - \frac{d_i d_j}{2m} \right)$$

- $d_i = \sum_j A_{ij}$, $m = \sum_{ij} A_{ij}$
- Find $\mathbf{z}$ such that $M_K(\mathbf{z})$ is maximal



Extensions and generalizations

- The Louvain algorithm is one example of graph-based clustering methods
- Widely used in single cell data analysis
- Many hyper parameters to tune
- Non Linear clustering is a very active field of research



Outline

Clustering analysis in a nutshell

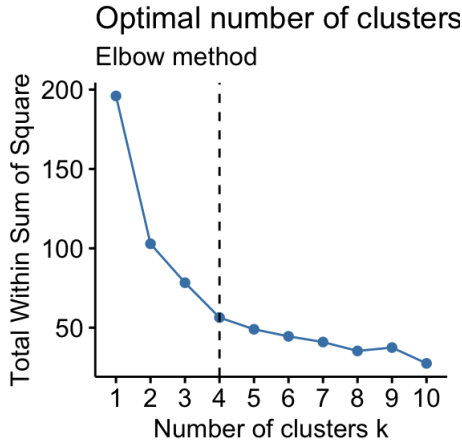
- choose a distance
- choose an algorithm
- choose the number of clusters K
- repeat the analysis for 1 to K_{max} clusters
- choose the number of clusters
- check the stability of clusters
- interpret the clusters
- Clustering is non supervised, part of the analysis is subjective, so we need guidelines

Elbow plot and model selection

- Choosing the number of clusters is a model selection task
- To choose a model we need a measure of quality of fit

$$\widehat{\mathbf{I}}_W(K) = \sum_{i=1}^{n} \sum_{k=1}^{K} \widehat{z}_{ik} d^2(\mathbf{x}_i, \overline{\mathbf{x}}_k)$$

- When K increases, $\widehat{\mathbf{I}}_W(K)$ decreases because clusters are more and more homogeneous
- The elbow plot consists in finding the best trade-off between quality of fit and a reasonable number of clusters



Intuitions for model selection

- Model selection is based on the bias-variance trade-off
- Bias : if a model has more parameters, it will approximate the data very precisely
- Variance : if a model has more parameters, the estimation error will increase
- How to find the best trade-off between both trends ?
- Model-selection criteria are based on penalized criteria:

$$C_K + \lambda \, \text{pen}(K)$$

- C_K is a contrast that decreases with the dimension
- $\text{pen}(K)$ is a penalty that increases with the dimension of the model
- λ is a penalty constant that tunes the trade-off

Examples of model selection criteria (not exhaustive)

- The Akaike Information Criterion

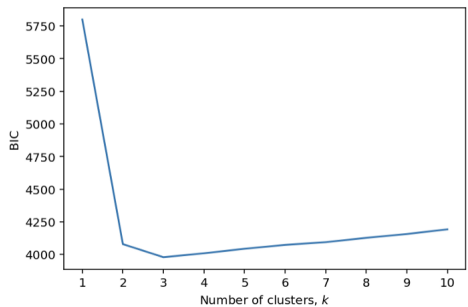
$$\mathrm{AIC}_K = -2 \log \widehat{\mathbf{l}}_W(K) + 2K$$

- The Bayesian Information Criterion

$$\mathrm{BIC}_K = -2 \log \widehat{\mathbf{l}}_W(K) + K \log(n)$$

- The Integrated Classification Likelihood

$$\mathrm{ICL}_K = -2 \log \widehat{\mathbf{l}}_W(K) + K \log(n) + \sum_{k=1}^{K} n_k \log n_k$$



Assessing cluster separation with the Silhouette score

- Consider clustering results into K clusters with inferred labels $(\widehat{\mathbf{z}}_1, \ldots, \widehat{\mathbf{z}}_K)$
- For point i that has been assigned to cluster k, compute the distance with points of the same cluster

$$a_i = \frac{1}{n_k - 1} \sum_{j \neq i} \widehat{z}_{ik} \widehat{z}_{jk} d^2(\mathbf{x}_i, \mathbf{x}_j)$$

- Compute the distance with points of other clusters

$$b_i = \min_{\ell} \left\{ \frac{1}{n_\ell} \sum_{j} \widehat{z}_{ik} \widehat{z}_{j\ell} d^2(\mathbf{x}_i, \mathbf{x}_j) \right\}$$

- Compute the silhouette score for each point

$$s_i = \frac{b_i - a_i}{\max\{a_i, b_i\}} \in [-1, 1]$$

Assessing cluster stability

- Consider two clustering results $\widehat{\mathcal{P}}_n^K, \widehat{\mathcal{P}}_n'^{K'}$
- (In)Stability of clustering results is defined as the expectation of distances between partitions (like adjusted Rand Index)

$$\mathbb{E}\left\{d\left(\widehat{\mathcal{P}}_n^K, \widehat{\mathcal{P}}_n'^{K'}\right)\right\}$$

- Use sub-sampling to perturb the data

$$\frac{1}{B^2}\sum_{b,b'} d\left(\widehat{\mathcal{P}}_{(b)}^K, \widehat{\mathcal{P}}_{(b')}'^{K'}\right)$$

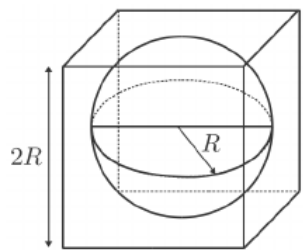
- To perturb the data (for instance): resampling, adding noise, use different dimension reduction methods

High dimensional setting

- distance-based methods are sensitive to increases in dimension
- The geometry of data is modified in high dimension
- Consider a sphere $S(\mathbf{x}, R)$ and a cube $C(\mathbf{x}, R)$ centered on $\mathbf{x} \in \mathbb{R}^p$ with radius R

$$\frac{\text{Vol}\left[C(\mathbf{x}, R)\right]}{\text{Vol}\left[S(\mathbf{x}, R)\right]} = \frac{2^p R^p}{2R^p \pi^{p/2} / p\Gamma(p/2)}$$

$$\frac{\text{Vol}\left[C(\mathbf{x}, R)\right]}{\text{Vol}\left[S(\mathbf{x}, R)\right]} \underset{p \to \infty}{\to} 0$$



From C.Azencott

Many points are needed to fill the space

- Number of points $(\mathbf{x}_1, \ldots, \mathbf{x}_n)$ needed to fill the cube $[0,1]^p$ by $S(\mathbf{x}_1, 1), \ldots, S(\mathbf{x}_n, 1)$

p	20	30	50	100
n	39	45,630	$5.7\ 10^{12}$	$42\ 10^{39}$

- High dimensional spaces are empty !
- Points are far apart

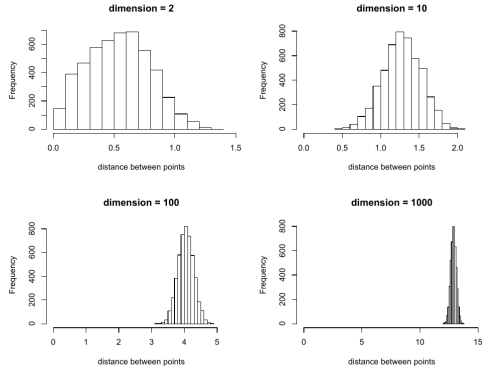
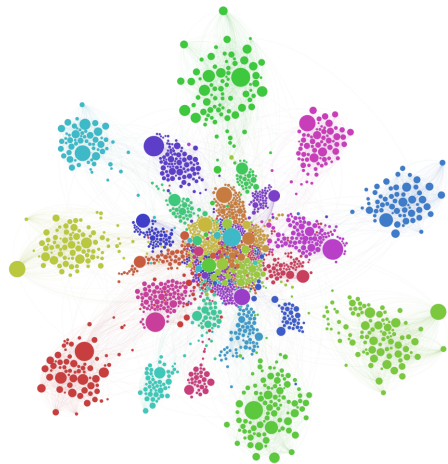


Figure 1.3 *Histograms of the pairwise-distances between $n = 100$ points sampled uniformly in the hypercube $[0,1]^p$, for $p = 2, 10, 100$, and 1000.*

From C.Giraud

Clustering in high dimension

- Dimension reduction is mandatory for clustering in high dimension
- Combine DR + clustering
- Use feature selection
- Try different DR methods
- Try different clustering methods
- Interpret clusters in the input space !



References

- `https://towardsdatascience.com/`
- Introduction to Machine Learning (C. Azencott)
- Introduction to High Dimensional Statistics (C. Giraud)

[1] V. Y. Kiselev, T. S. Andrews, and M. Hemberg. Challenges in unsupervised clustering of single-cell RNA-seq data. *Nat Rev Genet*, 20(5):273–282, 05 2019.

[2] J. Lever, Krzywinski M., and N. Altman. Principal component analysis. *Nat Methods*, 14:641–642, 2017.