① Soft k-means

$(X,Y) \in \mathbb{R}^d \times \{1,\dots,k\}$

$\begin{cases} X|Y \sim \mathcal{N}(\mu_Y, \Sigma_Y) \\ Y \sim \mathcal{D}(\pi) \quad \mathbb{P}(Y=j)=\pi_j \end{cases}$

Observat° : $X \quad (X_1,\dots,X_n)$

$X \sim \sum_{j=1}^{k} \pi_j \mathcal{N}(\mu_j, \Sigma_j)$

$\theta = (\pi_j, \mu_j, \Sigma_j, j=1,\dots,k)$

$\hat{\theta}$: EMV

$(\hat{\theta}_t)_{t\geq 1}$ vraisemblance $(\hat{\theta}_t)\nearrow$

② k-means

$(C_1,\dots,C_k)$ k cellules de $\mathbb{R}^d$

Distort°

$D_n(C_1,\dots,C_k) = \sum_{j=1}^{k} \sum_{X_i \in C_j} \|X_i - \mu_j\|_2^2$ ①

ici $\mu_j = \frac{1}{\#\{X_i \in C_j\}} \sum_{X_i \in C_j} X_i$ ②

→ Algo simple qui donne une approximat° de la solut°.

Algo EM mélanges de gaussiennes (Soft k-means)

Initialisn $\pi_j = 1/k$, μ_j au hasard dans $\{X_1,\dots,X_n\}$, $\Sigma_j = Id_d$

Itérer

"α distance de Mahalanobis" "Σ_i"

$p_{ij} \simeq \mathbb{P}(Y=j \mid X=X_i)$ ↓ Assignat° souple

π_j

μ_j = moyenne pondérée des X_i grâce à p_{ij}

$\Sigma_i = (X_i-\mu_j)(X_i-\mu_j)^T$

Algo des k-moyennes

Initialisat° μ_j pris au hasard dans $\{X_1,\dots,X_n\}$

Itérer

$(C_1,\dots,C_k)$ partitionnt de Voronoï ↙ Assignat° rigide

$p_{ij} = \mathbb{1}_{X_i \in C_j}$

μ_j = moyenne de $X_i \in C_j$

moyenne pondérée des X_i grâce à p_{ij}



Lecture 9: Spectral clustering

9.1 Cost minimization methods

9.1.1 Point-based objectives

In this section, we address the point of view opposite to the k-means approach: our aim is to devise a method that separates dissimilar points thanks to a paired criterion. For this purpose, let us assume that observations come from a probabilistic space $\mathcal{X}$, which may be more general than $\mathbb{R}^d$. For a partition $(C_1, \dots, C_k) \in \mathfrak{P}(\mathcal{X})$ and a similarity function $s\colon \mathcal{X} \times \mathcal{X} \to [0,1]$, the distortion to minimize is then

$$D(C_1, \dots, C_k) = \mathbb{E}\left[\sum_{j=1}^{k} s(X, X')\mathbf{1}_{X \in C_j \cap X' \notin C_j}\right],$$

where X' is an *iid* copy of X. Moving to estimation based on *iid* observations $X_1, \dots, X_n$, the previous distortion has a natural counterpart:

$$D_n(C_1, \dots, C_k) = \sum_{j=1}^{k} \sum_{\substack{1 \le i \le n \\ 1 \le \ell \le n}} s(X_i, X_\ell)\mathbf{1}_{X_i \in C_j \cap X_\ell \notin C_j}.$$

Then, it is often convenient to represent the relationships between training points by a similarity graph, in which each vertex represents a data point X_i and vertices are connected by an edge whose weight is their similarity. Such a graph can be defined by the similarity (or adjacency) matrix $W = (s(X_i, X_j))_{1 \le i,j \le n}$. Given the partition of the indexes $(I_1, \dots, I_k) \in \mathfrak{P}([n])$, defined by $i \in I_j \iff X_i \in C_j$, the previous point-based distortion reads:

$$D_n(C_1, \dots, C_k) = \sum_{j=1}^{k} \sum_{\substack{i \in I_j \\ \ell \notin I_j}} W_{i,\ell}.$$

Minimizing $D_n(C_1, \dots, C_k)$ is often referred to as the *graph cut problem*.

Remark 9.1.1. *As we will see after, the approach described in this section makes it possible to use non-Euclidean distances, such as graph distances, which are estimations of the intrinsic geodesic distance (on the potential manifold supporting the data). An example of such a distance for an unweighted graph is the length of the shortest path to go from a point to another in the graph:*

$$d(X_i, X_j) = \inf\left\{m \in \mathbb{N}^* : \{x_1, \dots, x_m\} \subset \{X_1, \dots, X_n\}, s(x_i, x_{i+1}) > 0, x_1 = X_i, x_m = X_j\right\} - 1$$

(with convention $\inf \emptyset = \infty$*). With this in mind, spectral clustering is more powerful than k-means.*

9.1.2 Similarity graphs

To be a bit more formal, we consider a *similarity graph* $G = (V, E)$, for which the vertices $V = (v_1, \dots, v_n)$ represent the points $(X_1, \dots, X_n)$. Tow vertices v_i and v_j are connected if the similarity $s(X_i, X_j) > 0$ and the edge between these two vertices is weighted by their similarity $s(X_i, X_j)$. The weighted adjacency matrix is $W = (s(X_i, X_j))_{1 \leq i,j \leq n}$.

The graph G is assumed undirected, which is equivalent to W being symmetric. In practice, this comes from considering a symmetric similarity measure s.

Definition 9.1.1. *The degree of a vertex* $v_i \in V$ *is* $d_i = \sum_{\ell=1}^{n} W_{i,\ell}$.

Given $A \subset V$*, we call* size *of A, and denote* $|A|$*, the number of its vertices, and volume of A:* $\mathrm{vol}(A) = \sum_{i \in [n]: v_i \in A} d_i$.

A is said connected if any two vertices of A can be joined by a path such that all intermediate points also lie in A.

A is called a connected component if it is connected and if there are no connections between vertices in A and $V \backslash A$.

When constructing a similarity graph, the goal is to model the local relationships between data points. In the forthcoming paragraphs, we describe four popular similarity graphs based on a given distance $d : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$.

$s(X_i, X_j) = \begin{cases} 1 & \text{si } \|X_i - X_j\|_2 \leq \varepsilon \\ 0 & \end{cases}$

The ϵ-neighborhood graph Two points X_i and X_j are connected if and only if their distance is less than a positive threshold ϵ: $d(X_i, X_j) \leq \epsilon$. If ϵ is small enough, all connected points are roughly at the same distance. Therefore, the weights assigned are $W_{i,j} = 1$ (if $d(X_i, X_j) \leq \epsilon$) and 0 otherwise. Given this rule, the ϵ-neighborhood graph is usually considered as an unweighted graph.

$s(X_i, X_j) = \begin{cases} 1 & \text{si } X_i \text{ est kppV de } X_j \text{ au sinon} \\ 0 & \end{cases}$

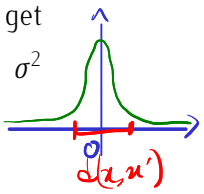
k-nearest neighbor graph Two points X_i and X_j are connected if and only if X_i is among the k-nearest neighbors of X_j or the other way around. Similarly to the ϵ-neighborhood graph, the weights are $W_{i,j} = 1$ if X_i and X_j are connected and 0 otherwise.

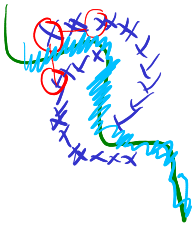
et

Mutual k-nearest neighbor graph Two points X_i and X_j are connected if and only if X_i is among the k-nearest neighbors of X_j and X_j is among the k-nearest neighbors of X_i. The weights are assigned similarly to the k-nearest neighbor graph.

The fully connected graph Points are connected if they have a positive similarity $s(X_i, X_j)$ and the edges are weighted by $s(X_i, X_j)$. It is important to note that since a similarity graph is supposed to reflect the local relationships, the similarity should be defined accordingly. In practice, it is asked to s to be fulfill $s(x, x) = 1, \forall x \in \mathcal{X}$ and $s(x, x')$ $(x' \in \mathcal{X})$ to decrease quickly to 0 when x and x' get away from each other. A popular choice is the Gaussian similarity: $s(x, x') = e^{-\frac{d(x,x')^2}{2\sigma^2}}$, in which σ^2 plays a role similar to ϵ and k for the similarity graphs introduced previously.

$s(X_i, X_j) = 1 - \|X_i - X_j\|$





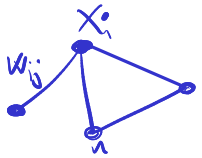
9.1.3 Spectral clustering

For $k = 2$, finding a minimal cut of a graph is a relatively easy problem and can be solved efficiently (for instance thanks to the Stoer-Wagner algorithm). However, it often results in separating an individual vertex from the rest of the graph, which is not satisfactory. Several solutions to this problem have been suggested but the most common ones are to normalize the empirical distortion either by the size of the clusters:

$$\text{RatioCut}(C_1, \dots, C_k) = \sum_{j=1}^{k} \frac{1}{|I_j|} \sum_{\substack{i \in I_j \\ \ell \notin I_j}} W_{i,\ell},$$

(let us remind that the partition of the indexes $(I_1, \dots, I_k) \in \mathfrak{P}([n])$ is defined by $i \in I_j \iff X_i \in C_j$) or by their volume:

$$\text{NormCut}(C_1, \dots, C_k) = \sum_{j=1}^{k} \frac{1}{\text{vol}(I_j)} \sum_{\substack{i \in I_j \\ \ell \notin I_j}} W_{i,\ell}.$$



These objectives are respectively called *Ratio Cut* and *Normalized Cut*. Unfortunately, the balancing introduced by the cluster importance makes the minimization problem computationally hard to solve. Therefore, from now on, we describe a relaxation procedure resulting in the so called *spectral clustering* algorithm.

To this end, we assume that the similarity measure $s\colon \mathcal{X} \times \mathcal{X} \to [0, 1]$ is such that:

$$\begin{cases} \forall (x, x') \in \mathcal{X}^2 : & s(x, x') = s(x', x) \\ \forall x \in \mathcal{X} : & s(x, x) > 0. \end{cases}$$

Definition 9.1.2 (Unnormalized graph Laplacian). *Let $W \in \mathbb{R}^{n \times n}$ be a symmetric matrix. The diagonal matrix $D \in \mathbb{R}^{n \times n}$ such that $D_{i,i} = \sum_{j=1}^{n} W_{i,j}$, $\forall i \in [n]$ and $L = D - W$ are respectively called the* degree matrix *and the* Laplacian *of the graph defined by W.*

Proposition 60. *Let W and L be respectively the adjacency matrix and the Laplacian of the similarity graph of $(X_1, \dots, X_n)$. For any positive integer k and for all partitioning $(C_1, \dots, C_k)$ of $(X_1, \dots, X_n)$, we have*

$$\text{RatioCut}(C_1, \dots, C_k) = \text{tr}(H^\top L H),$$

where $H = \left(\frac{1}{\sqrt{|I_j|}} \mathbb{1}_{i \in I_j} \right)_{\substack{1 \le i \le n \\ 1 \le j \le k}}$.

In addition, the columns of H are orthonormal to each other ($H^\top H = I_k$).

The proof will be done during the class.

Remark 9.1.2. *H represents the* one-hot-encoding *of the clusters under the normalization constraint $H^\top H = I_k$. For example, for $k = 3$, if we reorganize the sample $(X_1, \dots, X_n)$ such that*

Preuve Proposition 6.9

Pour $(C_1, \dots, C_k)$ donné et $H = \left(\frac{1}{\sqrt{|I_j|}} \mathbb{1}_{i \in I_j} \right)_{\substack{1 \le i \le n \\ 1 \le j \le k}}$ = □

On va montrer que $\mathrm{tr}(H^T L H) = \dots = \sum_{j=1}^{k} \frac{1}{|I_j|} \sum_{\substack{i \in I_j \\ e \notin I_j}} w_{ie}$.

- On écrit $H = [h_1 | \dots | h_k]$

$$\mathrm{tr}(H^T L H) = \mathrm{tr}\left(\begin{bmatrix} h_1^T \\ \vdots \\ h_k^T \end{bmatrix} [Lh_1 | \dots | Lh_k] \right) = \mathrm{tr}\left((h_i^T L h_j)_{1 \le i,j \le k} \right)$$

$$= \sum_{j=1}^{k} h_j^T L h_j$$

- $L = D - W$

$\forall u \in \mathbb{R}^n$, $u^T L u = u^T D u - u^T W u$

≥ 0

$\Rightarrow L$ SDP.

$$= \sum_i u_i D_{ii} u_i - \sum_{i,e} w_{ie} u_i u_e \qquad (D \text{ diagonale})$$

$D_{ii} = \sum_e w_{ie}$

$$= \sum_{i,e} w_{ie} u_i^2 - \sum_{i,e} w_{ie} u_i u_e$$

$$= \frac{1}{2} \sum_{i,e} w_{ie} u_i^2 + \frac{1}{2} \sum_{i,e} w_{ie} u_e^2 \qquad (W \text{ symétrique})$$

$$= \frac{1}{2} \sum_{i,e} w_{ie} (u_i - u_e)^2$$

- $u = h_j = \left(\mathbb{1}_{x_i \in C_j} / \sqrt{|I_j|} \right)_{1 \le i \le n}$

$$h_j^T L h_j = \frac{1}{2} \sum_{i,e} w_{ie} \left[(h_j)_i - (h_j)_e \right]^2$$

$$= \begin{cases} 0 & \text{si } x_i \notin C_j \text{ et } x_e \notin C_j \text{ ou si } x_i \in C_j \text{ et } x_e \in C_j \\ 1 & \text{si } x_i \in C_j \text{ et } x_e \notin C_j \text{ ou si } x_i \notin C_j \text{ et } x_e \in C_j \end{cases}$$

$$= \frac{1}{2} \sum_{\substack{i \in I_j \\ e \notin I_j}} \frac{w_{ie}}{|I_j|} + \frac{1}{2} \sum_{\substack{i \notin I_j \\ e \in I_j}} \frac{w_{ie}}{|I_j|}$$

$$= \sum_{\substack{i \in I_j \\ e \notin I_j}} \frac{w_{ie}}{|I_j|} \qquad (W \text{ est symétrique})$$

- Donc $\mathrm{tr}(H^T L H) = \sum_j \frac{1}{|I_j|} \sum w_{ie} = \mathrm{RatioCut}(C_1, \dots, C_k)$.

C_1 *appears first, then* C_2 *and so on, we get*

$$H = \begin{pmatrix} \frac{1}{\sqrt{|I_1|}} & 0 & 0 \\ \vdots & \vdots & \vdots \\ \frac{1}{\sqrt{|I_1|}} & 0 & 0 \\ 0 & \frac{1}{\sqrt{|I_2|}} & 0 \\ \vdots & \vdots & \vdots \\ 0 & \frac{1}{\sqrt{|I_2|}} & 0 \\ 0 & 0 & \frac{1}{\sqrt{|I_3|}} \\ \vdots & \vdots & \vdots \\ 0 & 0 & \frac{1}{\sqrt{|I_3|}} \end{pmatrix}$$

Owing to Proposition 60, the Ratio Cut problem

$$\underset{(I_1,\dots,I_k)\in\mathfrak{P}([n])}{\text{minimize}} \sum_{j=1}^{k} \frac{1}{|I_j|} \sum_{\substack{i\in I_j \\ \ell\notin I_j}} W_{i,\ell}$$

is equivalent to

$$\underset{\substack{H\in\mathbb{R}^{n\times k} \\ \alpha\in(\mathbb{R}_+^*)^k}}{\text{minimize}} \operatorname{tr}(H^\top L H) \quad \text{s.t.} \quad \begin{cases} H^\top H = I_k \\ \forall (i,j)\in[n]\times[k],\ H_{i,j}\in\{0,\alpha_j\}. \end{cases}$$

This optimization problem is a discrete programming problem. Unfortunately such problems are known (or at least were used to be known) to be difficult to solve numerically. Therefore, we relax the problem by discarding the last two constraints. Unnormalized spectral clustering boils down to solving

$$\underset{H\in\mathbb{R}^{n\times k}}{\text{minimize}} \operatorname{tr}(H^\top L H) \quad \text{s.t.} \quad H^\top H = I_k. \tag{P21}$$

The forthcoming theorem tells that a solution to (P21) can be obtained easily by a spectral decomposition.

Theorem 61. *Let* $C \in \mathbb{R}^{d\times d}$ *be a symmetric matrix and let us denote* $C = \sum_{i=1}^{d} \lambda_i v_i v_i^\top$ *its eigendecomposition, where for all* $i \in [d]$, $v_i \in \mathbb{R}^d$ *and* $\lambda_i \in \mathbb{R}$, *with sorted eigenvalues* $\lambda_1 \le \cdots \le \lambda_d$. *For any* $k \in [d]$, *let us denote* $V_- = [v_1|\dots|v_k] \in \mathbb{R}^{d\times k}$ *and* $V_+ = [v_{d-k+1}|\dots|v_d] \in \mathbb{R}^{d\times k}$, *respectively the matrices of the minor and major eigenvectors.*

Then,

$$\inf_{\substack{U\in\mathbb{R}^{d\times k} \\ U^\top U = I_k}} \operatorname{tr}(U^\top C U) = \operatorname{tr}(V_-^\top C V_-)$$

and

$$\sup_{\substack{U\in\mathbb{R}^{d\times k} \\ U^\top U = I_k}} \operatorname{tr}(U^\top C U) = \operatorname{tr}(V_+^\top C V_+).$$

Preuve

C symétrique, $C = V \Lambda V^T$ $\quad V = [v_1 | \dots | v_d]$ $\quad V \in \mathbb{R}^{d \times d}$ orthogonale

$V^T V = V V^T = I_d$

$\Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_d \end{pmatrix}$ $\quad \lambda_1 \leq \dots \leq \lambda_d$

$C v_i = \lambda_i v_i$

$C = \sum_{i=1}^{d} \lambda_i v_i v_i^T$

$V_- = [v_1 | \dots | v_k]$

Th° V_- est solution de $\min_{\substack{U \in \mathbb{R}^{d \times k} \\ U^T U = I_k}} \operatorname{tr}(U^T C U)$

- V_- est admissible car $V_-^T V_- = I_k$

  Il reste à montrer $\forall U$ admissible $\operatorname{tr}(U^T C U) \geq \operatorname{tr}(V_-^T C V_-)$

- ① $\operatorname{tr}(U^T C U) \geq \inf_{\beta \in S} \sum_{i=1}^{d} \lambda_i \beta_i$ $\quad , S = \{\beta \in [0,1]^d, \ \mathbb{1}^T \beta = k\}$

  ② $\inf_{\beta \in S} \sum_{i=1}^{d} \lambda_i \beta_i = \sum_{i=1}^{k} \lambda_i$

  ③ $\sum_{i=1}^{k} \lambda_i = \operatorname{tr}(V_-^T C V_-)$

- ① $\forall U \in \mathbb{R}^{d \times k}$ tq $U^T U = I_k$.

$\operatorname{tr}(U^T C U) = \operatorname{tr}(U^T V \Lambda V^T U) = \operatorname{tr}(B^T \Lambda B) = \sum_{j=1}^{k} b_j^T \Lambda b_j$ $\quad \left(\Lambda = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_d \end{pmatrix}\right)$

$= B \in \mathbb{R}^{d \times p}$

$B = [b_1 | \dots | b_p]$

$= \sum_j \sum_i \lambda_i \underbrace{(b_j)_i^2}_{B_{ij}^2}$

$= \sum_i \lambda_i \left(\sum_j B_{ij}^2 \right)$

$= \sum_i \lambda_i \beta_i$

avec $\beta_i = \sum_{j=1}^{k} B_{ij}^2$

- $\beta_i \geq 0$ ok.
- $\beta_i \leq 1$?

$i \to$ [....]

$B^T B = U^T V V^T U = U^T U = I_k$ $\quad B$ est orthonormée

On peut compléter $b_1, \dots, b_p$ par $a_{p+1}, \dots, a_d$ tq

$\{b_1, \dots, b_p, a_{p+1}, \dots, a_d\}$ BON de $\mathbb{R}^d$ $\quad A = [a_{p+1} | \dots | a_d]$

Alors $[B|A]$ orthogonale.

$[B|A][B|A]^T = I_d$ $\Leftrightarrow$ lignes orthonormées

i^e ligne $\sum_{j=1}^{k} B_{ij}^2 + \sum_{j=k+1}^{d} A_{ij}^2 = 1$ Ainsi $\beta_i \le 1$.

est normée ≤ 1 ≥ 0

- $\sum_{i=1}^{d} \beta_i = k$?

$$\sum_{i=1}^{d} \beta_i = \sum_{i=1}^{d} \sum_{j=1}^{k} B_{ij}^2 = \mathrm{tr}(B^T B) = \mathrm{tr}(I_k) = k.$$

Au final, $\beta \in S = \{\beta' \in [0,1]^d, \sum_{i=1}^{d} \beta_i' = k\}$

Puis $\mathrm{tr}(U^T C U) \ge \inf_{\beta \in S} \sum_{i=1}^{d} \lambda_i \beta_i$.

- (2) $\sum_{i=1}^{k} \lambda_i = \sum_{i=1}^{d} \lambda_i \beta_i^*$ avec $\beta^* = \begin{pmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{matrix} \uparrow \\ k \\ \downarrow \\ \uparrow \\ d-k \\ \downarrow \end{matrix} \in S$

$\forall \beta \in S$: $\sum_{i=1}^{d} \lambda_i \beta_i - \sum_{i=1}^{d} \lambda_i \beta_i^* \ge 0$?

$$= \sum_{i=1}^{k} \lambda_i (\beta_i - 1) + \sum_{i=k+1}^{d} \lambda_i \beta_i$$

≤ 0 ≥ 0

$\lambda_1 \le \dots \le \lambda_k \le \dots \le \lambda_d$

$$\ge \sum_{i=1}^{k} \lambda_k (\beta_i - 1) + \sum_{i=k+1}^{d} \lambda_k \beta_i$$

$$= \lambda_k \underbrace{\left(\sum_{i=1}^{d} \beta_i - k \right)}_{=0 \text{ car } \beta \in S}$$

$$= 0$$

- (3) $\mathrm{tr}(V_-^T C V_-) = \mathrm{tr}\left(\underbrace{(V_-^T V)}_{=S} \Lambda (V^T V_-) \right)$

$$= \sum_{i=1}^{k} \lambda_i$$

$$S = \begin{pmatrix} - & v_1^T & - \\ & \vdots & \\ - & v_d^T & - \end{pmatrix} \begin{matrix} \\ \\ d \\ \\ \\ \end{matrix} \quad \begin{matrix} k \\ \begin{pmatrix} | & & | \\ v_1 & \cdots & v_k \\ | & & | \end{pmatrix} \end{matrix}$$

$$\begin{matrix} I_k & 0 \\ 0 & 0_{d-k} \end{matrix} \quad \begin{matrix} k \\ d-k \end{matrix}$$

$$V^T V = \begin{pmatrix} I_k & 0 \\ 0 & 0_{d-k} \end{pmatrix} \quad \begin{matrix} k \\ d-k \end{matrix}$$

Retour à 16h05.

The proof will be done during the class.

By Theorem 61, (P21) is solved by the matrix H for which the columns are the minor eigenvectors of L. The resulting algorithm (see Algorithm 11) is called *Unnormalized spectral clustering*. It proceeds by mapping the data $(X_1, \dots, X_n)$ to the rows of the k minor eigenvectors of L and then by performing a vanilla k-means. Contrarily to the intuition raised by Remark 9.1.2, the clustering step performed by k-means cannot be replaced by a simple arg max operation. This is so since the Laplacian L has always $\mathbb{1}$ as eigenvector (with eigenvalue 0), so the mapping performed by spectral clustering cannot by thought as a one-hot-encoding of the expected partition cells (see also Figure 9.1).

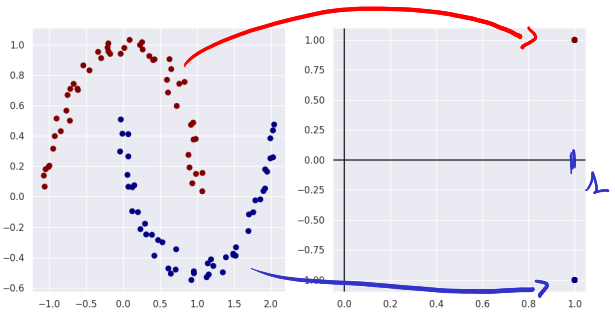


Figure 9.1: Example mapping performed by spectral clustering (fully connected Gaussian graph).

Algorithm 11 Unnormalized spectral clustering.

Input: $W \in \mathbb{R}^{n \times n}$ (adjacency matrix).
$L \leftarrow$ Laplacian of W
$H \leftarrow k$ minor eigenvectors of L as columns
$Y_i \leftarrow i^{th}$ row of H (for all $i \in [n]$) ($Y_i \in \mathbb{R}^k$)
$(\hat{C}_1, \dots, \hat{C}_k) \leftarrow$ output of k-means algorithm based on $(Y_1, \dots, Y_n)$
Output: $(\hat{C}_1, \dots, \hat{C}_k)$.

Proposition 62. *Let W and L be respectively the adjacency matrix and the Laplacian of the similarity graph of $(X_1, \dots, X_n)$. For any positive integer k and for all partitioning $(C_1, \dots, C_k)$ of $(X_1, \dots, X_n)$, we have*

$$\mathrm{NormCut}(C_1, \dots, C_k) = \mathrm{tr}(H^\top L H),$$

where $H = \left(\frac{1}{\sqrt{\mathrm{vol}(I_j)}} \mathbb{1}_{i \in I_j} \right)_{\substack{1 \le i \le n \\ 1 \le j \le k}}$.

In addition, the columns of $D^{\frac{1}{2}} H$ are orthonormal to each other ($H^\top D H = I_k$).

The proof is a good exercise.

Owing to Proposition 62, the Normalized Cut problem

$$\underset{(I_1, \dots, I_k) \in \mathfrak{P}([n])}{\text{minimize}} \sum_{j=1}^{k} \frac{1}{\mathrm{vol}(I_j)} \sum_{\substack{i \in I_j \\ \ell \notin I_j}} W_{i,\ell}$$

is equivalent to

$$\begin{array}{l} \underset{\substack{H\in\mathbb{R}^{n\times k}\\ \alpha\in(\mathbb{R}_+^*)^k}}{\text{minimize}} \ \operatorname{tr}(H^\top L H) \\ \text{s. t.} \quad \begin{cases} H^\top D H = I_k \\ \forall (i,j)\in[n]\times[k],\ H_{i,j}\in\{0,\alpha_j\}. \end{cases} \end{array}$$

and can be relaxed to

$$\begin{array}{l} \underset{H\in\mathbb{R}^{n\times k}}{\text{minimize}} \ \operatorname{tr}(H^\top L H) \\ \text{s. t.} \quad H^\top D H = I_k. \end{array} \tag{P22}$$

Since D is invertible (by assumptions on s), (P22) can be reformulated

$$\begin{array}{l} \underset{H\in\mathbb{R}^{n\times k}}{\text{minimize}} \ \operatorname{tr}(U^\top L_s U) \\ \text{s. t.} \quad \begin{cases} H = D^{-\frac{1}{2}} U \\ U^\top U = I_k, \end{cases} \end{array}$$

where $L_s = D^{-\frac{1}{2}} L D^{-\frac{1}{2}}$. Therefore (P22) is solved by the matrix U for which the columns are the minor eigenvectors of L_s, which corresponds to H for which the columns are the minor eigenvectors of $L_w = D^{-1}L$, adequately normalized such that $H^\top D H = I_k$. In practice, the latter normalization is omitted (we consider 1-norm eigenvectors of L_w) and the resulting algorithm (see Algorithm 12) is called *Normalized spectral clustering*.

Remark 9.1.3. *$\lambda \in \mathbb{R}_+$ is eigenvalue of L_w with eigenvector u if and only if λ and u solve the generalized eigenvalue problem $Lu = \lambda D u$.*

Algorithm 12 Normalized spectral clustering (with L_w).

Input: $W \in \mathbb{R}^{n\times n}$ (adjacency matrix).

$L_w \leftarrow$ Laplacian of W

$H \leftarrow k$ minor eigenvectors of L_w as columns *(similar to the generalized eigenproblem $Lu = \lambda D u$)*

$Y_i \leftarrow i^{th}$ row of H (for all $i \in [n]$) *($Y_i \in \mathbb{R}^k$)*

$(\hat{C}_1, \dots, \hat{C}_k) \leftarrow$ output of k-means algorithm based on $(Y_1, \dots, Y_n)$

Output: $(\hat{C}_1, \dots, \hat{C}_k)$.

Remark 9.1.4. *Comparing Ratio Cut and Normalized Cut leads to a very interesting discovery. First, let us remark that, as already mentioned, both objective functions encodes the second part of the intuitive definition of clustering: points separated into different clusters should be dissimilar.*

In addition, the balancing introduced takes into account the importance *of the clusters, either through their size or their volume. This is so since minimizing the min cut objectives leads to minimizing the cuts between the clusters while maximizing their importance. However, Ratio Cut and Normalized Cut behave differently concerning cluster importance. Indeed, it is easy to see that, for all $j \in [k]$:*

$$\sum_{\substack{i\in I_j \\ \ell\in I_j}} W_{i,\ell} = \operatorname{vol}(I_j) - \sum_{\substack{i\in I_j \\ \ell\notin I_j}} W_{i,\ell}.$$

In other words, the intra-cluster similarity is maximized as soon as the volume of the cluster is

maximized and the cut with the rest of the vertices is minimized; which is what is achieved by Normalized Cut minimization. On the other hand, the size $|I_j|$ of a cluster is not necessarily related to the intra-cluster similarity.

In this sense, Normalized Cut minimization addresses both parts of the clustering definition.

Moreover, it can be shown that, L_w behaves as expected when $n \to \infty$ and so it is for the resulting partitioning provided by normalized spectral clustering. On the contrary, L can lead to completely unreliable results, even for small sample size [von Luxburg, 2007].

There exists another popular normalized spectral clustering algorithm (see Algorithm 13) based on the third Laplacian that popped up during this analysis: L_s.

Algorithm 13 Normalized spectral clustering (with L_s).

Input: $W \in \mathbb{R}^{n \times n}$ (adjacency matrix).

$L_s \leftarrow$ Laplacian of W

$H \leftarrow k$ minor eigenvectors of L_s as columns

$Y_i \leftarrow i^{th}$ row of H normalized to 1 (for all $i \in [n]$) ($Y_i \in \mathbb{R}^k$, $\sum_{j=1}^{k} (Y_i)_j^2 = 1$)

$(\hat{C}_1, \dots, \hat{C}_k) \leftarrow$ output of k-means algorithm based on $(Y_1, \dots, Y_n)$

Output: $(\hat{C}_1, \dots, \hat{C}_k)$.

Remark 9.1.5. *First, there is no theoretical guarantees concerning the "quality" of these two relaxations.*

Second, there exist many other relaxations. Some of them rely on semidefinite programming.

Last but not least, spectral relaxations are not appealing for the quality of the solutions they provide but for the simplicity of the problem in which they results (standard linear algebra – eigenvalue – problems).

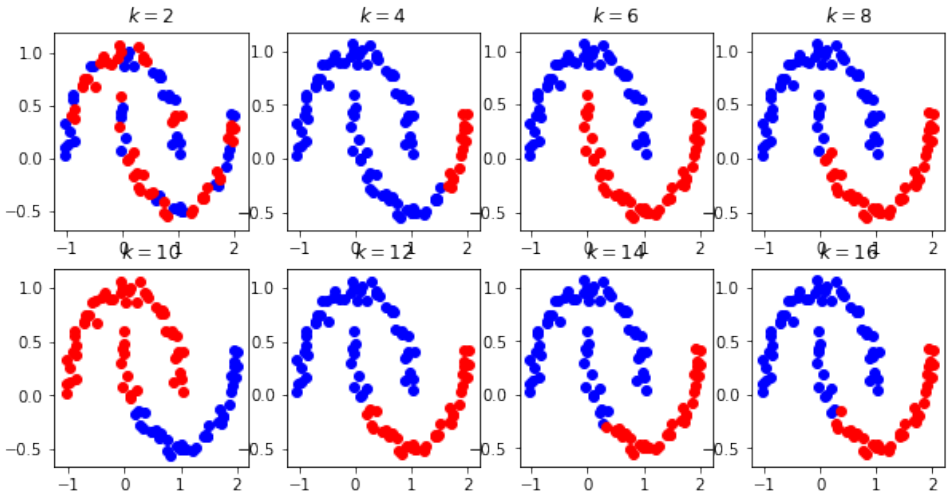


Figure 9.2: Example of spectral clustering (k-nearest neighbor graph).

9.1.4 (⋆) Properties of graph Laplacians

Let us consider $W \in \mathbb{R}_+^{n \times n}$ a symmetric adjacency matrix and $D \in \mathbb{R}^{n \times n}$ its degree matrix. So far,

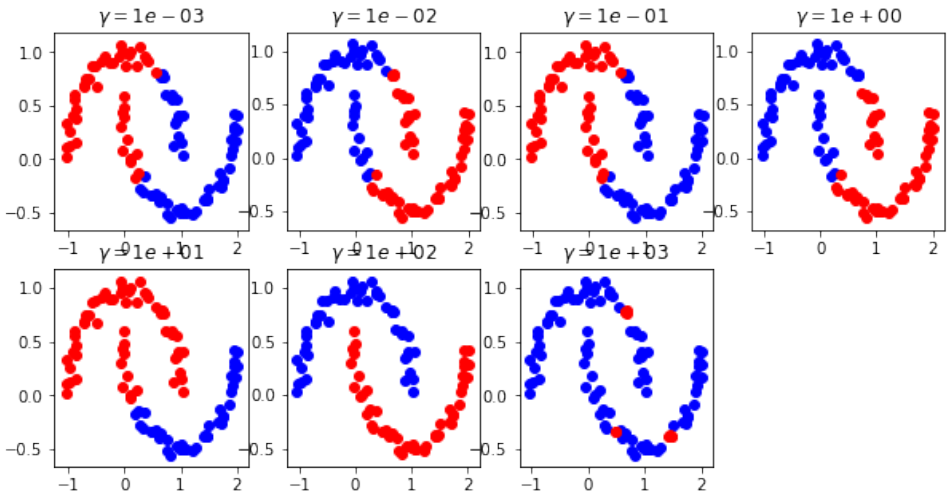


Figure 9.3: Example of spectral clustering (fully connected Gaussian graph).

we have seen three Laplacians, summed up in the following definition.

Definition 9.1.3.

Unnormalized Laplacian: $L = D - W$;
Nomalized Laplacian 1: $L_s = D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$;
Nomalized Laplacian 2: $L_w = D^{-1} L = I - D^{-1} W$.

Normalized Laplacians are subscripted by s and w because they are respectively symmetrically normalized by $D^{-\frac{1}{2}}$ (on left and right) and whitened by D.

Graph Laplacians have many properties. In the next proposition, we recover in particular the two properties bridging the gap between graph cut and eigenvalue decomposition (Item 1) and discover that 0 is an eigenvalue of L and L_w with eigenvector $\mathbb{1}$.

Proposition 63.

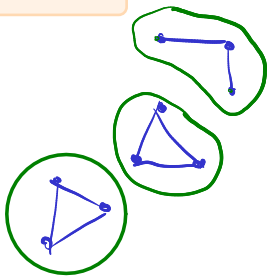
1. *One has,* $\forall u \in \mathbb{R}^n$*:*

$$u^\top L u = \frac{1}{2} \sum_{1 \leq i, \ell \leq n} W_{i,\ell} (u_i - u_\ell)^2$$

$$u^\top L_s u = \frac{1}{2} \sum_{1 \leq i, \ell \leq n} W_{i,\ell} \left(\frac{u_i}{\sqrt{D_{i,i}}} - \frac{u_\ell}{\sqrt{D_{\ell,\ell}}} \right)^2 .$$

2. *0 is eigenvalue of L and L_w with eigenvector $\mathbb{1}$. 0 is eigenvalue of L_s with eigenvector $D^{\frac{1}{2}}\mathbb{1}$.*
3. *$\lambda \in \mathbb{R}_+$ is eigenvalue of L_w with eigenvector u if and only if λ is eigenvalue of L_s with eigenvector $D^{\frac{1}{2}} u$.*
4. *L, L_s and L_w are symmetric PSD matrices.*

The proof is a good exercise.



Proposition 64. *Let G be an undirected graph with non-negative weights. Then, the multiplicities of the eigenvalue 0 of L, L_s and L_w are the same and equal the number k of connected components $(A_1, \dots, A_k)$ in G.*

In addition, the eigenspace of 0 for both L and L_w is spanned by $\{\mathbb{1}_{A_1}, \dots, \mathbb{1}_{A_k}\}$ and the eigenspace of 0 for L_s is spanned by $\{D^{-\frac{1}{2}}\mathbb{1}_{A_1}, \dots, D^{-\frac{1}{2}}\mathbb{1}_{A_k}\}$.

We refer to [von Luxburg, 2007, Fig. 1] for an illustration of eigenvector properties.

9.1.5 (⋆) Practical details

Similarity graph ϵ-neighborhood cannot handle *different scales* (different distances between data points) in different regions of the space. k-nearest neighbor graph can and connects regions of high and low densities. On the contrary, mutual k-nearest neighbor graph does not connect regions of high and low densities. It can be used to detect clusters of different densities.

k-nearest neighbor graph is a good starting point.

Connectivity parameter For k-nearest neighbor graph, choose k such that the graph is connected or has significantly fewer connected components than clusters to detect. Otherwise, spectral clustering will trivially return connected components as clusters. Some asymptotic connectivity results suggest to choose k in the order of $\log(n)$.

Very generally, we can observe that the mutual k-nearest neighbor graph has much fewer edges than the k-nearest neighbor graph for the same parameter k. This suggest to choose k larger for the mutual k-nearest neighbor graph.

For the ϵ-neighborhood graph, ϵ should be chosen such that the graph is connected. The smallest value of ϵ for which the graph is connected can be estimated by the length of the longest edge in a minimal spanning tree covering the fully connected graph of the data. However this method is very sensitive to outliers and isolated tight clusters.

For a fully connected graph, σ can be chose as the mean distance of a point to its k-nearest neighbors, where k is chosen similarly as above (k of the order $\log(n)$).

Number of clusters The gap heuristic: choose k such that all eigenvalues $\lambda_1, \dots, \lambda_k$ are very small and λ_{k+1} is relatively large (see Figures 9.4 and 9.5). A justification of this procedure, coming from perturbation theory, is that in the ideal case of k completely disconnected clusters, the eigenvalue 0 has multiplicity k and $\lambda_{k+1} > 0$.

Graph Laplacian One may look at the degree distribution of the similarity graph. If most vertices have approximately the same degree, then all graph Laplacians will perform equally. However, if the degrees of the graph are very broadly distributed, then Laplacians differ considerably and we suggest to choose normalized Laplacians, and particularly L_w rather than L_s.

This choice is justified first by Remark 9.1.4.

9.2 (⋆) Hierarchical clustering

Hierarchical methods for clustering aim at answering a major drawback of k-means: the lack of

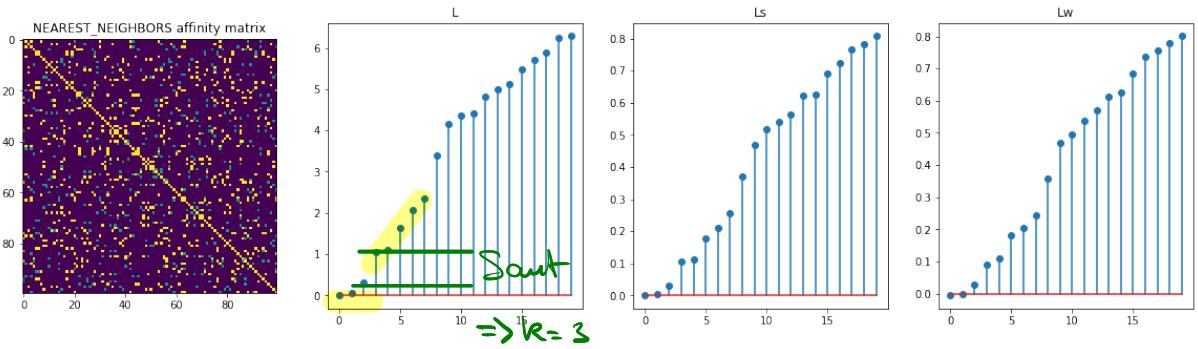


Figure 9.4: Example of Laplacian eigenvalues (k-nearest neighbor graph).

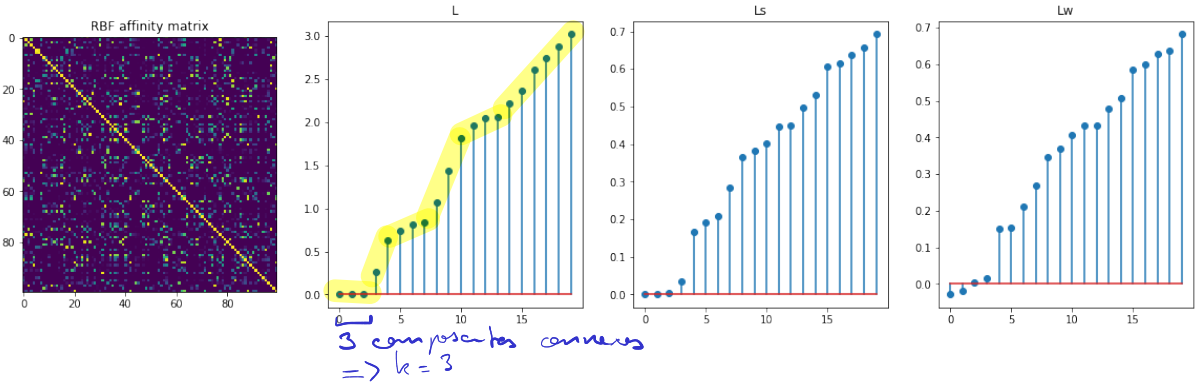


Figure 9.5: Example of Laplacian eigenvalues (fully connected Gaussian graph).

hierarchy in clusters (*i.e.* decreasing k does not lead to merging clusters). This section introduces very simple methods based on measuring the similarity (or linkage) between clusters. We focus on *agglomerative* approaches (which are based on merging clusters) and put *divisive* ones aside (based on splitting clusters).

9.2.1 Agglomerative approaches

Linkage-based methods are probably the simplest and most intuitive paradigm of clustering. In their agglomerative version, they start from the partitioning of the training set $(X_1, \dots, X_n)$ in which each cluster is a unit set $\{X_i\}$ (for $i \in [n]$) and merge successively the *closest* clusters. Straightforwardly, the number of clusters decreases at each iteration and clusters are nested: each cluster $\hat{C}^t$ at iteration t is either the same as at iteration $t-1$ ($\hat{C}^t = \hat{C}^{t-1}$) or the union of two previous clusters ($\hat{C}^t = \hat{C}_1^{t-1} \cup \hat{C}_2^{t-1}$).

Two parameters need to be defined in such a procedure: the (dis)similarity (or linkage) between two clusters and the merging stopping rule. To make the first point precise, let $d\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$ be a dissimilarity and consider two subsets A and B of $(X_1, \dots, X_n)$. We now give common examples of cluster dissimilarities $D\colon P(\{X_1, \dots, X_n\})^2 \to \mathbb{R}_+$.

Single linkage

$$D(A, B) = \min_{x \in A, y \in B} d(x, y).$$

Complete linkage

$$D(A, B) = \max_{x \in A, y \in B} d(x, y).$$

Average linkage

$$D(A, B) = \frac{1}{|A||B|} \sum_{x \in A, y \in B} d(x, y).$$

Ward's minimum variance

Given the intraclass inertia for a generic subset $C \subset (X_1, \ldots, X_n)$:

$$I(C) = \sum_{x \in C} d\left(x, m_C\right)^2,$$

where $m_C = \frac{1}{|C|} \sum_{y \in C} y$, the cluster dissimilarity in Ward's method is

$$D(A, B) = I(A \cup B) - I(A) - I(B),$$

which is the increase of intraclass inertia when merging A and B. For the Euclidean distance,

$$D(A, B) = \frac{|A||B|}{|A| + |B|} \left\| m_A - m_B \right\|_{\ell_2}^2.$$

Since Ward's method merges clusters by minimizing the increase in the total intraclass inertia, it is very similar to k-means but approximates a minimizer of the k-means objective with an agglomerative hierarchical procedure. Indeed, let us remind the distortion used in k-means: for a partition $(C_1, \ldots, C_k) \in \mathfrak{P}$,

$$D_n(C_1, \ldots, C_k) = \sum_{j=1}^{k} \sum_{i=1}^{n} \left\| X - \mu_n(C_j) \right\|_{\ell_2}^2 \mathbf{1}_{X_i \in C_j} = \sum_{j=1}^{k} I(\hat{C}_j),$$

where $\mu_n(C_j) = \frac{1}{|\{i \in [n] : X_i \in C_j\}|} \sum_{i=1}^{n} X_i \mathbf{1}_{X_i \in C_j}$ and $\hat{C}_j = C_j \cap \{X_1, \ldots, X_n\}$. At the beginning of the agglomerative procedure, the empirical partition is $(\hat{C}_1, \ldots, \hat{C}_n) = (\{X_1\}, \ldots, \{X_n\})$ and (with a slight abuse of notation)

$$D_n(\{X_1\}, \ldots, \{X_n\}) = 0.$$

Then, given an empirical partition $(\hat{C}_1, \ldots, \hat{C}_m)$, the agglomerative clustering technique merges the two clusters $\hat{C}_i$ and $\hat{C}_j$ such that merging two components of the summation bellow produces a minimal increase:

$$D_n(\hat{C}_1, \ldots, \hat{C}_m) = I(\hat{C}_1) + I(\hat{C}_2) + \cdots + I(\hat{C}_{m-1}) + I(\hat{C}_m).$$

Consequently, the distortion increases slightly at each iteration a reaches at the end a value of distortion, which is close to that obtained by k-means (and likely bigger).

Remark 9.2.1. *Linkage methods can be used with a variety of distances (or affinities), in particular:*

- ◇ *Euclidean distance (or l2);*
- ◇ *Manhattan distance (or Cityblock, or l1);*

◇ *cosine distance;*
◇ *any precomputed affinity matrix.*

If the agglomerative procedure runs until the end, all points share the same large cluster. The resulting sequence of partitioning can be represented as a tree, called a dendrogram, the root of which is the unique cluster that gathers all points (the final cluster) and the leaves of which are the unit set clusters (algorithm initialization).

If one is more interested in a useful partitioning instead of the clustering dendrogram, one needs to employ a stopping rule, which may be:

◇ a fixed number of clusters;
◇ a distance upper bound $\bar{D}$ (or alternatively a *scaled distance upper bound* $\alpha \in \mathbb{R}_+$ such that $\bar{D} = \alpha \max_{1 \leq i,j \leq n} d(X_i, X_j)$ for single, complete and average linkages).

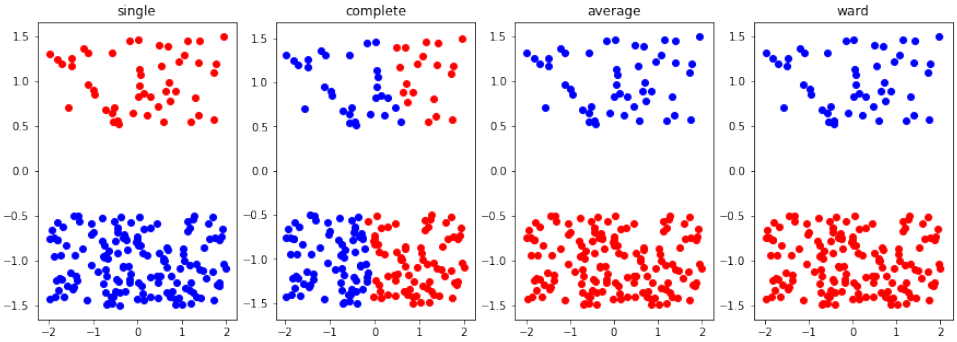


Figure 9.6: Example of agglomerative clustering (two rectangles).

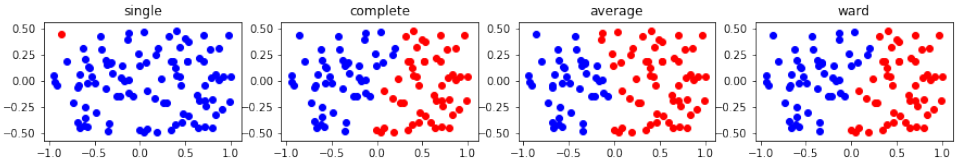


Figure 9.7: Example of agglomerative clustering (single rectangle).

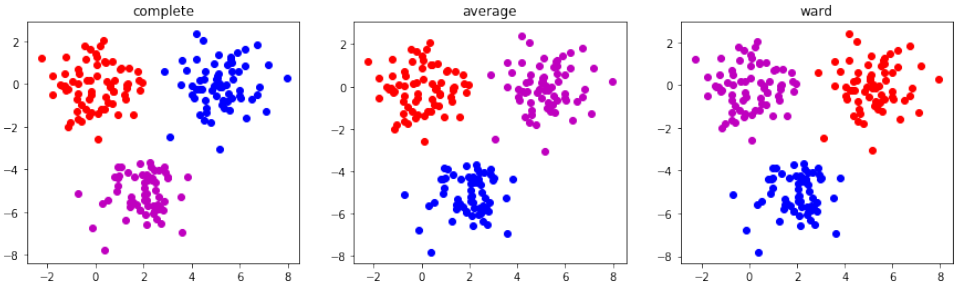


Figure 9.8: Example of agglomerative clustering (Gaussian clusters).

9.2.2 Connection with minimum spanning trees

Given a connected edge-weighted undirected graph $G = (V, E)$, with weight function $d \colon V \times V \to \mathbb{R}_+$, the problem of minimum spanning tree (MST) is to find a subgraph $T = (V, E')$ that connects

all vertices with minimal sum of weights $\sum_{\{u,v\}\in E'} d(u,v)$. It is easy to see that such a subgraph T is necessarily a tree. Indeed, if there where a cycle in T, we could remove an edge on the cycle to get a new subgraph connecting all vertices and with fewer sum of weights.

There are several classic techniques for finding an MST: Borůvka's, Kruskal's, Prim's and reverse-delete algorithms. All are greedy methods. In particular, Kruskal's algorithm consists in adding edges in increasing weight (at the initialization, we consider an edge with minimal weight), skipping those whose addition would create a cycle.

Back to clustering and similarly to spectral methods, let us consider the similarity graph, in which each vertex represents a data point and vertices X_i and X_j are all connected by an edge whose weight is their distance $d(X_i, X_j)$. Then, applying Kruskal's algorithm to this graph is exactly performing a single linkage agglomerative clustering on the set $(X_1, \dots, X_n)$. Indeed, picking edges in increasing weight in Kruskal's algorithm corresponds to merging the closest clusters in single linkage. In addition, discarding edges that would create a cycle is exactly saying that we measure distances and merge two different clusters in single linkage.

To complete the comparison, once an MST T is created, deleting the $k-1$ most expensive edges in T produces k connected subgraphs, which are exactly the clusters produced by single linkage.

9.3 (⋆) Density-based clustering

The algorithm called density-based spatial clustering of applications with noise (DBSCAN) assumes that clusters are dense regions separated by low-density corridors. It is one of the most common clustering algorithms because of its efficiency and its ability to automatically determine the number of clusters.

Given two parameters, a radius $\epsilon > 0$ and a minimal number of neighbors m, DBSCAN considers three types of points:

1. core points are points that have at least m neighbors within a distance ϵ (the ϵ-neighborhood), including themselves (the ϵ-neighborhood is at least a unit set);
2. reachable points are non-core points that fall in the neighborhood of a core point;
3. outliers are other points.

Clusters are formed by core points that fall in the neighborhoods of each other and by their reachable points.

Remark 9.3.1. *With $m = 2$, DBSCAN performs the same clustering as single linkage for which the dendrogram has been cut at height ϵ.*

Advantages

- ◇ DBSCAN makes assumption on cluster density, thus it determines automatically the number of clusters.
- ◇ There is no shape restriction for discovered clusters (while k-means requires convex clusters).
- ◇ DBSCAN prevents the single-link effect (different clusters being connected by a thin line of points).
- ◇ There is a notion of outliers/noise.
- ◇ DBSCAN is mostly insensitive to sample ordering.

Algorithm 14 DBSCAN.

Input: $\epsilon > 0$ (neighborhood radius), $m \in \mathcal{N}$ (minimal number of neighbors), $\{X_i\}_{1 \leq i \leq n}$ (training sample).

```
T ← {X_i}_{1≤i≤n} (unlabeled points)
k ← 0 (current number of clusters)
while T ≠ ∅ do
    pick X in T
    N ← ε-neighborhood of X
    if |N| ≥ m then
        k ← k + 1
        initialize a new cluster Ĉ_k = ∅
        move X from T to Ĉ_k
        S ← (N\{X}) ∩ T (unlabeled neighbors)
        while S ≠ ∅ do
            pick Y in S
            move Y from S to Ĉ_k (and remove Y from T)
            N' ← ε-neighborhood of Y
            if |N'| ≥ m then
                S ← S ∪ (N' ∩ T) (unlabeled neighbors)
            end if
        end while
    end if
end while
```

Output: $(\hat{C}_1, \ldots, \hat{C}_k, T)$ *(k clusters and a set of outliers)*

Drawbacks

- ◇ DBSCAN is only deterministic on core and noise points. Border points that are reachable from several clusters can be part of either cluster, depending on the order the data is processed.
- ◇ DBSCAN cannot cluster data sets well with large differences in densities, since the parameters ϵ and m cannot be chosen appropriately for all clusters.

Choice of the parameters

1. Because of the curse of dimensionality, m should be chosen of the order of $2d$.
2. The radius ϵ can be chosen at the elbow of the monotonic curve of maximum distances between a point and its $m - 1$ nearest neighbors.

DBSCAN in action

1. A fancy demo of DBSCAN.

9.4 (⋆) Clustering evaluation

When the ground truth is known, several criteria can be used for evaluating a clustering performance: adjusted rand index, normalized and adjusted mutual informations, V-measure, Fowlkes-Mallows score. All scores measure the similarity between class labels ignoring permutation. Besides, adjusted indexes

and Fowlkes-Mallows score have chance normalization: random uniform label assignment gets a 0 score.

Now, we briefly present some methods for assessing a clustering performance when the ground truth is not known. These methods can be used to select the number of clusters k or other parameters.

9.4.1 Elbow method

The elbow method is a method of validation of a partitioning based on the intraclass inertia. It is often used to choose an appropriate number of clusters and is particularly suited for convex clusters (due to the nature of its criterion).

The elbow method looks at the intraclass inertia (or inversely at the percentage of variance explained, that is the ratio of the interclass inertia to the total inertia) as a function of the number of clusters (see Figures 9.9 and 9.10).

The idea of the elbow method is that such a curve has two regimes:

- ◇ going from 1 to 2 (then 3) clusters will decrease a lot the intraclass inertia since these clusters help discovering groups;
- ◇ once we have more clusters than actual groups, there is a very limited gain (in the intraclass inertia) of adding a new cluster.

Therefore, intuitively, there should be an angle in the graph, separating the two regimes mentioned above. The point where this angle appears is called an *elbow* and precisely indicates the number of clusters to choose. However, in practice this elbow cannot always be unambiguously identified.

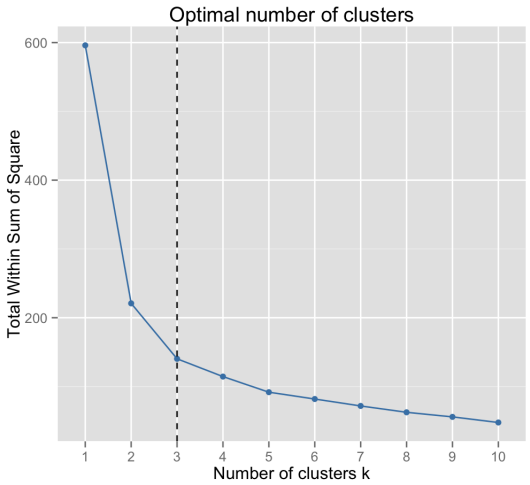


Figure 9.9: The elbow method (in theory). Courtesy of WWW.

9.4.2 Silhouette coefficient

For all $i \in [i]$, let $\hat{C}_j$ be the cluster associated to X_i and

$$a_i = \frac{1}{|\hat{C}_j| - 1} \sum_{\substack{Y \in \hat{C}_j \\ Y \neq X_i}} d(X_i, Y),$$

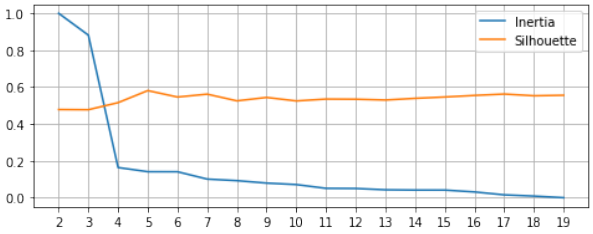


Figure 9.10: The elbow method (in practice).

as well as

$$b_i = \min_{\substack{1\leq \ell \leq k \\ \ell \neq j}} \frac{1}{|\hat{C}_\ell|} \sum_{Y \in \hat{C}_j} d(X_i, Y)$$

being respectively the average distance of X_i to its companions and to the members of the *neighboring cluster*. These values can be interpreted as how well X_i is compliant with its cluster and different from the neighboring cluster. The silhouette value for X_i is

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

The average s_i over all data of a cluster $\frac{1}{|\hat{C}_j|} \sum_{\substack{1\leq i \leq n \\ X_i \in \hat{C}_j}} s_i$ measures how tightly grouped all the data in the cluster are, and how distant from the neighboring cluster they are. In this sense, it is a density-based index, which is close to 1 when $\hat{C}_j$ is a dense group separated by a low-density corridor from its neighbor (similarly to the operating of DBSCAN).

The silhouette coefficient is well suited for choosing the number k of clusters: if there are too many or too few clusters, some of them will typically display much narrower silhouettes than the others, meaning that they should or should not be split (see for instance this interesting example).

Averaging the silhouette coefficients over all data produces a measure of how appropriate (the higher, the better) is the partitioning:

$$s = \frac{1}{n} \sum_{i=1}^{n} s_i.$$

9.4.3 Calinski-Harabasz index

Another density-based index is the Calinski-Harabasz coefficient, which boils down to a normalized ratio of the between-cluster dispersion (or interclass inertia) and the within-cluster dispersion (or intraclass inertia):

$$s = \frac{n-k}{k-1} \frac{b}{w},$$

where b is the between-cluster dispersion

$$b = \sum_{j=1}^{k} \frac{|\hat{C}_j|}{n} \left\| \hat{\mu} - \hat{\mu}_j \right\|_{\ell_2}^2,$$

and w is the within-cluster dispersion:

$$w = \frac{1}{n} \sum_{j=1}^{k} \sum_{X \in \hat{C}_j} \left\| X - \hat{\mu}_j \right\|_{\ell_2}^2,$$

where μ is the global mean and $\hat{\mu}_j$ is the mean of the cluster $\hat{C}_j$.

Remark 9.4.1. *Paradoxically, both previous indexes are generally higher for convex clusters than for other concepts of clusters, such as those produced by DBSCAN.*

Classement	Date	Participant(s)	Score
1	16 octobre 2026 21:04	yg.yao (yuguang.yao@etu.sorbonne-universite.fr)	6,0332
2	18 octobre 2026 18:33	OverSoul (thomas.anato@student-cs.fr)	6,0863
3	16 octobre 2026 21:11	fb_2110 (felix.bos8723@gmail.com)	6,2044
4	20 octobre 2026 16:39	martindelamare (martindelamare@icloud.com)	7,7074
5	9 octobre 2026 00:28	funnyvamp (1298926429@qq.com)	8,0283
6	18 octobre 2026 11:20	yassha26 (yassine.harrara@edhec.com)	8,0372
7	19 octobre 2026 09:53	juan_nicolas.mendoza_roncancio (Juan.Mendoza_Roncancio@etu.sorbonne-universite.fr)	8,0575
8	18 octobre 2026 22:00	140bricks (thomas.rze@icloud.com)	8,5213
9	18 octobre 2026 12:24	maevanoukoua (maeva-sandy.noukoua@ensta.fr)	8,5319
10	16 octobre 2026 08:10	Aliette (Xiran.Pang@etu.sorbonne-universite.fr)	8,6581
11	23 octobre 2026 11:01	matleniz (mat24.niz@gmail.com)	8,8839
12	20 octobre 2026 22:31	JohnnyLi (johnny.li@etu.sorbonne-universite.fr)	9,0493
13	18 octobre 2026 18:09	YUWenbo (y3082812594@outlook.com)	9,3987
14	22 octobre 2026 16:31	RFitah (rajaobelinafitahiry@yahoo.com)	9,5098
15	19 octobre 2026 17:31	Beedigoua1 (gouabeedicd@gmail.com)	9,6496

Fin dimanche 8 novembre - 22h.

Lecture 10: Dimensionality reduction

Dimensionality reduction is related to the concept of lossy compression in information theory. It consists in transforming data from a high-dimensional space to new data from a lower-dimensional space, with as few loss of information as possible.

Dimensionality reduction is motivated by:

- ◇ computational challenges;
- ◇ poor generalization ability (for example, for nearest neighbors classifiers, the sample complexity increases exponentially with the dimension);
- ◇ interpretability of the data (finding a meaningful structure, displaying the data).

Theorem 65 ([Shalev-Shwartz and Ben-David, 2014, Theorems 19.4 and 19.5]). *Let $\mathcal{X} = [0,1]^d$, $\mathcal{Y} = \{\pm 1\}$, $k \in \mathbb{N}^*$ and $(X_1, Y_1), \dots, (X_n, Y_n), (X, Y)$ iid. We note $g_n : \mathcal{X} \to \mathcal{Y}$ the k-nearest neighbor classifier based on $(X_1, Y_1), \dots, (X_n, Y_n)$, $g^* : \mathcal{X} \to \mathcal{Y}$ a Bayes classifier for (X, Y) and $\eta : \mathcal{X} \to \mathbb{P}(Y = 1 \mid X = x)$ the regression function.*

For any distribution on (X, Y) such that η is L-Lipschitz continuous for some $L > 0$ and for any $k \geq 10$:

$$\mathbb{P}(g_n(X) \neq Y) \leq \left(1 + \sqrt{\frac{8}{k}}\right) \mathbb{P}(g^*(X) \neq Y) + \frac{6L\sqrt{d} + k}{n^{\frac{1}{d+1}}}.$$

Moreover, for all $L > 1$ and $k \in \mathbb{N}^*$, there exists a distribution for (X, Y) such that η is L-Lipschitz continuous,

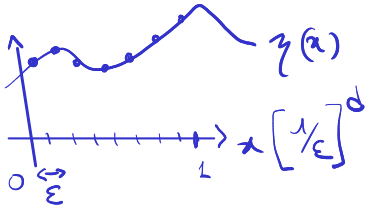
$$g^*(X) = Y \text{ a.s.} \qquad \text{and} \qquad \forall n \leq \frac{(L+1)^d}{2}, \ \mathbb{P}(g_n(X) \neq Y) \geq \frac{1}{4}.$$

It is easy to see that for the last term of the generalization bound in Theorem 65 to be smaller than $\epsilon > 0$, we should have $n \geq \left(\frac{6L\sqrt{d}+k}{\epsilon}\right)^{d+1}$. That is, the size of the training set should increase exponentially with the dimension. Besides, the rest of Theorem 65 tells us that this is not just an artifact of our upper bound, since there exists distributions for which we need exponentially big training samples to get low errors. This phenomenon is often referred to as the *curse of dimensionality*.

Other examples of the curse of dimensionality

Sampling

Sampling evenly a unit hypercube with a lattice that has a spacing of $\epsilon \in (0, 1]$, requires ϵ^{-d} points. For instance, for $\epsilon = 10^{-2}$, 100 points are required to sample the segment $[0, 1]$ while 10^{20} points are needed for the 10-dimensional unit hypercube.



Reciprocally, 100 points represent well the segment $[0,1]$, while they are really insufficient for "covering" $[0,1]^{10}$.

Distance functions

Given a radius $r > 0$, the ratio of the volumes of an inscribed hypersphere with radius r (in dimension d) to a hypercube with edges of length $2r$ is given by

$$\frac{\frac{2r^d\pi^{d/2}}{d\Gamma(d/2)}}{(2r)^d} = \frac{\pi^{d/2}}{2^{d-1}d\Gamma(d/2)} = \left(\frac{\sqrt{\pi}}{2}\right)^d \frac{2}{d\Gamma(d/2)},$$

where Γ is the gamma function and $\sqrt{\pi}/2 \approx 0.9$. It is easy to see that this quantity goes exponentially fast to 0 when d grows. As a consequence, we are used to say that "points are concentrated in the corners" of the hypercube. This assertion is enforced by the fact that the distance between a corner and the hypercube center is $r\sqrt{d}$. In this sens, the major part of the high-dimensional space is far away from the centre.

The Volume is in a narrow annulus

The ratio of the volume of a sphere of radius $(1-\epsilon)r$ ($\epsilon \in (0,1]$, $r > 0$) to the volume of a sphere of radius r is

$$\frac{\frac{2((1-\epsilon)r)^d\pi^{d/2}}{d\Gamma(d/2)}}{\frac{2r^d\pi^{d/2}}{d\Gamma(d/2)}} = (1-\epsilon)^d,$$

which decreases exponentially fast to 0 as d goes to infinity. In other words, in high dimensions, all of the volume of the sphere is concentrated in a narrow annulus at the surface.

Going back to dimensionality reduction, this chapter mainly focuses on linear methods, that is, finding a matrix $W \in \mathbb{R}^{p\times d}$, where d is the input dimension and p is the desired reduced dimension, that induces the mapping $\varphi : x \in \mathbb{R}^d \mapsto Wx \in \mathbb{R}^p$. Without supervised information, a natural criterion for choosing W is that the reduction mapping enables a reasonable recovery of the original data x.

The fundamental hypothesis underlying dimensionality reduction is that the observation X (with values in $\mathbb{R}^d$) is a noisy version of a latent variable Z (say $X = Z + \epsilon$) living in a manifold[1] $\mathcal{M} \subseteq \mathbb{R}^d$ of dimension p lower than d (in other words, "the data does not fill the entire space"). In that, dimensionality reduction aims at (estimating the manifold $\mathcal{M}$ and) "projecting" the data with few distortion of the geometry (this corresponds to $\psi \circ \varphi$ here after). More practically, if the manifold can be described by $\mathcal{M} = \psi(\mathbb{R}^p)$ (where $\psi\colon \mathbb{R}^p \to \mathcal{M}$ is a homeomorphism with particular features such as isometry), dimensionality reduction aims at reducing the data in $Y = \varphi(X)$ such that $X \approx \psi(Y)$.

10.1 Linear methods

10.1.1 Principal component analysis

As explained before, the most straightforward approach of dimensionality reduction is to find

- ◇ a compression function $\varphi\colon \mathbb{R}^d \to \mathbb{R}^p$;

[1] Informally, a manifold is a (topological) space, that is locally homeomorphic to the Euclidean space of dimension p (which is also the dimension of the manifold). The locally feature means that for any point, there exists a neighborhood (hence the need of the topology), that is homeomorphic to the Euclidean space.

- a decompression function $\psi : \mathbb{R}^p \to \mathbb{R}^d$,

where $p < d$ is the reduced dimension, such that φ enables a reasonable recovery of the original data, *i.e.* that $\psi \circ \varphi$ has minimal reconstruction error. Given the random variable of interest $X \in \mathbb{R}^d$ ($X \in L^2$), a natural variational formulation, of dimensionality reduction is

$$\underset{\substack{\varphi:\mathbb{R}^d \to \mathbb{R}^p \\ \psi:\mathbb{R}^p \to \mathbb{R}^d}}{\text{minimize}} \ \mathbb{E}\left[\left\|X - (\psi \circ \varphi)(X)\right\|_{\ell_2}^2\right].$$

Since we are interested in linear methods, we assume that φ and ψ are two linear functions. Then, the problem boils down to finding a compression matrix $W \in \mathbb{R}^{p \times d}$ such that $\varphi : x \in \mathbb{R}^d \mapsto Wx \in \mathbb{R}^p$ and a recovery matrix $U \in \mathbb{R}^{d \times p}$ such that $\psi : y \in \mathbb{R}^p \mapsto Uy \in \mathbb{R}^d$. The optimization problem thus becomes

$$\underset{U \in \mathbb{R}^{d \times p},\, W \in \mathbb{R}^{p \times d}}{\text{minimize}} \ \mathbb{E}\left[\left\|X - UWX\right\|_{\ell_2}^2\right]. \tag{P23}$$

Remark 10.1.1. *When φ and ψ are parameterized by neural networks, this approach corresponds to auto-encoders.*

Property 66. *If $p \leq d$ and (P23) admits a solution, then there exists $V \in \mathbb{R}^{d \times p}$, such that $V^\top V = I_p$ and the pair $(V, V^\top)$ is also a solution to (P23),* i.e. *the compression $\varphi : x \in \mathbb{R}^d \mapsto V^\top x \in \mathbb{R}^p$ and recovery $\psi : y \in \mathbb{R}^p \mapsto Vy \in \mathbb{R}^d$ functions are optimal for linear dimensionality reduction.*

The proof will be done during the class.

Owing to the preceding lemma, (P23) boils down to minimizing $\mathbb{E}\left(\left\|X - UU^\top X\right\|_{\ell_2}^2\right)$ with respect to $U \in \mathbb{R}^{d \times p}$ such that $U^\top U = I_p$. In addition, by simple algebra, one has

$$\begin{aligned}
\mathbb{E}\left(\left\|X - UU^\top X\right\|_{\ell_2}^2\right) &= \mathbb{E}\left(\left\|X\right\|_{\ell_2}^2 + X^\top UU^\top UU^\top X - 2X^\top UU^\top X\right) \\
&= \mathbb{E}\left(\left\|X\right\|_{\ell_2}^2 - X^\top UU^\top X\right) && (U^\top U = I_p) \\
&= \mathbb{E}\left(\left\|X\right\|_{\ell_2}^2\right) - \mathbb{E}\left(X^\top UU^\top X\right) \\
&= \mathbb{E}\left(\left\|X\right\|_{\ell_2}^2\right) - \mathbb{E}\left(\operatorname{tr}\left(X^\top UU^\top X\right)\right) && (X^\top UU^\top X \text{ is a scalar}) \\
&= \mathbb{E}\left(\left\|X\right\|_{\ell_2}^2\right) - \mathbb{E}\left(\operatorname{tr}\left(U^\top XX^\top U\right)\right) && (\star) \\
&= \mathbb{E}\left(\left\|X\right\|_{\ell_2}^2\right) - \operatorname{tr}\left(U^\top \mathbb{E}\left(XX^\top\right) U\right),
\end{aligned}$$

where we have used that the trace is invariant under cyclic permutations ($\star$). Therefore, once again, (P23) boils down to

$$\underset{U \in \mathbb{R}^{d \times p};\, U^\top U = I_p}{\text{maximize}} \ \operatorname{tr}\left(U^\top \mathbb{E}\left(XX^\top\right) U\right). \tag{P24}$$

By Theorem 61, it results that a solution to (P24) is $U = V_+$, where $V_+ \in \mathbb{R}^{d \times p}$ is the matrix of the leading eigenvectors of $\mathbb{E}\left[XX^\top\right]$. Then, dimensionality reduction mapping is $\varphi : x \in \mathbb{R}^d \mapsto V_+^\top x \in \mathbb{R}^p$ and the reconstruction mapping $\psi : y \in \mathbb{R}^d \mapsto V_+ y \in \mathbb{R}^d$. This approach is called PCA, due to its relation to eigendecomposition.

Preuve

Appelons $(U, W) \in \mathbb{R}^{d\times p} \times \mathbb{R}^{p\times d}$ une solut° de (P23)

But : on va montrer qu'on peut construire $V \in \mathbb{R}^{d\times p}$ tq

$V^T V = I_p$ et $\mathbb{E}\left[\|X - UWX\|_2^2\right] \geq \mathbb{E}\left[\|X - VV^T X\|_2^2\right]$. ←

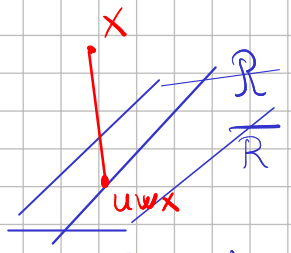
$UWX \in R$

$R = \text{range}(UW) = \{UWx, x \in \mathbb{R}^d\}$ $r = \text{rang}(UW) \leq p \ (\leq d)$.

$\dim(R) = r$, R sev de $\mathbb{R}^d$

$(v_1, \dots, v_r)$ BON de R.

Complète en $\underbrace{(v_1, \dots, v_r, v_{r+1}, \dots, v_d)}_{p}$ BON de $\mathbb{R}^d$.



$\overline{R} = \text{span}\{v_1, \dots, v_p\} = \text{range}(V)$ avec $V = [v_1 | \dots | v_p] \in \mathbb{R}^{d\times p}$

et $V^T V = I_p$.

Si P le projecteur ⊥ sur $\overline{R}$.

$UWX \in R \subset \overline{R}$ donc $\|X - UWX\|_2 \geq \|X - PX\|_2$

D'où $\mathbb{E}\left[\|X - UWX\|_2^2\right] \geq \mathbb{E}\left[\|X - PX\|_2^2\right] = \mathbb{E}\left[\|X - VV^T X\|_2^2\right]$.

Ici $P = V \underbrace{(V^T V)^{-1}}_{I_p} V^T = VV^T$

Remark 10.1.2. *The proof of Theorem 61 tells us that*

$$\operatorname{tr}\left(V_+^\top \mathbb{E}\left[XX^\top\right] V_+\right) = \sum_{i=d-p+1}^{d} \lambda_i \leq \sum_{i=1}^{d} \lambda_i,$$

where $0 \leq \lambda_1 \leq \cdots \leq \lambda_d$ *are the sorted eigenvalues of* $\mathbb{E}\left[XX^\top\right]$. *Since the bound is attained for* $p = d$ *(that is, there is no dimensionality reduction), the maximal value* $\operatorname{tr}\left(V_+^\top \mathbb{E}\left[XX^\top\right] V_+\right)$ *serves as an indicator of the "quality" of the approximation made when reducing the dimensionality of the data.*

Let us remark that we can go back to (P23):

$$\begin{aligned}
\mathbb{E}\left[\left\|X - V_+ V_+^\top X\right\|_{\ell_2}^2\right] &= \mathbb{E}\left[\|X\|_{\ell_2}^2\right] - \operatorname{tr}\left(V_+^\top \mathbb{E}\left[XX^\top\right) V_+\right] \\
&= \mathbb{E}\left[\operatorname{tr}(XX^\top)\right] - \operatorname{tr}\left(V_+^\top \mathbb{E}\left[XX^\top\right) V_+\right] \\
&= \operatorname{tr}\left[\mathbb{E}(XX^\top)\right] - \operatorname{tr}\left(V_+^\top \mathbb{E}\left[XX^\top\right) V_+\right] \\
&= \sum_{i=1}^{d-p} \lambda_i.
\end{aligned}$$

It comes then that the relative error of reconstruction is:

$$\frac{\mathbb{E}\left[\left\|X - (\psi \circ \varphi)(X)\right\|_{\ell_2}^2\right]}{\mathbb{E}\left[\|X\|_{\ell_2}^2\right]} = \frac{\sum_{i=1}^{d-p} \lambda_i}{\sum_{i=1}^{d} \lambda_i}.$$

In addition, for all $x \in \mathbb{R}^d$ *and* $j \in [p]$, $\left(V_+^\top x\right)_j = \sum_{i=1}^{d} (V_+)_{ij} x_i$. *In particular, for all* $i \in [d]$, $(V_+)_{ij}$ *describes the "influence" of the explicative variable* x_i *to the* j^{th} *component. Therefore, if* $|(V_+)_{ij}|$ *is large, it likely explains disparities of points in the* j^{th} *direction of the reduced space.*

10.1.2 Link with variance maximization

Up to now, PCA has been defined as building linear functions for compression and reconstruction (in other words, as projecting a random vector on a linear subspace with minimal error) but there is no reason not to consider an affine transformations (that is projecting on an affine subspace) :

$$\underset{\substack{U \in \mathbb{R}^{d\times p},\, W \in \mathbb{R}^{p\times d} \\ \mu \in \mathbb{R}^p,\, \lambda \in \mathbb{R}^d}}{\text{minimize}} \quad \mathbb{E}\left[\left\|X - \{U(WX + \mu) + \lambda\}\right\|_{\ell_2}^2\right].$$

It comes trivially that, for (U, W) fixed, the optimal pairs (μ, λ) are

$$\begin{cases} \mu \in \mathbb{R}^p \\ \lambda = \mathbb{E}\,X - UW\,\mathbb{E}\,X - U\mu. \end{cases}$$

In particular, we choose $\mu = 0$, which leads to $\lambda = \mathbb{E}\,X - UW\,\mathbb{E}\,X$ and to the affine PCA problem:

$$\underset{U \in \mathbb{R}^{d\times p},\, W \in \mathbb{R}^{p\times d}}{\text{minimize}} \quad \mathbb{E}\left[\left\|(X - \mathbb{E}\,X) - UW(X - \mathbb{E}\,X)\right\|_{\ell_2}^2\right].$$

That is why, it is a common practice to center the data before applying PCA, namely considering the random variable $Z = X - \mathbb{E}\,X$ instead of X. Then, PCA aims at solving

$$\underset{U \in \mathbb{R}^{d\times p}: U^\top U = I_p}{\text{maximize}} \ \operatorname{tr}\left(U^\top \mathbb{E}\left[ZZ^\top\right] U\right) = \operatorname{tr}\left(U^\top \mathbb{V}(X) U\right),$$

where $\mathbb{V}(X) = \mathbb{E}\left[(X - \mathbb{E}\,X)(X - \mathbb{E}\,X)^\top\right]$ is the covariance matrix of X. If follows that the principal components (the column vectors of V_+) are the orthonormal vectors that "maximize the variance of X".

To be more formal, let us describe what we mean by "maximizing" the variance of X. For this purpose, we define recursively the sequence $(u_1, \ldots, u_p)$. Let $u_1 \in \mathbb{R}^n$ be the normalized vector (direction) that maximizes the unidirectional variance of X, namely a solution to:

$$\begin{array}{ll} \underset{u\in\mathbb{R}^d}{\text{maximize}} & \mathbb{V}(u^\top X) \\ \text{s. t.} & \|u\|_{\ell_2} = 1. \end{array} \tag{P25}$$

Then, for all $k \in [p-1]$, given $(u_1, \ldots, u_k)$, let u_{k+1} be solution to

$$\begin{array}{ll} \underset{u\in\mathbb{R}^d}{\text{maximize}} & \mathbb{V}(u^\top X) \\ \text{s. t.} & \begin{cases} \|u\|_{\ell_2} = 1 \\ \forall j \in [k]\colon u^\top u_j = 0. \end{cases} \end{array} \tag{P26}$$

Now, we will show by induction that $(u_p, \ldots, u_1)$ corresponds to the principal components, namely the column vectors of V_+, which are also the leading eigenvectors of $\mathbb{V}(X)$.

First, since $\mathbb{V}(u^\top X) = u^\top \mathbb{V}(X) u = \operatorname{tr}\left(u^\top \mathbb{V}(X) u\right)$, by Theorem 61, the leading eigenvector of $\mathbb{V}(X)$ is solution to (P25). So we can set $u_1 = v_d$ (with the preceding notation).

Second, for any $k \in [p-1]$, let us assume that $(u_k, \ldots, u_1) = (v_{d-k+1}, \ldots, v_d)$. Since $(u_1, \ldots, u_k)$ are fixed, maximizing $\mathbb{V}(u^\top X) = u^\top \mathbb{V}(X) u$ is the same as maximizing $u^\top \mathbb{V}(X) u + \sum_{j=1}^k u_j^\top \mathbb{V}(X) u_j$. Therefore, (P26) is similar to

$$\begin{array}{ll} \underset{u\in\mathbb{R}^d,\, U\in\mathbb{R}^{d\times(k+1)}}{\text{maximize}} & \displaystyle\sum_{j=1}^{k} u_j^\top \mathbb{V}(X) u_j + u^\top \mathbb{V}(X) u = \operatorname{tr}\left(U^\top \mathbb{V}(X) U\right) \\ \text{s. t.} & \begin{cases} U = [u|u_k|\ldots|u_1] = [u|v_{d-k+1}|\ldots|v_d] \\ U^\top U = I_{k+1}. \end{cases} \end{array} \tag{P27}$$

Let us remark that (P27) has a solution since $(v_{d-k+1}, \ldots, v_d)$ are orthonormal. Now, let us consider $V_+ = [v_{d-k}|v_{d-k+1}|\ldots|v_d]$. Then, by Theorem 61, $\operatorname{tr}\left(V_+^\top \mathbb{V}(X) V_+\right) \geq \operatorname{tr}\left(U^\top \mathbb{V}(X) U\right)$ for all $U \in \mathbb{R}^{d\times(k+1)}$ such that $U^\top U = I_{k+1}$. In particular, this is also true for all U that fulfill the constraint of (P27) (said *admissible*). Since V_+ is also admissible, this proves that V_+, along with its first column $u = v_{d-k}$, is a maximizer of (P26).

Remark 10.1.3. *As explained previously, $\sum_{i=d-p+1}^{d} \lambda_i$ measures the quality of the approximation made by PCA with p components. In view of the variance maximization paradigm, $r = \frac{\sum_{i=d-p+1}^{d} \lambda_i}{\sum_{i=1}^{d} \lambda_i}$ is often called the "ratio of explained variance" and is a normalized indicator of the quality of approximation.*

10.1.3 Link with the Gram matrix

In practice, we are provided with a sample $\{X_1, \dots, X_n\} \subseteq \mathbb{R}^d$. Then, considering the empirical twin of $\mathbb{E}[XX^\top]$, which is $\frac{1}{n}\sum_{i=1}^n X_i X_i^\top$, PCA boils down to finding the p leading eigenvectors of the empirical (and scaled by n) covariance matrix $C = \sum_{i=1}^n X_i X_i^\top = \mathbf{X}^\top \mathbf{X} \in \mathbb{R}^{d\times d}$, where $\mathbf{X} \in \mathbb{R}^{n\times d}$ is the matrix whose rows are the observations X_i ($i \in [n]$).

As we are interested in dimensionality reduction, it is quite licit to assume that the dimension d is very big and that $d \geq n$. Therefore, PCA has to diagonalize a (big) $d \times d$ matrix, while saving only the p leading eigenvectors. The forthcoming property tells us that (as long as the reduced dimension $p \leq n$) PCA can be implemented in a cheaper manner by diagonalizing the (small) Gram matrix $K = \left(X_i^\top X_j\right)_{1\leq i,j\leq n} = \mathbf{X}\mathbf{X}^\top \in \mathbb{R}^{n\times n}$.

Property 67. *Let us assume that $d \geq n$ and note $r = \operatorname{rank}(K)$. If $\lambda_1 \geq \cdots \geq \lambda_n$ are the eigenvalues of K with orthonormal eigenvectors $(v_1, \dots, v_n)$, then there exists $(u_{r+1}, \dots, u_d) \in (\mathbb{R}^d)^{d-r}$ such that $\lambda_1 \geq \cdots \geq \lambda_r > 0 \geq \cdots \geq 0$ are eigenvalues of C with orthonormal eigenvectors $(\mathbf{X}^\top v_1/\sqrt{\lambda_1}, \dots, \mathbf{X}^\top v_r/\sqrt{\lambda_r}, u_{r+1}, \dots, u_d)$.*

The proof will be done during the class.

Remark 10.1.4. *Assuming that p is less than the rank of C (or K) and denoting*

$$V_+ = \left[\frac{\mathbf{X}^\top v_{n-p+1}}{\sqrt{\lambda_{n-p+1}}} \middle| \dots \middle| \frac{\mathbf{X}^\top v_n}{\sqrt{\lambda_n}}\right] \in \mathbb{R}^{d\times p}$$

the matrix of the dimensionality reduction mapping $x \in \mathbb{R}^d \mapsto V_+^\top x \in \mathbb{R}^p$, the matrix of reduced representations becomes:

$$\begin{aligned}
\mathbb{Z} &= \left(V_+^\top \mathbf{X}^\top\right)^\top = \mathbf{X}V_+ \\
&= \left[\frac{K v_{n-p+1}}{\left\|\mathbf{X}^\top v_{n-p+1}\right\|_{\ell_2}} \middle| \dots \middle| \frac{K v_n}{\left\|\mathbf{X}^\top v_n\right\|_{\ell_2}}\right] \\
&= \left[\frac{\lambda_{n-p+1} v_{n-p+1}}{\sqrt{\lambda_{n-p+1}}} \middle| \dots \middle| \frac{\lambda_n v_n}{\sqrt{\lambda_n}}\right] \\
&= \left[\sqrt{\lambda_{n-p+1}} v_{n-p+1} \middle| \dots \middle| \sqrt{\lambda_n} v_n\right]
\end{aligned}$$

10.1.4 Link with singular values

The next theorem gives a more general solution to PCA.

Theorem 68 (singular value decomposition (SVD) [Shalev-Shwartz and Ben-David, 2014, Appendix C.4]). *Let m and n be two positive integers and $A \in \mathbb{R}^{m \times n}$. Let $r = \text{rank}(A)$ be the rank of A, then there exist $U \in \mathbb{R}^{m \times r}$, $D \in \mathbb{R}^{r \times r}$ and $V \in \mathbb{R}^{n \times r}$ such that*

$$A = UDV^\top,$$

and

- *the columns of U are orthonormal: $U^\top U = I_r$;*
- *D is diagonal with positive and uniquely defined entries $\sigma_1, \ldots, \sigma_r$ (called singular values);*
- *the columns of V are orthonormal: $V^\top V = I_r$.*

Furthermore, denoting $U = [u_1 | \ldots | u_r]$ and $V = [v_1 | \ldots | v_r]$, one has:

- *$A = \sum_{i=1}^{r} \sigma_i u_i v_i^\top$;*
- *for all $i \in [r]$, u_i and v_i are left and right singular vectors: $Av_i = \sigma_i u_i$ and $A^\top u_i = \sigma_i v_i$;*
- *for all $i \in [r]$, u_i is an eigenvector of $AA^\top$ with eigenvalue σ_i^2;*
- *for all $i \in [r]$, v_i is an eigenvector of $A^\top A$ with eigenvalue σ_i^2.*

In practice, SVD is a tool from linear algebra, that is more robust than eigendecomposition for solving PCA. It computes neither $\mathbf{X}^\top \mathbf{X}$ nor $\mathbf{X}\mathbf{X}^\top$. In addition, PCA is performed even faster by computing the top p singular values, along with the left and right singular vectors of $\mathbf{X}$ thanks to a truncated SVD. Let us remark that for computing the dimensionality reduction mapping (inductive learning), we need the eigenvectors of $C = \mathbf{X}^\top \mathbf{X}$, *i.e.* the right singular vectors, and for the matrix of reduced representations (transductive learning), the eigenvectors of $K = \mathbf{X}\mathbf{X}^\top$, *i.e.* the left singular vectors.

Algorithm 15 Reduced representation by PCA.

Input: $\mathbf{X} \in \mathbb{R}^{n \times d}$ (data matrix), p (reduced dimension).

Second order matrix

$C \leftarrow \mathbf{X}^\top \mathbf{X}$

$V \leftarrow p$ leading eigenvectors of C

$U \leftarrow \mathbf{X}V$

Gram matrix

$K \leftarrow \mathbf{X}\mathbf{X}^\top$

$\lambda_1, \ldots, \lambda_p \leftarrow p$ leading eigenvalues

$V \leftarrow p$ leading eigenvectors of K

$U \leftarrow \left[\sqrt{\lambda_1} v_1 | \ldots | \sqrt{\lambda_p} v_p\right]$

SVD

$\sigma_1, \ldots, \sigma_p \leftarrow p$ leading singular values of $\mathbf{X}$

$V \leftarrow p$ leading left singular vectors of $\mathbf{X}$

$U \leftarrow [\sigma_1 v_1 | \ldots | \sigma_p v_p]$

Output: $U \in \mathbb{R}^{n \times p}$.

10.1.5 Random projection

Although PCA is very appealing by its simplicity and probabilistic interpretation, it requires computing singular vectors, which can be very expensive for very high-dimensional data, or very big samples. That is why, we describe in this section a very cheap way of reducing dimension. Here, the criterion of interest is not a reasonable recovery of the original data but saving the original placement of the data points $\{X_1, \dots, X_n\}$ with respect to each other. In other words, we would like to preserve pairwise distances.

Roughly speaking, Theorem 70 states that if the reduced dimension p is proportional to $\log(n)/\epsilon^2$, then a random matrix $W \in \mathbb{R}^{p\times d}$ produces a dimensionality reduction mapping $\varphi : x \in \mathbb{R}^d \mapsto Wx \in \mathbb{R}^p$, that preserves pairwise distances up to an error ϵ.

Lemma 69 (Concentration of a chi-squared variable). *Let $p \in \mathbb{N}^*$ and $Z \sim \chi^2_p$. Then for all $\epsilon \in (0,1)$:*

$$\mathbb{P}\left(\left|\frac{Z}{p} - 1\right| > \epsilon\right) \leq 2\,\mathrm{e}^{-\frac{p\epsilon^2}{8}}.$$

The proof is a good exercise.

Theorem 70 (Johnson-Lindenstrauss Lemma). *Let $\mathcal{S} \subseteq \mathbb{R}^d$ be a finite set of vectors with cardinality $n \geq 2$ and $W \in \mathbb{R}^{p\times d}$ be a random matrix such that its entries $\{W_{ij}\}_{\substack{1\leq i\leq p\\ 1\leq j\leq d}}$ are* iid *and distributed according to $\mathcal{N}\left(0, \frac{1}{p}\right)$. For any $(\epsilon, \delta) \in (0,1)^2$, if*

$$p \geq 16\epsilon^{-2}\log\left(n/\sqrt{\delta}\right),$$

then with probability at least $1-\delta$ on the random matrix W,

$$\forall (x, x') \in \mathcal{S}^2 : \quad (1-\epsilon)\left\|x - x'\right\|_{\ell_2}^2 \leq \left\|Wx - Wx'\right\|_{\ell_2}^2 \leq (1+\epsilon)\left\|x - x'\right\|_{\ell_2}^2.$$

The mapping $x \in \mathbb{R}^d \mapsto Wx \in \mathbb{R}^p$ is called an ϵ-isometry on $\mathcal{S}$.

The proof will be done during the class.

Remark 10.1.6. *The underlying idea of this approach is that the reduction mapping $x \in \mathbb{R}^d \mapsto Wx \in \mathbb{R}^p$ is an exact isometry "in expectation":*

$$\forall x \in \mathbb{R}^d : \quad \mathbb{E}\left(\left\|Wx\right\|_{\ell_2}^2\right) = \|x\|_{\ell_2}^2,$$

since we have shown that, in the situation of Theorem 70, for all $x \in \mathbb{R}^d \backslash \{0\}$, $\frac{p\|Wx\|_{\ell_2}^2}{\|x\|_{\ell_2}^2} \sim \chi^2_p$.

Let us remark that it is enough for $\{W_{ij}\}_{\substack{1\leq i\leq p\\ 1\leq j\leq d}}$ to be independent with $\mathbb{E}\,W_{ij} = 0$ and $\mathbb{V}(W_{ij}) = \frac{1}{p}$ (for all $i \in [p]$ and $j \in [d]$) in order to get an exact isometry "in expectation". Indeed, denoting $Z = Wx$, we have for all $i \in [p]$, $\mathbb{E}\,Z_i = 0$ and $\mathbb{V}(Z_i) = \sum_{j=1}^d x_j^2\,\mathbb{V}(W_{ij}) = \frac{\|x\|_{\ell_2}^2}{p}$. Then, it

comes:

$$\mathbb{E}\left[\left\|Wx\right\|_{\ell_2}^2\right] = \mathbb{E}\left[\left\|Z\right\|_{\ell_2}^2\right] = \sum_{i=1}^{p} \mathbb{E}\left[Z_i^2\right] = \sum_{i=1}^{p} \left[\mathbb{V}(Z_i) + (\mathbb{E}\, Z_i)^2\right] = \|x\|_{\ell_2}^2 .$$

Remark 10.1.7. *It is remarkable that the requirement on the reduced dimension $p \geq 16\epsilon^{-2}\log(n/\sqrt{\delta})$ does not depend on the original dimension d. This means that we could consider data in infinite-dimensional Hilbert space.*

The forthcoming corollary goes a step further Theorem 70, telling us that we can find very quickly a linear dimensionality reduction mapping, that is an exact ϵ-isometry on a given dataset. This is done by sampling a matrix and checking that it provides the expected result.

Corollary 71. *Let $\mathcal{S} \subseteq \mathbb{R}^d$ be a finite set of vectors with cardinality $n \geq 2$. For any $\epsilon \in (0,1)$, let p be an integer such that*

$$p \geq 16\epsilon^{-2}\log(n),$$

then there exists a matrix $W \in \mathbb{R}^{p\times d}$ such that

$$\forall (x,x') \in \mathcal{S}^2 : \quad (1-\epsilon)\left\|x-x'\right\|_{\ell_2}^2 \leq \left\|Wx - Wx'\right\|_{\ell_2}^2 \leq (1+\epsilon)\left\|x-x'\right\|_{\ell_2}^2 .$$

In addition, such a matrix can be found by a randomized algorithm, for which the expected time is linear in n.

The proof will be done during the class.

Figure 10.1 illustrates Corollary 71 regarding the minimal dimension required to get an ϵ-isometry. We observe that this is quite huge. Thus, random projection is a workable method only for very high-dimensional data. Otherwise, PCA should be preferred.

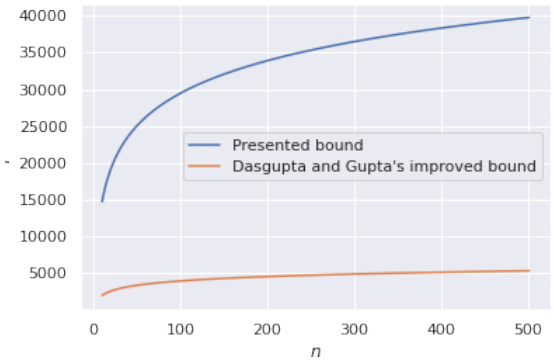


Figure 10.1: Curves $p = 16\epsilon^{-2}\log(n)$ (presented bound) and $p = \frac{4}{\epsilon^2/2-\epsilon^2/3}\log(n)$ (Dasgupta and Gupta's improved bound) for $\epsilon = 0.05$.

Besides, Figure 10.2 depicts the ratio $\frac{\|Wx_i - Wx_j\|_{\ell_2}^2}{\|x_i - x_j\|_{\ell_2}^2}$ for two trials of random matrices W and a dataset of 50 points uniformly sample on the hypercube $[0,1]^{10000}$. It appears that, in practice, a single trial is sufficient to get a suitable matrix W (which is dramatically faster than linear in n) and even if

the first trial does not work, the ϵ-isometry requirement is violated only for few points. A possible explanation is that the lower bound of the probability of success of finding a suitable matrix W $(1/n)$ is very loose because of the union bound used in the proof and of the lack of hypothesis on the data distribution.

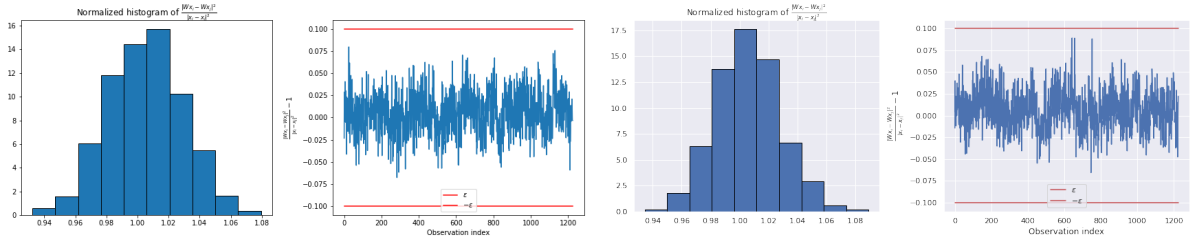


Figure 10.2: Analysis of the requirement $\left| \frac{\|Wx_i - Wx_j\|_{\ell_2}^2}{\|x_i - x_j\|_{\ell_2}^2} - 1 \right| \leq \epsilon$, which is fulfilled for the left trial and violated for the right one.

10.1.6 Reconstruction of random projections

When we moved from PCA to random projections, we changed the paradigm of dimensionality reduction from a *reasonable recovery* to *preserving pairwise distances*. It is entirely licit to wonder if one has a reasonable recovery of the original data when pairwise distances are preserved up to an error ϵ.

An answer comes from the mathematical domain of *compressed sensing*, which requires nevertheless to modify our assumptions: from now on, we do not consider being provided with a finite set of points $\mathcal{S}$ any longer, but, given an integer s, we focus on all s-sparse vectors. A vector $x \in \mathbb{R}^d$ is said s-sparse if its pseudo ℓ_0-norm is bounded by s:

$$\|x\|_{\ell_0} = \sum_{i=1}^{d} \mathbf{1}_{x_i \neq 0} \leq s.$$

Theorem 72 ([Shalev-Shwartz and Ben-David, 2014, Theorem 23.9]). *Let $W \in \mathbb{R}^{p \times d}$ be a random matrix such that its entries $\{W_{ij}\}_{\substack{1 \leq i \leq p \\ 1 \leq j \leq d}}$ are* iid *and distributed according to $\mathcal{N}\left(0, \frac{1}{p}\right)$, and $s \in [d]$ a sparsity level. For any $(\epsilon, \delta) \in (0,1)^2$, if*

$$p \geq 100 s \epsilon^{-2} \log(40d/(\delta\epsilon)),$$

then with probability at least $1 - \delta$ on the random matrix W,

$$\forall x \in \mathbb{R}^d : \|x\|_{\ell_0} \leq s, \quad (1-\epsilon)\|x\|_{\ell_2}^2 \leq \left\|Wx\right\|_{\ell_2}^2 \leq (1+\epsilon)\|x\|_{\ell_2}^2 .$$

The matrix W is said to have the (ϵ, s)-restricted isometry property (RIP).

Theorem 72 gives a condition on the reduced dimension p for the dimensionality reduction mapping $x \in \mathbb{R}^d \mapsto Wx \in \mathbb{R}^p$ to be an ϵ-isometry on the set of all s-sparse vectors "in expectation". It should be noticed that, contrarily to the Johnson-Lindenstrauss Lemma (Theorem 70), this condition on p depends on the dimension d of the original data. Figure 10.3 compares both requirements.

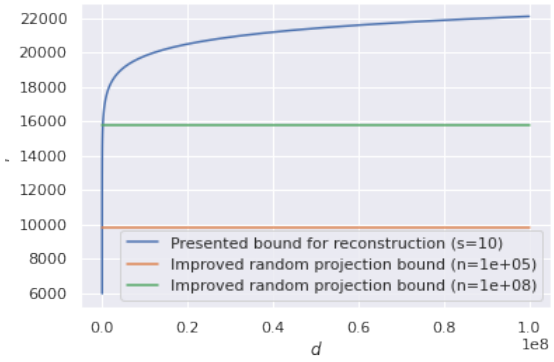


Figure 10.3: Curves $p = 100s\epsilon^{-2}\log(40d/(\delta\epsilon))$ (presented bound for reconstruction) and $p = \frac{4}{\epsilon^2/2 - \epsilon^2/3}\log(n)$ (improved random projection bound) for $\epsilon = 0.1$.

Now, Theorem 73 states that we can recover the original data $x \in \mathbb{R}^d$ from its Gaussian random projection $Wx \in \mathbb{R}^p$ by solving a convex optimization problem.

Theorem 73 ([Shalev-Shwartz and Ben-David, 2014, Theorem 23.7]). *Let $\epsilon \in (0, 2/5)$, $s \in [d]$ be a sparsity level and $W \in \mathbb{R}^{p\times d}$ be an $(\epsilon, 2s)$-RIP matrix. Then,*

$$\forall x \in \mathbb{R}^d : ||x||_{\ell_0} \leq s, \quad x \in \arg\min_{\substack{u\in\mathbb{R}^d: \\ Wu=Wx}} ||u||_{\ell_1},$$

where $||u||_{\ell_1} = \sum_{i=1}^d |u_i|$.

Remark 10.1.8. *It is quite natural to wonder which of PCA or random projection is preferable. To answer this question, one can focus on the recovery property of each method.*

On the one hand, PCA guarantees perfect recovery whenever the variable X lies in a linear subspace of $\mathbb{R}^d$, with dimension k less than the number p of selected components. Indeed, if $\mathcal{R}$ is the subspace in which lies X and $\mathcal{R}_\perp$ is its orthogonal subspace, the k directions $u \in \mathbb{R}^d$ ($||u||_{\ell_2} = 1$) that maximize the variance of $u^\top X$ are necessarily in $\mathcal{R}$ ($u^\top X = 0$ as soon as $u \in \mathcal{R}_\perp$). Thus, if we are interested in $p \geq k$ components, then the p directions that maximize the variance contain an orthornomal basis of $\mathcal{R}$, which guarantees perfect recovery of X from its projection.

On the other hand, Gaussian random projection guarantees perfect recovery whenever the original data is sparse (in a well chosen basis), which can be understood as lying in the union of linear subpsaces.

10.2 Exercises

10.2.1 Random projection

Exercise 10.1 (Concentration of a chi-squared variable). Let k be a positive integer and $Z \sim \chi^2_k$. The moment-generating function of Z is defined for all $\lambda < \frac{1}{2}$ by:

$$\mathbb{E}\left(e^{\lambda Z}\right) = (1 - 2\lambda)^{-\frac{k}{2}}.$$

1. Show that for all $x \leq \frac{1}{4}$:
$$\frac{1}{\sqrt{1-2x}} \leq e^{x+2x^2}\mathbf{1}_{x\geq 0} + e^{x+x^2}\mathbf{1}_{x<0}.$$
2. Prove that for all $\epsilon \in [0,1]$,
$$\mathbb{P}\left(\frac{Z}{k} - 1 \geq \epsilon\right) \leq e^{-k\frac{\epsilon^2}{8}}.$$
3. Prove that for all $\epsilon \geq 0$,
$$\mathbb{P}\left(-\left(\frac{Z}{k} - 1\right) \geq \epsilon\right) \leq e^{-k\frac{\epsilon^2}{4}}.$$
4. Deduce that $\forall \epsilon \in [0,1]$,
$$\mathbb{P}\left(\left|\frac{Z}{k} - 1\right| \geq \epsilon\right) \leq 2e^{-k\frac{\epsilon^2}{8}}.$$

10.3 ($\star$) Nonlinear methods

10.3.1 Kernel principal component analysis

With the kernel trick

Let $\{X_i\}_{1\leq i\leq n} \subset \mathbb{R}^d$ be *iid* copies of X and $k : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ a kernel with feature map $\phi : \mathbb{R}^d \to \mathcal{G}$, where $\mathcal{G}$ is an appropriate Hilbert space (of dimension D, potentially infinite). As a reminder, we have $\forall (x,x') \in \mathbb{R}^d \times \mathbb{R}^d : k(x,x') = \langle \phi(x), \phi(x')\rangle_{\mathcal{G}}$.

Similarly to kernel Fisher discriminant analysis (Section 1.1.4), we consider the general method of applying PCA to the random variable $Z = \phi(X) - \mathbb{E}(\phi(X)) \in \mathcal{G}$. As shown in Section 10.1.1, this boils down to diagonalizing $\mathbb{E}(ZZ^\top)$, which may be an infinitely dimensional matrix (as soon as $\mathcal{G}$ is of dimension ∞), that is a linear operator. From now on, we may use as a notation for all $x \in \mathbb{R}^d$ and $x' \in \mathbb{R}^d$, $\phi(x)^\top \phi(x') = \langle \phi(x), \phi(x')\rangle_{\mathcal{G}}$.

Even though it seems quite difficult, we rely on Section 10.1.3, in which we have shown that, in its empirical version, PCA can be performed by diagonalizing the Gram matrix $K_Z = \left(\langle Z_i, Z_j\rangle_{\mathcal{G}}\right)_{1\leq i,j\leq n}$ of the sample $\{Z_i\}_{1\leq i\leq n}$, where for each $i \in [n]$, $Z_i = \phi(X_i) - \frac{1}{n}\sum_{\ell=1}^n \phi(X_\ell)$.

Then, we can write the sample matrices
$$\mathsf{X} = [\phi(X_1)|\dots|\phi(X_n)]^\top \in \mathbb{R}^{n\times D}, \qquad \mathsf{Z} = [Z_1|\dots|Z_n]^\top \in \mathbb{R}^{n\times D},$$
the rows of which are the sample vectors. Since $\frac{1}{n}\sum_{\ell=1}^n \phi(X_\ell) = \mathsf{X}^\top \mathbb{1}/n$, it is easy to show that
$$\mathsf{Z} = \mathsf{X} - \left[\frac{1}{n}\sum_{\ell=1}^n \phi(X_\ell)|\dots|\frac{1}{n}\sum_{\ell=1}^n \phi(X_\ell)\right]^\top = \mathsf{X} - \mathbb{1}\left(\frac{1}{n}\sum_{\ell=1}^n \phi(X_\ell)\right)^\top = (I_n - M)\mathsf{X},$$
where $M = \mathbb{1}\mathbb{1}^\top/n \in \mathbb{R}^{n\times n}$. Therefore
$$K_Z = \mathsf{Z}\mathsf{Z}^\top = (I_n - M)K_X(I_n - M), \qquad n\times n$$
where $K_X = \left(\langle \phi(X_i), \phi(X_j)\rangle_{\mathcal{G}}\right)_{1\leq i,j\leq n} = \left(k(X_i, X_j)\right)_{1\leq i,j\leq n}$.

Remark 10.3.1. *More formally, one has, for all* $(i,j) \in [n]^2$*:*

$$\begin{aligned}
(K_Z)_{ij} &= \langle Z_i, Z_j \rangle_{\mathcal{G}} \\
&= \langle \phi(X_i), \phi(X_j) \rangle_{\mathcal{G}} + \left\| \frac{1}{n} \sum_{\ell=1}^{n} \phi(X_\ell) \right\|_{\mathcal{G}}^2 - \left\langle \phi(X_i), \frac{1}{n} \sum_{\ell=1}^{n} \phi(X_\ell) \right\rangle_{\mathcal{G}} - \left\langle \phi(X_j), \frac{1}{n} \sum_{\ell=1}^{n} \phi(X_\ell) \right\rangle_{\mathcal{G}} \\
&= \langle \phi(X_i), \phi(X_j) \rangle_{\mathcal{G}} + \frac{1}{n^2} \sum_{1 \le \ell, h \le n} \langle \phi(X_\ell), \phi(X_h) \rangle_{\mathcal{G}} - \frac{1}{n} \sum_{\ell=1}^{n} \langle \phi(X_i), \phi(X_\ell) \rangle_{\mathcal{G}} - \frac{1}{n} \sum_{\ell=1}^{n} \langle \phi(X_j), \phi(X_\ell) \rangle_{\mathcal{G}} \\
&= (K_X)_{ij} + (MK_XM)_{ij} - (KM)_{ij} - (MK)_{ij} \\
&= ((I_n - M)K_X(I_n - M))_{ij}.
\end{aligned}$$

Remark 10.3.2. *If the data is not centered, that is PCA is applied on the sample* $\{\phi(X_i)\}_{1 \le i \le n}$*, then the matrix to diagonalize is* K_X *instead of* K_Z*.*

Let now $(v_1, \dots, v_p) \subset \mathbb{R}^n$ be the leading unit eigenvectors of K_Z and assumes that their eigenvalues verify $0 < \lambda_1 \le \cdots \le \lambda_p$. The dimensionality reduction mapping is $\varphi : x \in \mathbb{R}^D \mapsto V_+^\top x \in \mathbb{R}^p$, where

$$V_+ = \left[\frac{1}{\left\| \mathbf{Z}^\top v_1 \right\|_{\ell_2}} \mathbf{Z}^\top v_1 \middle| \dots \middle| \frac{1}{\left\| \mathbf{Z}^\top v_p \right\|_{\ell_2}} \mathbf{Z}^\top v_p \right] \in \mathbb{R}^{D \times p}.$$

We have, for all $i \in [p]$,

$$\left\| \mathbf{Z}^\top v_i \right\|_{\mathcal{G}}^2 = v_i^\top K_Z v_i = \lambda_i \left\| v_i \right\|_{\ell_2}^2 = \lambda_i.$$

In addition, let $\mathbf{U} \in \mathbb{R}^{n \times p}$ be the matrix of reduced representations (that is $\left(V_+^\top \left[\phi(X_i) - \frac{1}{n} \sum_{\ell=1}^n \phi(X_\ell) \right] \right)_{1 \le i \le n}$ are the rows of $\mathbf{U}$) is

$$\mathbf{U} = \left(V_+^\top \mathbf{Z}^\top \right)^\top = \mathbf{Z} V_+ = \left[\frac{K_Z v_1}{\sqrt{\lambda_1}} \middle| \dots \middle| \frac{K_Z v_p}{\sqrt{\lambda_p}} \right] = \left[\sqrt{\lambda_1} v_1 \middle| \dots \middle| \sqrt{\lambda_p} v_p \right]. \tag{10.1}$$

Moreover, given a new point $x \in \mathbb{R}^d$, its reduced representation $u \in \mathbb{R}^p$ is $u = \varphi \left(\phi(x) - \frac{1}{n} \sum_{\ell=1}^n \phi(X_\ell) \right) =$

$V_+^\top \left(\phi(x) - \frac{1}{n}\sum_{\ell=1}^n \phi(X_\ell)\right)$, with, for all $j \in [p]$:

$$\begin{aligned}
u_j &= \lambda_j^{-1/2} v_j^\top \mathsf{Z} \left(\phi(x) - \frac{1}{n}\sum_{\ell=1}^n \phi(X_\ell)\right) \\
&= \lambda_j^{-1/2} \sum_{i=1}^n (v_j)_i \left(\phi(X_i) - \frac{1}{n}\sum_{\ell=1}^n \phi(X_\ell)\right)^\top \left(\phi(x) - \frac{1}{n}\sum_{\ell=1}^n \phi(X_\ell)\right) \\
&= \lambda_j^{-1/2} \sum_{i=1}^n (v_j)_i \left(k(x, X_i) + \frac{1}{n^2}\sum_{1\le \ell,\ell'\le n} k(X_\ell, X_{\ell'}) - \frac{1}{n}\sum_{\ell=1}^n (k(x, X_\ell) + k(X_i, X_\ell))\right) \\
&= \lambda_j^{-1/2} \left[\sum_{i=1}^n (v_j)_i \left(k(x, X_i) - \frac{1}{n}\sum_{\ell=1}^n k(X_\ell, X_i)\right) + \frac{\mathbb{1}^\top v_j}{n} \sum_{i=1}^n \left(\frac{1}{n}\sum_{\ell=1}^n k(X_i, X_\ell) - k(x, X_i)\right)\right] \\
&= \lambda_j^{-1/2} \left[\sum_{i=1}^n \left((v_j)_i - \frac{\mathbb{1}^\top v_j}{n}\right) k(x, X_i) + \frac{1}{n}\sum_{1\le i,\ell\le n} \left(\frac{\mathbb{1}^\top v_j}{n} - (v_j)_i\right) k(X_i, X_\ell)\right] \\
&= \sum_{i=1}^n (\alpha_j)_i k(x, X_i) - \frac{1}{n}\sum_{1\le i,\ell\le n} (\alpha_j)_i k(X_i, X_\ell),
\end{aligned}$$

where $\alpha_j = \lambda_j^{-1/2}\left(v_j - \left(\frac{1}{n}\mathbb{1}^\top v_j\right)\mathbb{1}\right) = \lambda_j^{-1/2}(I_n - M)v_j \in \mathbb{R}^n$ for all $j \in [p]$.

This derivation shows that, as expected, we only need the kernel k (and not the — possibly infinite dimensional — feature mapping ϕ) to apply PCA in the feature space $\mathcal{G}$.

Remark 10.3.3 (PCA and spectral clustering). *See Remark 10.1.5.*

RKHS point of view

For the sake of simplicity, let us assume that the dataset is centered in $\mathcal{G}$: $\frac{1}{n}\sum_{i=1}^n \phi(X_i) = 0$. The previous derivation boils down to $u_j = \sum_{i=1}^n (\alpha_j)_i k(x, X_i)$, where $\alpha_j = \lambda_j^{-1/2} v_j$ for all $j \in [p]$, *i.e.*

$$\varphi\left(\phi(x)\right) = \begin{pmatrix} h_1(x) \\ \vdots \\ h_p(x) \end{pmatrix},$$

for some $h_j \in \mathcal{H}$, where $\mathcal{H}$ is the RKHS associated to k.

Changing a bit the notation, let us consider the problem of determining a reduction mapping $\varphi : x \in \mathbb{R}^d \mapsto (h_1(x), \dots, h_p(x))$, where $h_j \in \mathcal{H}$ for all $j \in [p]$, and such that the components are orthonormal with maximal variance:

$$\begin{aligned}
&\underset{h_1,\dots,h_p \in \mathcal{H}}{\text{maximize}} \sum_{j=1}^p \mathbb{V}(h_j(X)) \\
&\text{s. t.} \quad \begin{cases} \forall j \in [p], \left\|h_j\right\|_{\mathcal{H}} = 1 \\ \forall i, j \in [p], i \neq j \implies \left\langle h_i, h_j \right\rangle_{\mathcal{H}} = 0. \end{cases}
\end{aligned}$$

Remarking that $\frac{1}{n}\sum_{i=1}^n \phi(X_i) = 0 \implies \frac{1}{n}\sum_{i=1}^n h(X_i) = 0, \forall h \in \mathcal{H}$, the empirical point of view of the latter problem is

$$\begin{aligned} \underset{h_1,\dots,h_p \in \mathcal{H}}{\text{maximize}} \quad & \sum_{\substack{1\le j\le p \\ 1\le i\le n}} h_j(X_i)^2 \\ \text{s. t.} \quad & \begin{cases} \forall j \in [p], \left\|h_j\right\|_{\mathcal{H}} = 1 \\ \forall i, j \in [p], i \neq j \implies \left\langle h_i, h_j \right\rangle_{\mathcal{H}} = 0, \end{cases} \end{aligned} \tag{P28}$$

the maximizers of which being chosen in $\operatorname{span}\{k(X_i, \cdot), i \in [n]\}$ (see Proposition 74).

Proposition 74 (Representer theorem for kPCA). *If Problem (P28) has a solution $(h_1, \dots, h_p)$ such that for all $j \in [p]$: $\sum_{i=1}^n h_j(X_i)^2 \neq 0$, then this solution verify: $h_j \in \operatorname{span}\{k(\cdot, X_i), i \in [n]\}$, for all $j \in [p]$.*

The proof will be done during the class.

Thus, considering $h_j = \sum_{i=1}^n (\alpha'_j)_i k(\cdot, X_i)$ for some $\alpha'_j \in \mathbb{R}^n$, the problem of maximal variance becomes

$$\begin{aligned} \underset{\alpha'_1,\dots,\alpha'_p \in \mathbb{R}^n}{\text{maximize}} \quad & \sum_{j=1}^p \alpha_j'^\top K_X^2 \alpha'_j \\ \text{s. t.} \quad & \begin{cases} \forall j \in [p], \alpha_j'^\top K_X \alpha'_j = 1 \\ \forall i, j \in [p], i \neq j \implies \alpha_i'^\top K_X \alpha'_j = 0. \end{cases} \end{aligned}$$

With the change of variable $v'_j = K_X^{\frac{1}{2}} \alpha'_j$ (assuming K_X invertible), this boils down to solve

$$\begin{aligned} \underset{v'_1,\dots,v'_p \in \mathbb{R}^n}{\text{maximize}} \quad & \sum_{j=1}^p v_j'^\top K_X v'_j \\ \text{s. t.} \quad & \begin{cases} \forall j \in [p], \left\|v'_j\right\|_{\ell_2} = 1 \\ \forall i, j \in [p], i \neq j \implies v_i'^\top v'_j = 0. \end{cases} \end{aligned}$$

Remembering that $K_Z = K_X$, it is clear that the p leading eigenvectors $v_1, \dots, v_p$ of K_Z are solution to the latter problem, meaning that $\alpha'_j = K_Z^{-\frac{1}{2}} v_j = \lambda_j^{-1/2} v_j = \alpha_j$ are solutions to the former problem.

To sum up, applying kernel PCA with centered data in $\mathcal{G}$ is equivalent to building a nonlinear reduction mapping $\varphi(x) = (h_1(x), \dots, h_p(x))$, where $h_1, \dots, h_p \in \mathcal{H}$ are such that the empirical variance of $h_j(X)$ is maximal and $h_1, \dots, h_p$ are orthonormal.

Remark 10.3.4. *When the data is not centered, it should be considered $\varphi_j = h_j - \frac{1}{n}\sum_{\ell=1}^p h_\ell$, with $h_j = \sum_{i=1}^n (\alpha'_j)_i k(\cdot, X_i)$ for some $\alpha'_j \in \mathbb{R}^n$, which leads for $\alpha'_1, \dots, \alpha'_p$ to be solution to*

$$\begin{aligned} \underset{\alpha'_1,\dots,\alpha'_p \in \mathbb{R}^n}{\text{maximize}} \quad & \frac{1}{n}\sum_{j=1}^p \alpha_j'^\top K_X (I_n - M) K_X \alpha'_j \\ \text{s. t.} \quad & \begin{cases} \forall j \in [p], \alpha_j'^\top K_X \alpha'_j = 1 \\ \forall i, j \in [p], i \neq j \implies \alpha_i'^\top K_X \alpha'_j = 0. \end{cases} \end{aligned}$$

It can be shown that $\alpha'_j = (I_n - M) K_Z^{-\frac{1}{2}} v_j = \alpha_j$ is solution to the latter problem, which is

consistent with the initial derivation.

10.3.2 Classical multidimensional scaling

In Section 10.1.5, we introduced the paradigm of preserving pairwise distances and showed that it was conceivable with random projections (based on Gaussian matrices). More formally, for any $\epsilon \in (0,1)$, we exhibited a matrix $W \in \mathbb{R}^{p\times d}$ such that for all pairs of points of interest $(x, x') \in \mathbb{R}^d \times \mathbb{R}^d$,

$$(1-\epsilon)\left\|x-x'\right\|_{\ell_2}^2 \leq \left\|Wx-Wx'\right\|_{\ell_2}^2 \leq (1+\epsilon)\left\|x-x'\right\|_{\ell_2}^2,$$

namely

$$\left|\left\|x-x'\right\|_{\ell_2}^2 - \left\|Wx-Wx'\right\|_{\ell_2}^2\right| \leq \epsilon\left\|x-x'\right\|_{\ell_2}^2.$$

The approach, called multidimensional scaling (MDS), goes a step further by building representations, not necessarily linear, that tend to preserve pairwise distances. Given a training sample $\{X_i\}_{1\leq i\leq n} \subseteq \mathbb{R}^d$, MDS proceeds by defining a *stress function* S and minimizing it over the reduced representations $\{z_i\}_{1\leq i\leq n}$.

Classical scaling considers that the distance of each pair of points should be preserved, regardless of how far points are, namely

$$\left|\left\|x-x'\right\|_{\ell_2}^2 - \left\|Wx-Wx'\right\|_{\ell_2}^2\right| \leq \epsilon.$$

Let $\mathbf{Z} \in \mathbb{R}^{n\times p}$ be the matrix of which the rows are the reduced representations $\{z_i\}_{1\leq i\leq n}$. A natural variational formulation of the preceding criterion is to minimize the stress function

$$S_C(\mathbf{Z}) = \sum_{1\leq i\neq j\leq n}\left(\left\|X_i-X_j\right\|_{\ell_2}^2 - \left\|z_i-z_j\right\|_{\ell_2}^2\right)^2.$$

Let $\mathbf{X} \in \mathbb{R}^{n\times d}$ be the matrix of which the rows are the training points $\{X_i\}_{1\leq i\leq n}$, $D_{\mathbf{X}} = \left(\left\|X_i-X_j\right\|_{\ell_2}^2\right)_{1\leq i,j\leq n}$ and $D_{\mathbf{Z}} = \left(\left\|z_i-z_j\right\|_{\ell_2}^2\right)_{1\leq i,j\leq n}$ be respectively the squared pairwise distances. Then, we have

$$S_C(\mathbf{Z}) = \left\|D_{\mathbf{X}} - D_{\mathbf{Z}}\right\|_F^2.$$

Since minimizing such a function with respect to $\mathbf{Z}$ may be difficult, classical scaling introduces the Gram matrices $K_{\mathbf{X}} = \mathbf{X}\mathbf{X}^\top \in \mathbb{R}^{n\times n}$ and $K_{\mathbf{Z}} = \mathbf{Z}\mathbf{Z}^\top \in \mathbb{R}^{n\times n}$. This makes the problem simple since as soon as we know $K_{\mathbf{Z}}$, $\mathbf{Z}$ can be obtained by factorization (see below).

It should be noticed that $D_{\mathbf{Z}}$ and $K_{\mathbf{Z}}$ are linked together: let $\delta_{\mathbf{X}} = \mathrm{diag}(K_{\mathbf{X}}) \in \mathbb{R}^n$ be the vector of diagonal items of $K_{\mathbf{X}}$. Then, one has

$$D_{\mathbf{X}} = \delta_{\mathbf{X}}\mathbb{1}^\top + \mathbb{1}\delta_{\mathbf{X}}^\top - 2K_{\mathbf{X}}. \tag{10.2}$$

However, obtaining $K_{\mathbf{X}}$ from $D_{\mathbf{X}}$ (what we need in practice) is not so easy. That is why we make use of the matrix of centered data:

$$\mathbf{X}' = \mathbf{X} - \mathbb{1}\bar{X}^\top,$$

where $\bar{X} = \frac{1}{n}\sum_{i=1}^n X_i$. Let now $K_{\mathbf{X}'} = \mathbf{X}'\mathbf{X}'^\top$ be the Gram matrix of the centered data.

Property 75. *One has*

$$K_{X'} = -\frac{1}{2} H D_X H,$$

where $H = I_n - \frac{1}{n}\mathbb{1}\mathbb{1}^\top \in \mathbb{R}^{n\times n}$.

The proof will be done during the class.

Remark 10.3.5. *H is the PSD matrix of the orthogonal projection onto the vector space orthogonal to* range$(\mathbb{1})$.

Property 76. *One has*

$$\left\|D_X - D_Z\right\|_F \leq 2(1+\sqrt{n})\left\|K_{X'} - K_Z\right\|_F.$$

The proof will be done during the class.

Property 76 tells us that minimizing the distance between $K_{X'}$ and K_Z makes the squared pairwise distances closer. Therefore, we now aim at minimizing the stress function

$$S'_C(Z) = \left\|K_{X'} - ZZ^\top\right\|_F^2 = \sum_{1\leq i,j\leq n}\left(\left\langle X_i - \bar{X}, X_j - \bar{X}\right\rangle_{\ell_2} - \left\langle z_i, z_j\right\rangle_{\ell_2}\right)^2.$$

Such a problem is a low rank approximation problem. As explained in the forthcoming theorem, it is solved by the truncation of the smallest singular values of $K_{X'}$.

Lemma 77. *Let $A \in \mathbb{R}^{m\times n}$ be a matrix of rank r, with $A = UDV^\top$ being its SVD. Then*

$$\left\|A\right\|_F^2 = \sum_{i=1}^{r} D_{ii}^2.$$

The proof will be done during the class.

Lemma 78 (Weyl's inequality)**.** *Let $A \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{m\times n}$ be two matrices and let us denote $q = \min(m,n)$. Let $(\sigma_1(\cdot), \ldots, \sigma_q(\cdot))$ be the set of singular values (of a given matrix) sorted in decreasing order and completed with 0 after the rank.*

Then, for all $(i,j) \in [q]^2$ such that $i + j - 1 \in [q]$,

$$\sigma_{i+j-1}(A+B) \leq \sigma_i(A) + \sigma_j(B).$$

Theorem 79 (Eckart–Young–Mirsky theorem)**.** *Let $A \in \mathbb{R}^{m\times n}$ be a matrix of rank r and let us*

denote $q = \min(m, n)$. Then, for any $p \in [q]$, a solution to

$$\underset{B \in \mathbb{R}^{m \times n} : \operatorname{rank}(B) \leq p}{\text{minimize}} \quad \|A - B\|_F$$

is $B^\star = A$ if $p \geq r$ and if $p < r$, $B^\star = UD'V^\top$, where

- *$A = UDV^\top$ is the SVD of A with singular values $D_{11} \geq \cdots \geq D_r$;*
- *$D' \in \mathbb{R}^{r \times r}$ is such that $D'_{ii} = D_{ii}$ for all $i \in [p]$ and 0 otherwise.*

The proof will be done during the class.

From Theorem 79, we obtain that S'_C can be minimized by computing a low rank approximation $\tilde{K}$ of $K_{X'}$ and by factorizing $\tilde{K}$ in $\mathbf{Z}\mathbf{Z}^\top$. This is described in Algorithm 16. We can remark that the result obtained for kernel PCA (or centered linear PCA solved with the Gram matrix) is similar to classical MDS (see Equation (10.1) and Remark 10.1.4). In fact, kernel PCA (with centered data) can be seen as classical MDS applied in the feature space $\mathcal{G}$. There is however a big difference: kernel PCA is a predictive (or inductive) model, while MDS is not (we have to know all points beforehand in order to transform them — it is a transductive method).

Algorithm 16 Classical multidimensional scaling.

Input: $D \in \mathbb{R}^{n \times n}$ (matrix of squared pairwise distances), $p \in [n]$ (reduced dimension).

$K_{X'} \leftarrow -\frac{1}{2} H D_X H$

Compute the eigendecomposition $\sum_{i=1}^{n} \lambda_i v_i v_i^\top$ of $K_{X'}$, with $\lambda_1 \geq \cdots \geq \lambda_n$

$\mathbf{Z} \leftarrow \left[\sqrt{\lambda_1} v_1 | \ldots | \sqrt{\lambda_p} v_p\right] \in \mathbb{R}^{n \times p}$

$\{z_i\}_{1 \leq i \leq n} \leftarrow$ rows of $\mathbf{Z}$

Output: $\{z_i\}_{1 \leq i \leq n}$.

Remark 10.3.6. *All this derivation is true for Euclidean distance matrices. However, classical MDS may be performed with simple dissimilarity matrices. One should only take care of negative eigenvalues.*

10.3.3 (⋆) Metric and nonmetric multidimensional scaling

Classical scaling is part of the family of metric scaling, because it tends to preserve pairwise distances. Two other approaches are included in this family.

Kruskal–Shepard

Kruskal–Shepard scaling is a variant of classical scaling, where squares have been dropped. The stress function to minimize is

$$S_{KS}(\mathbf{Z}) = \sum_{1 \leq i \neq j \leq n} \left(\|X_i - X_j\|_{\ell_2} - \|z_i - z_j\|_{\ell_2} \right)^2.$$

In practice, Kruskal–Shepard scaling is solved by the scaling by majorizing a complicated function (SMACOF) algorithm. It consists in majorizing S_{KS} by a convex quadratic function.

Property 80. *Let $\alpha \in \mathbb{R}^{n\times n}$ be a symmetric matrix. For any $\mathbf{Y} \in \mathbb{R}^{n\times p}$, with rows denoted $\{y_i\}_{1\leq i\leq n} \subseteq \mathbb{R}^p$,*

$$\sum_{1\leq i\neq j\leq n} \alpha_{ij}(z_i - z_j)^\top (y_i - y_j) = \operatorname{tr}(\mathbf{Z}^\top V \mathbf{Y}),$$

where $V \in \mathbb{R}^{n\times n}$ and for all $i \in [n]$, $V_{ii} = 2\sum_{\substack{1\leq j\leq n\\ j\neq i}} \alpha_{ij}$ and for all $j \in [n]$, such that $i \neq j$, $V_{ij} = -2\alpha_{ij}$.

The proof will be done during the class.

Let, for all $i \in [n]$ and $j \in [n]$, $d_{ij} = \left\|X_i - X_j\right\|_{\ell_2}$ and $\delta_{ij} = \left\|z_i - z_j\right\|_{\ell_2}$. Then,

$$\begin{aligned} S_{KS}(\mathbf{Z}) &= \sum_{1\leq i\neq j\leq n} \left(d_{ij} - \delta_{ij}\right)^2 \\ &= \sum_{1\leq i\neq j\leq n} d_{ij}^2 + \sum_{1\leq i\neq j\leq n} \delta_{ij}^2 - 2\sum_{1\leq i\neq j\leq n} d_{ij}\delta_{ij}. \end{aligned}$$

In addition,

$$\sum_{1\leq i\neq j\leq n} \delta_{ij}^2 = \sum_{1\leq i\neq j\leq n} (z_i - z_j)^\top (z_i - z_j) = \operatorname{tr}(\mathbf{Z}^\top V \mathbf{Z}),$$

where $V \in \mathbb{R}^{n\times n}$ has $2(n-1)$ on the diagonal and -2 elsewhere, and

$$\sum_{1\leq i\neq j\leq n} d_{ij}\delta_{ij} = \sum_{1\leq i\neq j\leq n} \frac{d_{ij}}{\delta_{ij}} \mathbf{1}_{\delta_{ij}\neq 0}\delta_{ij}^2 = \sum_{1\leq i\neq j\leq n} \frac{d_{ij}}{\delta_{ij}} \mathbf{1}_{\delta_{ij}\neq 0}(z_i - z_j)^\top (z_i - z_j) = \operatorname{tr}(\mathbf{Z}^\top V'(\mathbf{Z})\mathbf{Z}),$$

where $V'(\mathbf{Z}) \in \mathbb{R}^{n\times n}$ has $\left(2\sum_{\substack{1\leq j\leq n\\ j\neq i}} \frac{d_{ij}}{\delta_{ij}} \mathbf{1}_{\delta_{ij}\neq 0}\right)_{1\leq i\leq n}$ on the diagonal and $\left(-2\frac{d_{ij}}{\delta_{ij}} \mathbf{1}_{\delta_{ij}\neq 0}\right)_{1\leq i\neq j\leq n}$ elsewhere.

Thus,

$$S_{KS}(\mathbf{Z}) = \sum_{1\leq i\neq j\leq n} d_{ij}^2 + \operatorname{tr}(\mathbf{Z}^\top V \mathbf{Z}) - 2\operatorname{tr}(\mathbf{Z}^\top V'(\mathbf{Z})\mathbf{Z}).$$

But, for any $\mathbf{Y} \in \mathbb{R}^{n\times p}$,

$$\begin{aligned} \operatorname{tr}(\mathbf{Z}^\top V'(\mathbf{Y})\mathbf{Y}) &= \sum_{1\leq i\neq j\leq n} \frac{d_{ij}}{\left\|y_i - y_j\right\|_{\ell_2}} \mathbf{1}_{\|y_i - y_j\|_{\ell_2}\neq 0}(z_i - z_j)^\top (y_i - y_j) \\ &\leq \sum_{1\leq i\neq j\leq n} \frac{d_{ij}}{\left\|y_i - y_j\right\|_{\ell_2}} \mathbf{1}_{\|y_i - y_j\|_{\ell_2}\neq 0} \left\|z_i - z_j\right\|_{\ell_2} \left\|y_i - y_j\right\|_{\ell_2} \\ &= \sum_{1\leq i\neq j\leq n} d_{ij}\delta_{ij} \mathbf{1}_{\|y_i - y_j\|_{\ell_2}\neq 0} \\ &\leq \sum_{1\leq i\neq j\leq n} d_{ij}\delta_{ij}, \end{aligned}$$

by Cauchy-Schwarz and non-negativity of the weights inside the sum. As a consequence, by denoting

$$M_{KS}(\mathbf{Z}, \mathbf{Y}) = \sum_{1\leq i\neq j\leq n} d_{ij}^2 + \operatorname{tr}(\mathbf{Z}^\top V \mathbf{Z}) - 2\operatorname{tr}(\mathbf{Z}^\top V'(\mathbf{Y})\mathbf{Y}),$$

which is a convex quadratic function in $\mathbf{Z}$, we obtain, for all $\mathbf{Y} \in \mathbb{R}^{n \times p}$,

$$S_{KS}(\mathbf{Z}) \leq M_{KS}(\mathbf{Z}, \mathbf{Y}) \quad \text{and} \quad S_{KS}(\mathbf{Z}) = M_{KS}(\mathbf{Z}, \mathbf{Z}).$$

Given $\mathbf{Y} \in \mathbb{R}^{n \times p}$, the minimum of $M_{KS}(\cdot, \mathbf{Y})$ can be obtained by Fermat's rule:

$$0 = \nabla_{\mathbf{Z}} M_{KS}(\mathbf{Z}, \mathbf{Y}) = 2V\mathbf{Z} - 2V'(\mathbf{Y})\mathbf{Y},$$

by symmetry of V. Since V is not necessarily full rank, it cannot be inverted. That is why we resort to the Moore-Penrose inverse of V, denoted V^+, in order to obtain a minimizer of $M_{KS}(\cdot, \mathbf{Y})$:

$$\mathbf{Z} = V^+ V'(\mathbf{Y})\mathbf{Y}.$$

Algorithm 17 SMACOF.

Input: $d \in \mathbb{R}^{n \times n}$ (matrix of pairwise distances), $p \in [n]$ (reduced dimension).
$V \leftarrow$ matrix from $\mathbb{R}^{n \times n}$ with $2(n-1)$ on the diagonal and -2 elsewhere
$V^+ \leftarrow$ Moore-Penrose inverse of V
$\mathbf{Z} \leftarrow$ random matrix from $\mathbb{R}^{n \times p}$ *(initialization)*
while not converged **do**
$\{z_i\}_{1 \leq i \leq n} \leftarrow$ rows of $\mathbf{Z}$
$\delta_{ij} \leftarrow \left\| z_i - z_j \right\|_{\ell_2}$ for all $(i, j) \in [n]$
$V' \leftarrow$ matrix from $\mathbb{R}^{n \times n}$ with $\left(2 \sum_{\substack{1 \leq j \leq n \\ j \neq i}} \frac{d_{ij}}{\delta_{ij}} \mathbf{1}_{\delta_{ij} \neq 0}\right)_{1 \leq i \leq n}$ on the diagonal and $\left(-2 \frac{d_{ij}}{\delta_{ij}} \mathbf{1}_{\delta_{ij} \neq 0}\right)_{1 \leq i \neq j \leq n}$ elsewhere.
$\mathbf{Z} \leftarrow V^+ V' \mathbf{Z}$
end while
$\{z_i\}_{1 \leq i \leq n} \leftarrow$ rows of $\mathbf{Z}$
Output: $\{z_i\}_{1 \leq i \leq n}$.

The resulting procedure is described in Algorithm 17. It provides naturally a series of non-increasing stress values.

Sammon scaling

Sammon scaling goes back to the roots by considering the original criterion for preserving pairwise distances: for all pairs of points of interest $(x, x') \in \mathbb{R}^d \times \mathbb{R}^d$,

$$\left| \left\| x - x' \right\|_{\ell_2}^2 - \left\| Wx - Wx' \right\|_{\ell_2}^2 \right| \leq \epsilon \left\| x - x' \right\|_{\ell_2}^2.$$

A natural variational formulation is to minimize the Sammon mapping with respect to $\mathbf{Z}$:

$$S_S(\mathbf{Z}) = \sum_{1 \leq i \neq j \leq n} \frac{\left(\left\| X_i - X_j \right\|_{\ell_2} - \left\| z_i - z_j \right\|_{\ell_2} \right)^2}{\left\| X_i - X_j \right\|_{\ell_2}}.$$

Nonmetric scaling

In some applications, like wine tasting for instance, pairwise distances are not as important as ranking of them: if for some $(i, j, i', j') \in [n]^4$, $\left\|X_i - X_j\right\|_{\ell_2} \geq \left\|X_{i'} - X_{j'}\right\|_{\ell_2}$, we would like a representation $\mathbf{Z}$ that fulfills $\left\|z_i - z_j\right\|_{\ell_2} \geq \left\|z_{i'} - z_{j'}\right\|_{\ell_2}$. The major interest here is preserving the ordinal properties of the data. For this reason, nonmetric scaling aims at minimizing the stress function

$$S_{NM}(\mathbf{Z}, \varphi) = \frac{\sum_{1 \leq i \neq j \leq n} \left(\varphi\left(\left\|X_i - X_j\right\|_{\ell_2}\right) - \left\|z_i - z_j\right\|_{\ell_2}\right)^2}{\sum_{1 \leq i \neq j \leq n} \left\|z_i - z_j\right\|_{\ell_2}^2},$$

over representations $\mathbf{Z}$ and monotonically increasing functions φ.

A naive algorithm in order to approximate a minimizer of S_{NM} is to alternate minimization over $\mathbf{Z}$ (for instance thanks to a subgradient descent) for a fixed φ, and isotonic regression to approximate φ given $\mathbf{Z}$.

10.4 ($\star$) Other methods

10.4.1 Spectral embedding

As explained in Section 9.1.3, spectral clustering boils down to finding a novel representation of the training data and then performing a k-means. This new representation is in fact a dimensionality reduction technique, called *spectral embedding*.

10.4.2 Linear discriminant analysis

Dimensionality reduction can be performed naturally in a supervised manner, taking into consideration the derivation of multiclass discriminant analysis (Section 1.1.5). It has been shown that the (let us say, p) leading eigenvectors of $\Sigma^{-1}M$ (see notation in Section 1.1.5), denoted $(v_1, \ldots, v_p) \subseteq \mathbb{R}^n$, concentrate the variability between features. Thus, $x \in \mathbb{R}^d \mapsto [v_1|\ldots|v_p]^\top x \in \mathbb{R}^p$ defines a dimensionality reduction mapping. In fact, when $p = C - 1$ (C being the number of classes), this mapping projects the data onto the subspace spanned by the class centers, which is enough to discriminate points.