INF 556: Topological Data Analysis **Fall 2018**

Lecture 1: Introduction to dimensionality reduction

Lecturer: Steve Oudot *T.A.: Théo Lacombe*

Disclaimer:
Some typo and errors may remain. Please mention them at `theo.lacombe@polytechnique.edu`. Use these notes with caution, especially during the exam (we decline all responsibility linked with the use of these notes during the exam session).

Reminder: These notes are a concise summary of the lectures. They do not intend in any case to substitute to your personal notes and are just an additional support in order to clarify or insist on some points.

1.1 PCA vs MDS

In the following, we consider a dataset described by a matrix $P \in \mathbb{R}^{n\times d}$, where n is the number of points and d the dimension (i.e the number of *features* used to describe each point $p_i \in \mathbb{R}^d$). We assume that P is *centered*, that is, each column sums up to 0.[1]

Froebenius norm: We remind that the Froebenius norm of a square matrix is defined by:

$$||M||_F^2 = \sum_{i,j} M_{i,j}^2$$

	PCA	MDS
Optimize	$\underset{E\in Gr(d,k)}{\arg\min} \left\{ \frac{1}{n}\sum_{i=1}^{n} ||p_i - \underbrace{\pi_E(p_i)}_{\text{orth. proj.}} ||^2 \right\}$	$\underset{Y\in\mathbb{R}^{n\times k}}{\arg\min} \{||YY^T - PP^T||_F^2\}$
Matrix of interest	Covariance: $C := \frac{1}{n}P^TP \in \mathbb{R}^{d\times d}$	Gram: $G := PP^T \in \mathbb{R}^{n\times n}$
Diagonalization where	$C = Q^TDQ$ $Q = \begin{pmatrix} [e_1] \\ \vdots \\ [e_d] \end{pmatrix}, D = \left[\begin{array}{ccc\|c} \lambda_1 & & & \\ & \ddots & & \mathbf{0} \\ & & \lambda_r & \\ \hline & \mathbf{0} & & \mathbf{0} \end{array}\right]$ with $\lambda_1 \geqslant ... \geqslant \lambda_r > 0$	$G = R^TFR$ $F = \left[\begin{array}{ccc\|c} \delta_1 & & & \\ & \ddots & & \mathbf{0} \\ & & \delta_r & \\ \hline & \mathbf{0} & & \mathbf{0} \end{array}\right]$
Solution :	take $X = PQ^T$	take $Y = R^T\sqrt{F}$ (so that $YY^T = R^TFR = G$)

[1] This can basically be obtained by setting $P \leftarrow P - \frac{1}{n}\sum p_i$

These two methods have a similar outline, despite not optimizing the same quantity. When data are in $\mathbb{R}^d$, there is a clear link between PCA and MDS which can be highlighted by using the SVD (Singular Values Decomposition) of the matrix P. Indeed, one can always write:

$$P = U^T D' V$$

where:

$$D' = \left[\begin{array}{ccc|c} \mu_1 & & & \\ & \ddots & & \mathbf{0} \\ & & \mu_s & \\ \hline & \mathbf{0} & & \mathbf{0} \end{array}\right]$$

Then, with these notations, one can observe that PCA and MDS are written as:

	PCA	MDS
Matrix of interest	$C = \frac{1}{n}P^T P = \frac{1}{n}V^T D'^T D' V$	$G = PP^T = U^T D' D'^T U$
Uniqueness of eigenvalues:	$\Rightarrow \begin{cases} D = D'^T D'/n \\ r = s \\ \forall i, \mu_i^2 = n\lambda_i \end{cases}$	$\Rightarrow \begin{cases} F = D' D'^T \\ r = s \\ \forall i, \mu_i^2 = \delta_i \end{cases}$
Take: Link with previous solutions:	$X' = PV^T$ $X' \underbrace{(VQ^T)}_{\text{orth. transform.}} = PQ^T = X$	$Y' = U^T D'$ $\underbrace{(R^T U)}_{\text{orth. transform.}} Y' = R^T \sqrt{F} Id_{n,d} = Y I_{n,d}$

So we finally have that:

$$\begin{aligned} P = U^T D' V &\Rightarrow PV^T = U^T D' \\ &\Rightarrow X' = Y' \end{aligned}$$

That is, solutions provided by PCA and MDS written with **these coordinates** are exactly the same, and more generally, they are the same up to an orthogonal transformation.

In this framework, the core difference lies in the fact that PCA involves working with a $d \times d$ matrix while MDS considers a $n \times n$ matrix. Make the good choice regarding your dataset! Otherwise, you can also use the SVD formulation.

1.2 Metric MDS

However, a deeper difference between PCA and MDS is that PCA explicitly makes use of a description of your data in $\mathbb{R}^d$, while MDS only requires a (dis)similarity measure between your points.[2]

Consider given a dataset $p_1..p_n$ and a squared distance matrix $\Delta \in \mathbb{R}^{n \times n}$, that is:

$$\forall i, j, \Delta_{ij} = d(p_i, p_j)^2$$

[2]This is useful if you do not have access to a straightforward representation of your points in $\mathbb{R}^d$. E.g. your data are customers and you have access to a function $s(x, y)$ which represents how customers x and y are similar to each other (for instance a similarity derived from purchasing habits), but you have no canonical representation of your customers in $\mathbb{R}^d$.

Case 1: Your data are actually in a Euclidean space (i.e $\mathbb{R}^d$) and $d(p_i, p_j) = ||p_i - p_j||_2$. Then, you can set (*double centering*):

$$G := -\frac{1}{2} H \Delta H$$

where $H = I_n - \frac{1}{n} \begin{pmatrix} 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{pmatrix}$

Proposition 1. *G is then the inner product matrix:*

$$\forall i, j, G_{ij} = \langle p_i, p_j \rangle$$

So we can apply MDS on G and get an embedding X with $XX^T = G$.

Case 2: (general case) d is not a Euclidean distance. However, one can still set:

$$G := -\frac{1}{2} H \Delta H$$

Then, G is **not** an inner product matrix, but G remains symmetric thus diagonalizable (with potentially negative eigenvalues).

We can apply MDS on G and get an embedding X that minimizes $||XX^T - G||_F^2$, i.e that best preserves the Gram matrix (which can be understood as a measure of similarity between your data).

1.3 Isomap

Principle: Apply metric MDS to the matrix of distances along the (potentially curved) object.

Theorem 1 (De Silva, Langford, Tenenbaum - 2000)**.** *If the object S is defined as $S = \varphi(\Omega)$ where:*

- *Ω is a convex set in $\mathbb{R}^k$*
- *$\varphi : \Omega \to \mathbb{R}^d$ is an isometry (preserves distances)*

Then metric MDS applied to the matrix of pairwise squared distances along S gives an embedding $X \in \mathbb{R}^{n \times k}$ that preserves these distances exactly.

In practice:

- Approximate the pairwise distances along S by:
  1. computing some neighborhood graph (e.g. connect every data point to every other data point within Euclidean distance ε for some fixed $\varepsilon > 0$.
  2. Computing the distances in the obtained graph.
- Apply metric MDS to the resulting squared distance matrix.

Take home messages:

- PCA is good when you have a lot of data explicitly represented in $\mathbb{R}^d$. PCA is looking for *similarities between features* by computing the covariance matrix $C \in \mathbb{R}^{d \times d}$.
- MDS is looking for *similarities between data points* by computing the Gram matrix $G \in \mathbb{R}^{n \times n}$. It turns out to be more efficient than PCA if you have few data points in very high dimension (e.g. you can have 100 images represented in $\mathbb{R}^{d \simeq 10^6}$, each coordinate corresponding to a given pixel).
- MDS can be extended to be applied in metric spaces (with no explicit representation of your data in $\mathbb{R}^d$).
- Isomap consists in applying metric MDS to (an approximation of) the distance along an object.