

# INTRODUCTION AU MACHINE LEARNING

## RÉDUCTION DE DIMENSION

**Théo Lopès-Quintas**

BPCE Payment Services,  
Université Paris Dauphine

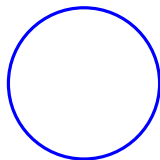
2022 - 2024

# FLÉAU DE LA DIMENSION

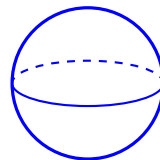
## HYPERSPHÈRE



(a) En dimension 1, volume = 2

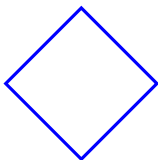


(b) En dimension 2, volume =  $\pi$

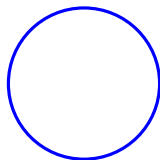


(c) En dimension 3, volume =  $\frac{4}{3}\pi$

**Figure** – Représentation et volume d'une hypersphère de rayon 1 dans 3 espaces de dimensions différentes



(a) Avec la norme 1



(b) Avec la norme 2



(c) Avec la norme infinie

**Figure** – Représentation d'une hypersphère de rayon 1 en dimension 2 pour 3 normes différentes

# FLÉAU DE LA DIMENSION

## VOLUME D'UNE HYPERSPHERE

On appelle *boule* ou hypersphère l'objet défini par :

$$B_n^p(R) = \{u \in \mathbb{R}^n, \|u\|_p^p \leq R^p\}$$

Et son volume par :

$$V_n^p(R) = \int_{B_n^p(R)} \bigotimes_{i=1}^n dx_i$$

### Proposition 1 (Volume d'une hypersphere)

Avec les notations précédentes, on a :

$$\begin{aligned} \forall R > 0, \forall n \geq 2, \forall p \geq 1, \quad V_n^p(R) &= \frac{\left(2R\Gamma\left(\frac{1}{p} + 1\right)\right)^n}{\Gamma\left(\frac{n}{p} + 1\right)} \\ &\sim \sqrt{\frac{p}{2\pi n}} \left[2R\Gamma\left(\frac{1}{p} + 1\right) \left(\frac{pe}{n}\right)^{\frac{1}{p}}\right]^n \end{aligned}$$

Avec la fonction  $\Gamma$  définie comme :

$$\Gamma(x) = \int_0^{+\infty} e^{-t} t^{x-1} dt$$

# FLÉAU DE LA DIMENSION

## CONCENTRATION DANS UNE HYPERSPHERE

On rappelle que :

$$\forall R > 0, \forall n \geq 2, \forall p \geq 1, \quad V_n^p(R) = \frac{\left(2R\Gamma\left(\frac{1}{p} + 1\right)\right)^n}{\Gamma\left(\frac{n}{p} + 1\right)}$$

### Exercice 1 (Concentration dans l'hypersphère)

Soit  $\varepsilon > 0$ . On considère une hypersphère de rayon  $R$ . Montrer que :

$$\frac{V_n^p(R - \varepsilon)}{V_n^p(R)} = \left(1 - \frac{\varepsilon}{R}\right)^n$$

# RÉDUCTION DE DIMENSION

## LEMME DE JOHNSON-LINDENSTRAUSS

Nous cherchons une fonction  $f : \mathbb{R}^d \rightarrow \mathbb{R}^k$  avec  $k \ll d$  telle que pour  $\varepsilon > 0$  et  $\forall (u, v) \in \mathcal{D}^2$ , nous ayons la propriété :

$$(1 - \varepsilon) \|u - v\|_2^2 \leq \|f(u) - f(v)\|_2^2 \leq (1 + \varepsilon) \|u - v\|_2^2 \quad (1)$$

**Distance dans l'espace de départ**

**Distorsion**

### Lemme 1 (Johnson-Lindenstrauss)

Soit  $\varepsilon > 0$ . Si  $k > \frac{24}{3\varepsilon^2 - 2\varepsilon^3} \ln(n)$ , alors il existe une fonction  $f : \mathbb{R}^d \rightarrow \mathbb{R}^k$  qui vérifie l'équation (1) pour tout  $(u, v) \in \mathcal{D}^2$ .

## RÉDUCTION DE DIMENSION

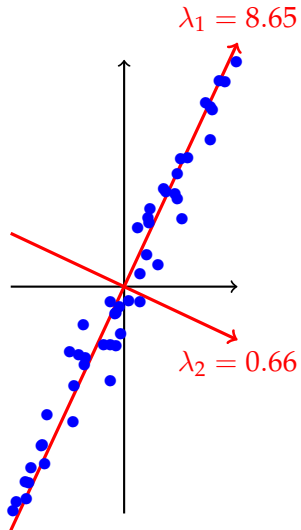
### LEMME DE JOHNSON-LINDENSTRAUSS

$n$	distorsion $\varepsilon$					
	0.1	0.2	0.3	0.4	0.5	0.8
500	5326	1434	690	423	298	166
1000	5920	1594	767	470	331	185
1500	6268	1687	812	498	351	195
2000	6515	1754	844	518	364	203
2500	6706	1805	869	533	375	209
3000	6862	1847	889	545	384	214
3500	6994	1883	906	556	391	218
4000	7109	1914	921	565	398	222
4500	7210	1941	934	573	403	225
5000	7300	1965	946	580	408	228
5500	7382	1987	956	587	413	230
6000	7456	2007	966	593	417	233
6500	7525	2026	975	598	421	235
7000	7588	2043	983	603	424	237

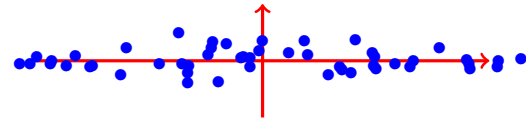
**Table** – Dimension minimale de l'espace de réduction pour plusieurs  $n$  et  $\varepsilon$

# RÉDUCTION DE DIMENSION

## ANALYSE PAR COMPOSANTE PRINCIPALE



(a) Dans l'espace de départ



(b) Dans l'espace engendré par les **vecteurs propres**

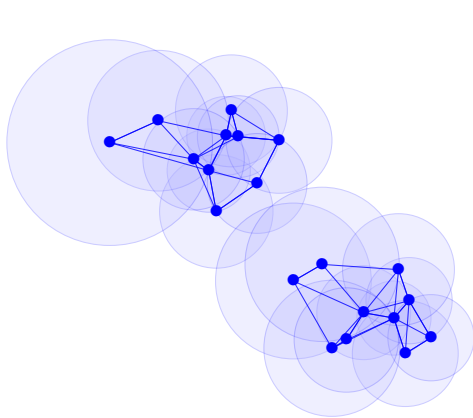
**Figure** – Projection d'une matrice  $X$  dans l'espace engendré par ses vecteurs propres

# ALGORITHME UMAP

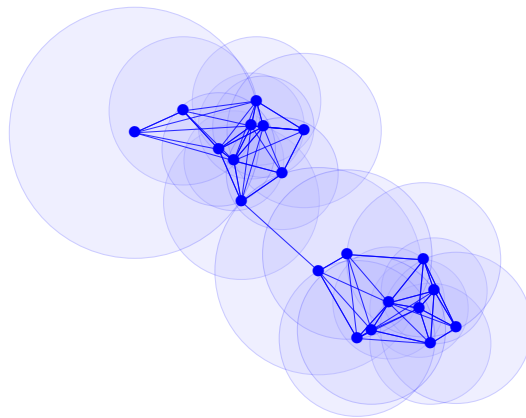
## HYPER-PARAMÈTRES

Leland McInnes, John Healy et James Melville publient en 2018 l'article *UMAP : Uniform manifold approximation and projection for dimension reduction*. Voici les principaux hyper-paramètres :

- ▶  $k$  : nombre de voisins à considérer dans l'espace de départ pour définir la structure des données
- ▶  $d$  : la dimension de l'ensemble de réductions
- ▶  $\text{min\_dist}$  : la séparation souhaitée entre deux points proches dans l'espace de réduction
- ▶  $\text{n\_epochs}$  : le nombre d'itérations d'optimisation pour la projection dans l'espace de réduction



(a) Pour  $k = 3$  voisins



(b) Pour  $k = 5$  voisins

**Figure** – Graphes appris pour deux valeurs de  $k$



# ALGORITHME UMAP

## CONSTRUCTION DU GRAPHE ORIENTÉ EN GRANDE DIMENSION

Pour chaque point  $x_i$ , on commence par trouver ses  $k$  voisins les plus proches selon la distance  $d$  que l'on aura sélectionné. On note cet ensemble  $\mathcal{N}(x_i) = \{x_i^{(1)}, \dots, x_i^{(k)}\}$ . On peut donc définir :

$$\rho_i = \min \left\{ d \left( x_i, x_i^{(j)} \right) \mid 1 \leq j \leq k, d \left( x_i, x_i^{(j)} \right) > 0 \right\}$$

Puis on définit un coefficient de normalisation  $\sigma_i$  qui est solution de l'équation :

$$\sum_{j=1}^k \exp \left( \frac{-\max \left\{ 0, d \left( x_i, x_i^{(j)} \right) - \rho_i \right\}}{\sigma_i} \right) = \frac{\ln(k)}{\ln(2)}$$

Ce coefficient permet de normaliser les distances locale pour chaque point  $x_i$ . Tout cela nous permet de définir le poids d'une arête comme :

$$w \left( \left( x_i, x_i^{(j)} \right) \right) = \exp \left( \frac{-\max \left\{ 0, d \left( x_i, x_i^{(j)} \right) - \rho_i \right\}}{\sigma_i} \right)$$

# ALGORITHME UMAP

## CONSTRUCTION DE LA MATRICE ADJACENTE SYMÉTRIQUE

Le poids d'une arête dans le graphe orienté  $\overline{G}$  appris en grande dimension est :

$$w\left(\left(x_i, x_i^{(j)}\right)\right) = \exp\left(\frac{-\max\left\{0, d\left(x_i, x_i^{(j)}\right) - \rho_i\right\}}{\sigma_i}\right)$$

### Exercice 2 (Valeur de $w$ )

*On reprend l'ensemble des notations jusqu'à présent.*

1. *Que cela signifie-t-il quand  $\max\left\{0, d\left(x_i, x_i^{(j)}\right) - \rho_i\right\} > 0$  ?*
2. *Quelle est la plus grande valeur que peut prendre  $w\left(\left(x_i, x_i^{(j)}\right)\right)$  ?*
3. *Est-ce que  $w$  est symétrique ?*

Nous pouvons définir un graphe symétrique  $G$  à partir de la matrice adjacente  $A$  du graphe  $\overline{G}$  comme :

$$B = A + A^t - A \circ A^t$$

# ALGORITHME UMAP

## RÉDUCTION DU GRAPHE

Une fois le graphe  $G$  appris, les données sont projetées à l'aide du *Spectral Embedding*. Après cette initialisation, les points vont être déplacé à l'aide d'une descente de gradient stochastique.