INTRODUCTION AU MACHINE LEARNING

UMAP : UNIFORM MANIFOLD APPROXIMATION AND PROJECTION

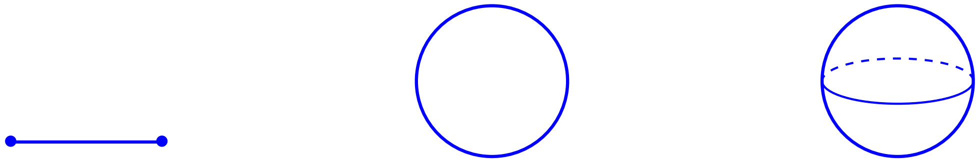
Théo Lopès-Quintas

BPCE Payment Services,
Université Paris Dauphine

2023

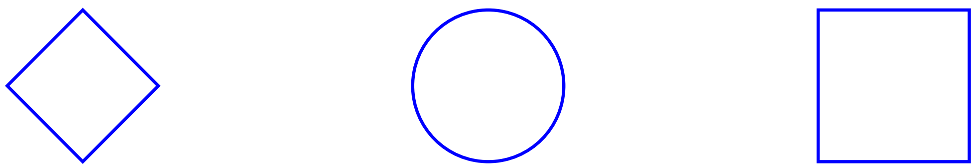
Fléau de la dimension

Hypersphère



(a) En dimension 1, volume = 2 **(b)** En dimension 2, volume = π **(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

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

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

Et son volume par :

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

Proposition 1 (Volume d'une hypersphere)

Avec les notations précédentes, on a :

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

Avec la fonction Γ *définie comme :*

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

On rappelle que :

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

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

Distance dans l'espace de départ

$$(1 - \varepsilon) \, \|u - v\|_2^2 \leqslant \|f(u) - f(v)\|_2^2 \leqslant (1 + \varepsilon) \, \|u - v\|_2^2 \tag{1}$$

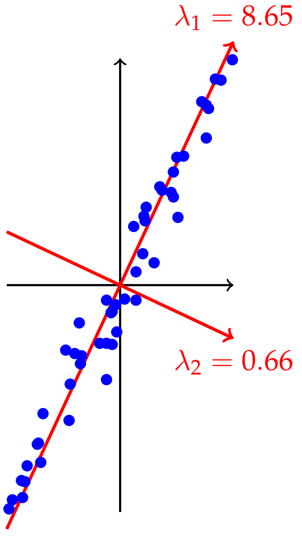
Distorsion

Lemme 1 (Johnson-Lindenstrauss)

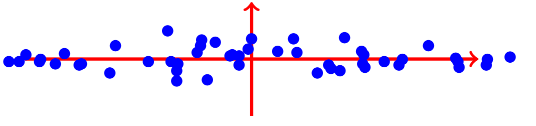
Soit $\varepsilon > 0$. Si $k > \dfrac{24}{3\varepsilon^2 - 2\varepsilon^3} \ln(n)$, alors il existe une fonction $f : \mathbb{R}^d \to \mathbb{R}^k$ qui vérifie l'équation (1) pour tout $(u, v) \in \mathcal{D}^2$.

Réduction de dimension

Analyse par composante principale



(a) Dans l'espace de départ



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

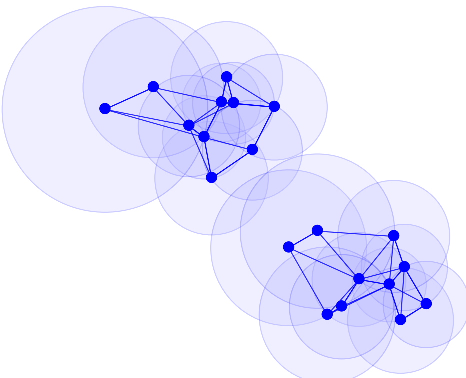
Figure – Projection d'une matrice X dans l'espace engendré par ses vecteurs propres

Algorithme UMAP

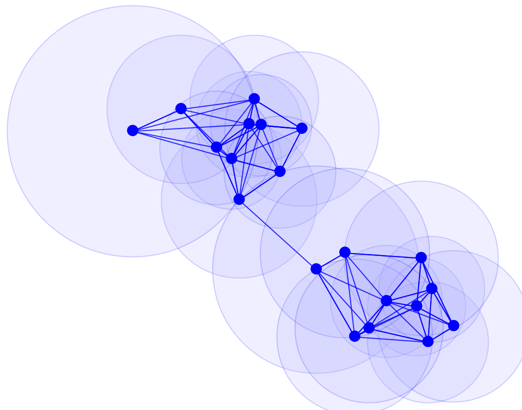
Hyper-paramètres

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
- min_dist : la séparation souhaitée entre deux points proches dans l'espace de réduction
- 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 :

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

Puis on définit un coefficient de normalisation σ_i qui est solution de l'équation :

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

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 :

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

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.