

REsearch and methodology in Data Science

Cours 3 – Réduction de dimension

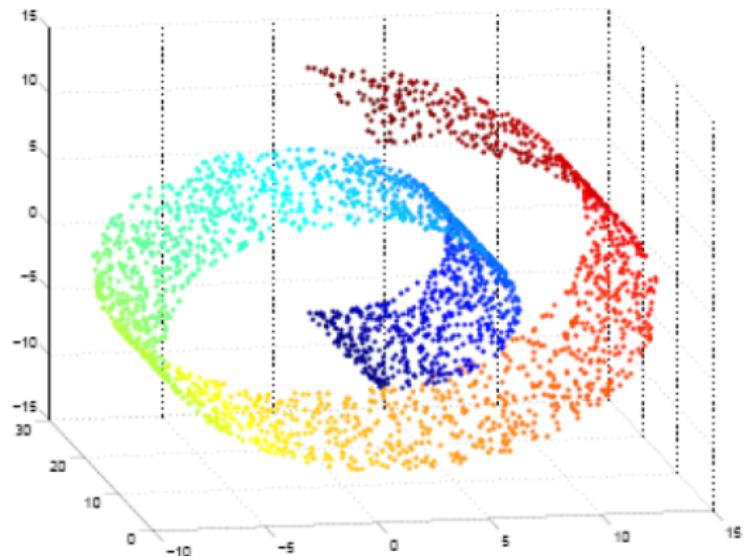
Olivier Schwander <olivier.schwander@lip6.fr>

Master DAC Data Science
UPMC - LIP6



2020-2021

Hypothèse fondamentale



Hypothèse fondamentale



Hypothèse fondamentale

La plupart des données en dimension D appartiennent en fait à un espace de dimension $d \ll D$

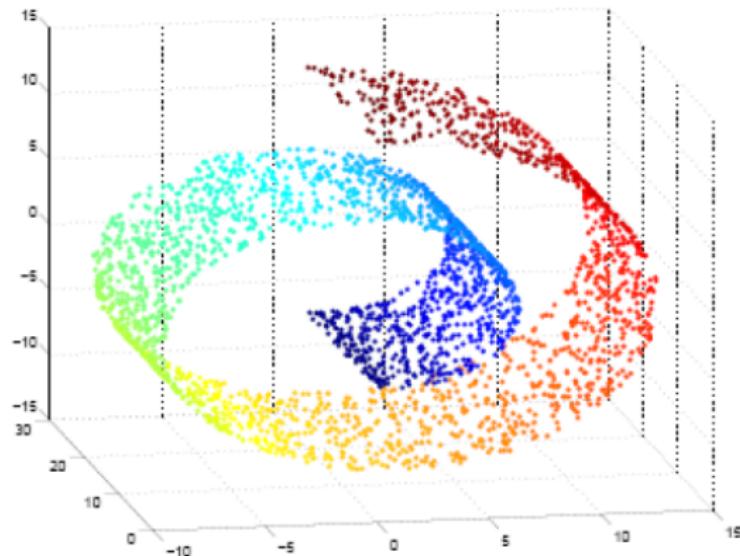
Questions

- ▶ Comment trouver cette dimension ?
- ▶ Comment trouver la "forme" de l'espace ?
- ▶ Comment inférer des propriétés intéressantes de cet espace ?

Variété

- ▶ Courbe: variété de dimension 1
- ▶ Surface: variété de dimension 2

Définition Une variété de dimension n est un espace topologique *localement euclidien*



Atlas

Ensemble de cartes locales

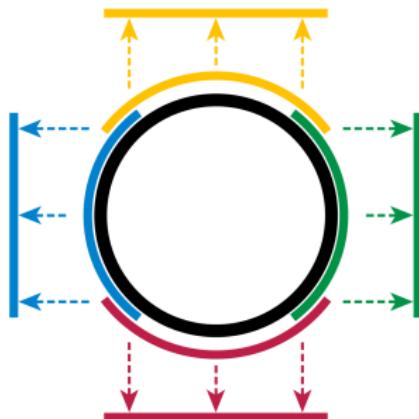


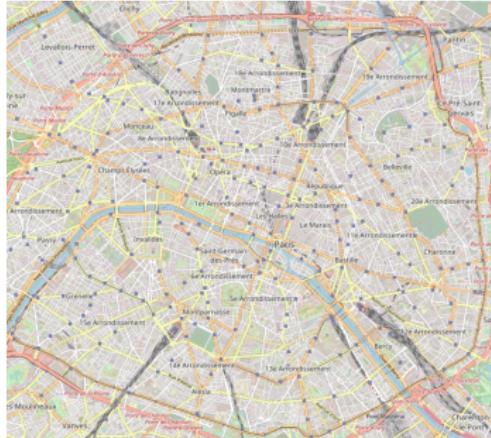
Figure Wikipédia

Système de coordonnées locales

Atlas



The Blue Marble / NASA

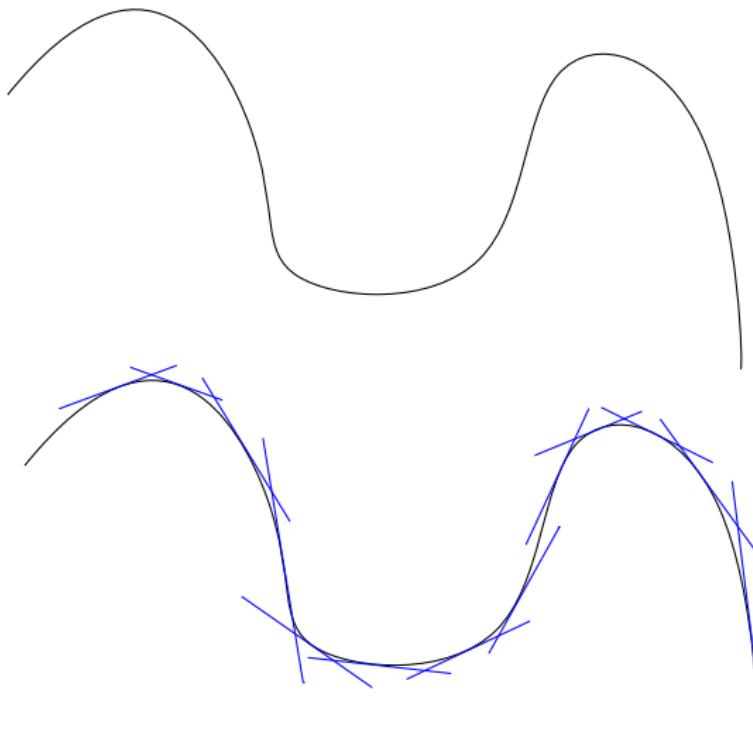


Openstreetmap

Variété riemannienne

Variété différentielle, avec une structure métrique

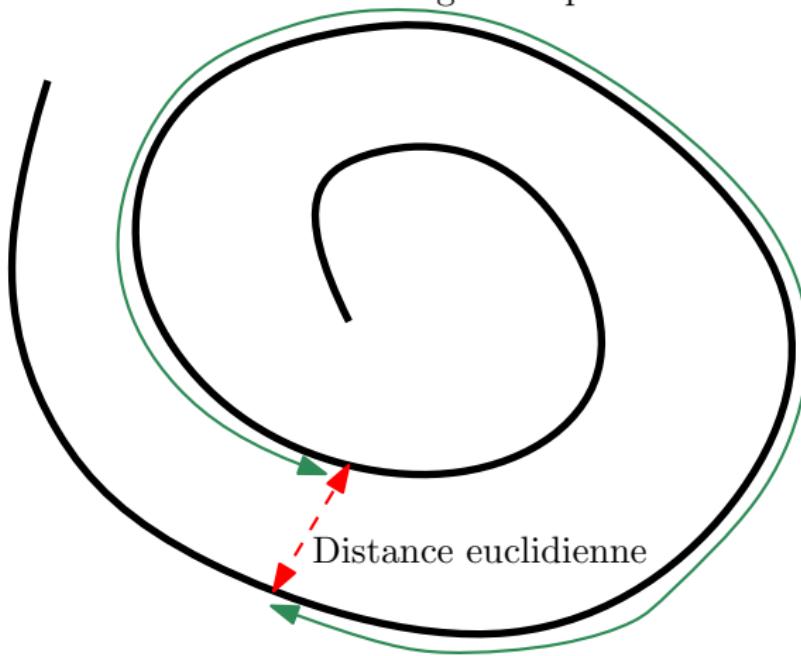
Atlas: espaces tangents, chacun avec un produit scalaire local



Distance géodésique

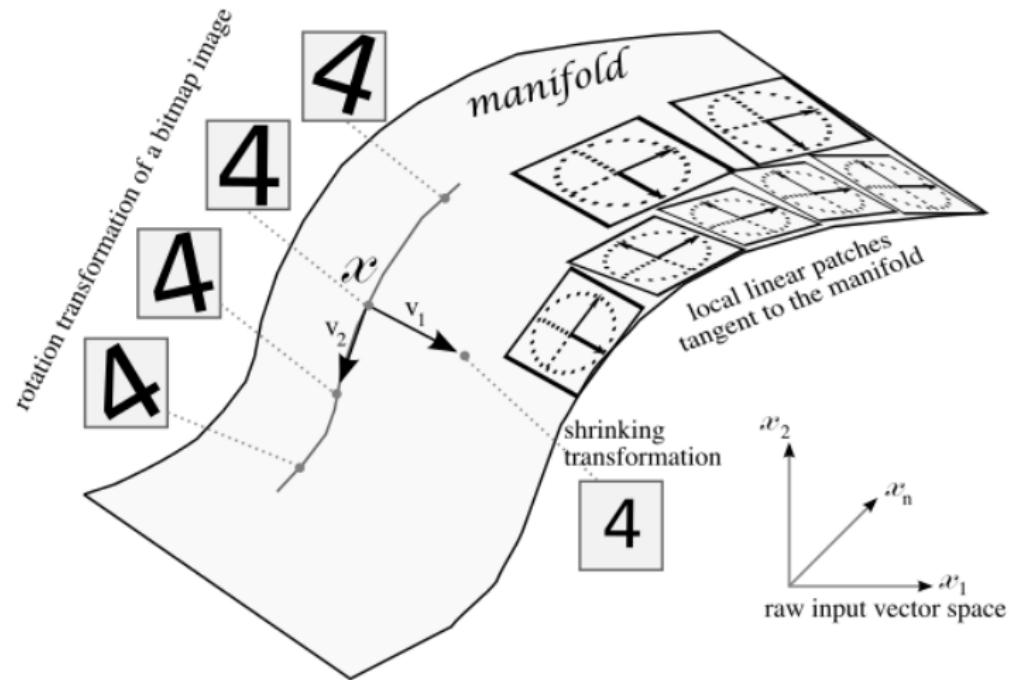
- ▶ Plus cours chemin sur la variété

Distance géodésique



Données et Variété

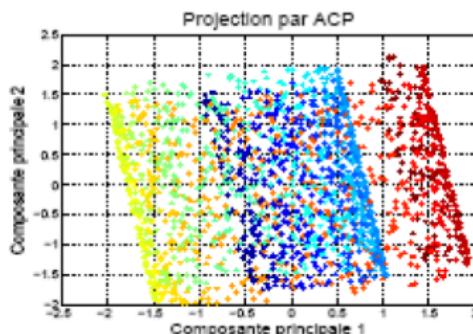
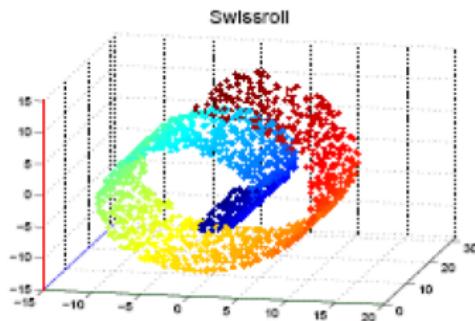
Idée: Les données sont organisées selon une variété



Visualisation de variétés

Question Comment visualiser des données organisées sur une variété ?

On sait faire, non ? Analyse en composante principale



Visualisation de variétés

Problème Projection linéaire inadaptée

Projection non linéaire

- ▶ Kernel PCA
- ▶ Multi-Dimensional Scaling (MDS)
- ▶ ISOMAP
- ▶ Locally Linear Embedding
- ▶ Random Projections
- ▶ t-SNE

Multi-Dimensional Scaling (MDS)

Entrée

- ▶ Distance entre points
- ▶ **Pas** les coordonnées des points

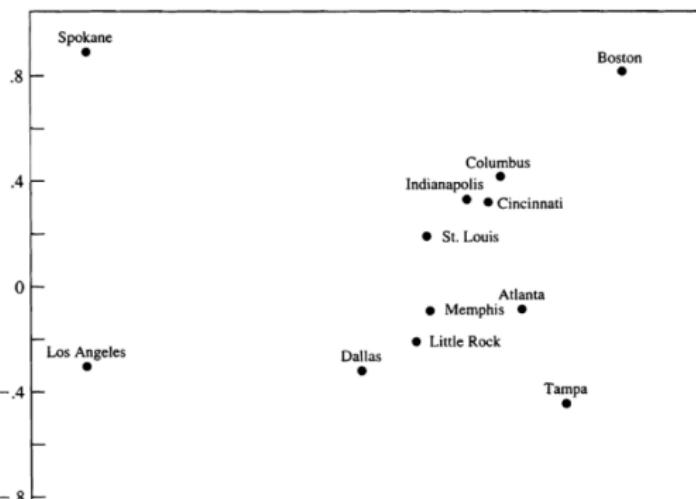
Objectif

- ▶ Reconstruire les coorodnnées

Remarque Une matrice de distance est invariante par translation/symétrie/rotation ⇒ on ne peut pas espérer retrouver la position "exacte" des individus dans l'espace d'origine.

Multi-Dimensional Scaling (MDS)

	Atlanta (1)	Boston (2)	Cincinnati (3)	Columbus (4)	Dallas (5)	Indianapolis (6)	Little Rock (7)	Los Angeles (8)	Memphis (9)	St. Louis (10)	Spokane (11)	Tampa (12)
(1)	0											
(2)	1068	0										
(3)	461	867	0									
(4)	549	769	107	0								
(5)	805	1819	943	1050	0							
(6)	508	941	108	172	882	0						
(7)	505	1494	618	725	325	562	0					
(8)	2197	3052	2186	2245	1403	2080	1701	0				
(9)	366	1355	502	586	464	436	137	1831	0			
(10)	558	1178	338	409	645	234	353	1848	294	0		
(11)	2467	2747	2067	2131	1891	1959	1988	1227	2042	1820	0	
(12)	467	1379	928	985	1077	975	912	2480	779	1016	2821	0



Multi-Dimensional Scaling (MDS)

Que faire si l'on ne dispose que d'une mesure de similarité ?

- ▶ $d_{ij} = \text{constant} - s_{ij}$
- ▶ $d_{ij} = s_{ii} + s_{jj} - 2s_{ij}$
- ▶ $d_{ij} = 1/s_{ij} - \text{constant}$
- ▶ ...

	Murder	Rape	Robbery	Assault	Burglary	Larceny	MVT
Murder	1.0000000	0.2260129	0.6991807	0.5022187	0.5129268	0.1642008	0.4784107
Rape	0.2260129	1.0000000	0.2390043	0.2937968	0.5179029	0.2972498	0.1771553
Robbery	0.6991807	0.2390043	1.0000000	0.6605061	0.5961077	0.1519019	0.7056040
Assault	0.5022187	0.2937968	0.6605061	1.0000000	0.6818327	0.5113177	0.5753717
Burglary	0.5129268	0.5179029	0.5961077	0.6818327	1.0000000	0.5891757	0.5771577
Larceny	0.1642008	0.2972498	0.1519019	0.5113177	0.5891757	1.0000000	0.2166964
MVT	0.4784107	0.1771553	0.7056040	0.5753717	0.5771577	0.2166964	1.0000000

Multi-Dimensional Scaling (MDS)

- ▶ On considère que D est une matrice de distance euclidienne ($N \times N$).
- ▶ Elle dérive d'une matrice X de points (inconnue) ($N \times q$) où q est la dimension de l'espace "initial"
- ▶ **Objectif:** approcher la distance originale dans l'espace latent
- ▶ Soit la matrice B (Matrice de gram) définie comme la matrice des produits scalaires des éléments de X :

$$B = XX^T (= XM(XM)^T) \text{ avec } M \text{ orthogonale}$$

$$b_{ij} = \sum_{k=1}^q x_{ik}x_{jk}$$

Alors, la matrice D peut s'écrire en fonction de la matrice B :

$$d_{ij}^2 = b_{ii} + b_{jj} - 2b_{ij}$$

Multi-Dimensional scaling (MDS)

Idée de MDS

- ▶ si l'on peut exprimer b en fonction de d , alors on pourra retrouver x facilement
- ▶ il n'y a pas de solution unique, il faut donc rajouter des contraintes.
- ▶ la contrainte classique est que la somme des colonnes de x vaut 0

$$\forall k \sum_{i=1}^n x_{ik} = 0$$

- ▶ donc la somme sur les lignes de b vaut 0

$$\sum_{j=1}^n b_{ij} = \sum_{j=1}^n \sum_{k=1}^q x_{ik} x_{jk} = \sum_{k=1}^q x_{ik} \left(\sum_{j=1}^n x_{jk} \right)$$

- ▶ et celle sur les colonnes aussi

Multi-Dimensional Scaling (MDS)

$$d_{ij}^2 = b_{ii} + b_{jj} - 2b_{ij}$$

$$\sum_{j=1}^n b_{ij} = \sum_{j=1}^n \sum_{k=1}^q x_{ik} x_{jk} = \sum_{k=1}^q x_{ik} \left(\sum_{j=1}^n x_{jk} \right)$$

Déduction :

- $\sum_{i=1}^n d_{ij}^2 = T + nb_{jj}$
- $\sum_{j=1}^n d_{ij}^2 = nb_{ii} + T$
- $\sum_{i=1}^n \sum_{j=1}^n d_{ij}^2 = 2nT$

avec T qui est la trace de B . On a donc:

$$b_{ij} = -\frac{1}{2} [d_{ij}^2 - d_{i\cdot}^2 - d_{\cdot j}^2 + d_{\cdot\cdot}^2]$$

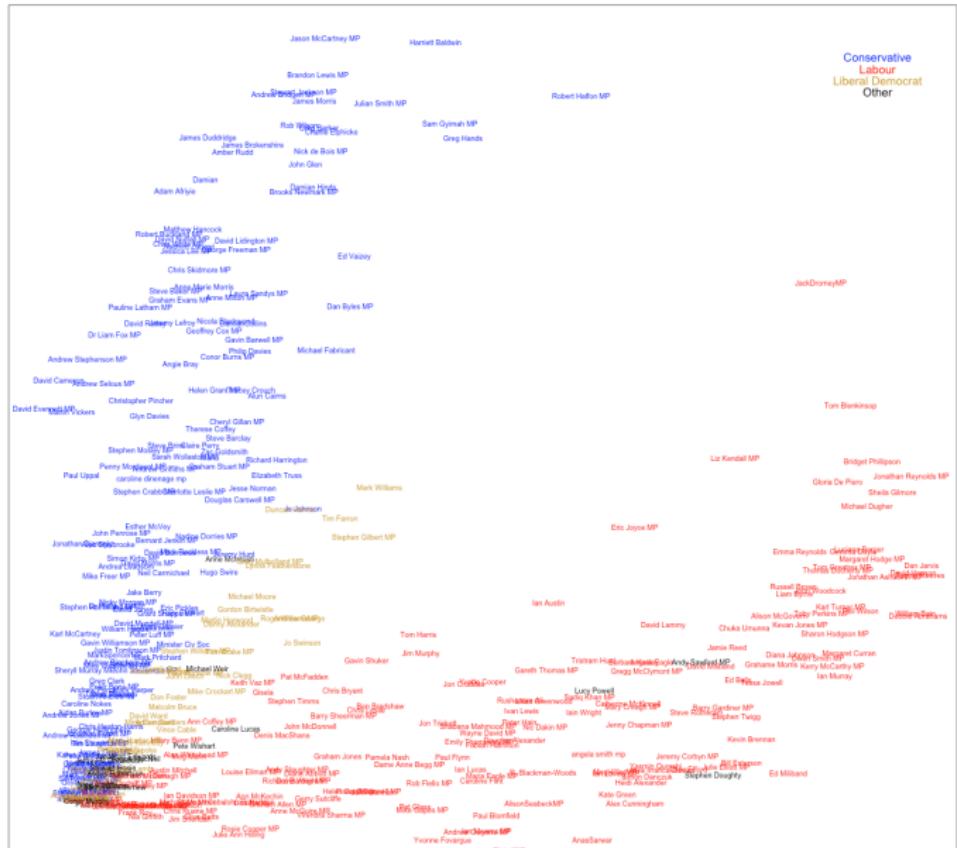
$$d_{i\cdot}^2 = (\sum_{j=1}^n d_{ij}^2)/n, \quad d_{\cdot j}^2 = (\sum_{i=1}^n d_{ij}^2)/n, \quad d_{\cdot\cdot}^2 = (\sum_{i=1}^n \sum_{j=1}^n d_{ij}^2)/n^2$$

Multi-Dimensional Scaling (MDS)

- ▶ B peut aussi s'écrire $B = U\Lambda U^T$ où Λ est la matrice diagonale des valeurs propres de B et U est la matrice des vecteurs propres (normalisés) de B
- ▶ B est de rang q vu que les données X sont dans un espace de taille q . On ne garde donc que les Q valeurs/vecteurs propres
- ▶ Projection $X = U\sqrt{\Lambda}$
- ▶ La qualité de la solution est $\frac{\sum\limits_{i=1}^q \lambda_i}{\sum\limits_{i=1}^N \lambda_i}$

Multi-Dimensional Scaling (MDS)

Two dimensional clustering of UK Members of Parliament



ISOMAP

- ▶ Données sur une variété
- ▶ Variété inconnue

MDS sur la distance géodésique ?

- ▶ Pas besoin de distance euclidienne
- ▶ Mais comment calculer la distance géodésique ?

Graphe de voisinage

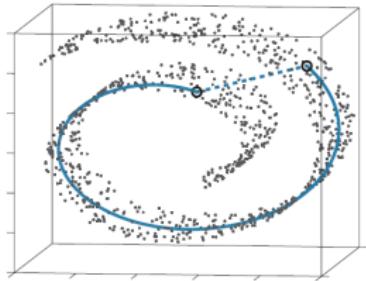
- ▶ ε -voisinage
- ▶ k -plus proches voisins

Approximation de la distance géodésique

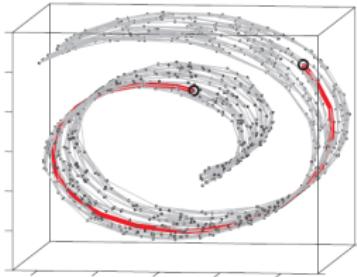
- ▶ Localement: distance euclidienne
- ▶ Globalement: plus court chemin sur le graphe de voisinage
(Dijkstra par exemple)

ISOMAP

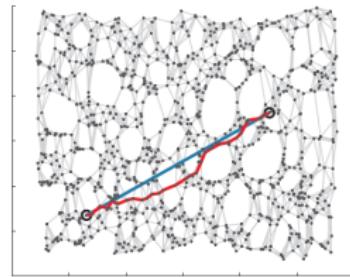
A



B



C



ISOMAP

Step

1 Construct neighborhood graph

Define the graph G over all data points by connecting points i and j if [as measured by $d_X(i,j)$] they are closer than ϵ (ϵ -Isomap), or if i is one of the K nearest neighbors of j (K -Isomap). Set edge lengths equal to $d_X(i,j)$.

2 Compute shortest paths

Initialize $d_G(i,j) = d_X(i,j)$ if i,j are linked by an edge; $d_G(i,j) = \infty$ otherwise. Then for each value of $k = 1, 2, \dots, N$ in turn, replace all entries $d_G(i,j)$ by $\min\{d_G(i,j), d_G(i,k) + d_G(k,j)\}$. The matrix of final values $D_G = \{d_G(i,j)\}$ will contain the shortest path distances between all pairs of points in G (16, 19).

3 Construct d -dimensional embedding

Let λ_p be the p -th eigenvalue (in decreasing order) of the matrix $\tau(D_G)$ (17), and v_p^i be the i -th component of the p -th eigenvector. Then set the p -th component of the d -dimensional coordinate vector \mathbf{y}_i equal to $\sqrt{\lambda_p} v_p^i$.

ISOMAP

Inconvénients

- ▶ Complexité en temps de calcul...
- ▶ N'intègre pas facilement de nouveaux points
- ▶ Il faut que l'espace soit "dense" pour que l'approximation de la distance géodésique soit de bonne qualité

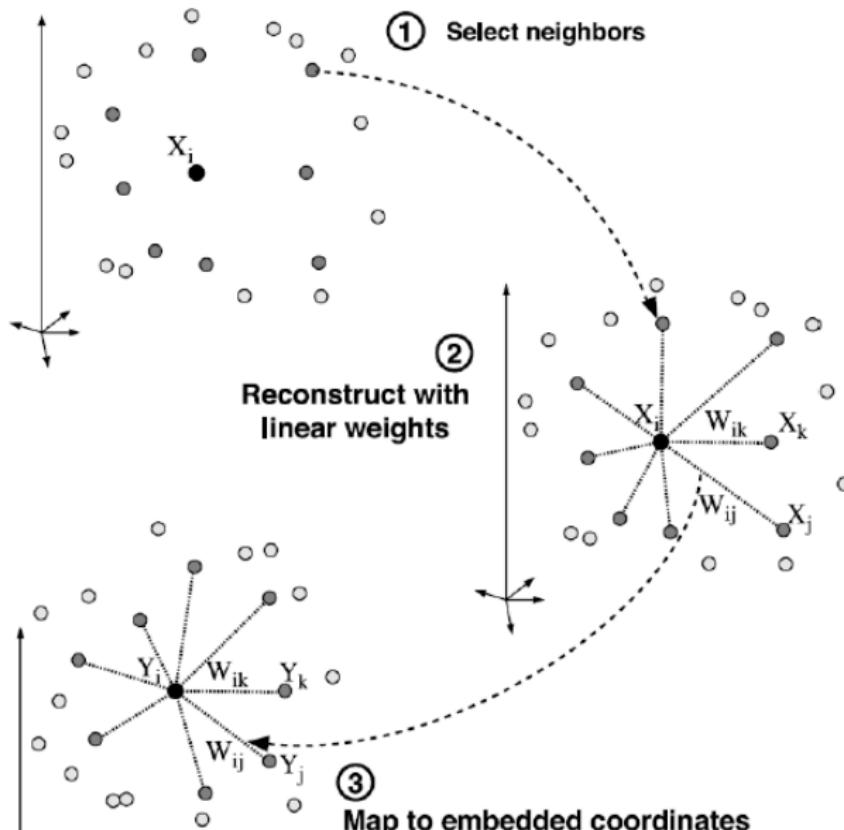
Locally Linear Embedding (LLE)

- ▶ L'espace est localement linéaire \Rightarrow un point de l'espace peut être estimé par une transformation linéaire de ses voisins les plus proches
- ▶ La même transformation linéaire peut être utilisée dans l'espace de représentation des données (pour les embeddings)

LLE ALGORITHM

1. Compute the neighbors of each data point, \vec{X}_i .
2. Compute the weights W_{ij} that best reconstruct each data point \vec{X}_i from its neighbors, minimizing the cost in eq. (1) by constrained linear fits.
3. Compute the vectors \vec{Y}_i best reconstructed by the weights W_{ij} , minimizing the quadratic form in eq. (2) by its bottom nonzero eigenvectors.

Locally Linear Embedding (LLE)



LLE: approximation dans l'espace d'origine

Soit x_1, \dots, x_N l'ensemble des points

Objectif: trouver W tq

- ▶ $x_i = \sum_j W_{ij}x_j$
- ▶ sous la contrainte $\sum_j W_{ij} = 1$
- ▶ et $W_{ij} = 0$ si j n'est pas voisin de i

Erreur résiduelle

- ▶ $E(W) = \sum_i \|x_i - \sum_{j \in N_i} W_{ij}x_j\|^2$

Optimisation

- ▶ par rapport à W
- ▶ solution analytique

LLE: plongement dans l'espace latent

Soit la matrice W apprise précédemment

Objectif: trouver y_1, \dots, y_N tq

- ▶ $y_i = \sum_j W_{ij}y_j$
- ▶ sous la contrainte $YY^T = 1$
- ▶ et $\sum y_i = 0$

Erreur résiduelle

- ▶ $E(Y) = \left\| y_i - \sum_{j \in N_i} W_{ij}y_j \right\|^2$

Optimisation

- ▶ par rapport à Y
- ▶ solution analytique

Locally Linear Embedding

input: $x_1, \dots, x_n \in \mathbb{R}^D$, d , k

1. *Compute reconstruction weights.* For each point x_i , set

$$W_i := \frac{\sum_k C_{jk}^{-1}}{\sum_{lm} C_{lm}^{-1}}.$$

2. *Compute the low-dimensional embedding.*

- Let U be the matrix whose columns are the eigenvectors of $(I - W)^\top(I - W)$ with nonzero accompanying eigenvalues.
- Return $Y := [U]_{n \times d}$.

Stochastic Neighbor Embedding (SNE): espace d'origine

Probabilité

$$p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2 / 2\sigma_i^2)}$$

σ_i petit dans les régions denses de l'espace

Interprétation

Pour un x_i , $p_{j|i}$ est la probabilité de tirer x_j comme voisin

Stochastic Neighbor Embedding (SNE): espace latent Probabilité

$$q_{ij} = \frac{\exp(-\|\mathbf{y}_i - \mathbf{y}_j\|^2)}{\sum_{k \neq m} \exp(-\|\mathbf{y}_k - \mathbf{y}_m\|^2)}$$

Critère d'apprentissage

$$C = KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

Gradient

$$\frac{\delta C}{\delta y_i} = 2 \sum_j (p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j})(y_i - y_j).$$

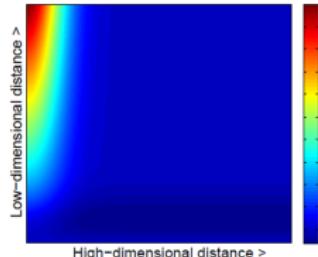
t-distributed Stochastic Neighbor Embedding (t-SNE)

Problèmes de l'approche précédente

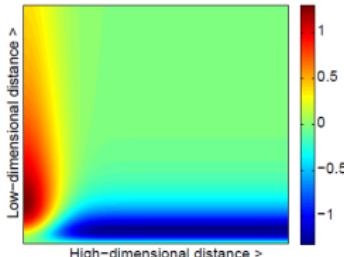
- ▶ Toutes les points éloignés exercent une force (faible) sur un point i
- ▶ Le point i est attiré vers le centre de l'espace
- ▶ Idée: Rajouter une (légère) force répulsive

Nouvel espace de sortie

$$q_{ij} = \frac{(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2)^{-1}}{\sum_{k \neq m} (1 + \|\mathbf{y}_k - \mathbf{y}_m\|^2)^{-1}}$$



(a) Gradient of SNE.

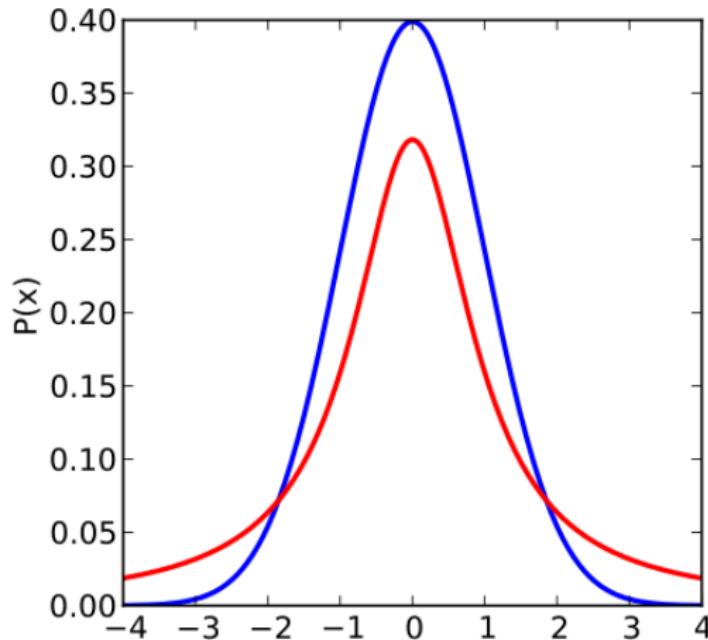


(c) Gradient of t-SNE.

Loi t de Student

Loi à queue lourde

- ▶ Queue non exponentiellement bornée
- ▶ Plus grande probabilité de tomber loin de la moyenne



t-distributed Stochastic Neighbor Embedding (t-SNE)

Algorithm 1: Simple version of t-Distributed Stochastic Neighbor Embedding.

Data: data set $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$,

cost function parameters: perplexity $Perp$,

optimization parameters: number of iterations T , learning rate η , momentum $\alpha(t)$.

Result: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, \dots, y_n\}$.

begin

 compute pairwise affinities $p_{j|i}$ with perplexity $Perp$ (using Equation 1)

 set $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$

 sample initial solution $\mathcal{Y}^{(0)} = \{y_1, y_2, \dots, y_n\}$ from $\mathcal{N}(0, 10^{-4}I)$

for $t=1$ **to** T **do**

 compute low-dimensional affinities q_{ij} (using Equation 4)

 compute gradient $\frac{\delta C}{\delta \mathcal{Y}}$ (using Equation 5)

 set $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\delta C}{\delta \mathcal{Y}} + \alpha(t) (\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)})$

end

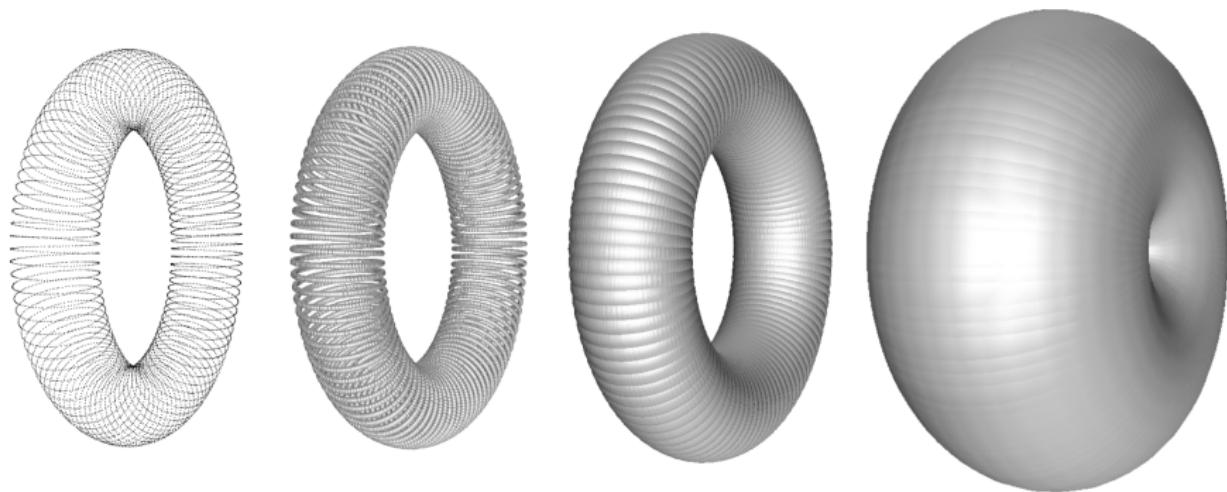
end

[http:](http://jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf)

//jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf

Inférence topologique

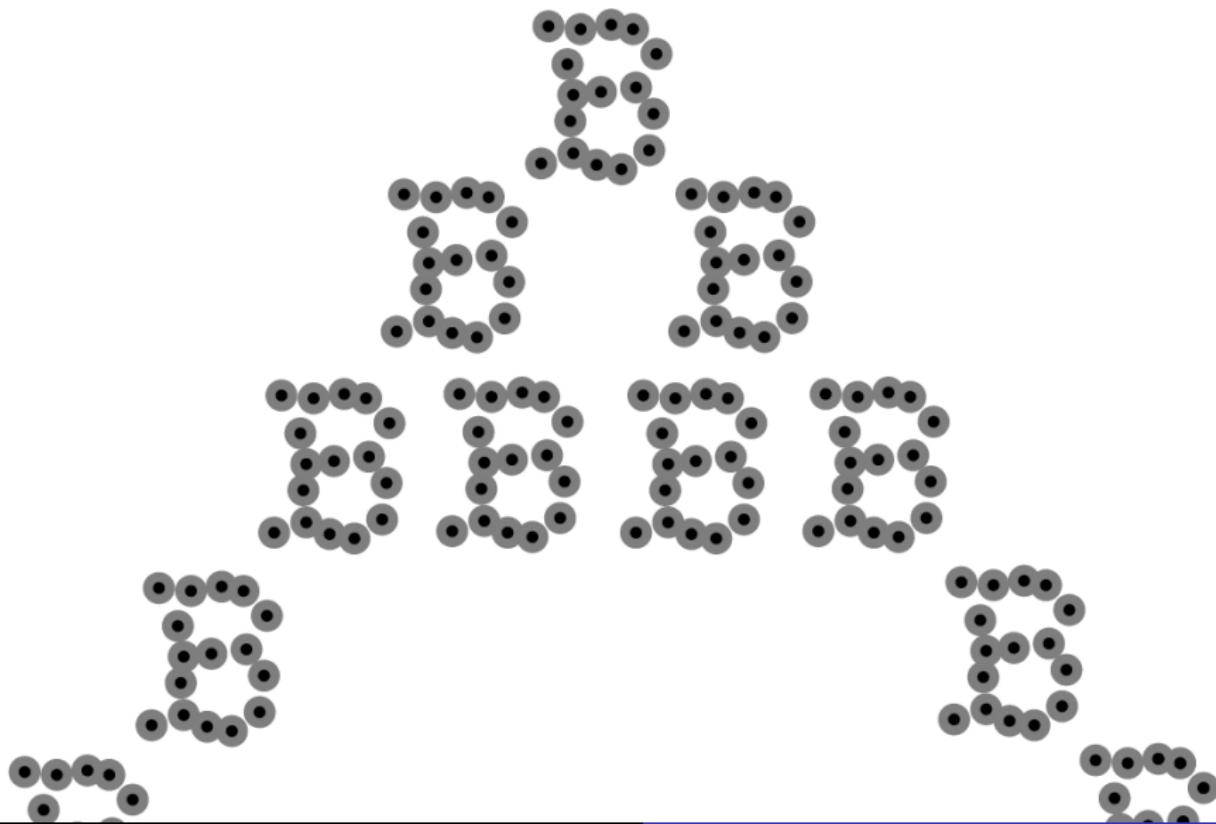
Reconstruire la topologie à partir des données



{Figures Steve Oudot}

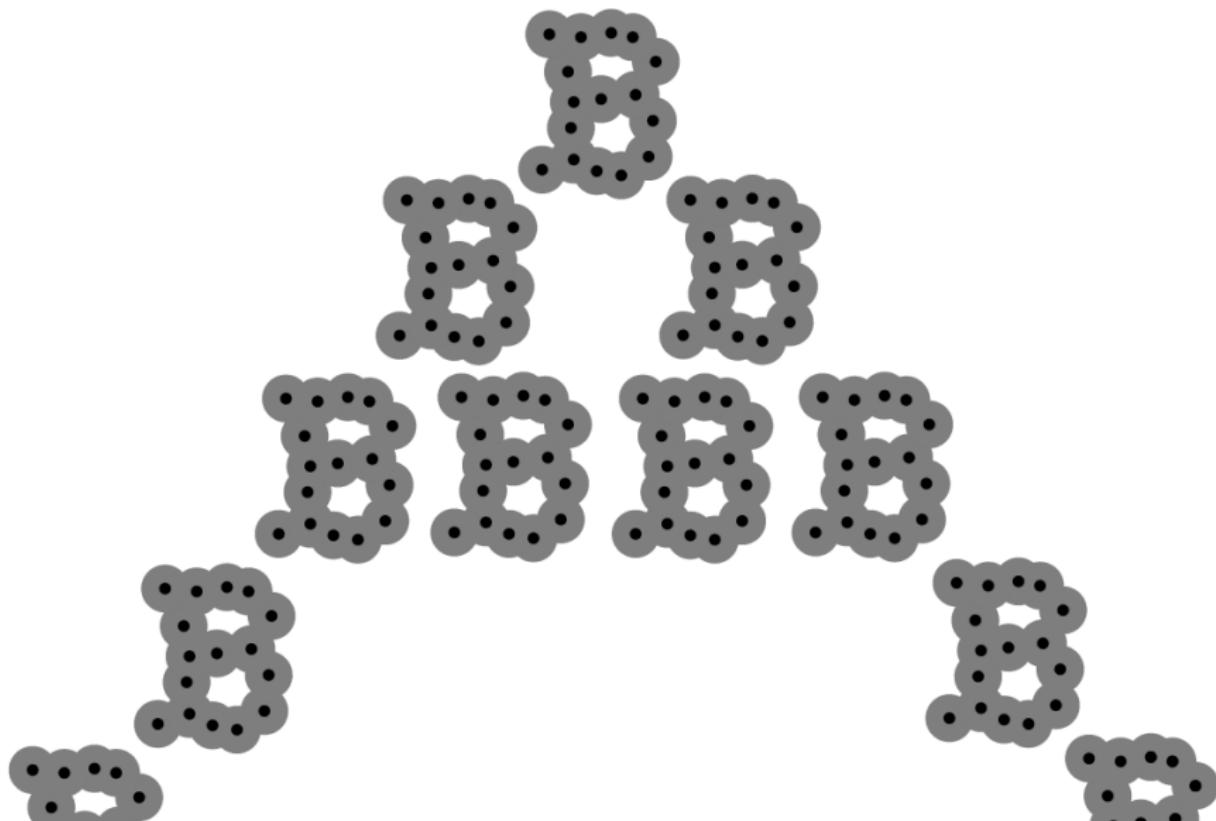
Inférence topologique

La topologie dépend de l'échelle



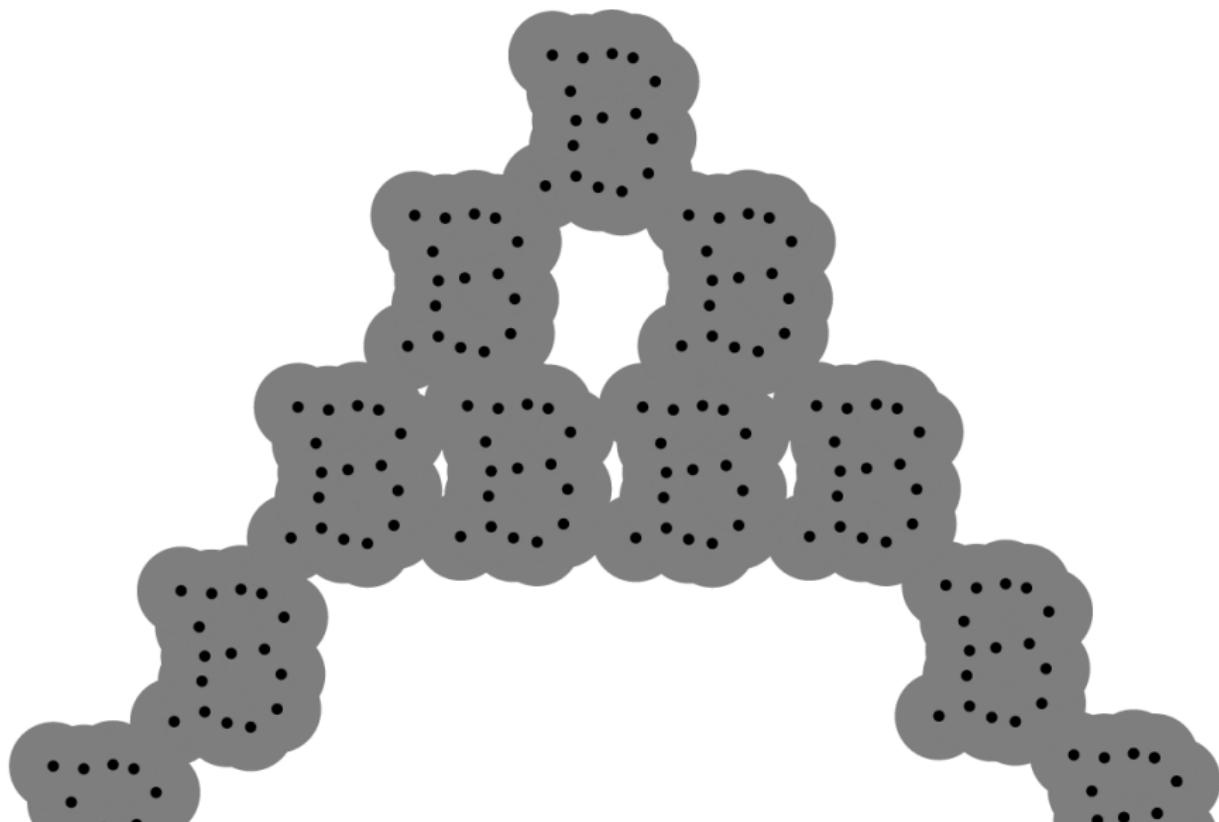
Inférence topologique

La topologie dépend de l'échelle



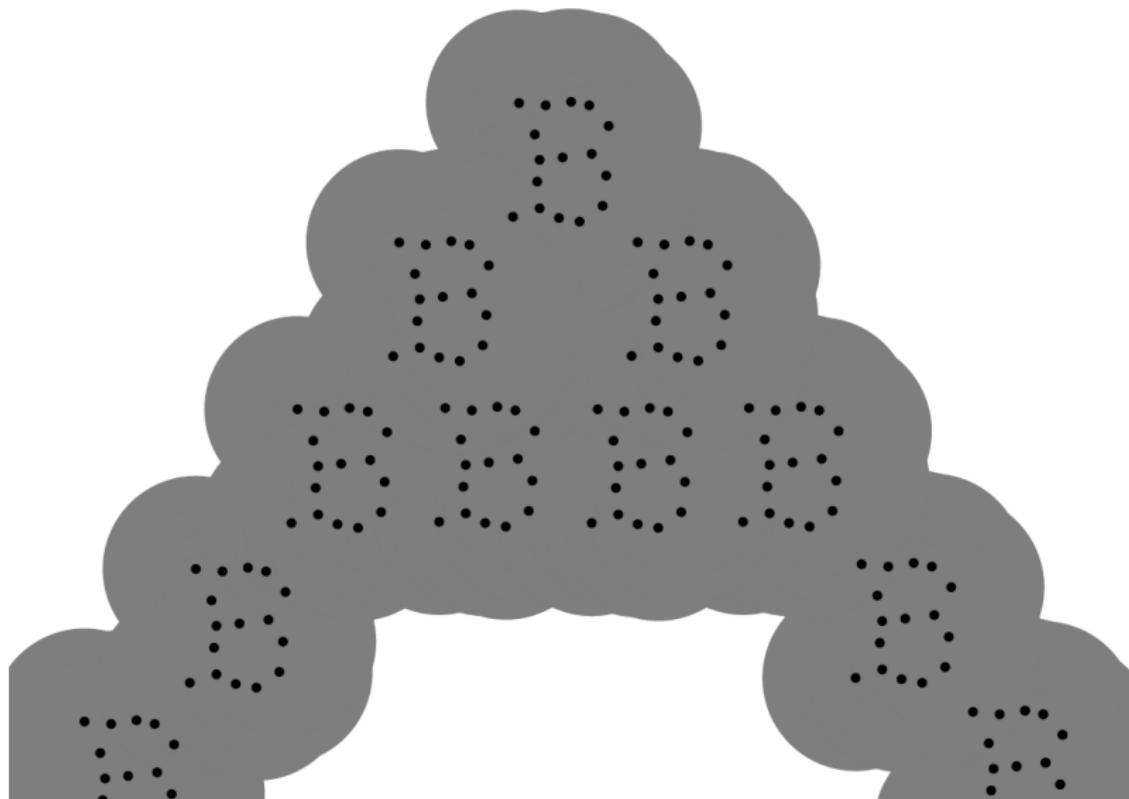
Inférence topologique

La topologie dépend de l'échelle

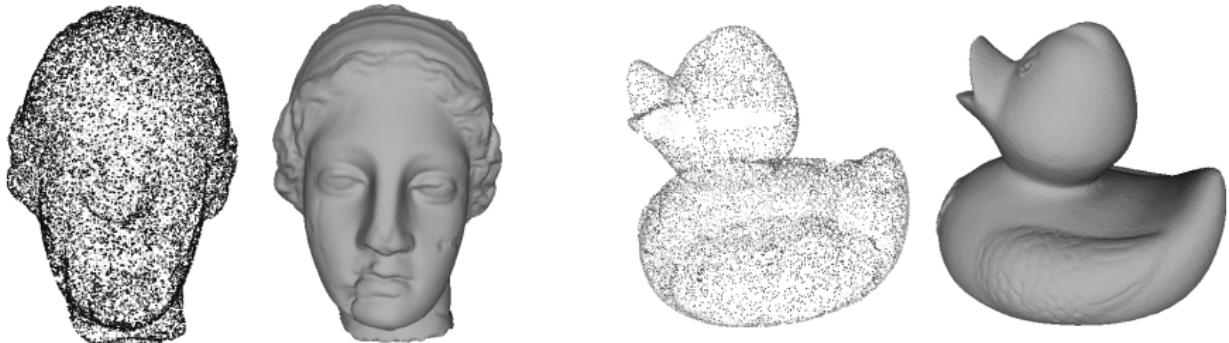


Inférence topologique

La topologie dépend de l'échelle



Reconstruction



{Figure https://elmoatazbill.users.greyc.fr/point/_cloud/index.html}

Conclusion

On a vu

- ▶ Réduction de dimension / visualisation
- ▶ Non-supervisé

Autres méthodes

- ▶ auto-encoders
- ▶ apprentissage de représentation

Pour aller plus loin

- ▶ Cours Fabrice Rossi
<http://apiacoa.org/teaching/visualization/index.fr.html>
- ▶ Cours Steve Oudot
<http://www.enseignement.polytechnique.fr/informatique/INF556>