Chapitre III: Manifold learning and visualization Embeddings

Hicham Janati

hjanati@insea.ac.ma

INSEA

Chapitre III: Manifold learning and visualization Embeddings

Qu'est-ce qu'un Embedding ?

(Définition 1). Une transformation pour réduire la dimension:

$$\mathbb{R}^d \rightarrow \mathbb{R}^k$$

(Définition 2). Une transformation pour encoder numériquement l'information:

$$\text{Texte, images, sons} \rightarrow \mathbb{R}^k$$

La PCA est un exemple d'embedding linéaire

INSEA

Propriétés et limites de l'ACP:

1. La PCA est une transformation linéaire
2. La PCA est la projection qui maximise la variance des données projetées
3. Elle ne peut pas identifier des structures complexes, non linéaires dans les données

Cherchons une méthode de projection qui préserve les similarités entre les points:

On cherche des projetés en faible dimension $\mathbf{z}_i \in \mathbb{R}^2$ tels que:

$$\langle \mathbf{x}_i, \mathbf{x}_j \rangle = \langle \mathbf{z}_i, \mathbf{z}_j \rangle$$

On cherche des projetés en faible dimension $\mathbf{z}_i \in \mathbb{R}^k$ tels que:

$$\langle \mathbf{x}_i, \mathbf{x}_j \rangle = \langle \mathbf{z}_i, \mathbf{z}_j \rangle$$

Étant données des observations $\mathbf{x}_1, \ldots, \mathbf{x}_n \in \mathbb{R}^d$:

$$\min_{\mathbf{z}_1,\ldots,\mathbf{z}_n \in \mathbb{R}^2} \sum_{i,j}^{n} (\mathbf{x}_i^\top \mathbf{x}_j - \mathbf{z}_i^\top \mathbf{z}_j)^2$$

(Classical) Multi-dimensional scaling — Classical MDS

Introduction à MDS

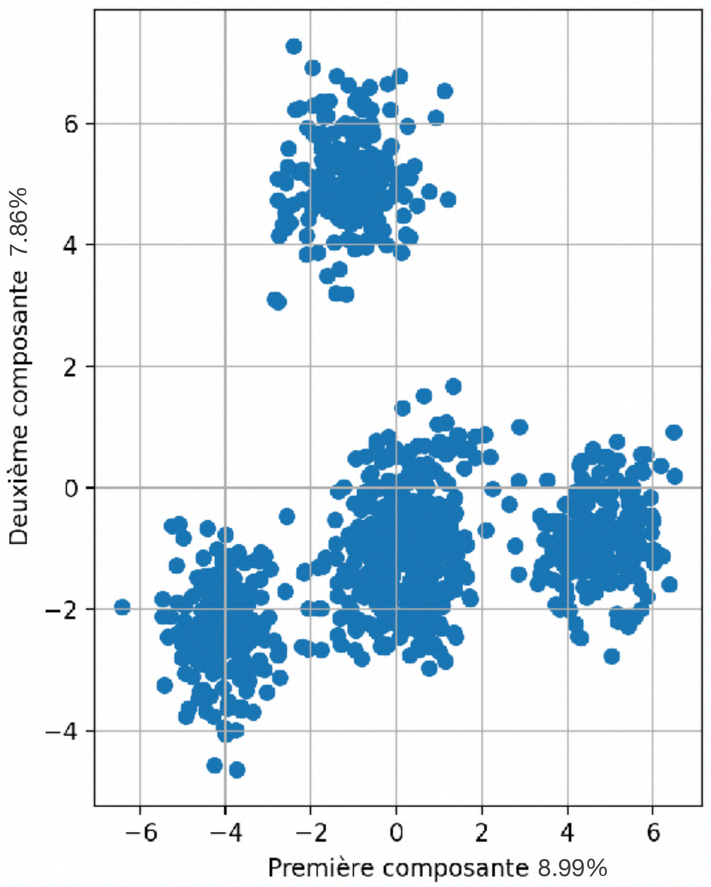
Classical MDS appliquée aux données “ratings”

Données de ratings du site e-commerce:

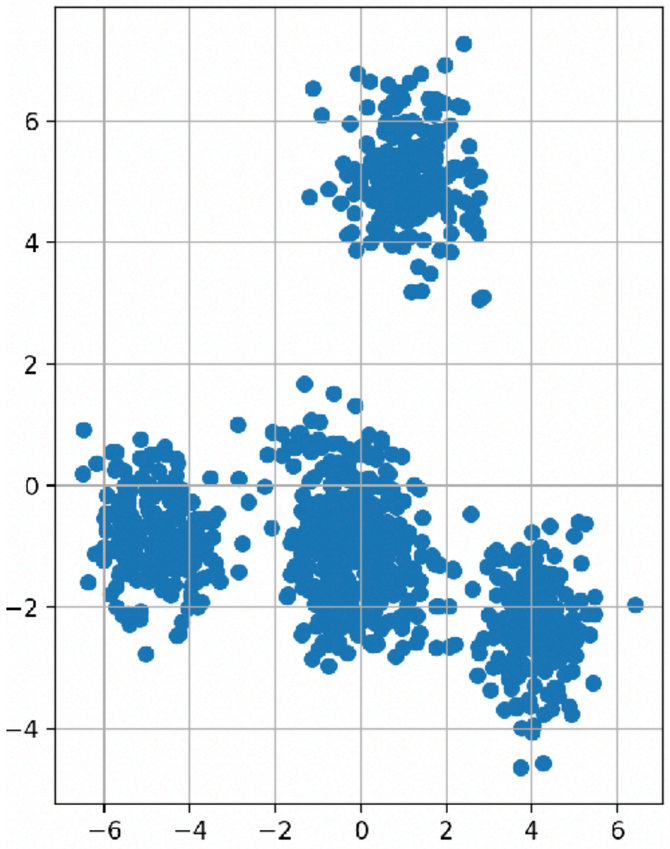
Customer	Product 1	Product 2	...
Customer 1	4.5	3.0	...
Customer 2	3.5	4.0	...
Customer 3	5.0	2.5	...
⋮	⋮	⋮	⋮
Customer n	4.0	4.5	...

Qu’en pensez-vous ?

PCA



Classical MDS



Les visualisations PCA et Classical MDS sont mathématiquement équivalentes (TD 3)

INSEA

(Classical) Multi-dimensional scaling

$$\min_{\mathbf{z}_1,\dots,\mathbf{z}_n \in \mathbb{R}^2} \sum_{i,j}^{n} (\mathbf{x}_i^\top \mathbf{x}_j - \mathbf{z}_i^\top \mathbf{z}_j)^2$$

Question

Écrire ce problème pour sous forme matricielle avec $X \in \mathbb{R}^{n \times d}$ et $Z \in \mathbb{R}^{n \times k}$

Suite (TD 3)

On peut généraliser cela en cherchant à avoir:

$$\text{distance}(\mathbf{x}_i, \mathbf{x}_j) \approx \text{distance}(\mathbf{z}_i, \mathbf{z}_j) \quad \text{au lieu de} \quad \langle \mathbf{x}_i, \mathbf{x}_j \rangle \approx \langle \mathbf{z}_i, \mathbf{z}_j \rangle$$

$$\min_{\mathbf{z}_1,\dots,\mathbf{z}_n \in \mathbb{R}^2} \sum_{i,j}^{n} (\text{dist}(\mathbf{x}_i, \mathbf{x}_j) - \|\mathbf{z}_i - \mathbf{z}_j\|)^2$$

Cette méthode s'appelle: (Metric) *Multi-dimensional scaling* (MDS)

Remarques

1. L'optimisation est directement sur les projetés $\mathbf{z}_i$: pas de fonction de projection générale comme avec l'ACP.
2. On peut formuler le problème pour réduire la dimension à un espace de dimension k avec $2 < k < d$
3. On n'a pas besoin de connaître les $\mathbf{x}_i$, il suffit d'avoir la matrice de taille $n \times n$ des distances $\text{dist}(\mathbf{x}_i, \mathbf{x}_j)$.

Pouvez-vous penser à une application de cette méthode où on a uniquement la matrice des distances ?

MDS sur un graphe (1)

Visualiser des données d'un graphe où on a uniquement des interactions entre des entités.

Exemple 1: données du nombres de "liens" entres des sites webs.

	Website 1	Website 2	Website 3	...	Website $n-1$	Website n
Website 1	0	3	5	...	2	1
Website 2	2	0	4	...	1	0
Website 3	1	2	0	...	3	2
⋮	⋮	⋮	⋮	⋱	⋮	⋮
Website $n-1$	0	1	2	...	0	4
Website n	3	0	1	...	5	0

Peut-on utiliser cette matrice comme input de l'algorithme MDS ?

Non ! Cette matrice mesure une forme de **similarité**: le contraire de la distance !

En plus elle n'est pas symétrique ! Il faut d'abord la symétriser: en prenant $(A + A^T) / 2$

Il faut ensuite la transformer en **dissimilarité** en prenant l'inverse, ou exp(-.) et appliquer MDS par la suite.

INSEA

MDS sur un graphe (2)

Exemple 2: on a les données de Likes entre utilisateurs sur facebook

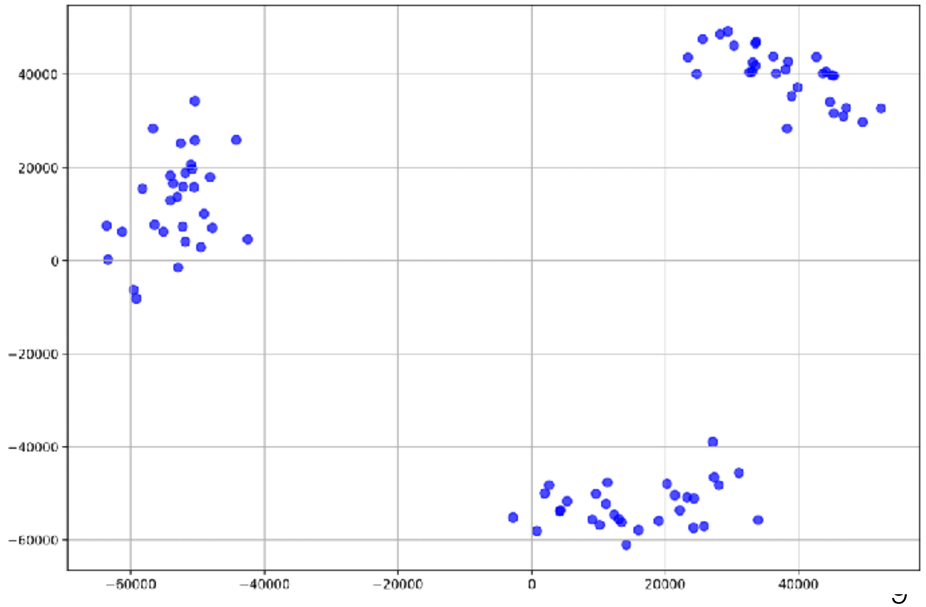
	User 1	User 2	...	User n
User 1	1	3	...	23
User 2	53	0	...	0
⋮	⋮	⋮	⋱	⋮
User n	0	187	...	0

Symétrie
+
inverse
+
diagonale = 0

→

	User 1	User 2	...	User n
User 1	0	1/56	...	1/23
User 2	1/56	0	...	1/187
⋮	⋮	⋮	⋱	⋮
User n	1/23	1/187	...	0

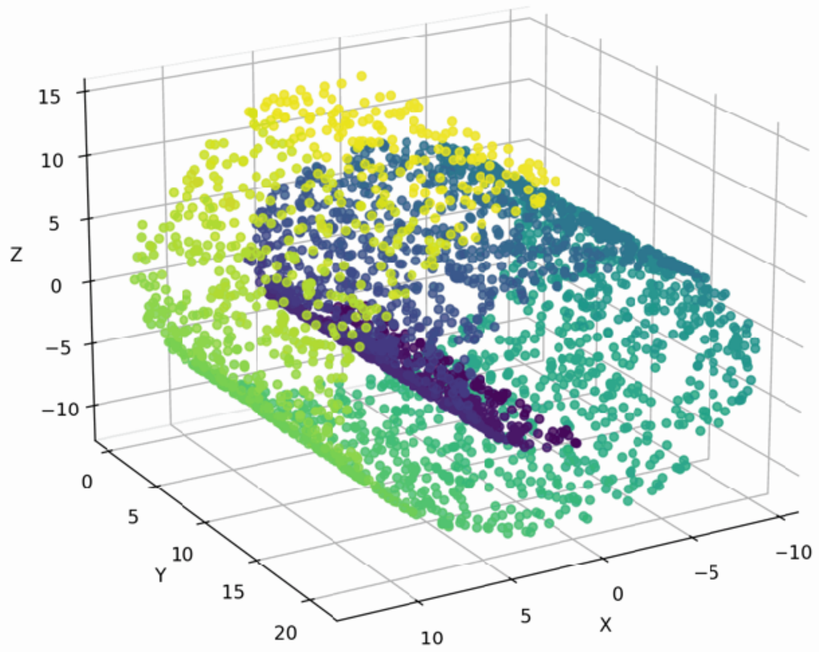
↙ MDS



INSEA

MDS sur les données Swiss roll

Données “Swiss roll”

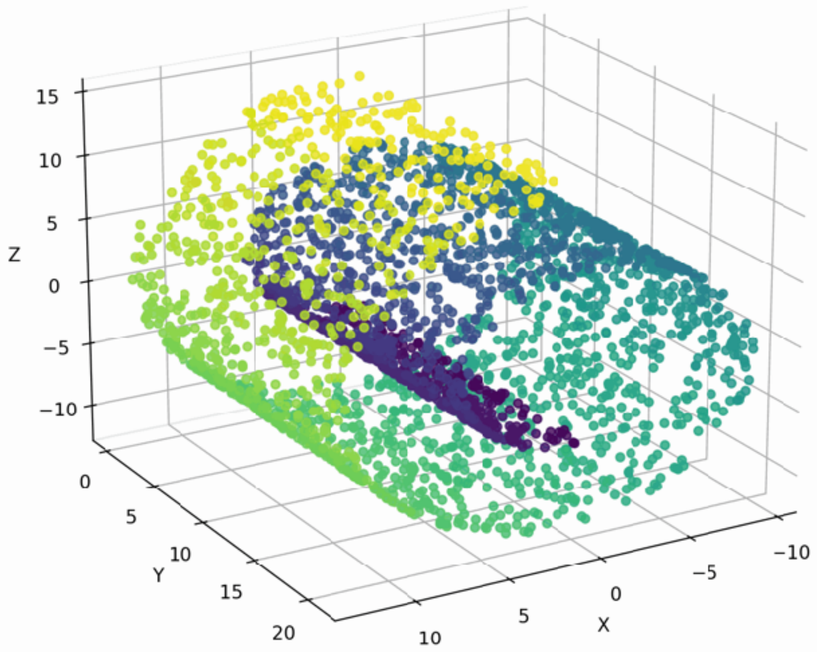


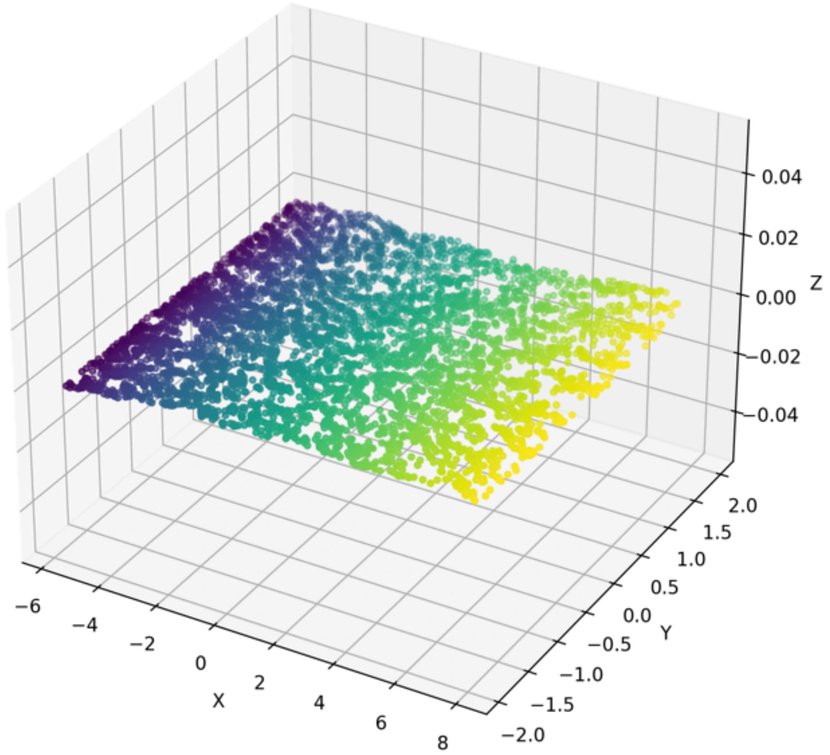
Quelle transformation intuitive simplifierait ces données ?

Étaler la génoise !

MDS sur les données Swiss roll

Données “Swiss roll”





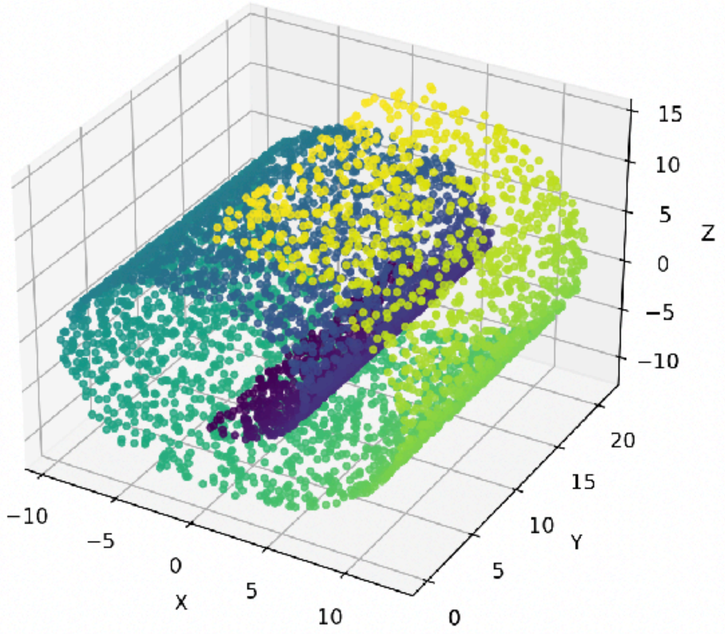
Quelle transformation intuitive simplifierait ces données ?

Étaler la génoise !

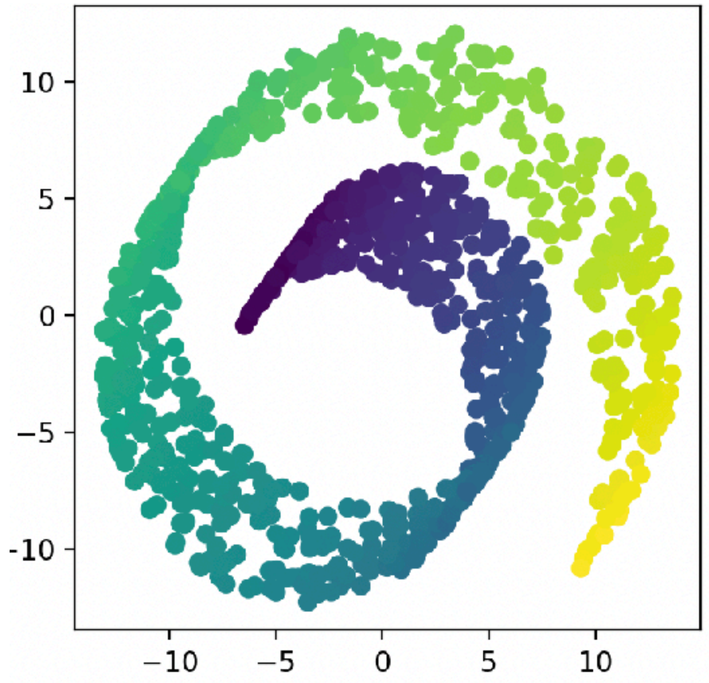
INSEA

MDS sur les données Swiss roll

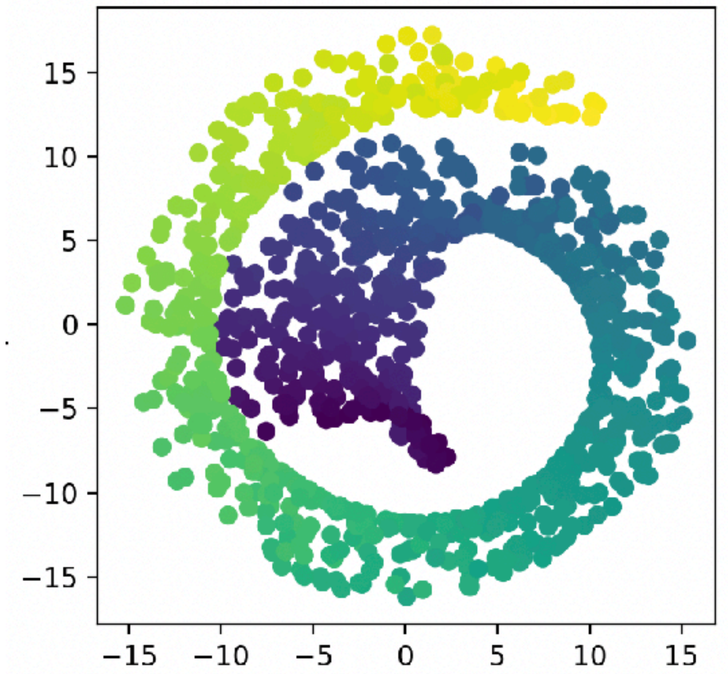
Données “Swiss roll”



PCA



MDS

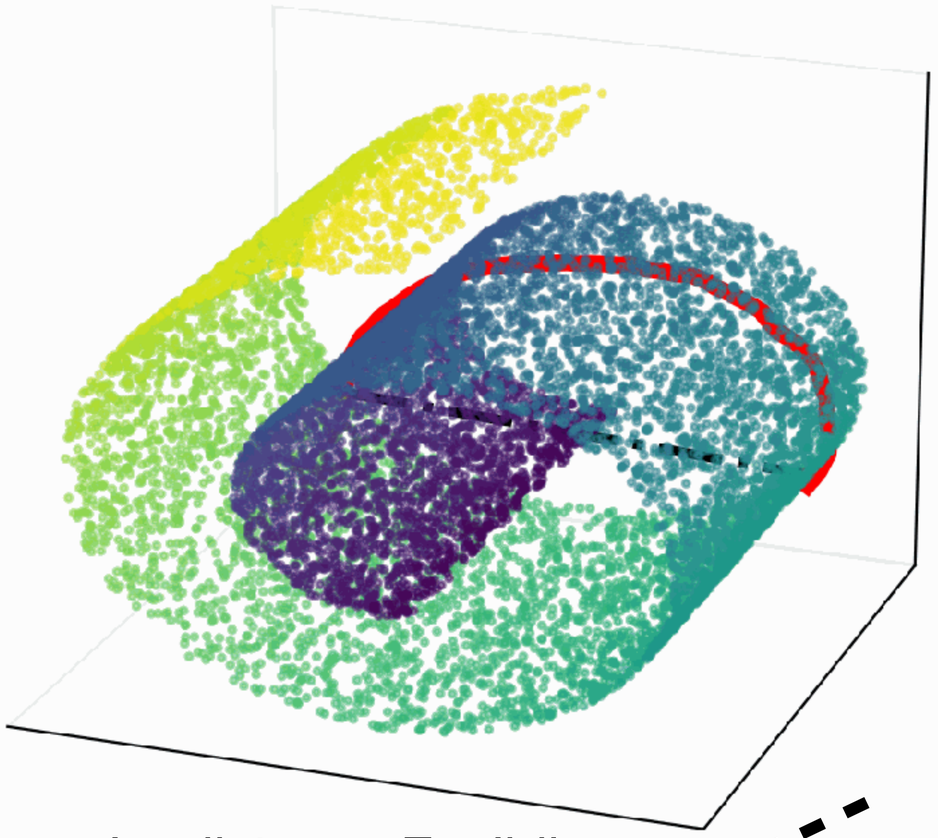


Peut-on améliorer cette MDS ?

On peut choisir la distance ! Quelle fonction de distance devrait-on choisir sur les données Swiss roll ?

INSEA

MDS et géodésiques



La distance Euclidienne

La géodésique

On aimerait bien pouvoir calculer les distances en “marchant” sur la surface courbe du roll

La distance entre deux points serait égale au trajet parcouru en prenant le plus court chemin sur cette surface

Cette distance s’appelle une géodésique.

Comment peut-on avoir une approximation de cette distance ?

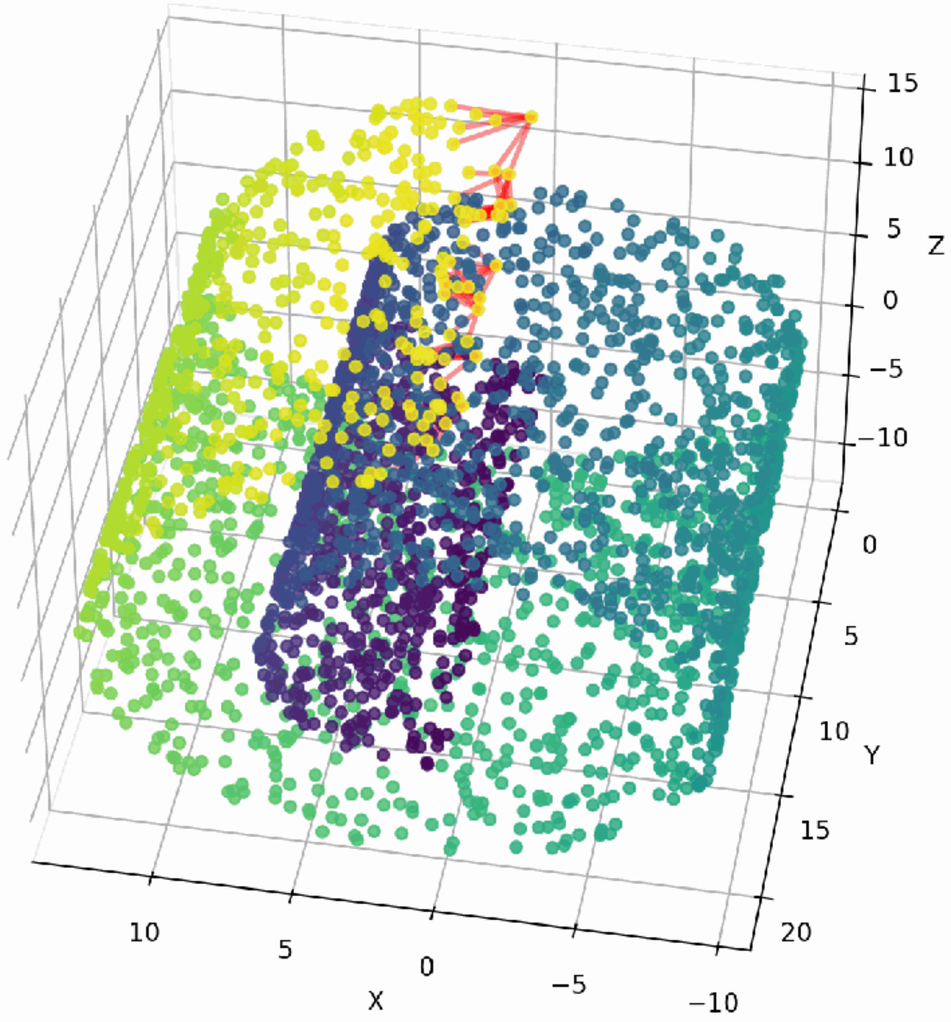
On doit approximer la surface avec un graphe

Graph des plus proches voisins

On doit approximer la surface avec un graphe

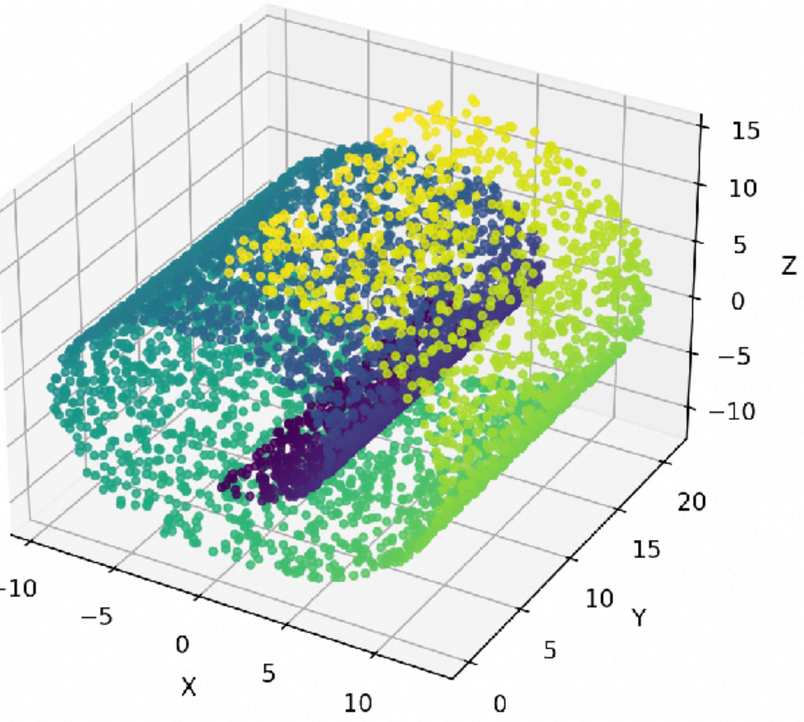
On crée un graphe en liant chaque point avec ses 5 plus proches voisins

La géodésique entre deux points est la longueur du chemin parcouru entre les deux points en restant sur le graphe



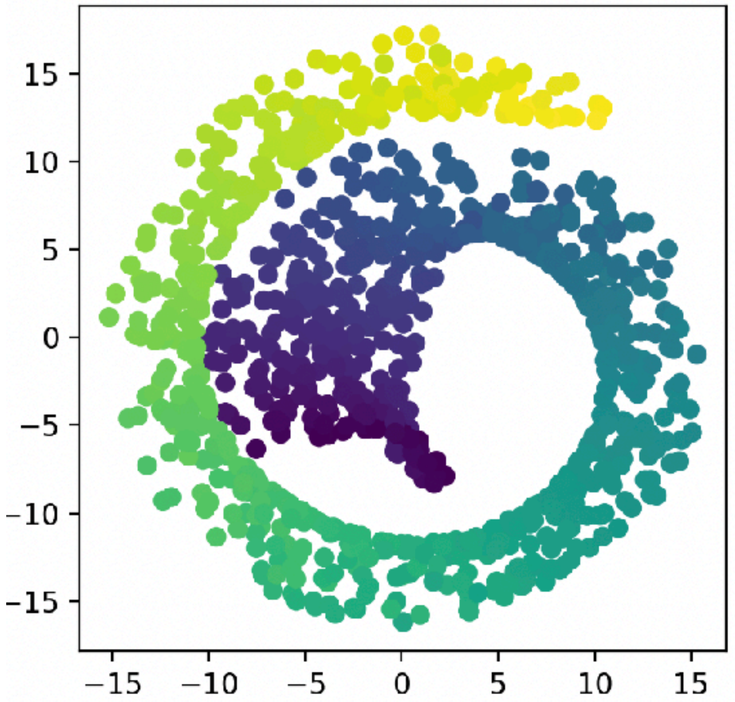
INSEA

Isomap



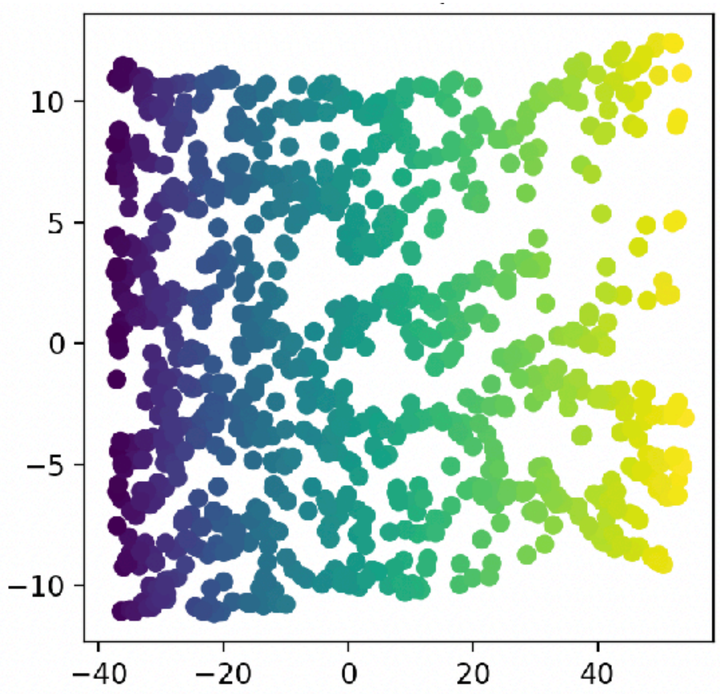
MDS

MDS (distances euclidiennes)



Isomap

MDS (distances géodésiques)



INSEA

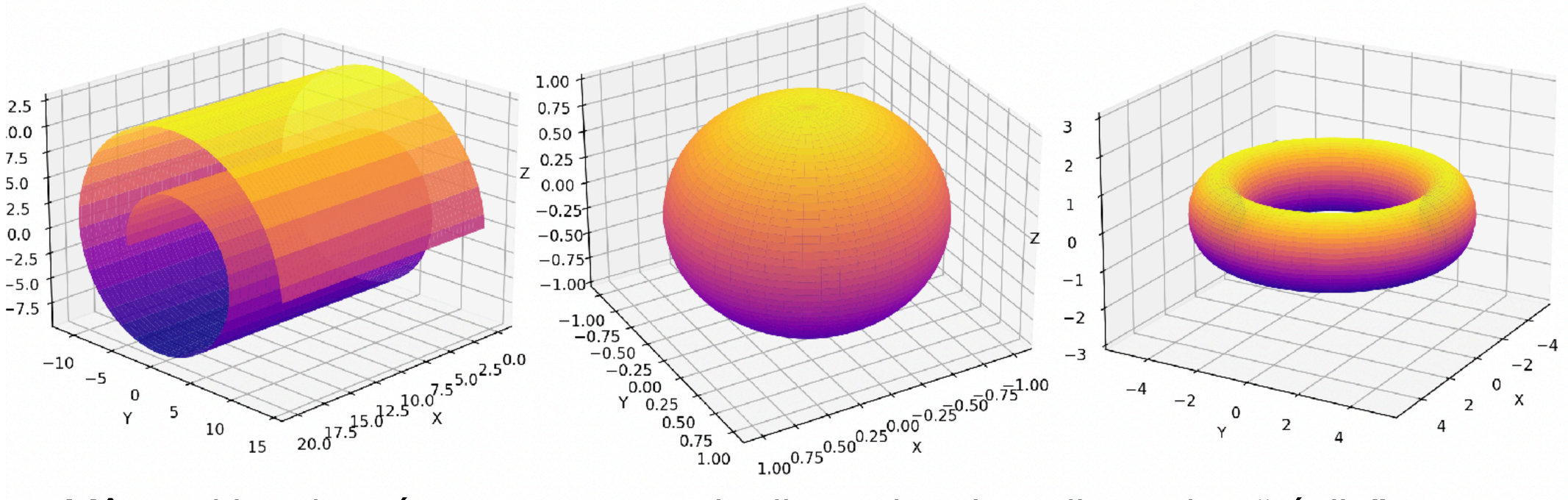
MDS: Résumé

1. Classical MDS projette des points en faible dimension tel que les produits scalaires entre les points sont préservés
2. Classical MDS donne des projections équivalentes à celles de la PCA.
3. MDS généralise Classical MDS et préserve les distances entre les points
4. MDS est surtout utilisé pour visualiser des graphes
5. MDS est peu utilisé en ML pour réduire la dimension: la PCA est en général préférable pour deux raisons:
6. a) MDS optimise les projetés directement et ne donne pas accès à une fonction de projection que l'on peut appliquer à des données futurs
7. b) Les axes MDS ne sont pas décorrélés et ne sont pas facilement interprétables
8. La puissance de MDS réside dans la possibilité d'utiliser des distances "sophistiquées" comme input
9. Si la distance en entrée est issue du plus court chemin sur un graphe, la méthode s'appelle Isomap

PCA, MDS, Isomap sont des méthodes de “Manifold learning”

Manifold: un sous espace “**localement**” vectoriel / Euclidien: une surface très lisse

Exemples:



Même si les données sont en grande dimension, leur dimension “réelle” est en général plus faible: elles sont situées sur un manifold

Quels manifolds la PCA peut-elle apprendre en une projection 3D -> 2D ?

Les plans uniquement !

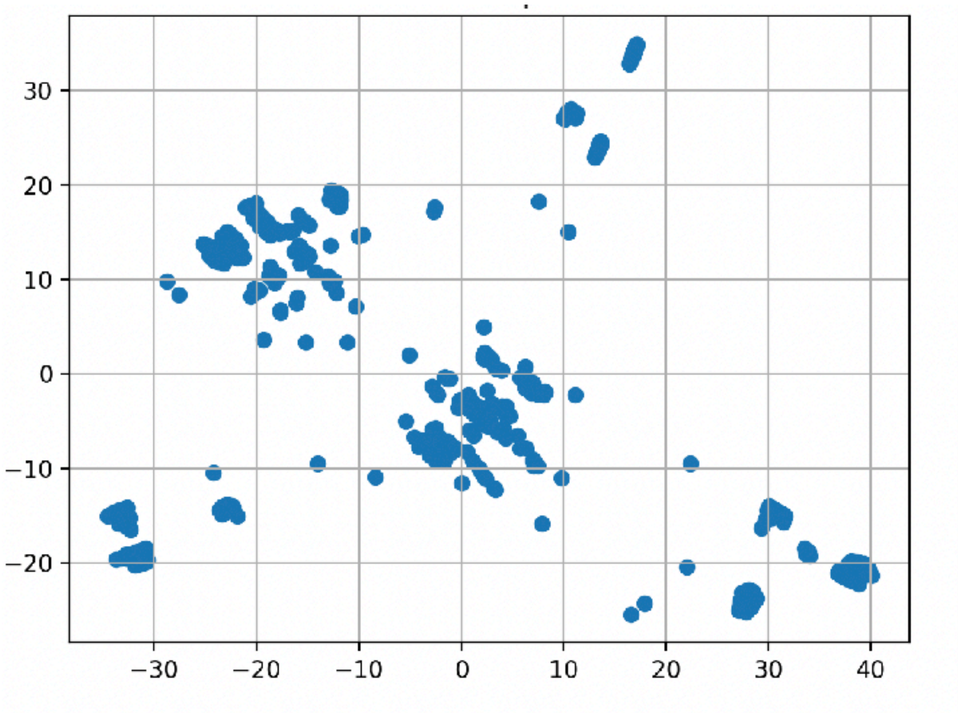
Isomap sur les données “ratings”

Données de ratings du site e-commerce:

Customer	Product 1	Product 2	...
Customer 1	4.5	3.0	...
Customer 2	3.5	4.0	...
Customer 3	5.0	2.5	...
⋮	⋮	⋮	⋮
Customer n	4.0	4.5	...

1. On construit le graphe des k plus proches voisins
2. On calcule les plus courts chemins sur le graphe
3. On donne cette distance à MDS

Isomap (MDS avec les géodésiques)

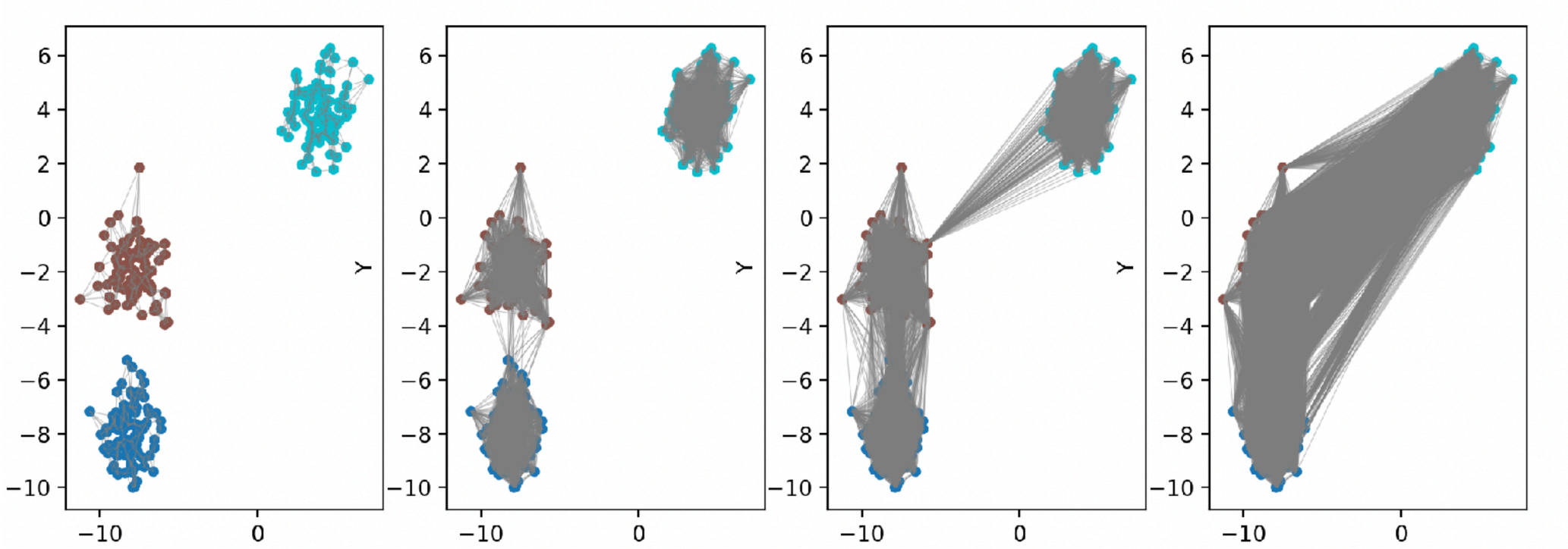


À vous de deviner: pourquoi Isomap a échoué sur des données “faciles” ?

Raison principal: graphe mal construit !

Le graphe dépend du nombre de voisins:

k = 5 | k = 50 | k = 100 | k = 200



Graphes non connectés:
Géodésique mal définie

Graphes “trop” connectés:
Géodésique converge vers
la distance euclidienne

Isomap: les plus et les moins

Avantages

1. Capable de retrouver des manifolds complexes, non linéaires
2. Correspond à une Metric MDS avec les géodésiques en input.

Inconvénients

1. Difficile d'avoir un graphe connecté (structure globale) sans sacrifier les structures locales (clusters proches).
2. Le calcul des géodésiques sur le graphe se fait avec l'algorithme Floyd-Warshall qui a une complexité $O(n^3)$.
3. Basé sur MDS, donc pas d'apprentissage d'une fonction de projection générale

INSEA

Comment préserver les structures locales ?

Objectif

Des points dans le même voisinage en grande dimension doivent rester voisins dans leur projection en 2D

Idée

1. Modéliser les voisinages avec une distribution de “saut” d’un point à l’autre
2. Garantir la même distribution de voisinage dans les projections 2D

Comment préserver les structures locales ?

Idée

1. Modéliser les voisinages avec une distribution de “saut” d’un point à l’autre

$\mathbb{P}(\mathbf{x}_j \text{ est voisin de } \mathbf{x}_i)$ doit être inversement proportionelle à $\|\mathbf{x}_i - \mathbf{x}_j\|$

$\mathbb{P}(\mathbf{x}_j \text{ est voisin de } \mathbf{x}_i)$ doit tendre rapidement vers 0 si $\|\mathbf{x}_i - \mathbf{x}_j\|$ est très grande (pas voisins)

Quelle fonction peut modéliser cette probabilité ?

$\mathbb{P}(j|i)$ proportionnel à $\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2)$ donc: $\mathbb{P}(j|i) = \dfrac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2)}$

1. Modéliser les voisinages avec une distribution de “saut” d’un point à l’autre

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

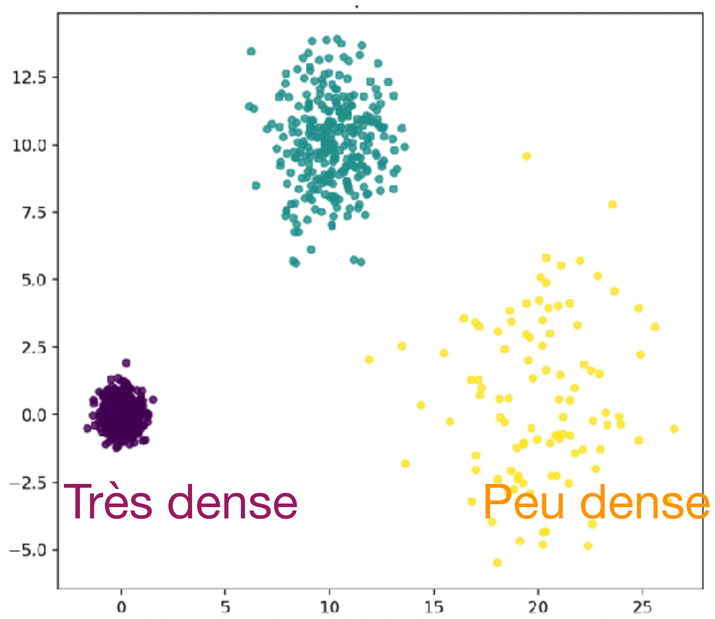
1. Ne tient pas compte de l’ordre de grandeur des distances
2. Les clusters doivent être de même variance

Quelles sont les inconvénients de ce modèle ?

Comment modifier cette fonction pour en tenir compte ?

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

Utiliser une variance différente pour chaque voisinage



On symétrise la distribution:

$$\mathbb{P}(i \text{ et } j \text{ sont voisins}) = \frac{1}{2N}\left(\mathbb{P}(i|j) + \mathbb{P}(j|i)\right)$$

INSEA

1. Modéliser les voisinages avec une distribution de “saut” d’un point à l’autre

$$\mathbb{P}(j|i) = \frac{\exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{{\sigma_i}^2}\right)}{\sum_{k \neq i} \exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_k\|^2}{{\sigma_i}^2}\right)} \qquad \mathbb{P}(i \text{ et } j \text{ sont voisins}) = \frac{1}{2N}\left(\mathbb{P}(i|j) + \mathbb{P}(j|i)\right)$$

2. Garantir la même distribution de voisinage dans les projections 2D

Optimiser les projetés $\mathbf{z}_1, \ldots, \mathbf{z}_n \in \mathbb{R}^2$ tels que:

$$\mathbb{P}(\mathbf{x}_i \text{ et } \mathbf{x}_j \text{ sont voisins}) \approx \mathbb{P}(\mathbf{z}_i \text{ et } \mathbf{z}_j \text{ sont voisins})$$

Quelle distribution de voisinage pour les projetés z ?

Le plus simple serait: $\text{proportionnel à } \exp(-\|\mathbf{z}_i - \mathbf{z}_j\|^2)$

(SNE): Stochastic Neighbor Embedding *2002 (G. Hinton et al.)*

INSEA

(SNE): Stochastic Neighbor Embedding

Avec cette distribution, les voisinages projetés en 2D sont “surchargés / superposés”

$\exp(-\|\mathbf{z}_i - \mathbf{z}_j\|^2)$ décroit trop rapidement vers 0 avec la taille du voisinage

Solution:

$$\mathbb{P}(\mathbf{z}_i \text{ et } \mathbf{z}_j \text{ sont voisins}) \quad \text{proportionnel à} \quad \frac{1}{1 + \|\mathbf{z}_i - \mathbf{z}_j\|^2}$$

Distribution t de Student

(t-SNE): t-Distributed Stochastic Neighbor Embedding

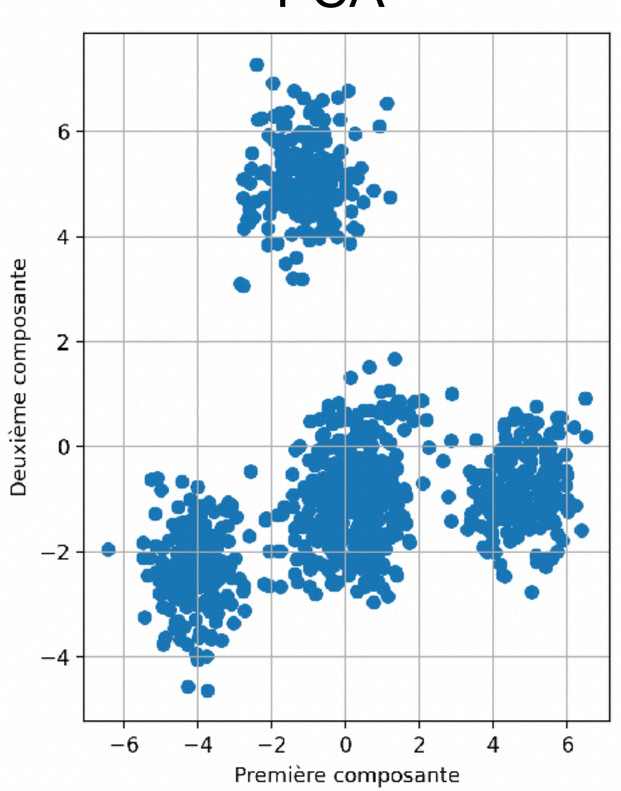
2008 (van der Marteen et al.)

t-SNE sur les données “ratings”

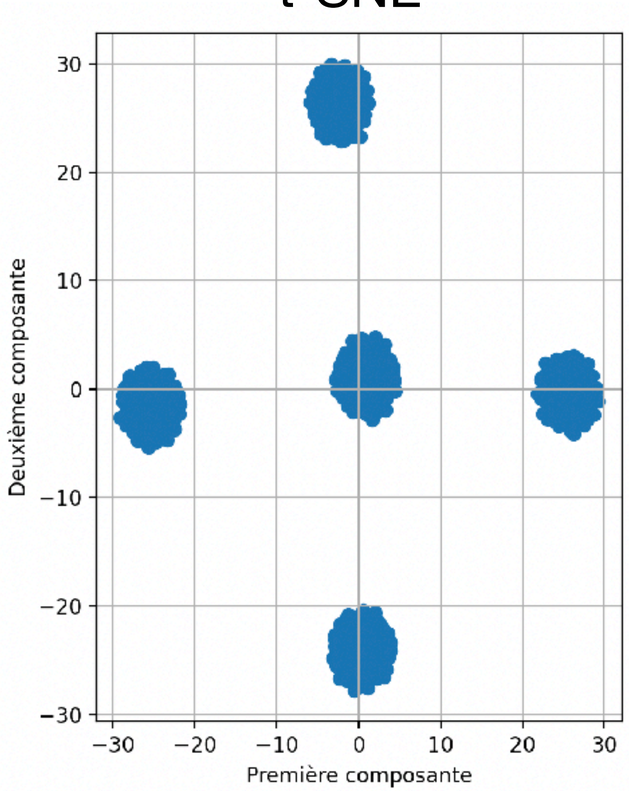
Données de “ratings”
du site e-commerce:

Customer	Product 1	Product 2	...
Customer 1	4.5	3.0	...
Customer 2	3.5	4.0	...
Customer 3	5.0	2.5	...
⋮	⋮	⋮	⋮
Customer n	4.0	4.5	...

PCA



t-SNE



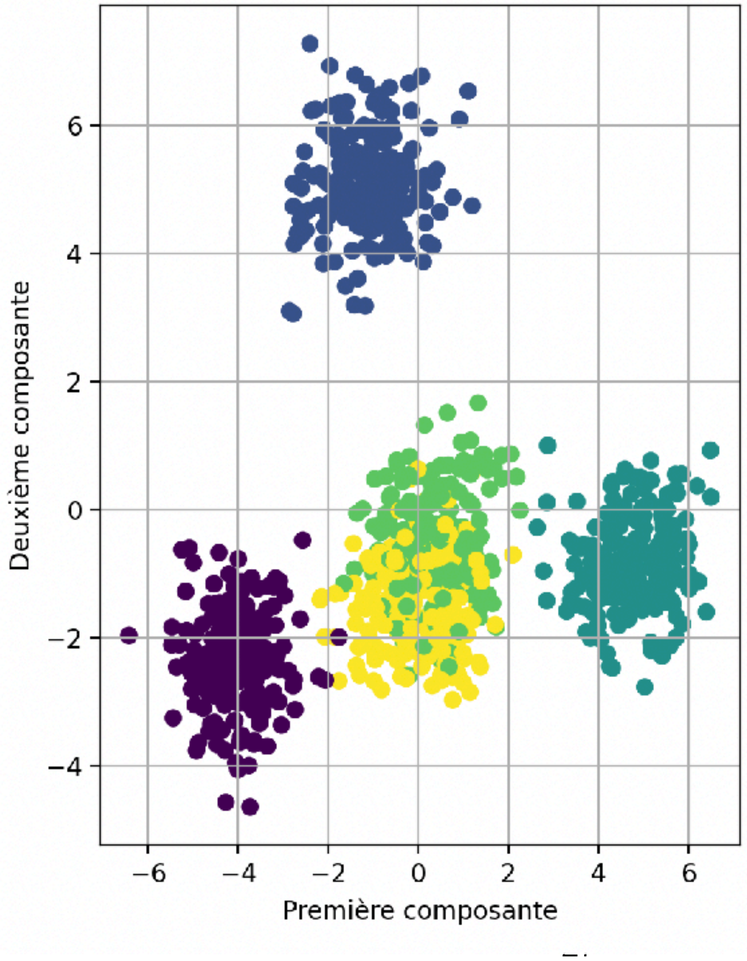
t-SNE a trouvé 5 clusters très distincts ! Comment cela est-il possible ?

t-SNE sur les données “ratings”

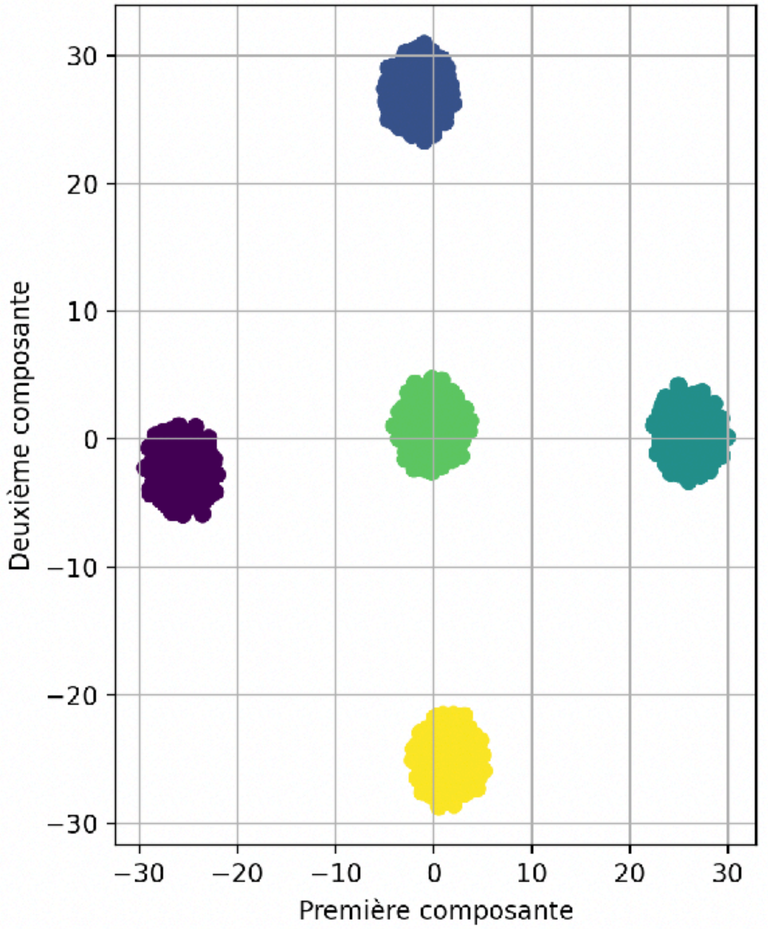
Ces données sont en réalité simulées avec 5 clusters.
En ajoutant l’information des labels (couleur):

La projection PCA a superposé deux clusters !

PCA



t-SNE



INSEA

t-SNE: le paramètre de perplexité

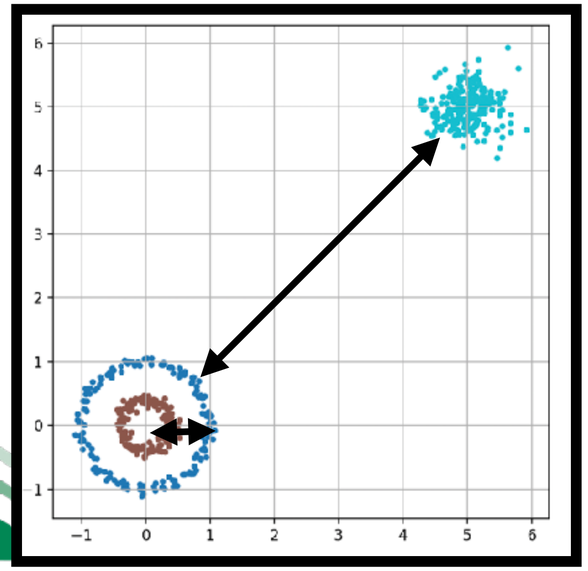
On modélise les voisinages avec une distribution de “saut” d’un point à l’autre:

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

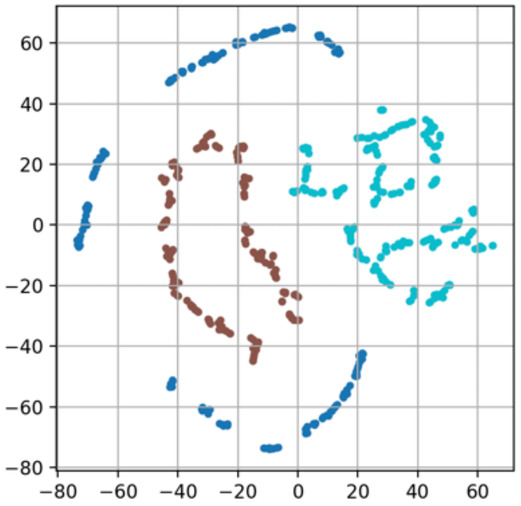
Remarque 1:

On ne choisit pas les variances une à une pour chaque point. On les contrôle indirectement en précisant un paramètre de “perplexité” similaire au nombre de voisins les plus proches. Plus la perplexité est grande, plus on conserve la structure globale des données.

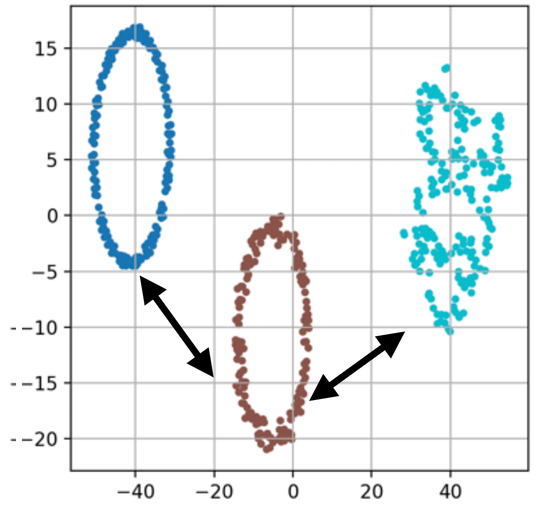
Données



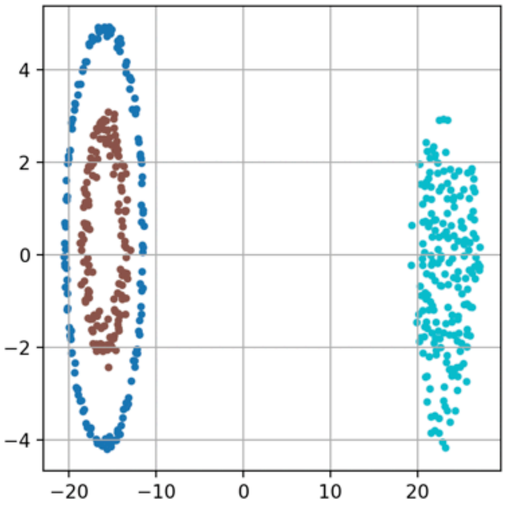
perp. = 5



perp. = 20



perp. = 100



On ne peut pas interpréter les distances entre clusters (structure globale)!

t-SNE: remarques

Remarque 2:

t-SNE repose sur plusieurs opérations à complexité quadratique $O(n^2)$: t-SNE n'est pas adapté pour des grands datasets. La complexité du calcul de la matrice des distances est $O(dn^2)$.

Remarque 3:

t-SNE est sensible au bruit: si la dimension est très grande, il est préférable d'utiliser d'abord une PCA pour la réduire (à $k < 1000$) avant de la réduire à 2 avec t-SNE.

INSEA

Comparaison des méthodes

Méthode	Manifold	Préserve	Coût de calcul	Année de publication
PCA	Linéaire	Structure Globale	$O(d^3)$	1901 (K. Pearson)
Classical MDS	Linéaire	Structure Globale	$O(n^3)$	1952 (W.S. Torgerson)
Metric MDS	Non-Linéaire	Structure Globale	$O(n\_iter \times n^2)$	1964 (Joseph B. Kruskal)
Isomap	Non-Linéaire	Structure Globale	$O(n^3)$	2000 (J.B. Tenenbaum et al.)
SNE	Non-Linéaire	Structure Locale	$O(n\_iter \times n^2)$	2002 (G. Hinton et al.)
t-SNE	Non-Linéaire	Structure Locale	$O(n\_iter \times n^2)$	2008 (van der Marteen et al.)

Une méthode préservant à la fois la structure locale et globale ?

UMAP	Non-Linéaire	Locale et globale	$O(n \log n + n \times n\_iter)$	2018 (Leland McInnes et al.)

INSEA

UMAP en bref

Une méthode préservant à la fois la structure locale et globale ?

UMAP	Non-Linéaire	Locale et globale	O(n log n)	2018 (Leland McInnes et al.)

UMAP: **U**niform **M**anifold **A**pproximation and **P**rojection

UMAP

1. Crée une approximation d'un graphe par voisins pour modéliser le manifold (comme Isomap) pour prendre en considération la structure globale.

2. Mais ne calcule pas les distances géodésiques: utilise le graphe pour modéliser les distributions des voisins (comme t-SNE) pour prendre en compte la structure locale.

Aussi / plus performant que t-SNE + plus rapide

Non inclus dans scikit-learn, package indépendant: pip install umap-learn

INSEA

Quelles sont les différences fondamentales entre :

MDS Isomap t-SNE UMAP

non-linéaire Pas de transformation générale

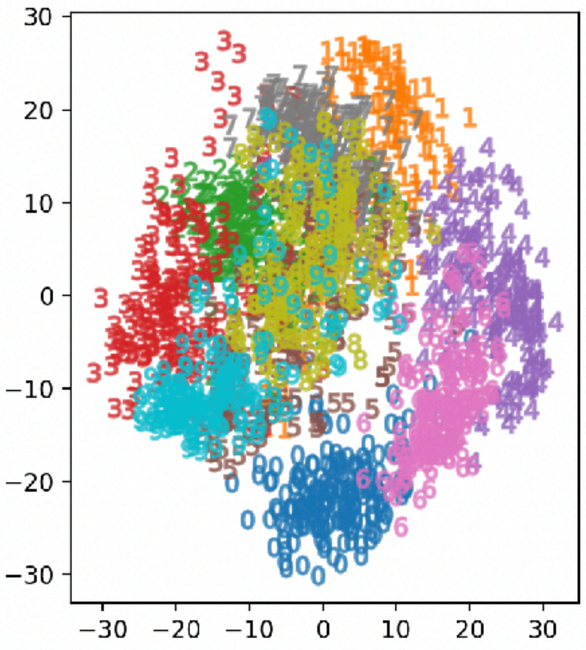
et

PCA

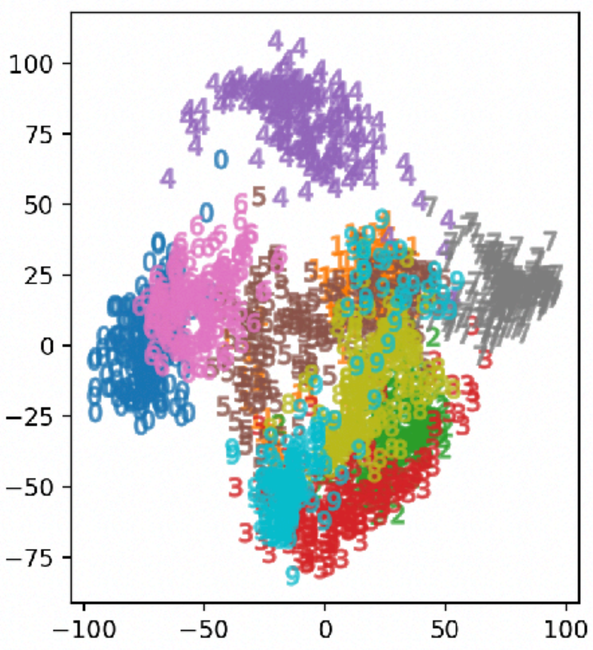
Linéaire Apprend une transformation générale

Visualisations des données digits

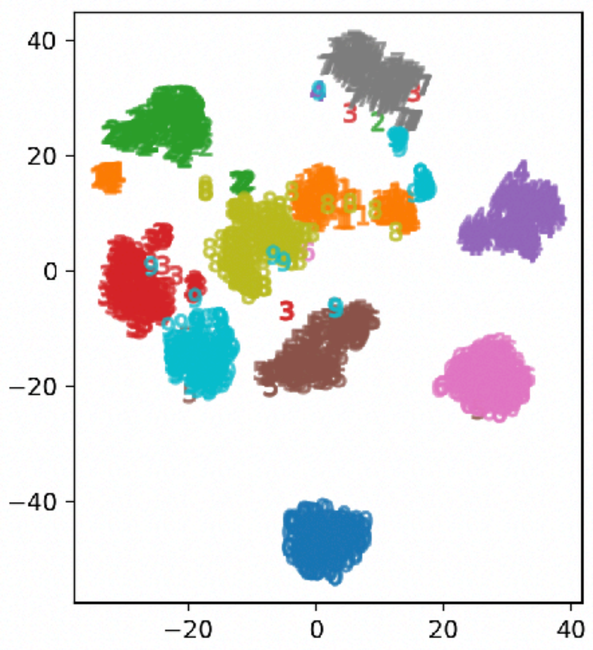
PCA (2ms)



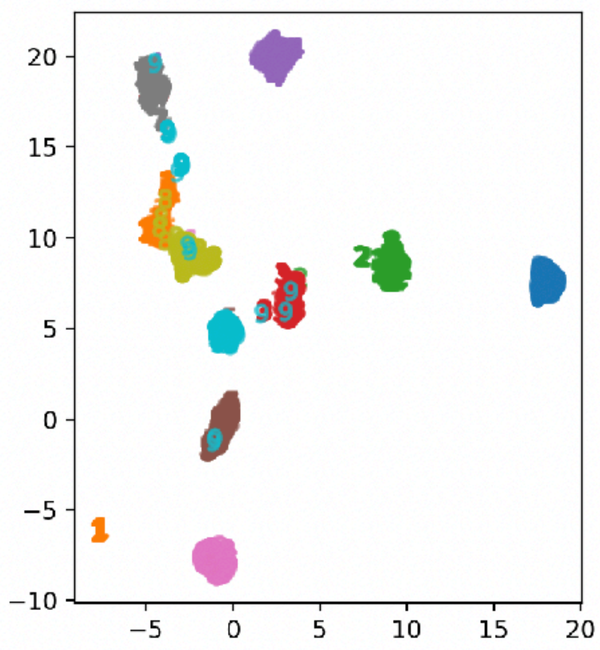
Isomap (4s)



t-SNE (29s)



UMAP (5s)



(TP cet après-midi)

INSEA