

Machine Learning

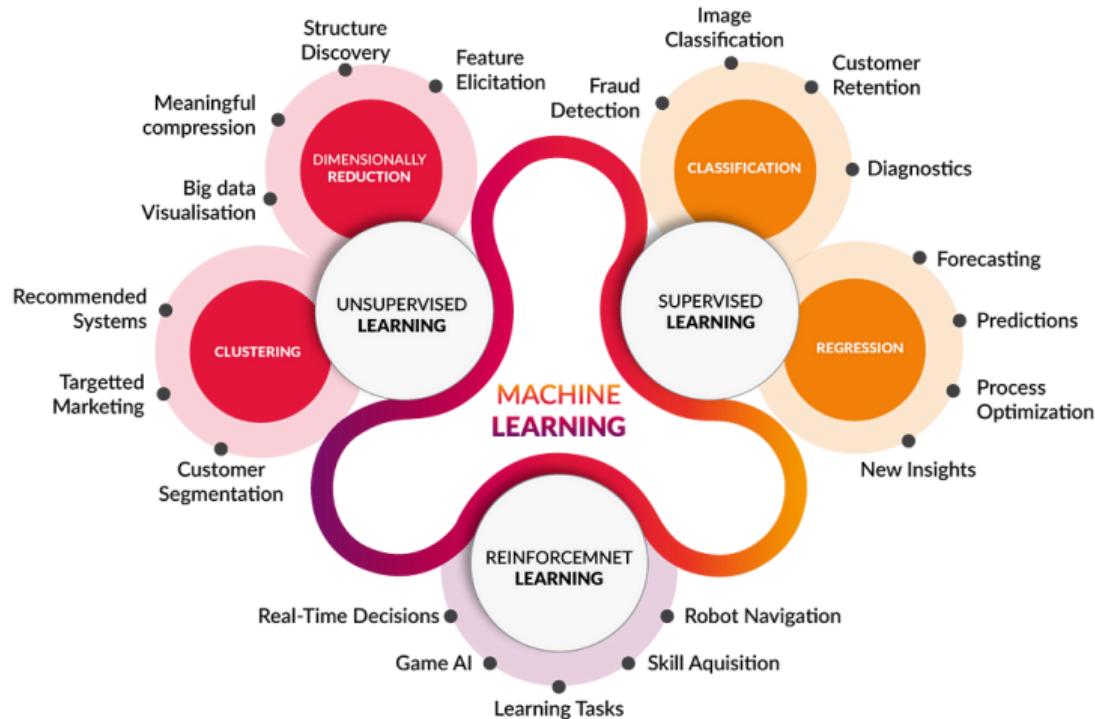
Apprentissage non supervisé

S. Herbin

stephane.herbin@onera.fr

Introduction

Les types d'apprentissage



<https://resources.explorify.co/a1-nl/coding-deep-learning-for-beginners-types-of-machine-learning/>

Apprentissage non-supervisé : contexte

Données de grande dimension :

- ▶ Ex : image = un point dans $[0, 255]^{1M}$
- ▶ Information redondante, non-pertinente
- ▶ Fléau de la dimension : les espaces de représentation sont vides (exemple : 50 dimensions, 20 niveaux par dimensions $\Rightarrow 20^{50}$ cellules...)
- ▶ Temps de traitement dépend de la taille des données

Grand volume de données :

- ▶ Stockage de l'information pertinente
- ▶ Visualisation globale des distributions
- ▶ Mais annotations plus rares (car chères)

Apprentissage non-supervisé : pourquoi ?

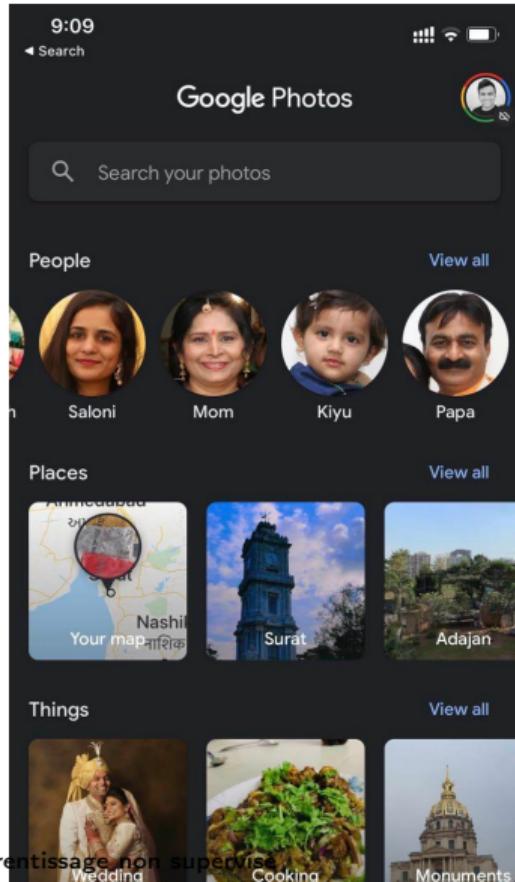
L'information dans les données peut être cachée

- ▶ Les données sont génériques : pas adaptées à tous les problèmes
- ▶ Elles contiennent souvent des mesures ou dimensions inutiles
- ▶ Il peut y avoir du bruit

Les données sont en général non annotées

- ▶ Car les annotations sont chères
- ▶ Mais leur distribution a une structure
- ▶ Elles peuvent *aider* l'apprentissage supervisé
- ▶ Les humains apprennent avec peu d'annotations !

Apprentissage non-supervisé : quelques applications I



Frequently bought together



Total price: £304.21

Add all three to Basket

i These items are dispatched from and sold by different sellers. [Show details](#)

- This item:** Canon EOS 4000D DSLR Camera and EF-S 18-55 mm f/3.5-5.6 III Lens - Black £255.00
- Canon ES100 Camera Bag - Black £38.63
- SanDisk Ultra SDXC Memory Card Up to 80 MB/s, Class 10, U1, 64 GB, Black/Grey £10.58

Apprentissage non-supervisé : quelques applications II

RESEARCH ARTICLE | BIOLOGICAL SCIENCES | 8

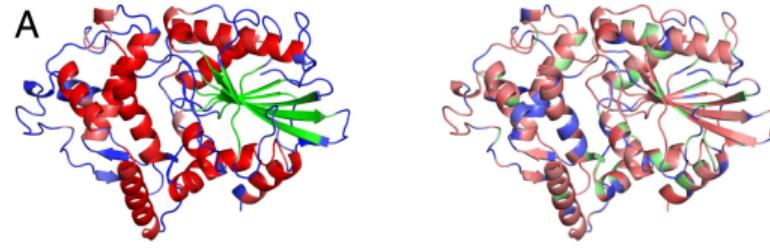


Biological structure and function emerge from scaling unsupervised learning to 250 million protein sequences

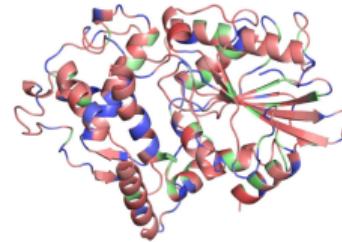
Alexander Rivero, Joshua Meier, Tom Serre, and Rob Fergus. [Authors Info & Affiliations](#)

Edited by David T. Jones, University College London, London, United Kingdom, and accepted by Editorial Board Member William H. Press December 16, 2020 (received for review August 6, 2020)

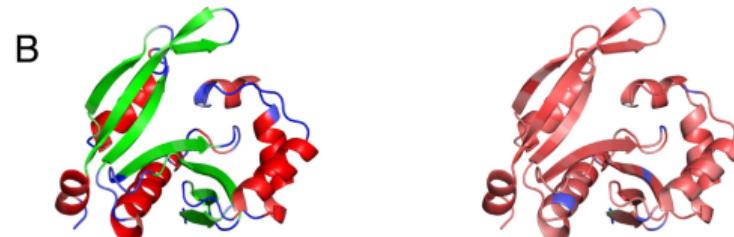
April 5, 2021 | 118 (15) e2016239118 | <https://doi.org/10.1073/pnas.2016239118>



With pre-training
8-class Acc: 70.6%



d1nt4a_ (Phosphoglycerate mutase-like fold)



With pre-training
8-class Acc: 82.4%

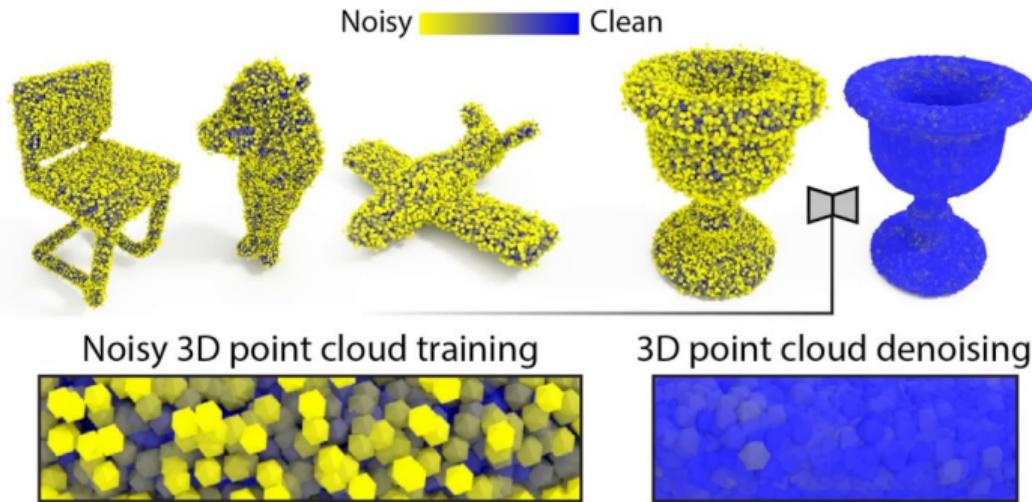


d3wr7a_ (Acyl-CoA N-acyltransferases fold)

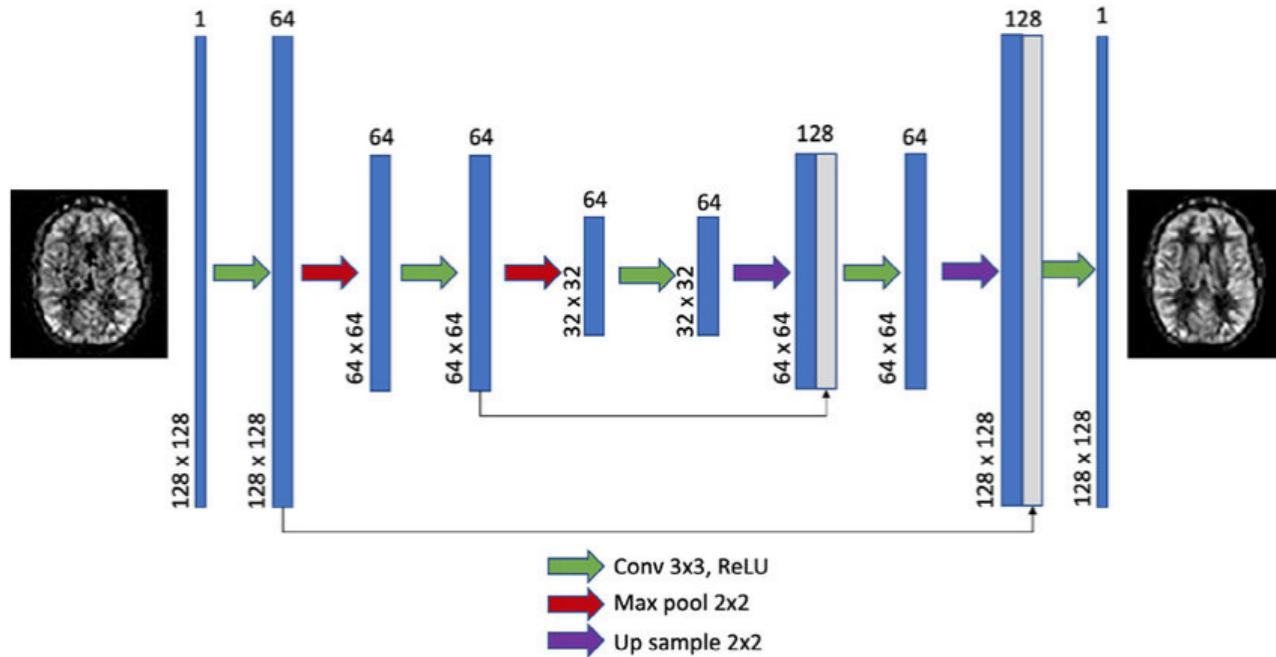


Apprentissage non-supervisé : quelques applications III

Unsupervised point cloud denoising



Apprentissage non-supervisé : quelques applications IV



Apprentissage non-supervisé : objectifs

Trois problématiques :

1. Découvrir des « structures » dans les données.
2. Extraire et représenter l'information utile.
3. Visualiser les distributions.

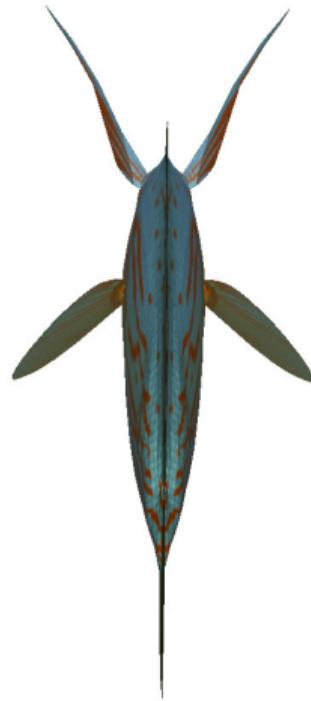
Apprentissage de représentation

Analyse en Composantes Principales

Comment bien dessiner un objet 3D ?

Analyse en Composantes Principales

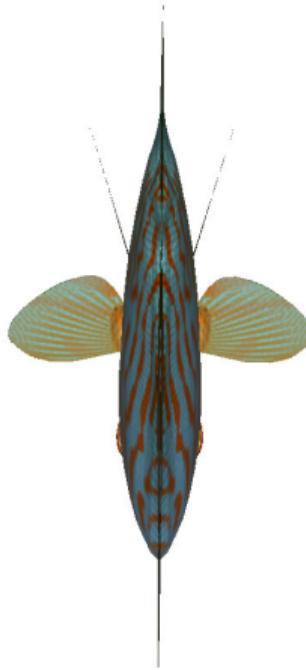
Dessinez un poisson...



Analyse en Composantes Principales

Dessinez un poisson...

- ▶ Les poissons vivent en 3D...



Analyse en Composantes Principales

Dessinez un poisson...

- ▶ Les poissons vivent en 3D...
- ▶ Comment les représenter sur une feuille 2D ?



Analyse en Composantes Principales

Dessinez un poisson...

- ▶ Les poissons vivent en 3D...
- ▶ Comment les représenter sur une feuille 2D ?
- ▶ En choisissant le meilleur point de vue



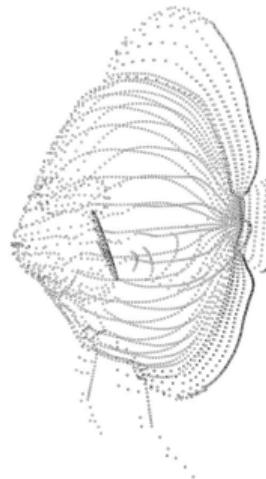
Analyse en Composantes Principales

Dessinez un poisson...

- ▶ Les poissons vivent en 3D...
- ▶ Comment les représenter sur une feuille 2D ?
- ▶ En choisissant le meilleur point de vue
- ▶ En perspective (tous les points de vue !)



Analyse en Composantes Principales



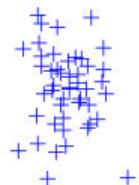
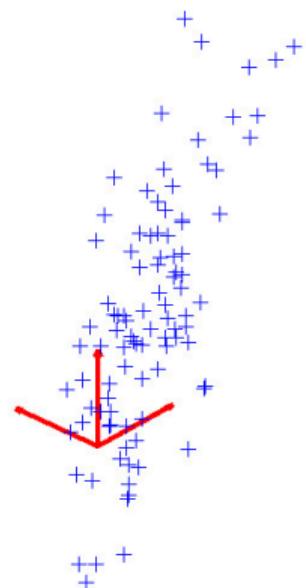
Analyse en Composantes Principales

L'ACP (ou *PCA - Principal Component Analysis*) est une méthode de projection qui permet de représenter au mieux les données d'origine en réduisant le nombre de dimensions.

Analyse en Composantes Principales : Formalisme

Algèbre Linéaire (*Rappel*)

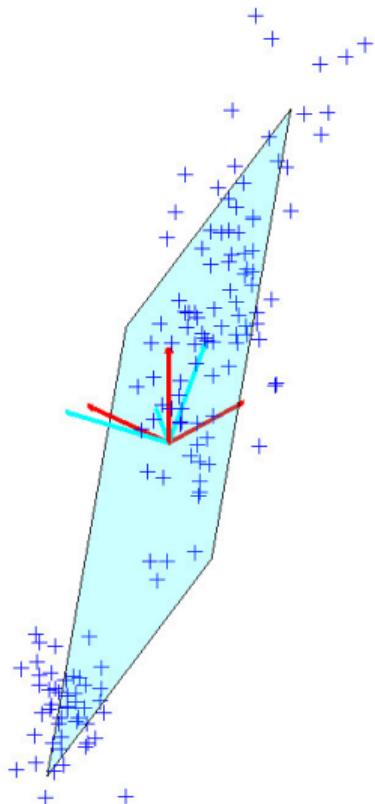
- ▶ Espace vectoriel E : structure permettant des combinaisons linéaires de *vecteurs*
 $x_k = (x_k^1, \dots, x_k^n)$
- ▶ Base B : famille de vecteurs *libre* et *génératrice*



Analyse en Composantes Principales : Formalisme

Algèbre Linéaire (*Rappel*)

- ▶ Espace vectoriel E : structure permettant des combinaisons linéaires de *vecteurs*
 $x_k = (x_k^1, \dots, x_k^n)$
- ▶ Base B : famille de vecteurs *libre* et *génératrice*
- ▶ Changement de base : endomorphisme
 $E \rightarrow E, B \mapsto B'$.
- ▶ Projection : Application linéaire de $E \rightarrow F$, ss-EV de E .



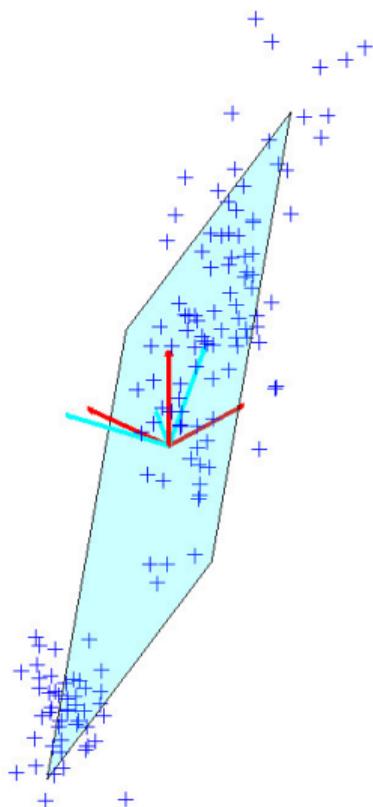
Analyse en Composantes Principales : Formalisme

Objectif géométrique de l'ACP

L'ACP est la recherche du sous-espace de projection qui permet la représentation la plus informative des variables dans un sous-espace de dimension réduite.

Deux principes de conception possibles

- ▶ Approximation : minimiser l'erreur de reconstruction
- ▶ Représentation : maximiser l'information



Analyse en Composantes Principales : principe I

On cherche la direction de projection u qui

- ▶ Maximise la variance de projection (l'« information »)

$$\frac{1}{n} \sum_{i=1}^n (u^T x_i)^2 = \frac{1}{n} u^T X X^T u$$

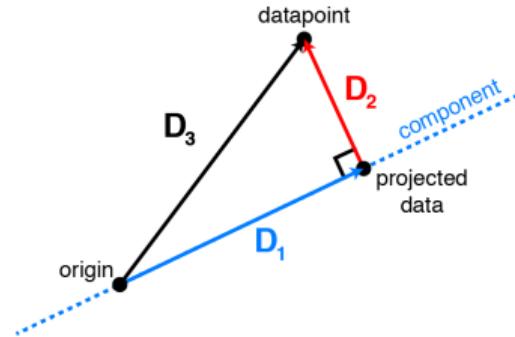
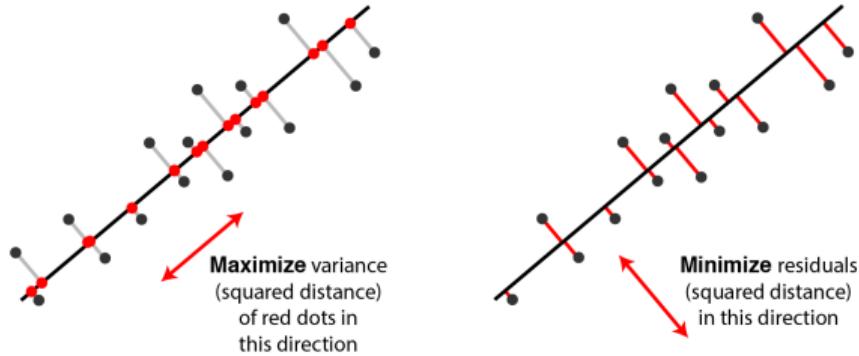
où $X = [x_1, \dots, x_n]$ est la matrice $d \times n$ des données, et XX^T la matrice de covariance.

- ▶ Minimise l'erreur de reconstruction

$$\frac{1}{n} \sum_{i=1}^n \|x_i - (u^T x_i)u\|^2$$

- ▶ Les deux principes aboutissent au même résultat !

Analyse en Composantes Principales : principe II



$$D_3^2 = D_1^2 + D_2^2$$

initial variance = remaining variance + lost variance

$$\|a_i\|^2 = \|w_i c\|^2 + \|a_i - w_i c\|^2$$

this is constant maximize this or minimize this

Analyse en Composantes Principales : Calcul

- ▶ On cherche à maximiser la variance projetée :

$$\mathbf{u}^* = \arg \max_{\|\mathbf{u}\|^2=1} \frac{1}{n} \sum_{i=1}^n (\mathbf{u}^T \mathbf{x}_i)^2$$

- ▶ En utilisant un Lagrangien pour l'optimisation sous contrainte :

$$\mathcal{L}(\mathbf{u}, \lambda) = \mathbf{u}^T \mathbf{X} \mathbf{X}^T \mathbf{u} - \lambda (\mathbf{u}^T \mathbf{u} - 1)$$

et en dérivant, on en déduit que la solution est un vecteur propre de $\mathbf{X} \mathbf{X}^T$.

- ▶ La matrice $\mathbf{X} \mathbf{X}^T$ est symétrique \implies elle est diagonalisable avec des vecteurs propres orthogonaux.

Analyse en Composantes Principales : Propriétés

- ▶ On peut montrer que si l'on projette sur les p premiers vecteurs propres (i.e. p composantes principales), l'erreur de reconstruction est :

$$\|X - PP^T X\|^2 = \sum_{c=p+1}^d \lambda_c^2$$

où $P = [u_1, \dots, u_d]$ est la matrice $d \times p$ des d vecteurs propres.

- ▶ On en déduit une décroissance des résidus d'approximation lorsque l'on ordonne les vecteurs propres de manière décroissante selon leur valeur absolue.
- ▶ Les valeurs propres décroissent assez vite dans beaucoup de problèmes.

Analyse en Composantes Principales : mise en œuvre

Première manière

1. Calcul de la matrice de covariance (centrée, c'est mieux)
2. Calcul des vecteurs et valeurs propres
3. Choix des composantes (nouvelle base) et projection des données sur cette base.

Deuxième manière

1. Minimisation directe de l'erreur de reconstruction
[Cunningham and Ghahramani, 2015]

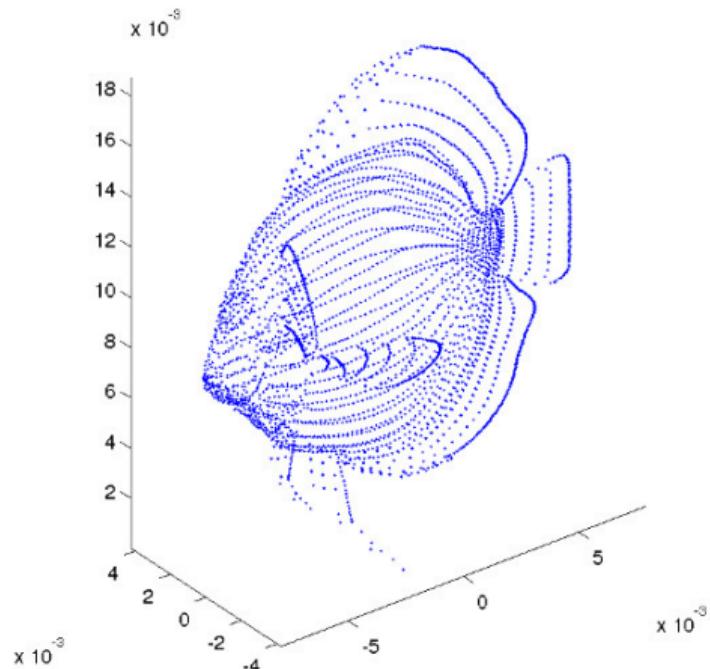
Trouver P qui minimise $\|X - PP^T X\|_F^2$ avec $P \in \mathcal{O}^{d \times p}$.

Peut être plus efficace pour de très grandes dimensions de l'espace de représentation.

Analyse en Composantes Principales : Exemples

Points $\in \mathbb{R}^3$ répartis sur la surface du Discus Alenquer

Variances :



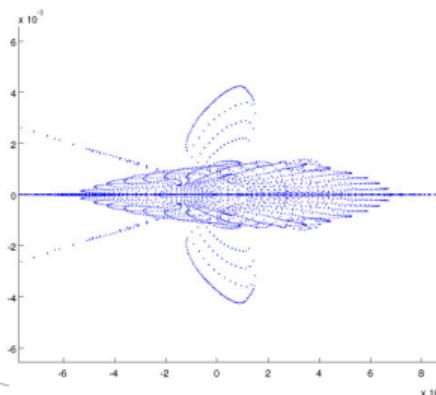
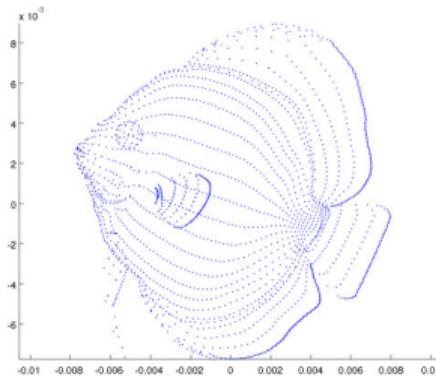
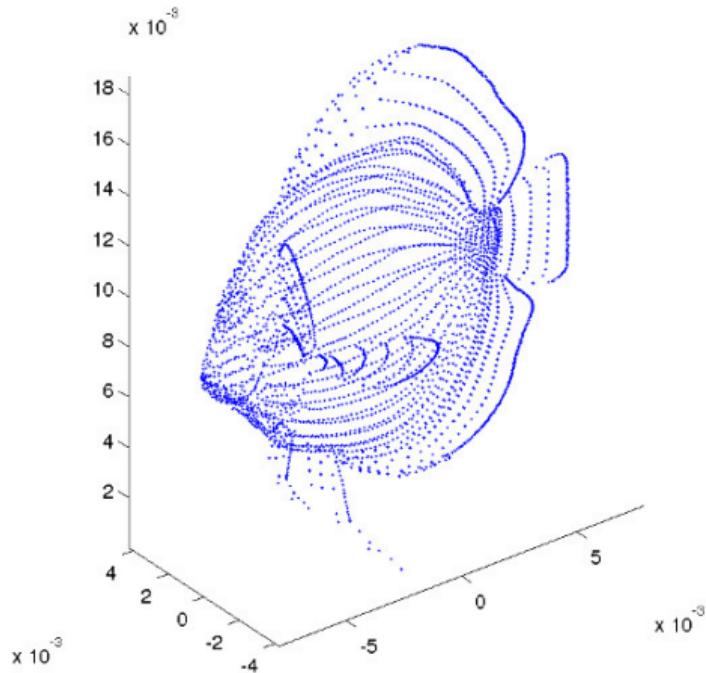
$$\Delta \propto \begin{pmatrix} 0.17 & 0 & 0 \\ 0 & 0.15 & 0 \\ 0 & 0 & 0.01 \end{pmatrix}$$

Base des vecteurs propres :

$$P = \begin{pmatrix} 0.91 & -0.42 & 0 \\ 0 & 0 & 1 \\ 0.42 & 0.91 & 0 \end{pmatrix}$$

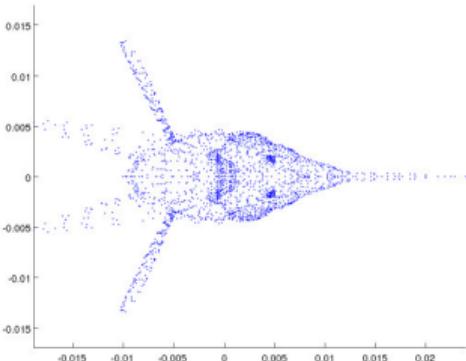
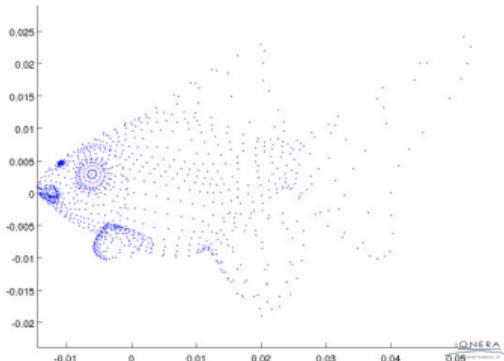
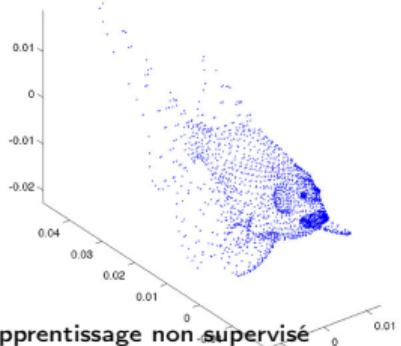
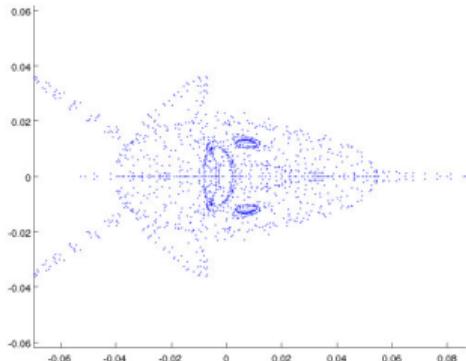
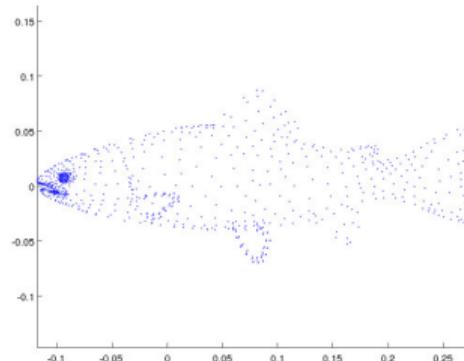
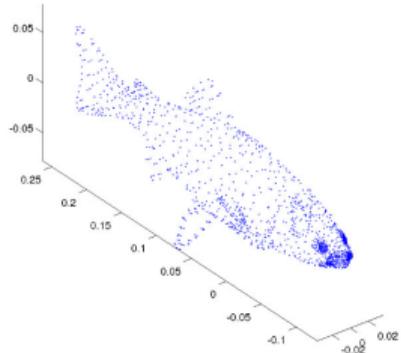
Analyse en Composantes Principales : Exemples

Projections sur les 2 premières (ou dernières) composantes



Analyse en Composantes Principales : Exemples

3D vs. 1ères CP (= représentation canonique) vs. dernières CP



Apprentissage non supervisé

ACP = Apprentissage de représentations

L'ACP est un moyen de représenter l'information utile contenue dans les données par décomposition sur une base orthogonale + mécanisme de reconstruction linéaire.

Au delà de l'ACP

D'autres techniques d'apprentissage non-supervisé de telles représentations existent.

- ▶ Représentation par dictionnaire : décomposition sur une base non orthogonale + reconstruction linéaire parcimonieuse
- ▶ Auto supervision : décomposition et reconstruction par « Deep Network »
- ▶ Kernel PCA [[Schölkopf et al., 1998](#)] : minimisation dans un espace de Hilbert (produit scalaire → noyau, le « kernel trick »)

Représentation par dictionnaire : formulation

- ▶ Dictionnaire : $D = [d_1, \dots, d_m] \in \mathbb{R}^{d \times m}$ et $\|d_j\|^2 \leq 1$ définit les éléments de représentation
- ▶ On cherche les coefficients (le code) $\alpha = [\alpha_1, \dots, \alpha_m]$ qui permet d'approximer chaque donnée x par le dictionnaire :

$$x \approx \sum_j \alpha_j \cdot d_j$$

- ▶ On recherche un codage parcimonieux (« sparse ») obtenu par pénalisation L_1 :

$$\alpha^*(x) = \arg \min_{\alpha \in \mathbb{R}^m} \|x - D\alpha\|_2^2 + \lambda \|\alpha\|_1$$

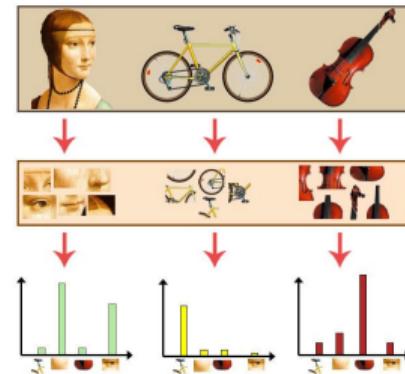
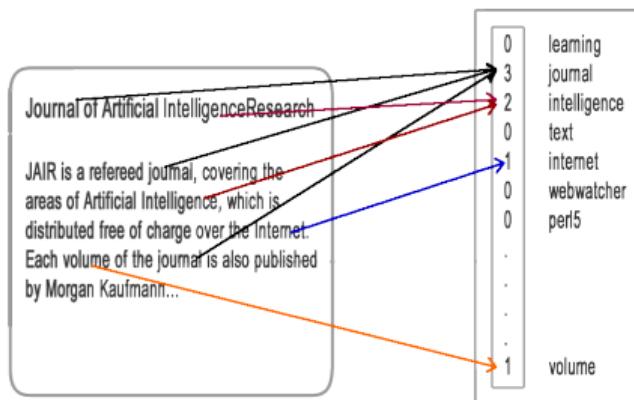
- ▶ L'apprentissage du dictionnaire demande d'optimiser conjointement les codes α et le dictionnaire D , éventuellement de manière incrémentale [[Mairal, 2015](#)] :

$$\min_{D, \alpha} \|X - D\alpha\|_2^2 + \lambda \|\alpha\|_1$$

Utilisation d'un dictionnaire

« Bag of words »

- ▶ Encode chaque donnée (« texte ») par la fréquence d'occurrence des mots du dictionnaire (histogramme)
- ▶ Le vocabulaire codant est souvent limité : représentation creuse, mais les coordonnées non nulles sont caractéristiques
- ▶ Utilisation pour représentation de texte ou d'image.



Utilisation d'un dictionnaire

Débruitage

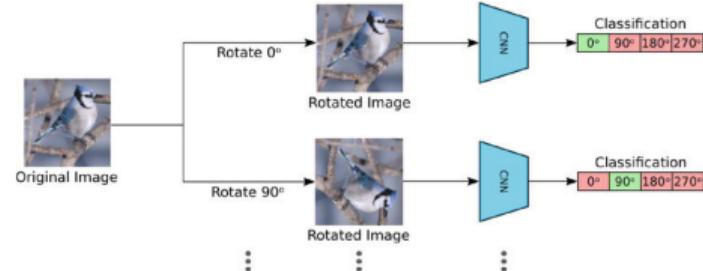
- ▶ Le codage permet de représenter les données dans un espace que l'on peut contrôler
- ▶ Codage
 - = projection dans cet espace
 - = reconstruction
 - = extraction de l'information utile
- ▶ On verra d'autres approches DL de débruitage (auto-encodeurs).



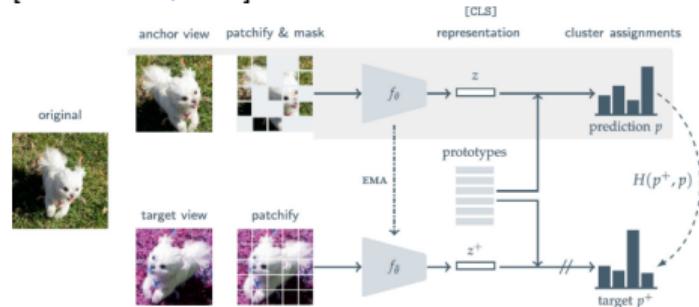
Auto-supervision I

Principe

- ▶ Créer un problème facile à annoter (*tâche prétexte*) qui demande d'extraire de l'information utile pour le résoudre
Ex : Rotation, puzzle, masquage
- ▶ Optimiser la représentation à partir d'un coût mesurant le succès de résolution du problème sur beaucoup de données
- ▶ Utiliser ensuite la représentation pour des tâches supervisées (avec peu/moins de données annotées)



[Gidaris et al., 2018]



[Assran et al., 2022]

Vers une représentation universelle ?

Apprentissage non supervisé

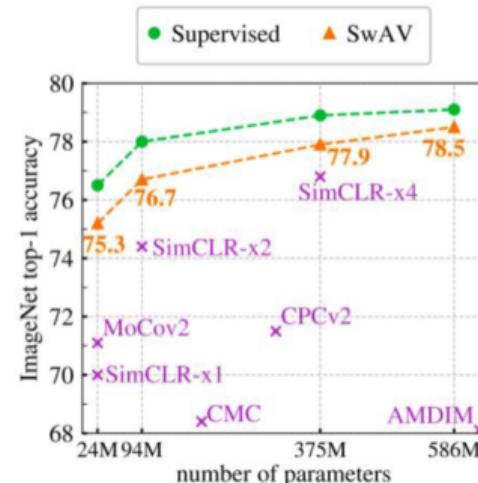
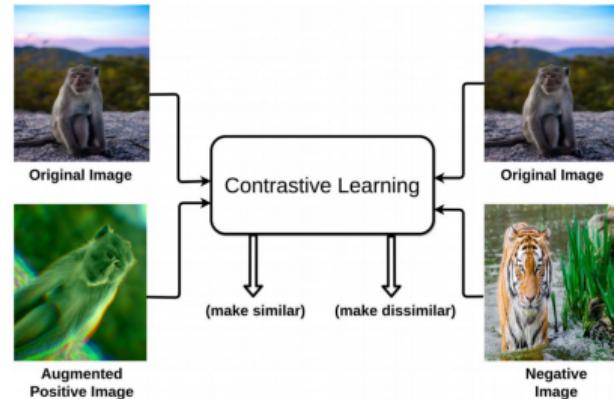
Auto-supervision II

Apprentissage contrastif

- ▶ Principe simple : augmenter l'écart entre ce que l'on souhaite et ce que l'on veut éviter !
- ▶ Ex : dans le cas d'apprentissage de représentation, augmenter la similarité pour des représentations que l'on sait équivalentes.

$$l(z_i, z_j) = -\log \frac{\exp[sim(z_i, z_j)/T]}{\sum_{j' \neq j} \exp[sim(z_i, z_{j'})/T]}$$

- ▶ La question est de bien générer les exemples similaires et différents.



[Caron et al., 2020]

Apprentissage de représentations : Résumé

Points clés

- ▶ Représenter et réduire des données de grande dimension
- ▶ Décorreler les variables (ACP)
- ▶ Extraire et coder l'information *utile*
- ▶ Plusieurs techniques optimales
- ▶ Des versions « deep learning » très performantes pour représenter les données

Utilisations

- ▶ Pré-traitement pour l'analyse de données (cf. cours précédents)
- ▶ Codage pour stockage et recherche de données par le contenu
- ▶ Débruitage

Catégorisation

Catégorisation

Définition

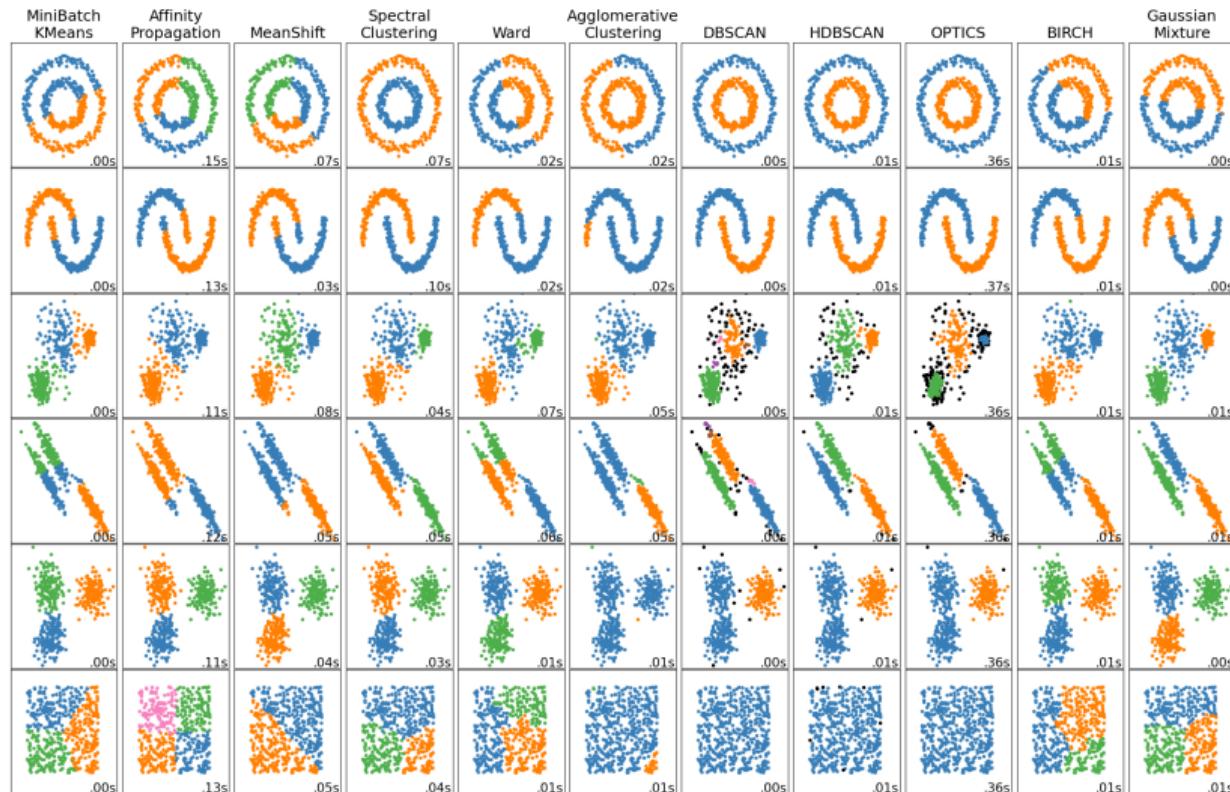
- ▶ Trouver des groupes d'objets, une structure dans la distribution des données
- ▶ Synonymes : partitionnement, *clustering*...
- ▶ ... classification **non-supervisée** : des données $\{x_i\}_{i=1}^n$ dans \mathbb{R}^d mais sans labels

Objectifs

Trouver les *groupes* de données dans \mathbb{R}^d selon des critères de

- ▶ proximité (métrique ou topologie)
- ▶ similarité (partage de caractéristiques)
- ▶ densité (fréquence d'occurrence)

Catégorisation



<https://scikit-learn.org>

Catégorisation : K-means

Objectifs

- ▶ Soit K le nombre de groupes cherchés.
- ▶ Un groupe (d'indice $j \in \{1 \cdots K\}$) = un ensemble de points.
- ▶ Soit $u_{ji} \in \{0, 1\}$ l'appartenance de chaque x_i au groupe j
- ▶ Soient $B = \{\beta_j | j \in \{1 \cdots K\}\}$ les prototypes qui caractérisent ces groupes.
- ▶ L'algorithme **K-means** cherche à minimiser :

$$J_{B,U}(X) = \sum_{j=1}^K \sum_{i=1}^n u_{ji} \|x_i - \beta_j\|_2^2$$

- ▶ « Nuées Dynamiques » en français

Catégorisation : K-means

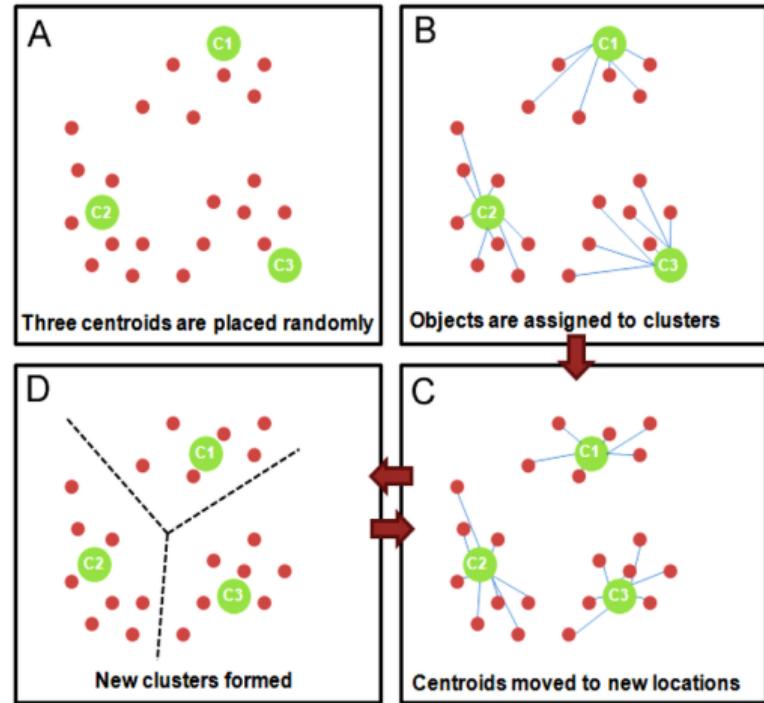
Algorithme

Initialiser les β_j , puis itérer :

1. Assigner chaque donnée x_i au plus proche β_j
2. Recalculer les prototypes selon la moyenne des données du groupe :

$$\beta_j = \frac{\sum_{i=1}^n u_{ji} * x_i}{\sum_{i=1}^n u_{ji}}$$

On recalcule les K moyennes de chaque groupe (cluster).



Catégorisation : K-means

Propriétés

- ▶ L'algorithme fait diminuer la fonction de coût $J_{B,U}(X)$ à chaque itération.
- ▶ Il y a un nombre fini de K partitions possibles, donc l'algorithme **converge**.
- ▶ Mais la solution peut ne pas être optimale (minimum local) !
 \Rightarrow importance de l'initialisation.
 Un remède : choisir β_j parmi les observations x_i .
- ▶ Hypothèse que les groupes sont compactes (i.e. ressemblent à des modèles gaussiens) : très dépendant de la forme des distributions

Catégorisation : K-means

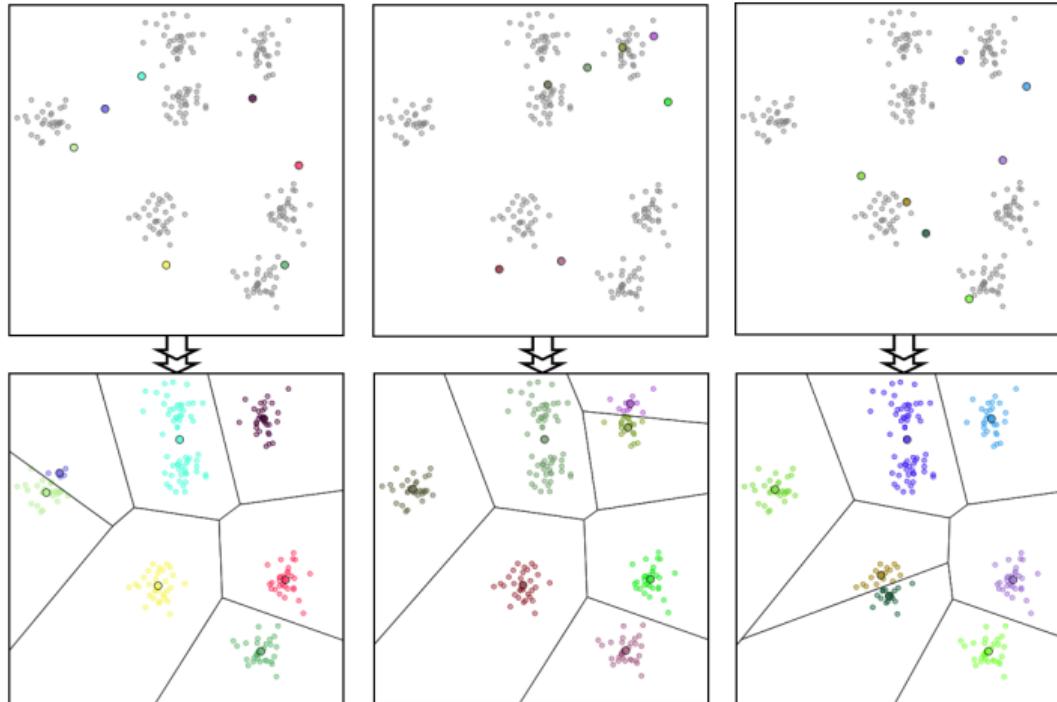


Figure 1 – Trois initialisations différentes

Catégorisation : K-means

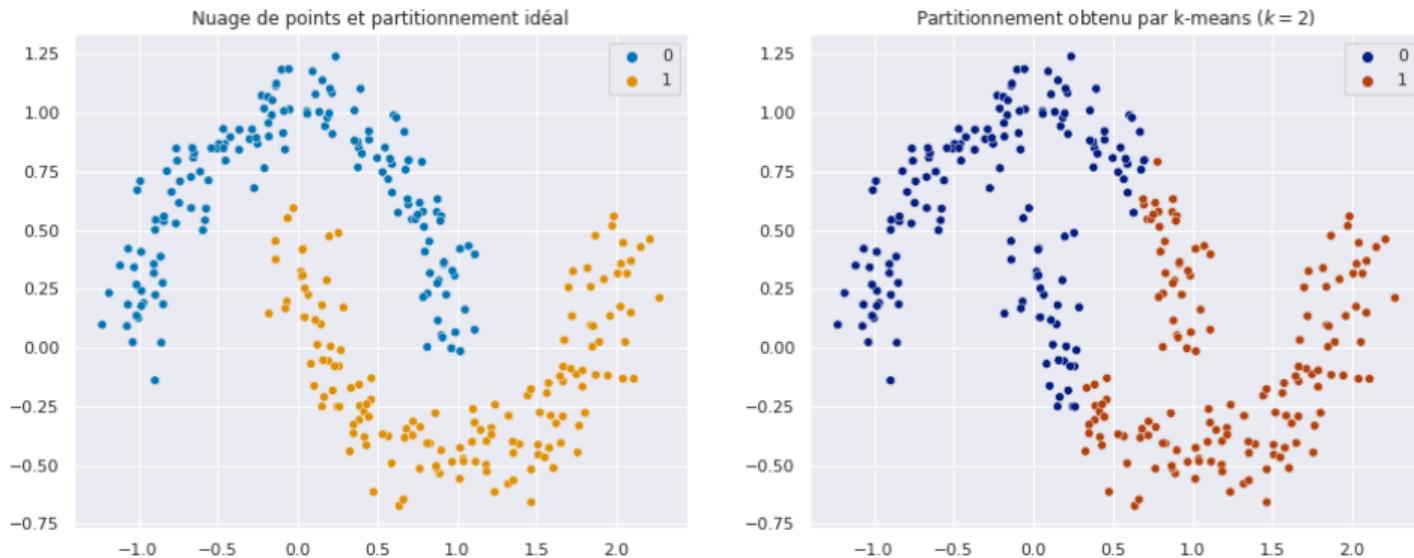
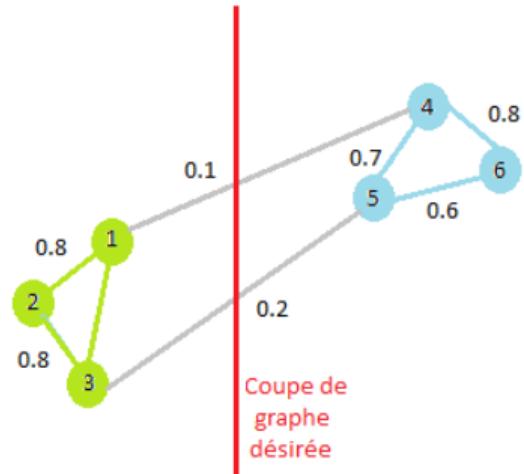


Figure 2 – Résultat de K-means sur demi-lunes

Catégorisation : méthodes alternatives I

Regroupement spectral : [von Luxburg, 2007] :

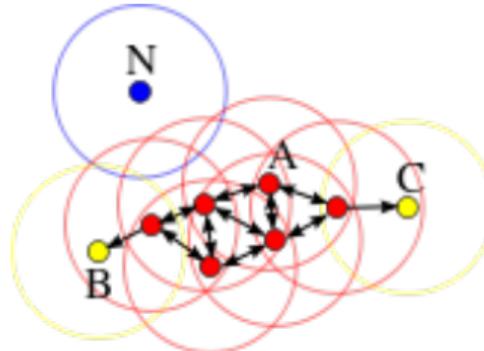
- ▶ Dépend de la matrice de similarité, éventuellement creuse, entre données : W où $w_{ij} \geq 0$ est une pondération de la relation entre données i et j
 - ▶ On définit la matrice diagonale des degrés D où $d_i = \sum_j w_{ij}$.
 - ▶ On transforme chaque donnée comme vecteurs propres associés aux plus *petites* valeurs propres de $L = D - W$ (matrice laplacienne).
 - ▶ On applique dans cet espace transformé un K-means.
- ⇒ *objets complexes, non vectoriels*



Catégorisation : méthodes alternatives II

DBSCAN : « Density-Based Spatial Clustering of Applications with Noise » [Ester et al., 1996]

- ▶ Se base sur des voisinages de taille ϵ (topologie) et un décompte minimal de points par voisinage $MinPts$ (densité)
- ▶ Parcours de proche en proche de tous les points d'un voisinage pour ajouter des points au groupe courant si localement denses.
- ▶ Arrêt de l'itération lorsque tous les points ont été visités.
- ▶ Deux paramètres à régler : ϵ et $MinPts$
 - ⇒ estime automatiquement le nombre catégories,
 - ⇒ gère les données aberrantes
 - ⇒ mieux adapté aux distributions complexes



<https://fr.wikipedia.org/wiki/DBSCAN>

Catégorisation : méthodes alternatives III

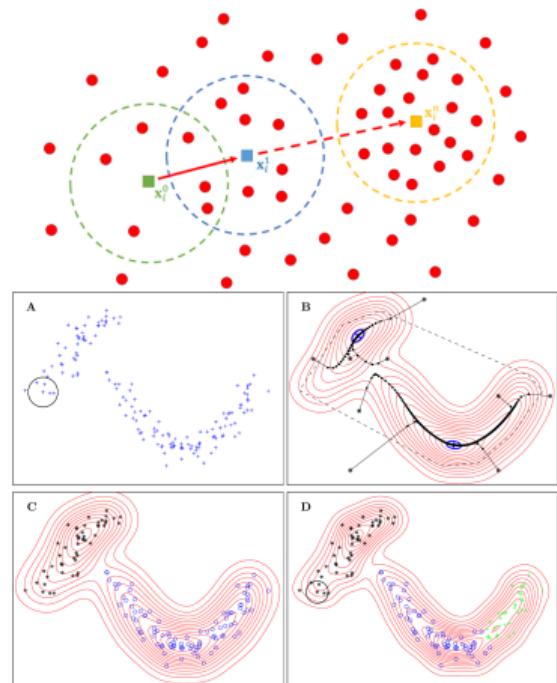
Mean Shift

- ▶ Algorithme itératif de détection des modes principaux d'une distribution
- ▶ Principe : construire une trajectoire pour chaque donnée convergeant vers un mode en partant d'un point :

$$x^{t+1} = m(x^t) = \frac{\sum_{i \in N(x^t)} x_i K(x^t, x_i)}{\sum_{i \in N(x^t)} K(x^t, x_i)}$$

où $K(x, x')$ est un noyau et $N(x)$ est l'ensemble des points dans le voisinage de x

- ▶ La catégorisation consiste à associer chaque donnée à un mode en suivant la trajectoire
- ⇒ pas de preuve de convergence



[Fukunaga and Hostetler, 1975,
Cheng, 1995,
Comaniciu et al., 2003]

Catégorisation : Résumé

Points clés du clustering

- ▶ Regrouper des données **non-labelisées** en catégories
- ▶ Notion de **distance** ou de **similarité** entre échantillons et de **densité**
- ▶ Un grand nombre d'algorithmes.

Utilisations

- ▶ Pré-traitement pour l'analyse de données : Bag-of-words, superpixels, etc.
- ▶ Classification **non-supervisée**

Visualisation

Réduction de dimension : Pour aller plus loin, t-SNE

Définition

- ▶ t-SNE = t-distributed stochastic neighbor embedding
- ▶ Approche **non-linéaire** de représentation de données, proposée par [Maaten and Hinton, 2008]

Objectifs

- ▶ Trouver une représentation des données en faible dimension qui reflète au mieux les **similarités** dans l'espace d'origine.

t-SNE

Formulation

Soient n observations (x_1, \dots, x_n) en grande dimension.

- ▶ On exprime les **similarités** entre observations comme une loi conditionnelle :
$$p[x_j|x_i] = p_{j|i} = \frac{\exp(-||x_i - x_j||^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-||x_i - x_k||^2/2\sigma_i^2)}$$
 et $p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$
- ▶ t-SNE cherche à construire une représentation d -dimensionnelle des données y_1, \dots, y_n (avec $y_i \in \mathbb{R}^d$) qui reflète au mieux les similarités p_{ij} .
- ▶ On transforme le problème en un apprentissage d'une association (« mapping ») $x_i \mapsto y_i$ qui minimise un écart entre distributions des similarités p_{ij} et q_{ij} dans l'espace projeté : $KL(P||Q)$.
- ▶ On modélise les similarités projetées par une distribution de Student :
$$q_{ij} = \frac{\sum_{k \neq j} 1 + ||y_k - y_j||^2}{1 + ||y_i - y_j||^2}$$

t-SNE

Algorithme

Soient n observations (x_1, \dots, x_n) en grande dimension.

1. Choix du paramètres du noyau de similarité : σ (« perplexité »)
~~ permet de révéler différents niveaux de structures dans les données
2. Calcul des probabilités de **similarité** des observations : p_{ij}
3. Minimisation de la divergence de Kullback-Leibler de Q par rapport à P :
$$KL(P||Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$
 pour trouver les y_i par **descente de gradient**
~~ sensibilité à l'initialisation des y_i

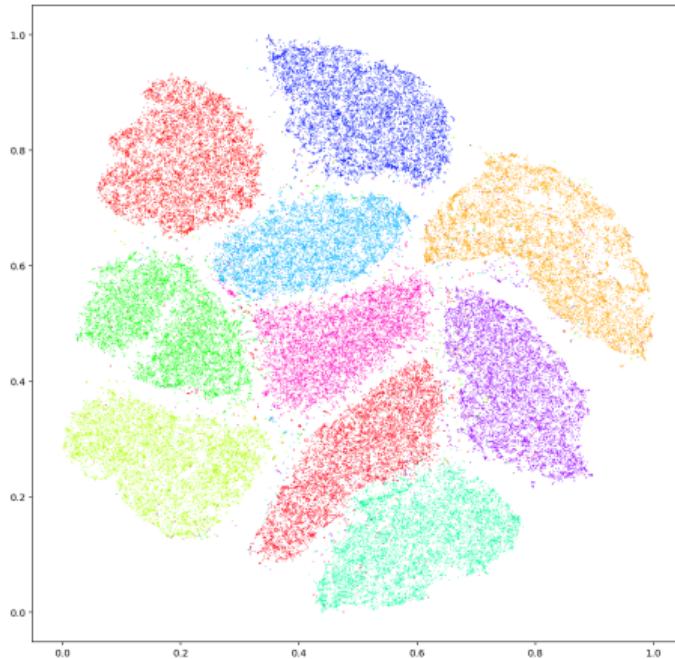
t-SNE

t-SNE sur base d'images

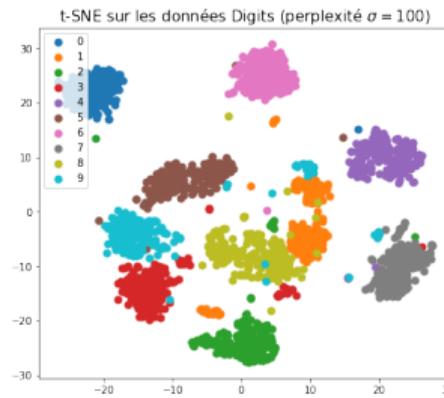
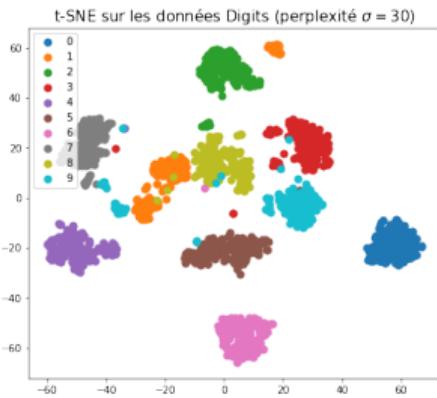
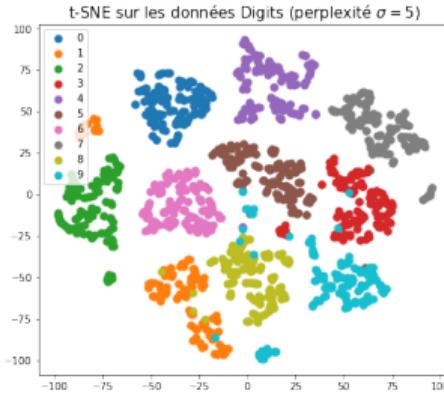
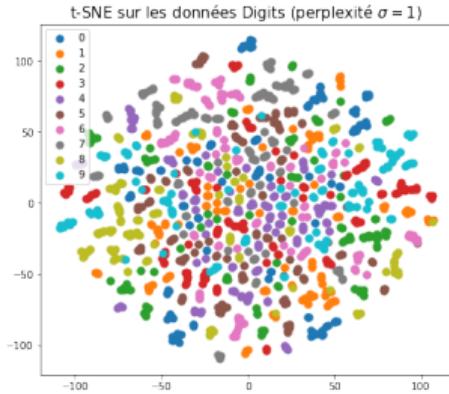


Van Maaten, <https://lvdmaaten.github.io/t-sne/>

t-SNE sur MNIST



t-SNE : impact de la perplexité σ



Variantes

D'autres approches qui calculent une représentation par minimisation globale sur la distribution de points :

- ▶ Multi-Dimensional Scaling [Kruskal, 1964] :

Trouve les points y_i qui minimisent $\sum_{i < j} (\|y_i - y_j\| - d_{ij})^2$

- ▶ Locally-Linear Embedding (LLE) [Roweis and Saul, 2000] :

calcule W minimisant $\sum_i \|x_i - \sum_{j \in N(x_i)} w_{ij}x_j\|^2$

puis calcule Y minimisant $\sum_i \|y_i - \sum_j w_{ij}y_j\|^2$

- ▶ Uniform Manifold Approximation and Projection (UMAP) [McInnes et al., 2018] : même principe que t-SNE mais avec d'autres mesures de similarité et d'autres distributions cibles q_{ij} .

Points clés

- ▶ Approche **non-linéaire** de visualisation de données
- ▶ Principe : accorder les **distributions des similarités** entre observations et projections
- ▶ Contraintes de représentation dans l'espace projeté (distribution de Student pour t-SNE)
- ▶ Optimisation par descente de gradient

Utilisations

- ▶ Visualisation
- ▶ Analyse de la structure globale des données
- ▶ **ATTENTION** : ce n'est pas une projection, juste une représentation globale des données qui respecte les proximités locales

Références |

Bertand Le Saux (<https://blesaux.github.io/teaching/I0GS-machine-learning>)

Nicolas Audebert (<https://cedric.cnam.fr/vertigo/Cours/ml/>)

[Assran et al., 2022] Assran, M., Caron, M., Misra, I., Bojanowski, P., Bordes, F., Vincent, P., Joulin, A., Rabbat, M., and Ballas, N. (2022).

Masked Siamese Networks for Label-Efficient Learning.

In Avidan, S., Brostow, G., Cissé, M., Farinella, G. M., and Hassner, T., editors, *Computer Vision – ECCV 2022*, Lecture Notes in Computer Science, pages 456–473, Cham. Springer Nature Switzerland.

[Caron et al., 2020] Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. (2020).

Unsupervised Learning of Visual Features by Contrasting Cluster Assignments.

In *Advances in Neural Information Processing Systems*, volume 33, pages 9912–9924. Curran Associates, Inc.

[Cheng, 1995] Cheng, Y. (1995).

Mean shift, mode seeking, and clustering.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 17(8) :790–799.

Conference Name : IEEE Transactions on Pattern Analysis and Machine Intelligence.

[Comaniciu et al., 2003] Comaniciu, D., Ramesh, V., and Meer, P. (2003).

Kernel-based object tracking.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 25(5) :564–577.

Conference Name : IEEE Transactions on Pattern Analysis and Machine Intelligence.

[Cunningham and Ghahramani, 2015] Cunningham, J. P. and Ghahramani, Z. (2015).

Linear Dimensionality Reduction : Survey, Insights, and Generalizations.

Journal of Machine Learning Research, 16(89) :2859–2900.

Références II

- [Ester et al., 1996] Ester, M., Kriegel, H.-P., Sander, J., Xu, X., and others (1996).
A density-based algorithm for discovering clusters in large spatial databases with noise.
In *kdd*, volume 96, pages 226–231.
Number : 34.
- [Fukunaga and Hostetler, 1975] Fukunaga, K. and Hostetler, L. (1975).
The estimation of the gradient of a density function, with applications in pattern recognition.
IEEE Transactions on Information Theory, 21(1) :32–40.
Conference Name : IEEE Transactions on Information Theory.
- [Gidaris et al., 2018] Gidaris, S., Singh, P., and Komodakis, N. (2018).
Unsupervised Representation Learning by Predicting Image Rotations.
- [Kruskal, 1964] Kruskal, J. B. (1964).
Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis.
Psychometrika, 29(1) :1–27.
- [Maaten and Hinton, 2008] Maaten, L. v. d. and Hinton, G. (2008).
Visualizing Data using t-SNE.
Journal of Machine Learning Research, 9(86) :2579–2605.
- [Mairal, 2015] Mairal, J. (2015).
Incremental majorization-minimization optimization with application to large-scale machine learning.
SIAM Journal on Optimization, 25(2) :829–855.
Publisher : SIAM.
- [McInnes et al., 2018] McInnes, L., Healy, J., Saul, N., and Großberger, L. (2018).
UMAP : Uniform Manifold Approximation and Projection.
Journal of Open Source Software, 3(29) :861.

Références III

[Roweis and Saul, 2000] Roweis, S. T. and Saul, L. K. (2000).

Nonlinear Dimensionality Reduction by Locally Linear Embedding.

Science, 290(5500) :2323–2326.

Publisher : American Association for the Advancement of Science.

[Schölkopf et al., 1998] Schölkopf, B., Smola, A., and Müller, K.-R. (1998).

Nonlinear Component Analysis as a Kernel Eigenvalue Problem.

Neural Computation, 10(5) :1299–1319.

[von Luxburg, 2007] von Luxburg, U. (2007).

A tutorial on spectral clustering.

Statistics and Computing, 17(4) :395–416.