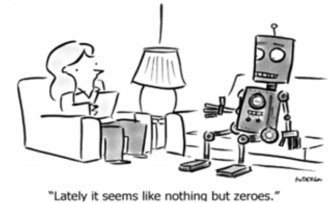
Unsupervised Learning

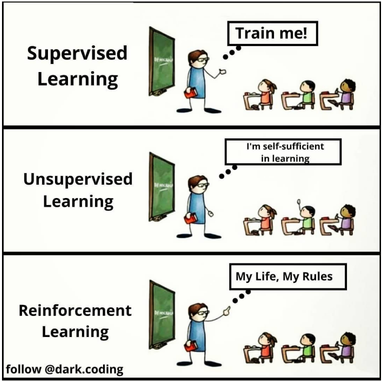
Gianni FRANCHI
ENSTA-Paris



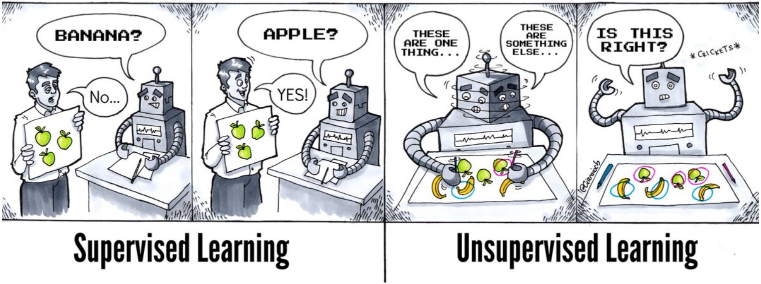
ENSTA
IP PARIS

Lecture outline

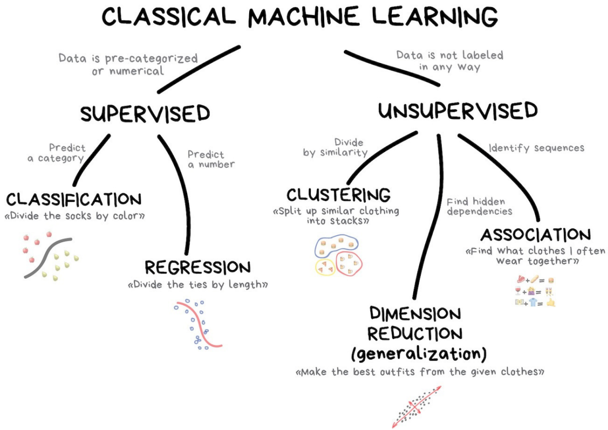
Machine learning



Machine learning



Machine learning

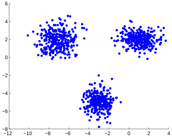


Content and Goals of the lecture

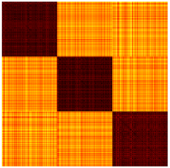
- Explain the interest of Unsupervised learning
- Introduce Dimensionality reduction via Principal Component Analysis
- Introduce Dimensionality reduction via Kernel Principal Component Analysis
- Introduce clustering methods
- Introduce Neural network and Unsupervised learning

Introduction to curse of dimensionality

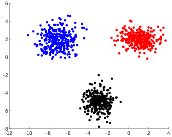
The curse of dimensionality refers to various phenomena that arise when analyzing and organizing data in high-dimensional spaces. Suppose that we have 900 data $v \in [0,1]^D$, where D is the dimension of the data space. Consider first a simple case where $D = 2$



(a)



(b)

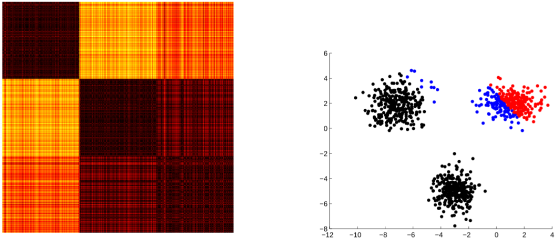


(c)

Figure: A set of 900 data of dimension 2. In (a) the data, in (b) their Gram matrix, in (c) the 3 clusters of the data.

Introduction to curse of dimensionality

Consider more comple case where $D = 10$



(a) (b)

Figure: A set of 900 data of dimension 100. In (a) the Gram matrix of the data. As we can see it is difficult to separate some classes. In (b) the 3 clusters of the data, the clusters are not perfect because of the curse of dimnensionality.

Introduction to curse of dimensionality

Moreover we can see that the ratio between the maximum euclidean distance and the minimum euclidean distance $R = \frac{\max_{(i,j)}\{\|v_i - v_j\|_2\}}{\min_{(i,j)} \|v_i - v_j\|_2\}}$ of the data tends to 1 when the D increases.

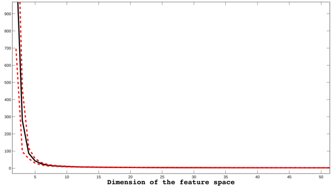


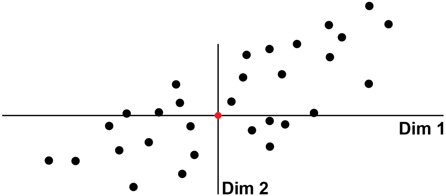
Figure: In this figure we have selected randomly 500 data $v_i \in [0,1]^D$ where D is the dimension of the feature space. We represent $R = \frac{\max_{(i,j)}\{\|v_i - v_j\|_2\}}{\min_{(i,j)} \|v_i - v_j\|_2\}}$ which represents the power of discrimination of the distance.

Introduction to Principal Component Analysis (PCA) [Jolliffe1986]

We start with a set of n points $F = \{v_i\}_{i=1}^n \in \mathbb{R}^D$. The PCA goal is to reduce the dimension of this vector space finding the basis that captures most of the variance of data set thanks to a projection on the principal component space, namely

$$F = \{v_i\}_{i=1}^n \longrightarrow F' = \{v_i'\}_{i=1}^n \quad (1)$$

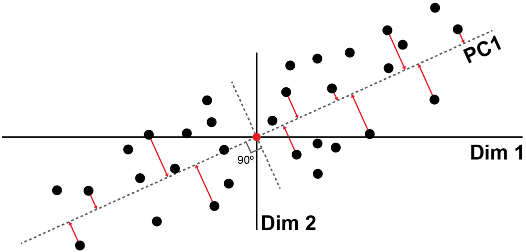
with $v_i' \in \mathbb{R}^d$, where $d \ll D$.



Introduction to Principal Component Analysis (PCA)[Jolliffe1986]

Let us call $w_j \in \mathbb{R}^D$ the j principal component. The aim of PCA is to find the set of vectors$\{w_j, 1 \leq j \leq D\}$ such as:

$$\arg\min_{w_j} \left[n^{-1} \sum_{i=1}^{n} \| v_i - < v_i, w_j > \frac{w_j}{\|w_j\|} \|^2 \right], \quad \forall 1 \leq j \leq D. \qquad (2)$$



Introduction to Principal Component Analysis (PCA)[Jolliffe1986]

we want to minimize:

$$\arg\min_{w_j} \left[n^{-1} \sum_{i=1}^{n} \| v_i - < v_i, w_j > \frac{w_j}{\|w_j\|} \|^2 \right], \quad \forall 1 \leq j \leq D. \qquad (3)$$

Developing now the distance we have: $\| v_i - < v_i, w_j > \frac{w_j}{\|w_j\|} \|^2 = 1 - 2\frac{< v_i, w_j >^2}{\|w_j\|} + < v_i, w_j >^2$, by adding the additional constraint that $\|w_j\|^2 = 1$, and replacing in (3) and keeping only terms that depend on w_j, we have the following new objective function:

$$\arg\max_{w_j, \|w_j\|^2 = 1} \; n^{-1} \sum_{i=1}^{n} < v_i, w_j >^2, \quad \forall 1 \leq j \leq D. \qquad (4)$$

Introduction to Principal Component Analysis (PCA)**[Jolliffe1986]**

Since :

$$\mathrm{var}(< v_i, w_j >) = n^{-1}\sum_{i=1}^{n}(< v_i, w_j >)^2 - (n^{-1}\sum_{i}^{n}(< v_i, w_j >))^2,$$

if we consider that the data F has been column-centered, which means that $\sum_{i=1}^{n} v_i = 0$, then :

$$\mathrm{var}(< v_i, w_j >) = n^{-1}\sum_{i=1}^{n}(< v_i, w_j >)^2.$$

Thus we can see that the goal of the PCA is to find principal components that maximize the variance.

Introduction to Principal Component Analysis (PCA)**[Jolliffe1986]**

The problem can be rewritten in a matrix way:

$$
\begin{aligned}
n^{-1} \sum_{i=1}^{n} < v_i, w_j >^2 &= n^{-1} (F w_j)^T (F w_j) \\
&= w_j^T (n^{-1} (F^T F)) w_j = w_j^T V w_j,
\end{aligned}
$$

where $V = n^{-1}(F^T F)$, $V \in M_{D,D}(\mathbb{R})$, is the covariance of F. Hence we should optimize:

$$
\underset{w_j, \|w_j\|^2 = 1}{\arg\max} \; w_j^T V w_j, \quad \forall 1 \leq j \leq D. \tag{5}
$$

Introduction to Principal Component Analysis (PCA)[Jolliffe1986]

So we want to maximize: $\arg\max_{w_j, \|w_j\|^2=1} w_j^T V w_j$ subject to the constraint $\|w_j\|^2 = 1$.

How can we solve that?

Lagrange multiplier

Definition(Local extremum under constraint)

Let f and g be two functions of two variables. Let $P_0 = (x_0, y_0)$ a point belonging to the domain definition of f denoted D_f and domain definition of g denoted D_g checking $g(x0, y0) = 0$. P_0 is a local maximum (resp. Local minimum) of f on $D = \{(x, y)|g(x, y) = 0\}$ if there is a neighboorhood V of P_0 such that for all (x, y) of V satisfying $g(x, y) = 0$, $f(x, y) \leq f(x0, y0)$ (resp. $f(x, y) \geq f(x0, y0)$).

Lagrange multiplier

Theorem(necessary condition of Local extremum)

Let f and g be two functions of two variables of C^1 (that is, having continuous first derivatives). Let $P_0 = (x_0, y_0)$ a point belonging to D_f and D_g checking $g(x0, y0) = 0$. If P_0 is a local extrama of f on $D = \{(x, y)|g(x, y) = 0\}$ and $\nabla g(x0, y0) \neq 0$ then $\nabla f(x0, y0)$ and $\nabla g(x0, y0)$ are aligned . That is to say: there exists a scalar $\lambda_0 \in \mathbb{R}$ such that

$$\nabla f(x0, y0) = \lambda_0 \nabla g(x0, y0)$$

P_0 is called a stationary point of $f$on D and λ_0 is called associated Lagrange multiplier.

Lagrange multiplier

Lagrange multipliers is used to find local maxima and minima of a function subject to equality constraints

Proposition Lagrange multiplier

P_0 is a local extrama of f on $D = \{(x,y)|g(x,y)=0\}$ associated with the Lagrange multiplier $\lambda_0 \in \mathbb{R}$ if and only if $(x0, y0, \lambda_0)$ is solution of :

$$\begin{cases} \frac{\partial \mathcal{L}}{\partial x}(x,y,\lambda) = 0 \\ \frac{\partial \mathcal{L}}{\partial y}(x,y,\lambda) = 0 \\ \frac{\partial \mathcal{L}}{\partial \lambda}(x,y,\lambda) = 0 \end{cases} \tag{6}$$

with $\mathcal{L} = f + \lambda g$

Introduction to Principal Component Analysis (PCA)**[Jolliffe1986]**

So we want to maximize: $\arg\max_{w_j, \|w_j\|^2=1} w_j^T V w_j$ subject to the constraint $\|w_j\|^2 = 1$.
Thanks to **Lagrange multiplier proposition** we can rewrite the objective function as:

$$\mathcal{L}(w_j, \lambda) = w_j^T V w_j - \lambda(w_j^T w_j - 1), \tag{7}$$

where $\lambda \in \mathbb{R}$. Since we want to maximize this function, we have to derive it and equal it to zero:

Introduction to Principal Component Analysis (PCA)

$$\frac{\partial \mathcal{L}}{\partial w_j}(w_j, \lambda) = 2Vw_j - 2\lambda w_j = 0.$$

So, we finally obtain as solution

$$Vw_j = \lambda w_j. \tag{8}$$

Thus, the principal component w_j that satisfies the objective function is an eigenvector of the covariance matrix V, and the one maximizing $\mathcal{L}(w_j, \lambda)$ is the one with the larger eigenvalue. Then we can have all the w_j by computing the SVD of V.

Covariance matrix

We have $F = \{v_i\}_{i=1}^{n} \in \mathbb{R}^D$. For $i \in [1, n]$ and $j \in [1, D]$ let us write $v_{i,j}$ the $j-$th coefficient of v_i. The empirical covariance matrix is

$$V = \begin{pmatrix} \mathrm{Var}(v_{.,1}) & \mathrm{Covar}(v_{.,1}, v_{.,2}) & \dots & \mathrm{Covar}(v_{.,1}, v_{.,D}) \\ \mathrm{Covar}(v_{.,2}, v_{.,1}) & \mathrm{Var}(v_{.,2}) & \dots & \mathrm{Covar}(v_{.,2}, v_{.,D}) \\ \vdots & \ddots & \vdots & \vdots \\ \mathrm{Covar}(v_{.,D-1}, v_{.,1}) & \dots & \mathrm{Var}(v_{.,D-1}) & \mathrm{Covar}(v_{.,D-1}, v_{.,D}) \\ \mathrm{Covar}(v_{.,D}, v_{.,1}) & \dots & \mathrm{Covar}(v_{.,D}, v_{.,D-1}) & \mathrm{Var}(v_{.,D}) \end{pmatrix}$$

with

$$\mathrm{Var}(v_{.,j}) = (1/n) \sum_{i}^{n} v_{i,j}^2 - \left((1/n) \sum_{i}^{n} v_{i,j} \right)^2$$

with

$$\mathrm{Covar}(v_{.,j_1}, v_{.,j_2}) = (1/n) \sum_{i}^{n} v_{i,j_1} v_{i,j_2} - \left((1/n) \sum_{i}^{n} v_{i,j_1} \right) \left((1/n) \sum_{i}^{n} v_{i,j_2} \right)$$

Introduction to Principal Component Analysis (PCA)

There are different approaches to choose the reduced dimension d. One technique consists of evaluating the proportion of the original variance kept

$$\text{Prop} = \sum_{j=1}^{d} \lambda_j / \sum_{j=1}^{D} \lambda_j$$

We will write W_d the square matrix of size D containing the d eigenvectors corresponding of the higher eigenvalues, and all the other columns are null. Then thanks to the Eckart-Young theorem **[Eckart1936]** it is possible to quantify the error of reduction of dimension such as :

$$Err_{\text{PCA}} = \|V - W_d^T V W_d\|_F^2 = \sum_{j=d+1}^{D} \lambda_j^2 \quad (9)$$

PCA Algorithm

Init. Start with initial data $F = \{v_i\}_{i=1}^{n} \in \mathbb{R}^D$.

PCA Evaluation

1. Calculate the covariance of F we call it V
2. Evaluate the SVD of V, we call $\{w_j\}_{j=1}^{D}$ the set eigenvectors and $\{\lambda_j\}_{j=1}^{D}$ the set eigenvalues.
3. Order the eigenvalues, eigenvectors in the descending order.
4. Take the d first eigenvectors such that Prop reaches your criterion
5. Project the data in your new basis.

Kernel trick [Smola1998]

Definition

By definition a kernel is a function $\mathcal{K} : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ which is symmetric and hermitian.

However most of the time we work with positive definite kernel kernel.

Definition

$\mathcal{K}$ is called a positive definite kernel if $\forall \{x_1, \ldots, x_n\} \in \mathcal{X}^n$ and $\forall \{\alpha_1, \ldots, \alpha_n\} \in \mathbb{R}^n$, the following non-negativity condition holds: $\sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j^* \mathcal{K}(x_i, x_j) \geq 0$.

Kernel trick [Smola1998]

Definition

A Hilbert space $\mathcal{H}$ is a vector space with a real or complex inner product space that is also a complete metric space with respect to the distance function induced by the inner product, that means that every Cauchy sequence in $\mathcal{H}$ a limit in $\mathcal{H}$.

Moore-Aronszajn Theorem

$\mathcal{K}$ is a positive definite kernel if and only if there exists a Hilbert space $\mathcal{H}$ and a mapping $\phi : \mathcal{X} \to \mathcal{H}$ such that
$\mathcal{K}(x_i, x_j) =< \phi(x_i), \phi(x_j) >_{\mathcal{H}}$

Kernel trick **[Smola1998]**

Kernel trick: Representer Theorem

Let $\mathcal{X}$ be a set endowed with a positive definite kernel $\mathcal{K}$, and $\mathcal{H}_\mathcal{K}$ the corresponding RKHS, and $x_1, \ldots, x_n \subset \mathcal{X}$ a finite set of points. Let $\Psi : \mathbb{R}^{n+1} \to \mathbb{R}$ be a function of n + 1 variables, strictly increasing with respect to the last variable. Then, any solution to the optimization problem:

$$\min_{g \in \mathcal{H}_\mathcal{K}} \Psi(g(x_1), \ldots, g(x_n), \|g\|_{\mathcal{H}_\mathcal{K}}), \tag{10}$$

admits a representation of the form:$\forall x \in \mathcal{X}$,
$g(x) = \sum_{i=1}^{n} \alpha_i \mathcal{K}(x_i, x)$ where $\|g\|_{\mathcal{H}_\mathcal{K}} = \sqrt{< g, g >_{\mathcal{H}_\mathcal{K}}}$.

Kernel PCA **[Smola1998]**

Let us consider a set vector $v_i \in \mathbb{R}^D$ $\forall i \in [1, n]$ where n represents the number of vectors. Let us map our data into another space $\mathcal{H}$, that may have some interesting properties :

$$\phi = \begin{cases} \mathbb{R}^D \rightarrow \mathcal{H} \\ v_i \rightarrow \phi(v_i) \end{cases} \tag{11}$$

Kernel PCA

The goal of the kernel PCA (KPCA) is to find the set $\{w_j, j \in [1, D]\}$ that minimize the quantity :

$$\min(\frac{1}{n} \times \sum_{i}^{n} \|\phi(v_i) - < \phi(v_i), w_j >_{\mathcal{H}_\mathcal{K}} \cdot \frac{w_j}{\|w_j\|^2_{\mathcal{H}_\mathcal{K}}} \|^2_{\mathcal{H}_\mathcal{K}}) \ \forall j \in [1, P] \quad (12)$$

Kernel PCA

By doing the same calculus as on the PCA we have:

$$\mathcal{L}(w_j, \lambda) = \frac{1}{n} \times \sum_{i}^{n} < \phi(v_i), w_j >^2_{\mathcal{H}_\mathcal{K}} - \lambda.(\|w_j\|^2_{\mathcal{H}_\mathcal{K}} - 1) \quad (13)$$

where $\lambda \in \mathbb{R}$. Thanks to the Representer Theorem w_j can be written as:

$$w_j = \sum_{l=1}^{n} \alpha_{l,j} \phi(v_l) \quad (14)$$

$$\mathcal{L}(\alpha_j, \lambda) = \frac{1}{n} \times \sum_{i}^{n} (\sum_{l=1}^{n} \alpha_{l,j} < \phi(v_i), \phi(v_l) >_{\mathcal{H}_\mathcal{K}})^2 - \lambda. \sum_{(k,l) \in [1,n]^2} \alpha_{l,j} \alpha_{k,j} \mathcal{K}(v_l, v_k) - 1)$$

Kernel PCA [Smola1998]

The problem can be rewrite in a matrix way by:

$$\mathrm{L}(\alpha_j, \lambda) = \frac{1}{n}\alpha_j^t \times \mathcal{K}^2 \times \alpha_j - \lambda.(\alpha_j^t \times \mathcal{K} \times \alpha_j - 1) \tag{15}$$

with $\alpha_j \in \mathbb{R}^D$

Importance of the Kernel choice

The advantage of the kernel trick is that we can use different kernels without having to compute explicitly the mapping $\phi(v_i)$. Thanks to that, we can use a huge variety of kernels. The most popular kernels are:

- The polynomial kernel : $\mathcal{K}(v_i, v_j) = (< v_i, v_j >_{\mathbb{R}^D} + c)^P$, where P is the degree of the kernel and c is a constant;
- The rbf kernel or gaussian kernel : $\mathcal{K}(v_i, v_j) = e^{\frac{-\|v_i - v_j\|^2_{\mathbb{R}^D}}{2\sigma^2}}$, with parameter σ. This kernel bring the data in a space of infinite dimension.

Importance of the Kernel choice

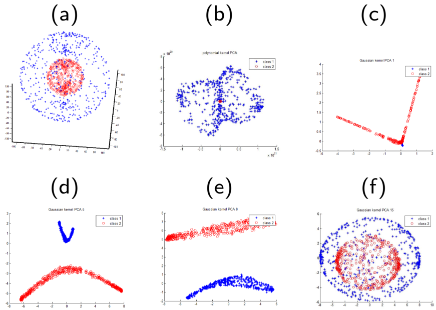


Figure: (a) Two concentric spheres synthetic manifold, (b) Polynomial KPCA with $p = 5$, (c) Gaussian KPCA with σ,(d) Gaussian KPCA with $5.\sigma$,(e) Gaussian KPCA with $8.\sigma$, (f) Gaussian KPCA with $15.\sigma$.

Importance of the Kernel choice

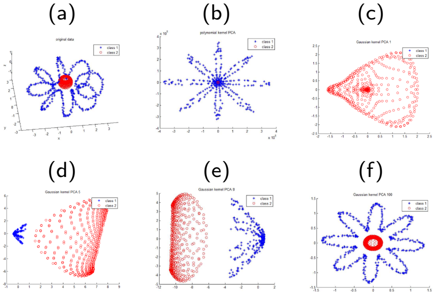
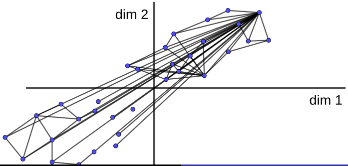


Figure: (a) The flower synthetic manifold, (b) Polynomial KPCA with $p = 5$, (c) Gaussian KPCA with σ,(d) Gaussian KPCA with $5.\sigma$,(e) Gaussian KPCA with $8.\sigma$, (f) Gaussian KPCA with $100.\sigma$.

Multi Dimensional Scaling MDS [Cox2008]

Multidimensional scaling is an data mining technique used to decrease the dimensionality of the data by retaining the pairwise distance between the data so : $\mathcal{F} = \{v_i\}_{i=1}^n \longrightarrow \mathcal{F}' = \{v_i'\}_{i=1}^n$, with $\|v_i - v_j\| \simeq \|v_i' - v_j'\| \; \forall i,j \in [1,n]^2$, where $\|v_i - v_j\|$ represents the euclidean distance between v_i and v_j. So the **main objective function** is:

$$\Phi(\mathcal{F}') = \sum_{i,j \in [1,n]^2} (\|v_i' - v_j'\|_2^2 - \|v_i - v_j\|_2^2) \tag{16}$$



Other classical dimensionality reduction

- Independent component analysis (ICA) **[Comon1994]**
- Factor Analysis **[Harman1976]**
- Local linear embeddings (LLE) **[Chenping2009]**
- t-distributed stochastic neighbor embedding (TSNE) **[Laurens2008]**

t-distributed Stochastic Neighbor Embedding (TSNE) [Maaten2008]

t-SNE is an unsupervised machine learning algorithm for visualizing high-dimensional data by projecting each point into a two/three-dimensional map. This method can find non-linear connections contrary to PCA. It relies on **three steps**:

- We calculate the similarities of points in the initial large-dimensional space.
- We create a smaller dimensional space in which we will represent our data.
- We optimize the mapping of points on the lower dimension space.

TSNE - step 1 [Maaten2008]

Given a set of n high-dimensional objects $v_i \in \mathbb{R}^D \ \forall i \in [1, n]$, the first step computes probabilities p_{ij} that are proportional to the similarity of objects v_i and v_j.
For $i \neq j$, they define

$$p_{j|i} = \frac{\exp(-\|\mathrm{v}_i - \mathrm{v}_j\|^2/2\sigma^2)}{\sum_{k \neq i} \exp(-\|\mathrm{v}_i - \mathrm{v}_k\|^2/2\sigma^2)}$$

Note that $p_{ij} = \frac{p_{i|j} + p_{j|i}}{2n}$. The bandwidth of the Gaussian kernels σ_i is called the perplexity and it is a parameter.

TSNE - step 2 **[Maaten2008]**

The goal of t-SNE is to learn a mapping $v_i' \in \mathbb{R}^d \ \forall i \in [1, n]$ that reflects the similarities p_{ij} as well as possible. Usually, d is set to 2 or 3 if we want to use the dimension reduction for visualization. Similarly to step 1, we calculate the similarities q_{ij} of the points in the newly created space by using a t-Student distribution instead of a Gaussian one. In the same way, we obtain a list of similarities q_{ij} :

$$q_{ij} = \frac{(1 + \|\mathrm{v}_i' - \mathrm{v}_j'\|^2)^{-1}}{\sum_k \sum_{l \neq k} (1 + \|\mathrm{v}_k' - \mathrm{v}_l'\|^2)^{-1}}$$

TSNE - step 3 **[Maaten2008]**

We use the Kullback-Leiber divergence to make the joint probability distribution of the new data points v'_i in the low dimension as similar as possible to the one from the original dataset. Hence, we minimize the Kullback Leibler divergence between the distributions P and Q:

$$\mathrm{KL}\left(P \parallel Q\right) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

The minimization of the Kullback Leibler divergence is done thanks to the gradient descent algorithm.

Autoencoder [Hinton2006]

Autoencoder is a neural network designed to learn an identity function in an unsupervised way to reconstruct the original input while compressing the data in the process

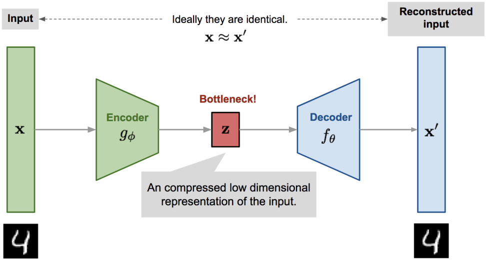


Figure: A simple autoencoder [1].

[1] https://lilianweng.github.io/

Autoencoder [Hinton2006]

- Let us consider $\mathcal{F} = \{v_i\}_{i=1}^n \longrightarrow \mathcal{F}' = \{v_i'\}_{i=1}^n$
- Let us write g_ϕ the encoder DNN. ϕ represents the weights of the DNN.
- Let us write f_θ the decoder DNN. θ represents the weights of the DNN.
- $v_i' = f_\theta(g_\phi(v_i))$

There are various metrics to quantify the difference between two vectors, such as cross entropy when the activation function is sigmoid, or as simple as MSE loss:

$$\mathcal{L}(\phi, \theta) = 1/n \sum_i^n \| v_i - f_\theta(g_\phi(v_i)) \|^2 \tag{17}$$

Robust Autoencoder [Vincent2008]

Since the autoencoder might be facing the risk of "overfitting" when there are more network parameters than the number of data. A solution : corrupt partially the input (adding noises or random masking of input values).

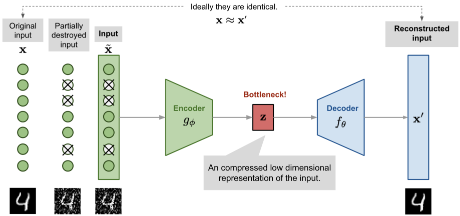


Figure: A Robust autoencoder [2].

[2]https://lilianweng.github.io/

Clustering

- Let us consider $\mathcal{F} = \{v_i\}_{i=1}^{n}$
- We consider that there is a set of C distributions P_k with $k \in [1, K]$
- We consider that all the v_i $i \in [1, n]$ are a realisation or of one of the P_k with $k \in [1, K]$
- we don't have information on K on the general case and on the P_k.

Our goal:

1. Identify the number of clusters. (At least have a number of cluster that make sense)
2. Gather the data of F into clusters without having any information.

K-means [Kanungo2002]

Let us assume we have choosen a value for K.
Let us build a new variable z_i with $i \in [1, n]$, that assign to each v_i a cluster.

$$\forall i \in [1, n] \; z_i = k \text{ if we assign } v_i \text{ to the class k.} \tag{18}$$

The objective in K-means can be written as follows:

$$\mathcal{L}(z, \mu) = \underset{z,\mu}{\arg\min} \|v_i - \mu_{z_i}\|^2 \text{ with } \mu_k = \frac{1}{|C_k|} \sum_{i \in C_k} v_i \tag{19}$$

with $C_k = \{v_i, \forall i \in [1, n] \mid z_i = k\}$.

K-means Algorithm **[Kanungo2002]**

Init. F (n nb of variables), K nb of clusters
Initialize each centroid with random values

Repeat (For a given number of iterations)

1. Assignment. Assign each observation to the group with the closest centroid
2. Update. Recalculate centroids from individuals attached to the groups
3. Evaluate if the loss has reached a threshold value.

K-means [Kanungo2002]

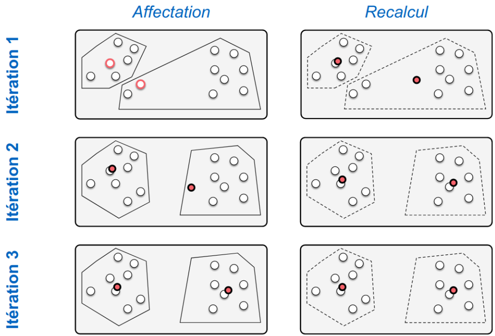


Figure: Example K-means [3].

[3]https://www.irit.fr/ Yoann.Pitarch/

K-means [Kanungo2002]

Advantages:

1. Scalability: Ability to process very large dataset. Only the centroids coordinates must be stored in memory.
2. Easy to understand and interpret

Disadvantages

1. The computing time may be high because we process many times each individual.
2. There is no guarantee that the algorithm reaches the global optimum of the loss.
3. The solution depends on the initial values of the centroids.

K-means issues [Kanungo2002]

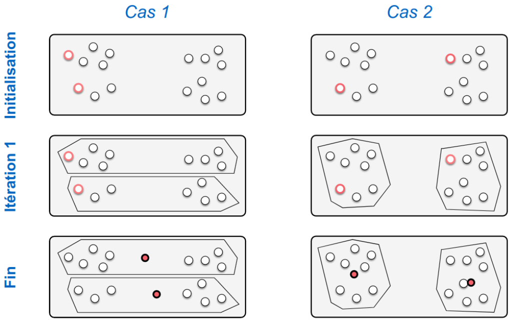


Figure: Example of a bad initialization [4].

[4]https://www.irit.fr/ Yoann.Pitarch/

K-means and dissimalarities [Kanungo2002]

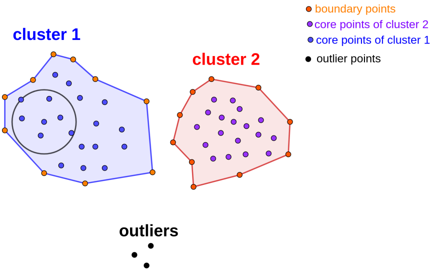
We have illustrated the K-means with the Euclidean distance, yet other dissimilarity measures can be used:

1. Cosine distance: It determines the cosine of the angle between the point vectors of the two points in the n dimensional space
$$d(x,y) = \frac{x.y}{\|x\| * \|y\|}$$
2. Manhattan distance: It computes the sum of the absolute differences between the co-ordinates of the two data points.
$$d(x,y) = \sum_n |x_i - y_i|$$
3. Minkowski distance: It is also known as the generalised distance metric. It can be used for both ordinal and quantitative variables.
$$d(x,y) = (\sum_n |x_i - y_i|^{\frac{1}{p}})^p$$

DBSCAN [Martin1996]

Basic idea

- Clusters are dense regions in the data space, separated by regions of lower object density
- A cluster is defined as a maximal set of density-connected points
- Discovers clusters of arbitrary shape and number of cluster



Neighborhood

To measure the density of a point we need to define the Neighborhood.

Definition

The $\epsilon-$ Neighborhood of a point v for a given distance d, is the composed of all the point within a radius ϵ from v. Hence we can write this set:

$$N_\epsilon(v) = \{x, d(x, v) \leq \epsilon\}$$

Core points

Given ϵ and an integer MinPts, DBSCAN categorizes the points into three exclusive categories (core points, outliers, and border points)

Definition

A point is a core point if it has more than a specified number of points (MinPts) within $\epsilon-$ Neighborhood.

So v is a core point if $N_\epsilon(v) > MinPts$.
These are points that are at the interior of a cluster

Density-reachability

Definition

An object q is directly density-reachable from object p if p is a core object and q is in p's $\epsilon-$ Neighborhood.

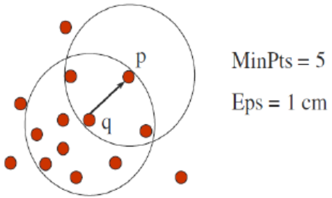


Figure: p is directly density-reachable from q. q is not directly density-reachable from p. (https://cse.buffalo.edu/ jing/)

Density-reachability

Two points p and q are directly density-reachable if there is a chain that connect these points.

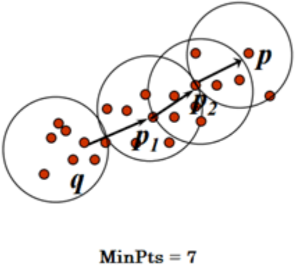


Figure: p is directly density-reachable from q. q is not directly density-reachable from p.(https://cse.buffalo.edu/ jing/)

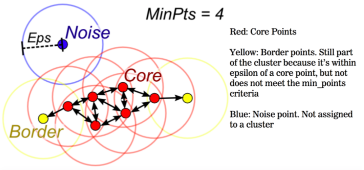
Border points- Outlier points

Definition

A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point.

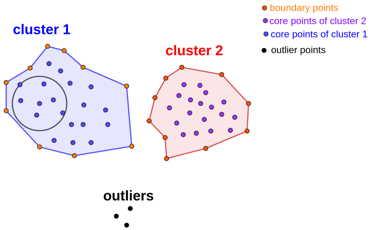
Definition

An outlier (noise) point is any point that is not a core point nor a border point.



Clustering with DBSCAN

A cluster C is a maximal subset of point v such that all points of C are density-connected two by two. A set is said to be maximals if any reachable density point from an element of this set also belongs to this same cluster.



Results of DBSCAN

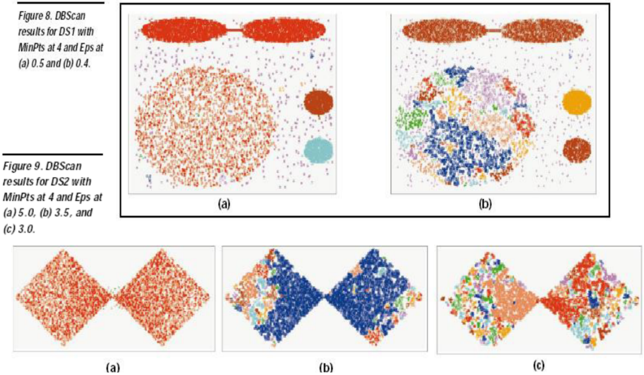


Figure: from https://cse.buffalo.edu/ jing/

DBSCAN vs kmeans

K-Means algorithm is sensitive towards outlier. Outliers can skew the clusters in K-Means in very large extent.

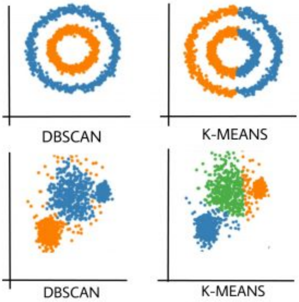


Figure: from https://www.geeksforgeeks.org/

Bibliography

- **[Jolliffe1986]** Jolliffe, Ian T. "Principal components in regression analysis." Principal component analysis. Springer, New York, NY, 1986. 129-155.
- **[Kanungo2002]** Kanungo, Tapas, et al. "An efficient k-means clustering algorithm: Analysis and implementation." IEEE transactions on pattern analysis and machine intelligence 24.7 (2002): 881-892.
- **[Martin1996]** Ester, Martin, et al. "A density-based algorithm for discovering clusters in large spatial databases with noise." Kdd. Vol. 96. No. 34. 1996.

Bibliography

- **[Smola1998]** Smola, Alex J., and Bernhard SchÃ¶lkopf. Learning with kernels. Vol. 4. GMD-Forschungszentrum Informationstechnik, 1998.
- **[Hinton2006]** Hinton, Geoffrey E., and Ruslan R. Salakhutdinov. "Reducing the dimensionality of data with neural networks science 313.5786 (2006): 504-507.
- **[Vincent2008]** Vincent, P., Larochelle, H., Bengio, Y., & Manzagol, P. A. "Extracting and composing robust features with denoising autoencoders." Proceedings of the 25th international conference on Machine learning. 2008.

Bibliography

- **[Cox2008]** Cox, Michael AA, and Trevor F. Cox. "Multidimensional scaling." Handbook of data visualization. Springer, Berlin, Heidelberg, 2008. 315-347.
- **[Laurens2008]** Van der Maaten, Laurens, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of machine learning research 9.11 (2008).
- **[Chenping2009]** Hou, Chenping, et al. "Local linear transformation embedding." Neurocomputing 72.10-12 (2009): 2368-2378.
- **[Harman1976]** Harman, Harry H. Modern factor analysis. University of Chicago press, 1976.
- **[Comon1994]** Comon, Pierre. "Independent component analysis, a new concept?." Signal processing 36.3 (1994): 287-314.

Bibliography

- **[Eckart1936]** Eckart, Carl, and Gale Young. "The approximation of one matrix by another of lower rank." Psychometrika 1.3 (1936): 211-218.
- **[Maaten2008]** Van der Maaten, Laurens, and Geoffrey Hinton. "Visualizing data using t-SNE." Journal of machine learning research 9.11 (2008).