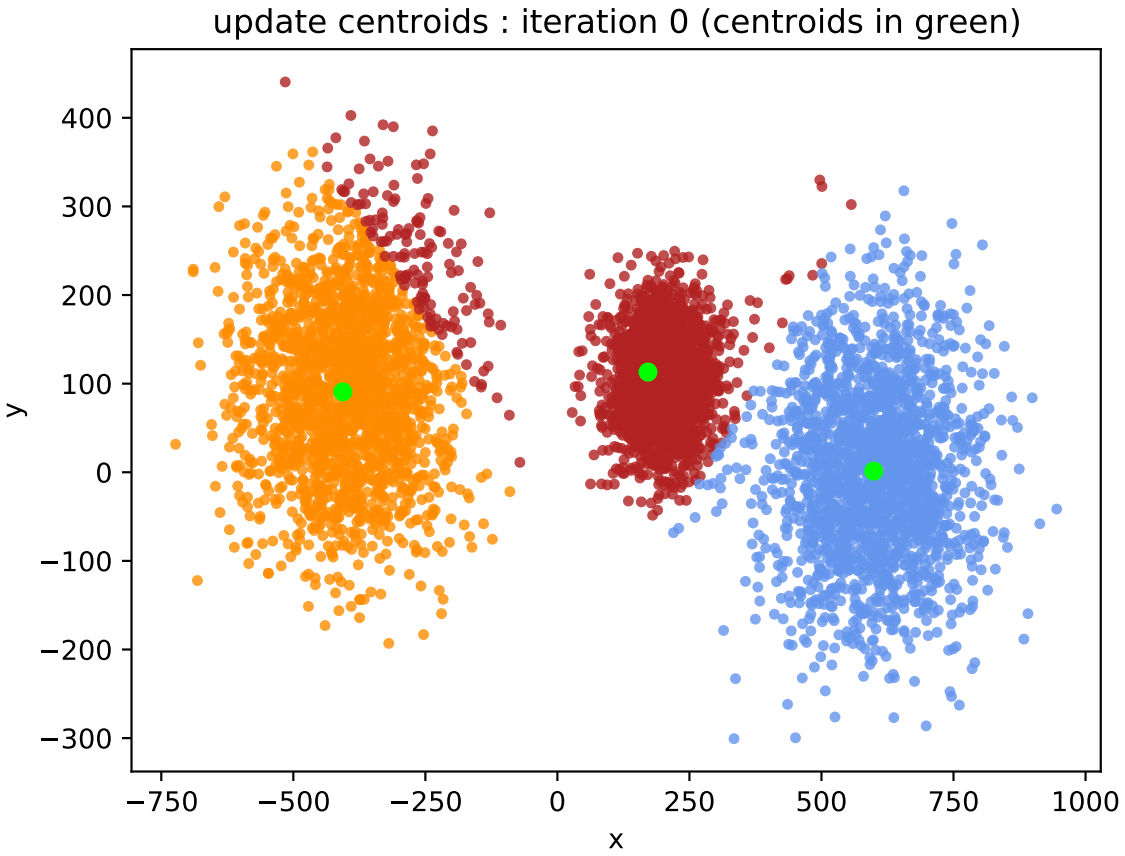
Fondamentaux théoriques du machine learning



Unsupervised learning

Clustering

- Motivation
- Vector quantization
- K-means clustering
  - Optimizations of K-means
- Hierarchical clustering
- Spectral clustering
- Evaluating the quality of clustering

Dimensionality reduction

- Motivation
- Principal component analysis
  - Applications of PCA
- Nonlinear dimensionality reduction

Density estimation

- Motivation
- Kernel density estimation

Unsupervised learning

From a number of samples x_i, you want to retrieve information on their structure : **modelisation**.

Unsupervised learning

From a number of samples x_i, you want to retrieve information on their structure : **modelisation**. The three main unsupervised learning problems are :

- clustering
- density estimation
- dimensionality reduction

Clustering

Clustering consists in partitioning the data. $\forall i, x_i \in \mathcal{X}^n$.

$$D_n = \{(x_i)_{i \in [1,\dots,n]}\} \tag{1}$$

Clustering

Clustering consists in partitioning the data. $\forall i, x_i \in \mathcal{X}^n$.

$$D_n = \{(x_i)_{i \in [1,\dots,n]}\} \tag{2}$$

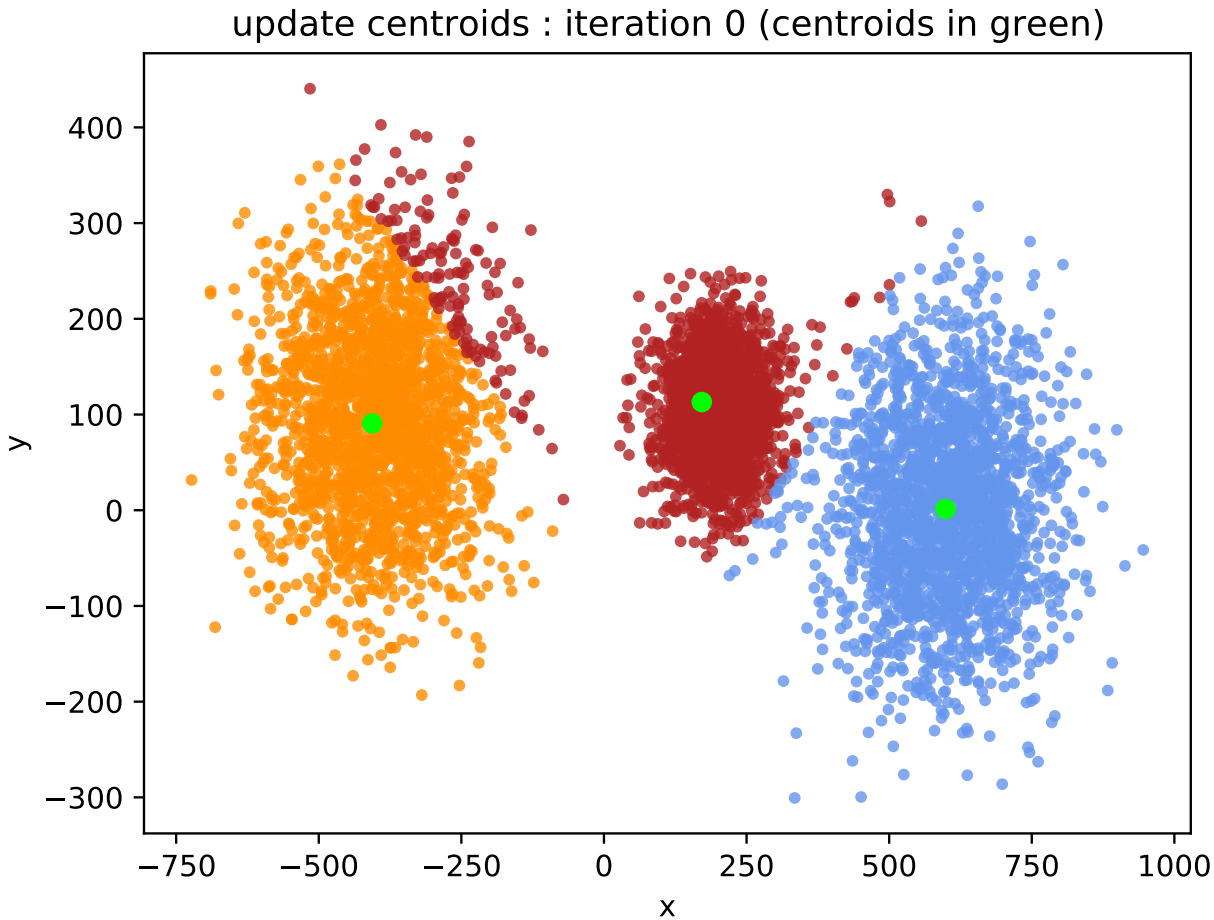
A **partition** is a set of K subsets $A_k \subset D_n$, such that

- $$\cup_{k \in [1,\dots,K]} A_k = D_n \tag{3}$$
- $$\forall k \neq k', A_k \cap A_{k'} = \emptyset \tag{4}$$

Partitions

- **Example 1 :** A is the set of even integers, B the set ot odd integers. Is (A, B) a partition of $\mathbb{N}$?
- **Example 2 :** C is the set of multiples of 2, D the set of multiples of 3. Is (C, D) a partition of $\mathbb{N}$?

Example : partition of data



Applications of clustering

Example applications :

- spam filtering [Sharma and Rastogi, 2014,]
- fake news identification
  [Hosseinimotlagh and Papalexakis, 2018,]
- marketing and sales
- document analysis [Zhao and Karypis, 2002,]
- traffic classificaiton [Woo et al., 2007,]

Some of these applications can be considered to be semi-supervised learning.

Vector quantization

Vector quantization consists in computing **prototypes** $\Omega = (\omega_k)_{k\in[1,\dots,K]} \in \mathcal{X}^K$ that represent the data well.
This implies that a **metric** is defined on $\mathcal{X}$.
Most often, this is interesting if $K << n$.

Choice of the metric

In some contexts, some usual metrics such as $L2$ might not be meaningful.

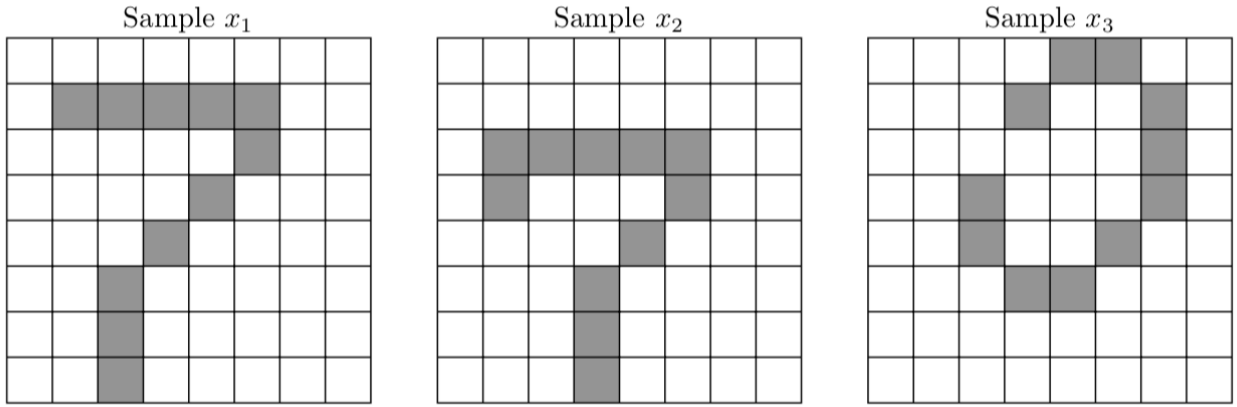


Figure – In $\mathbb{R}^{64}$, those three points form an equilateral triangle, [Fix et al., ,]

Voronoï subsets

We assume a loss L is defined on $\mathcal{X}$. The Voronoï subset of ω is defined as

$$V(\omega) = \{x \in D_n, \arg\min_{\omega' \in \Omega} L(\omega', x) = \omega\} \tag{5}$$

- We assume that arg min returns one single element.
- The Voronoï subsest form a partition of D_n.

Distortion

To measure the quality of a Voronoï partition, we introduce the **distortion** $R(\Omega)$.
For each x, we note $h_\Omega(x) = \arg\min_{\omega' \in \Omega} L(\omega', x)$.

$$R(\Omega) = \frac{1}{n} \sum_{i=1}^{n} L(x_i, h_\Omega(x_i)) \tag{6}$$

Distortion

For each x, we note $h_\Omega(x) = \arg\min_{\omega' \in \Omega} L(\omega', x)$.

$$\begin{aligned} R(\Omega) &= \frac{1}{n} \sum_{i=1}^{n} L(x_i, h_\Omega(x_i)) \\ &= \frac{1}{n} \sum_{\omega \in \Omega} \sum_{x \in V(\omega)} L(x_i, h_\Omega(x_i)) \\ &= \frac{1}{n} \sum_{\omega \in \Omega} V_\Omega(\omega) \end{aligned} \tag{7}$$

with

$$V_\Omega(\omega) = \sum_{x \in V(\omega)} L(x_i, h_\Omega(x_i)) \tag{8}$$

Minimum of distortion

We want to find the prototypes for which the distortion is **minimal**.

- The set of prototypes minimizing distorsion might not be unique.
- We need to tune K (number of prototypes).

Vector quantization techniques

- K-means
- Growing neural gas (GNG)
- Self-organizing maps

K-means clustering

- $\mathcal{X} = \mathbb{R}^d$.
- $L(x, y) = ||x - y||^2$.

Objective function

With

- $\Omega = \{\omega_1, \ldots, \omega_K\} \in \mathbb{R}^{K,d}$.
- $z_i^k = 1$ if $h_\Omega(x_i) = \omega_k$, $z_i^k = 1$ otherwise. $z = (z_i^k) \in \mathbb{R}^{n,K}$.

we define the objective function

$$J(\Omega, z) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_i^k ||x_i - \omega_k||^2 \tag{9}$$

It is also called the **inertia.**

K-means algorithm

Result: $\Omega \in \mathbb{R}^{K,d}$
$\Omega \leftarrow$ Random initialization;
$z = M$ where M is the $n \times K$ matrix with 0's;
while *Convergence criteria is not satisfied* **do**
  a] Minimize J with respect to z;
  b] Minimize J with respect to Ω;
end
return Ω

Algorithm 1: K-means (Loyd algorithm)

Stopping criterion

To stop the algorithm, the norm of the difference between Ω_t and Ω_{t+1} must be smaller than a given tolerance. (e.g. $1e^{-4}$). Here it is a norm between matrix (Frobenius norm) :

$$||A||_F = \sqrt{\sum_{i=1}^{n} A_{ij}^2} \tag{10}$$

Minimization

We focus on step b]. How can we minimize J with respect to Ω ?

Minimization

Exercice 1 : **Convexity :**
Show that $J(\Omega, z)$ is convex with respect to Ω.

$$J(\Omega, z) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_i^k ||x_i - \omega_k||^2 \tag{11}$$

Minimization

Exercice 1 : **Convexity :**
Show that $J(\Omega, z)$ is convex with respect to Ω.

$$J(\Omega, z) = \sum_{i=1}^{n} \sum_{k=1}^{K} z_i^k ||x_i - \omega_k||^2 \tag{12}$$

Hence, to minimize J with respect to Ω, we just need to cancel the gradient.

Minimization

Exercice 2 : **Gradient :**
Compute the gradient of $J(\Omega, z)$ with respect to Ω and deduce the minimizer Ω^*.

- z is fixed
- we can see Ω has a vector of $\mathbb{R}^{Kd}$.

Minimization

Exercice 2 : **Gradient :**

Compute the gradient of $J(\Omega, z)$ with respect to Ω and deduce the minimizer Ω^*.

- z is fixed
- we can see Ω has a vector of $\mathbb{R}^{Kd}$.

It is sufficient to compute the gradient with respect to all the ω_k separately.

Minimization

$$
\begin{aligned}
\nabla_{\omega_{k_0}} J &= \nabla_{\omega_{k_0}} \sum_{i=1}^{n} \sum_{k=1}^{K} z_i^k ||x_i - \omega_k||^2 \\
&= \sum_{i=1}^{n} \nabla_{\omega_{k_0}} \sum_{k=1}^{K} z_i^k ||x_i - \omega_k||^2 \\
&= \sum_{i=1}^{n} z_i^{k_0} \nabla_{\omega_{k_0}} ||x_i - \omega_{k_0}||^2 \\
&= \sum_{i=1}^{n} z_i^{k_0} 2(x_i - \omega_{k_0})
\end{aligned}
\tag{13}
$$

Gradient cancellation

This gradient cancels if

$$2\omega_{k_0} \sum_{i=1}^{n} z_i^{k_0} = 2 \sum_{i=1}^{n} z_i^{k_0} x_i \tag{14}$$

or equivalently

$$\omega_{k_0} = \frac{\sum_{i=1}^{n} z_i^{k_0} x_i}{\sum_{i=1}^{n} z_i^{k_0}} \tag{15}$$

Hence, the minimizer $w_{k_0}^*$ is the average of its cluster.

Convexity

Is $J(\Omega, z)$ convex in z ?

Convexity

Is $J(\Omega, z)$ convex in z ?
No, as z is not even defined on a convex set.
Hence, the function might have **local minima**.

Suboptimal clustering

Exercice 3 : **Local minima** : propose a setting (dataset, initialization) for which the algorithm outputs a bad set of centroids.

Suboptimal clustering

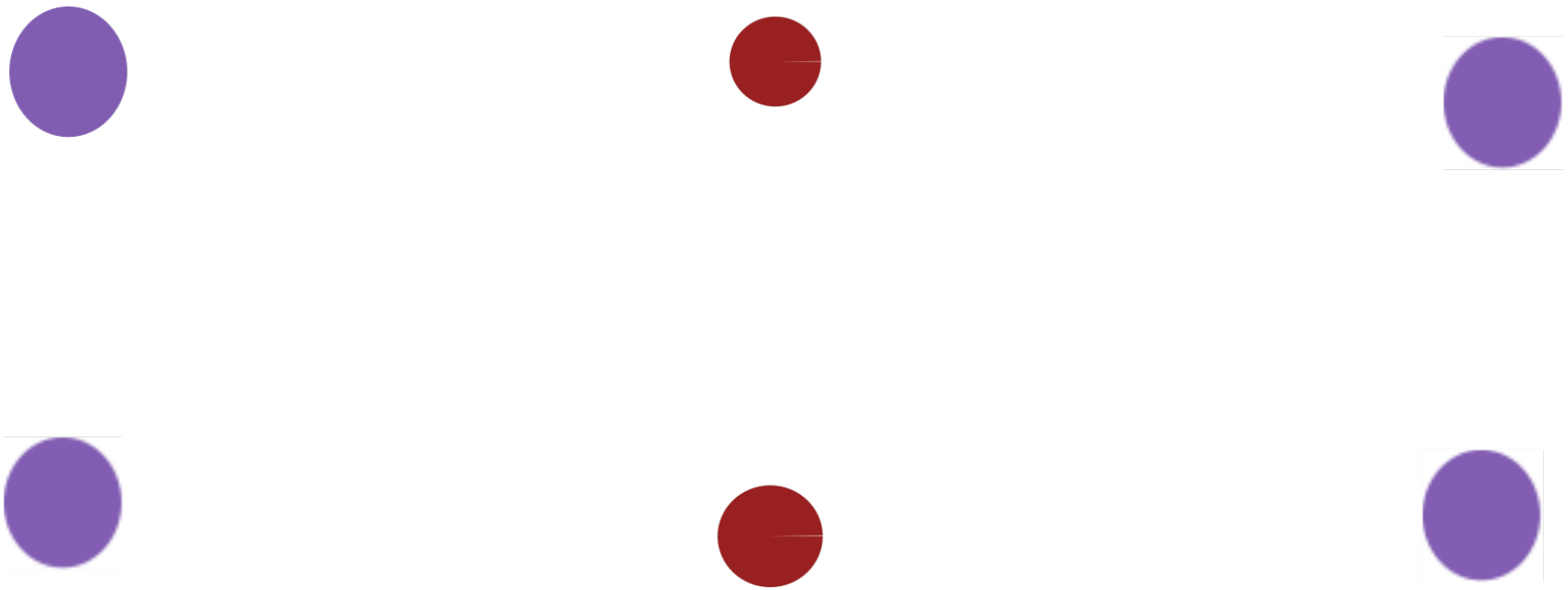


Figure – Initialization of centroids in red.

Random initialization

Hence, the result of K-means strongly depends on the initial position of the centroids.
A common approach is to restart the algorithm several times and select the result with lowest inertia.

Drawbacks of inertia minimization

K-means is based on the minimization of the inertia and hence on the euclidean distance. **However**, in some contexts, the euclidean distance is not the adapted metric.

`https://scikit-learn.org/stable/modules/clustering.html`

Number of clusters

Another drawback of Kmeans is that we have to input the number of clusters.

Number of clusters

Exercice 4 : **Hyperparameter :** can you propose a heuristic based on the inertia in order to choose a relevant K ?

Knee detection

`https://github.com/arvkevi/kneed`

K-means ++

K-means++ is a different initialization procedure.

- Choose one random sample x_{i_0} as first centroid.
- Then repeat until K initial centroid have been chosen :
  - For each x_i, compute the distance between x_i and its nearest centroid, noted $D_{\mu_i}(x_i)$.
  - Choose a new centroid among the data randomly, but with a non-uniform probability distribution. The probability of choosing x_i is proportionnal to $D_{\mu_i}(x_i)$.

This initialization tends to spread out the initial centroids.

X-means

Hierarchical clustering

This method build a hierarchy of clusters. It can be :

- agglomerative
- divisive

Example applications :

- phylogenetics
- document analysis

Agglomerative clustering

Initalization : one cluster per data point;
while *The number of clusters is* $> p$ **do**
| Merge the two closest clusters according to the linkage criterion.
end

Algorithm 2: Agglomerative clustering

Linkage criteria

Several **linkage critera** may be used to define the metric on the clusters, which means define a distance $d(A, B)$ between clusters A and B. For instance :

- single-linkage :

$$d(A, B) = \min\{d(x, y), x \in A, y \in B\} \tag{16}$$

- complete-linkage :

$$d(A, B) = \max\{d(x, y), x \in A, y \in B\} \tag{17}$$

- average linkage :

$$d(A, B) = \frac{1}{|A|.|B|} \sum_{x \in A} \sum_{x \in B} d(x, y) \tag{18}$$

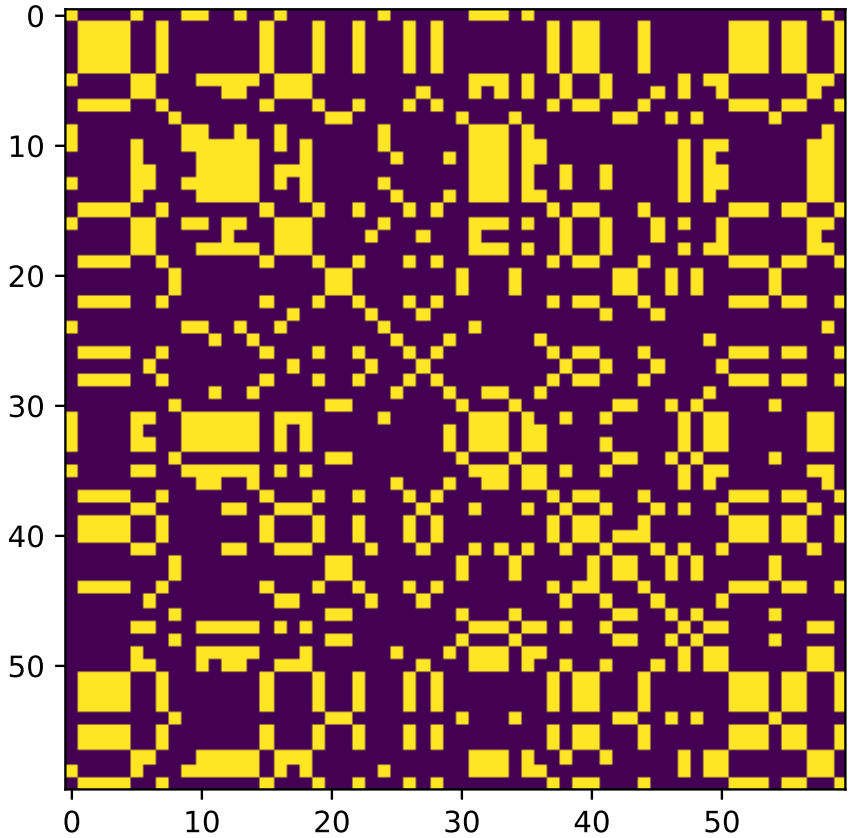
Similarities

- The K-means algorithm is based on the euclidean **distance** between points.
- **However,** in some situations, we do not have access to a distance between the points.
- We need to work with a quantity that is more general, for instance a **similarity**. Equivalently, we can consider dissimilarities.

Example of similarity : adjacency

- An example of similarity is the relationship of **adjacency**.
- If i and j are related by an edge, $S_{ij} = 1$.
- Otherwise $S_{ij} = 0$.

Adjacency matrix



Similarities

Differences between similarities and distances :

- A similarity S is not always symmetrical.

Similarities

Differences between similarities and distances :

- A similarity S is not always symmetrical.
- Indeed, in a **directed graph**, having a directed edge between i and j does not mean that we have an edge between j and i.

Similarities

- A similarity is a more general notion than a distance. Given a similarity between two points, we can deduce a similarity.
- For instance the **Gaussian similarity** , if d_{ij} is the distance betwwen i and j :

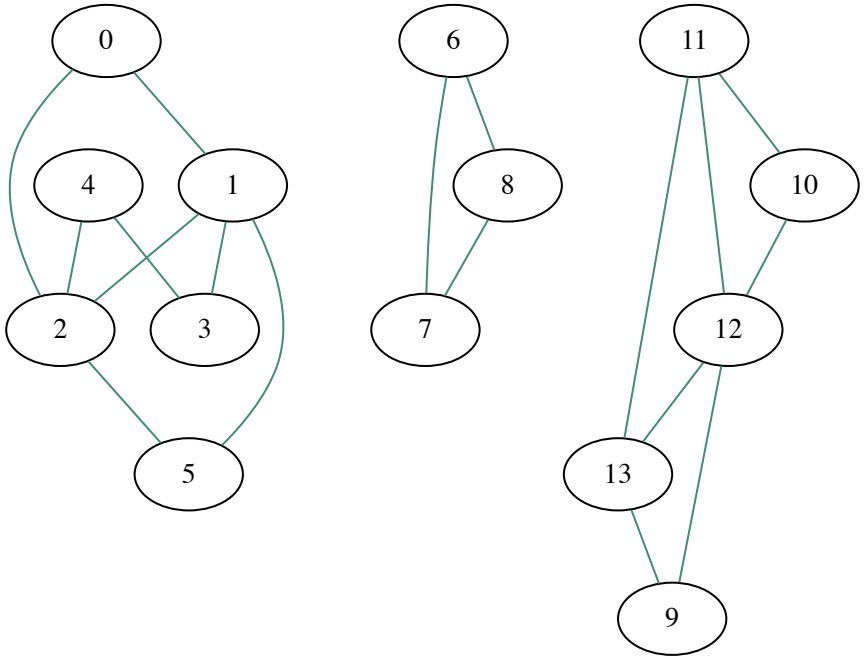
$$S_{ij} = \exp(-d_{ij}) \tag{19}$$

Spectral Clustering

- Spectral clustering is a method that works with similarities instead of distances.
- It performs a low dimensional embedding (representation, image by a map) of the **Laplacian** of similarity matrix, followed by a K-means.
  `https://en.wikipedia.org/wiki/Laplacian_matrix`

Exercise

It allows to cluster for instance graphs :



Spectral clustering

- Again, we need to provide the number of clusters.
- Not adapted to a large number of clusters.
- There is a k-means step : so the result depends on a random initialization.

Heuristic

- We would like a critetion in order to justify the number of clusters used.

Normalized cut : a measurement of the quality of a clustering

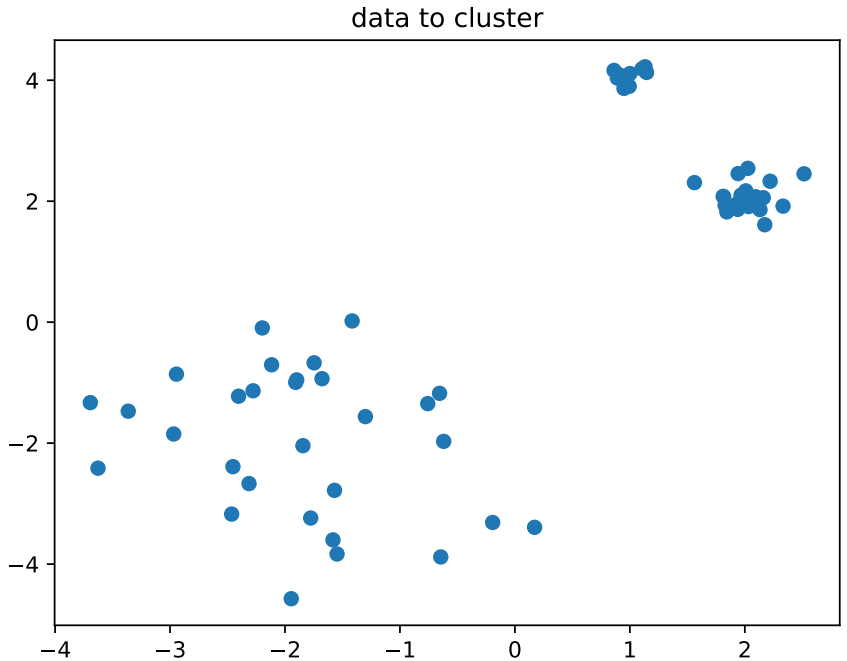
- The **cut of a cluster** is the number of outside connections (connections with other clusters), we note it $Cut(C_k, V \backslash C_k)$.
- The **degree** of a node is its number of adjacent edges
- The **degree of a cluster** is the sum of the degrees of its nodes, we note it d_{C_k} for cluster C_k.
- We define the **normalized cut** of a clustering as :

$$NCut(\mathcal{C}) = \sum_{k=1}^{K} \frac{Cut(C_k, V \backslash C_k)}{d_{C_k}} \tag{20}$$

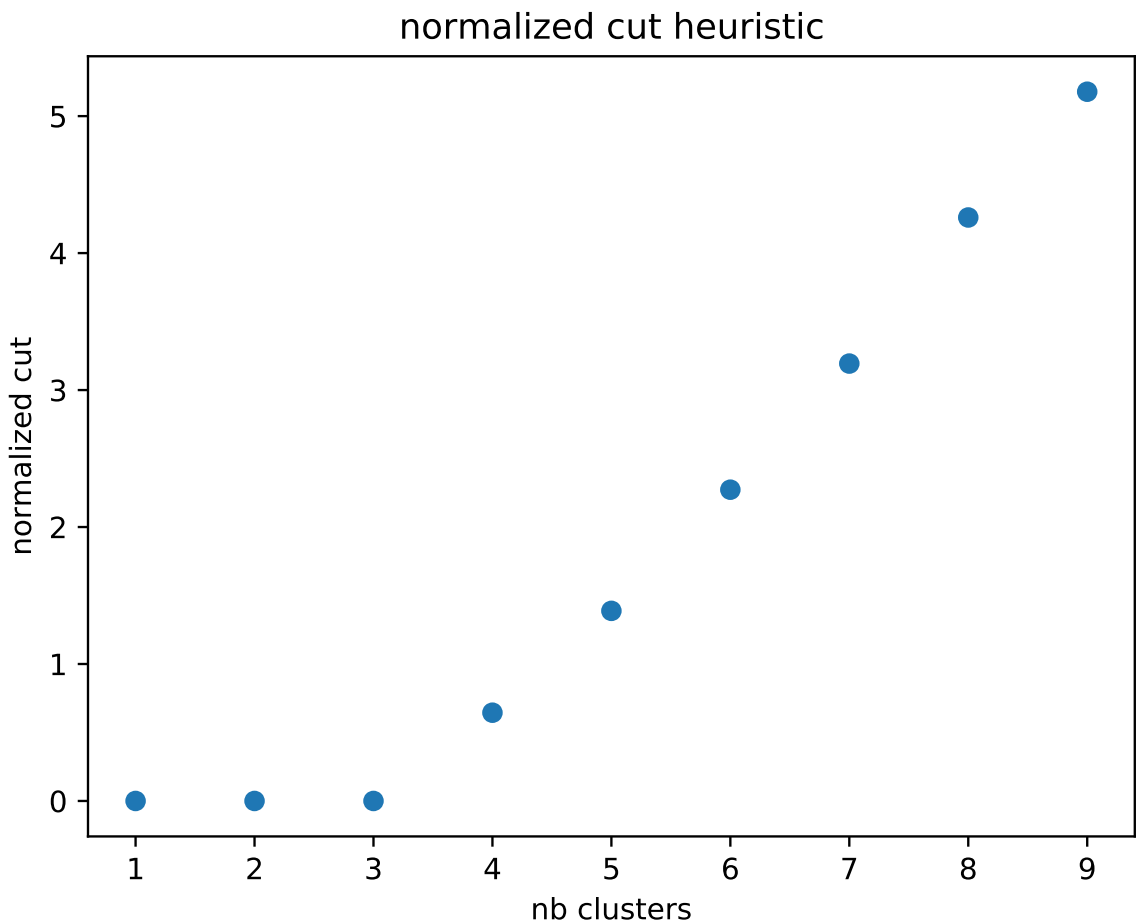
Normalization

- The normalization is useful in order to take the **weight** (degree) of a cluster into account.

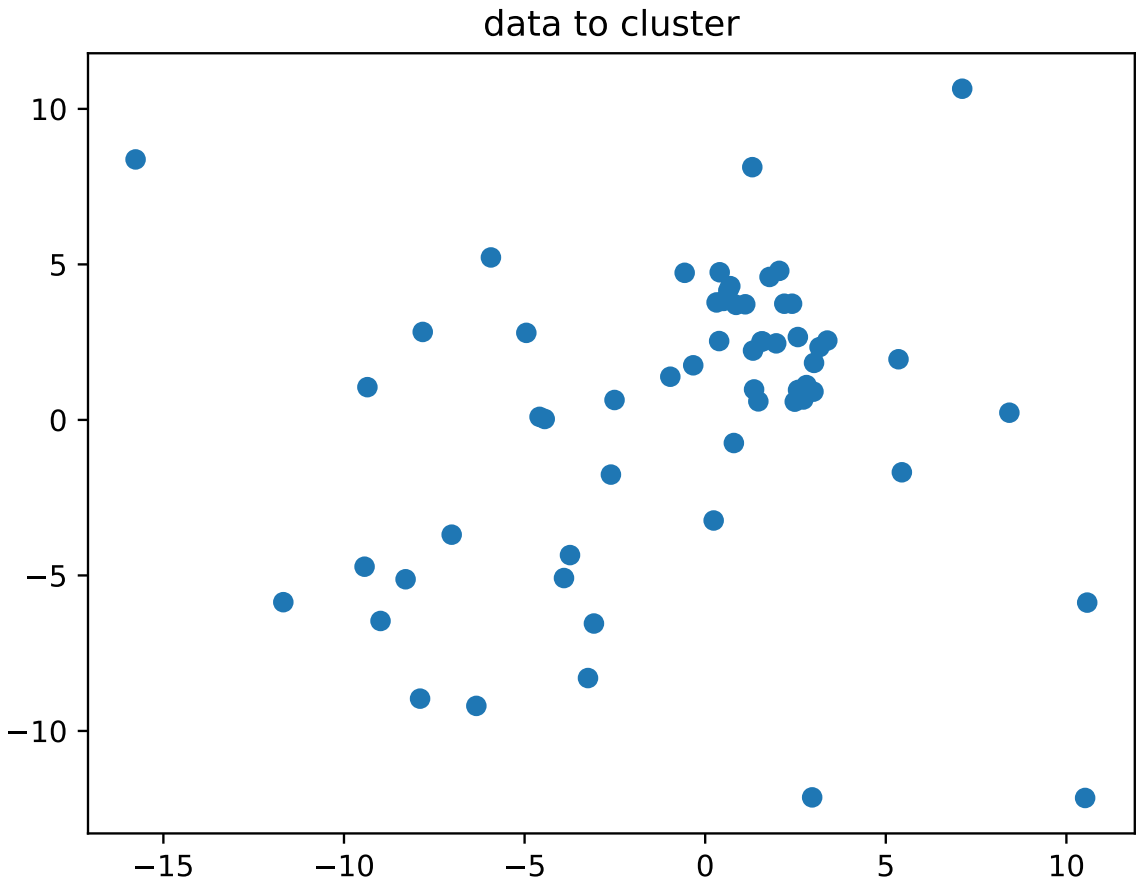
Heuristic



Normalized cuts



Heuristic



Normalized cuts

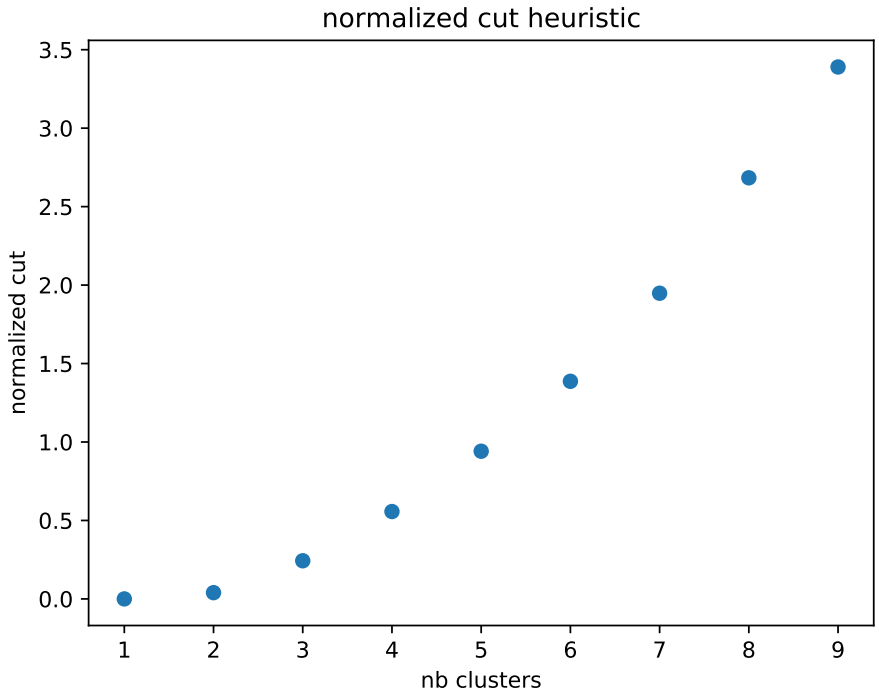


Figure – If the standard deviations in the dataset are larger, it is harder to identify a relevant number of clusters.

Other tools for the valuation of the quality of a clustering

- Stability of the result when lauching the algorithm many times
- Separation of the clusters (the mean distance between pairs of centroids is large)
- Ratio inter / intra
- Silhouette coefficient

Dimensionality reduction

We consider the space $\mathcal{X}$ that contains the data, for either supervised or unsupervised learning. In machine learning, we often have $\mathcal{X} \in \mathbb{R}^d$.

- If d is large (e.g. $\geq 10^4$), the algorithms that run on the data might become too slow to be used, as their algorithmic complexity depends on d (potentially in a quadratic or exponential way, curse of dimensionality)

Dimensionality reduction

We consider the space $\mathcal{X}$ that contains the data, for either supervised or unsupervised learning. In machine learning, we often have $\mathcal{X} \in \mathbb{R}^d$.

- If d is large (e.g. $\geq 10^4$), the algorithms that run on the data might become too slow to be used, as their algorithmic complexity depends on d (potentially in a quadratic or exponential way, curse of dimensionality)
- **However**, often the data might actually occupy a **subspace** of lower dimension q, or it may be possible to project the data on such a subspace without loosing too much information.
  - Working in a subspace of lower dimension might speed up the algorithms.
  - It may also allow visualization of the data.

Main methods of dimensionality reduction

- **feature selection** : selecting a subset of the original dimensions.
- **feature extraction** : computing new features from the original features.

Principal component analysis (PCA)

- PCA is a **linear feature extraction** technique.
- Points in $\mathbb{R}^d$ are linearly projected on a well chosen affine subspace of $\mathbb{R}^q$, with $q \leq d$.

Formalization as a variance maximimisation problem

Without loss of generality, we assume the data are **centered**, which means that

$$\bar{x} = \sum_{i=1}^{n} x_n = 0 \in \mathbb{R}^d \tag{21}$$

We note X is the design matrix as in OLS. The **first principal componenent** is a vector $w \in \mathbb{R}^d$, with $||w|| = 1$, such that $\hat{Var}(w^T x)$ is maximal, where $\hat{Var}$ denotes the empirical variance.

Variance

$$\overline{w^T x} = w^T \overline{x} = 0 \tag{22}$$

Hence,

$$\begin{aligned} \hat{Var}(w^T x) &= \frac{1}{n-1} \sum_{i=1}^{n} \left((w^T x)_i - \overline{w^T x} \right)^2 \\ &= \frac{1}{n-1} \sum_{i=1}^{n} \left(w^T x_i \right)^2 \end{aligned} \tag{23}$$

Variance maximisation problem

We can then formulate the problem as finding w, $||w|| = 1$ such that

$$\sum_{i=1}^{n} \left(w^T x_i\right)^2 \tag{24}$$

is maximal.

First principal component

We look for w, $||w|| = 1$ such that

$$\sum_{i=1}^{n} \left(w^T x_i\right)^2 \tag{25}$$

is maximal.

Proposition

w is the eigenvector of $X^T X$ with largest eigenvalue λ_{max}.

First principal component

We look for w, $||w|| = 1$ such that

$$\sum_{i=1}^{n} \left(w^T x_i\right)^2 \tag{26}$$

is maximal.

Proposition

w is the eigenvector of $X^T X$ with largest eigenvalue λ_{max}.

Exercice 5 : Show the proposition.

First principal component

$$\sum_{i=1}^{n}(w^T x_i)^2 = ||Xw||^2$$
$$= \langle Xw, Xw \rangle$$
$$= \langle (X^T X)w, w \rangle$$

This quantity is always smaller that λ_{max}, and it attained for an eigenvector in the eigenspace with norm 1, since we impose that $||w|| = 1$.

Reconstruction error

Alternately, we can formulate the problem as a **reconstruction error minimization.**

$$\mathbb{R}^d = \text{Vect}(w) \bigoplus \text{Vect}(w)^{\perp} \tag{27}$$

and if $||w|| = 1$,

$$\begin{aligned} \forall x \in \mathbb{R}^d, ||x||^2 &= ||(x^T w)w||^2 + ||x - (x^T w)w||^2 \\ &= (x^T w)^2 + ||x - (x^T w)w||^2 \end{aligned} \tag{28}$$

Reconstruction error

We can formulate the problem as a **reconstruction error minimization.** If $||w|| = 1$,

$$\begin{aligned} \forall x \in \mathbb{R}^d, ||x||^2 &= ||(x^T w)w||^2 + ||x - (x^T w)w||^2 \\ &= (x^T w)^2 + ||x - (x^T w)w||^2 \end{aligned} \tag{29}$$

Hence,

$$\begin{aligned} \sum_{i=1}^{n} ||x_i||^2 &= \sum_{i=1}^{n} (x_i{}^T w)^2 + \sum_{i=1}^{n} ||x_i - (x_i{}^T w)w||^2 \\ &= \hat{Var}(w^T x) + \sum_{i=1}^{n} ||x_i - (x_i{}^T w)w||^2 \end{aligned} \tag{30}$$

Reconstruction error

$$\sum_{i=1}^{n} ||x_i||^2 = \hat{Var}(w^T x) + \sum_{i=1}^{n} ||x_i - (x_i^T w)w||^2 \qquad (31)$$

We can see $\sum_{i=1}^{n} ||x_i - (x_i^T w)w||^2$ as a **reconstruction error**, when the data are projected on Vect(x).
Maximizing the variance of the projections is equivalent to minimizing the reconstruction errors obtained by projection.

Several principal components

Most of the time, we project the data on several principal components.

- 1] compute the first principal component w_1
- 2] project the data on $\text{Vect}(w_1)^\perp$
- 3] start again on the projected data

Reconstruction error

The interpretation stays the same. If $p_F(x)$ is the projection of x on the subspace spanned by the principal components :

$$||x||^2 = ||p_F(x)||^2 + ||x - p_F(x)||^2 \tag{32}$$

The principal components are the largest eigenvectors of $X^T X \in \mathbb{R}^d, d$ with norm 1. They are **orthogonal** to each other.

Inertia

We can again define an inertia :

$$I_F = \sum_{i=1}^{n} ||x - p_F(x)||^2 \tag{33}$$

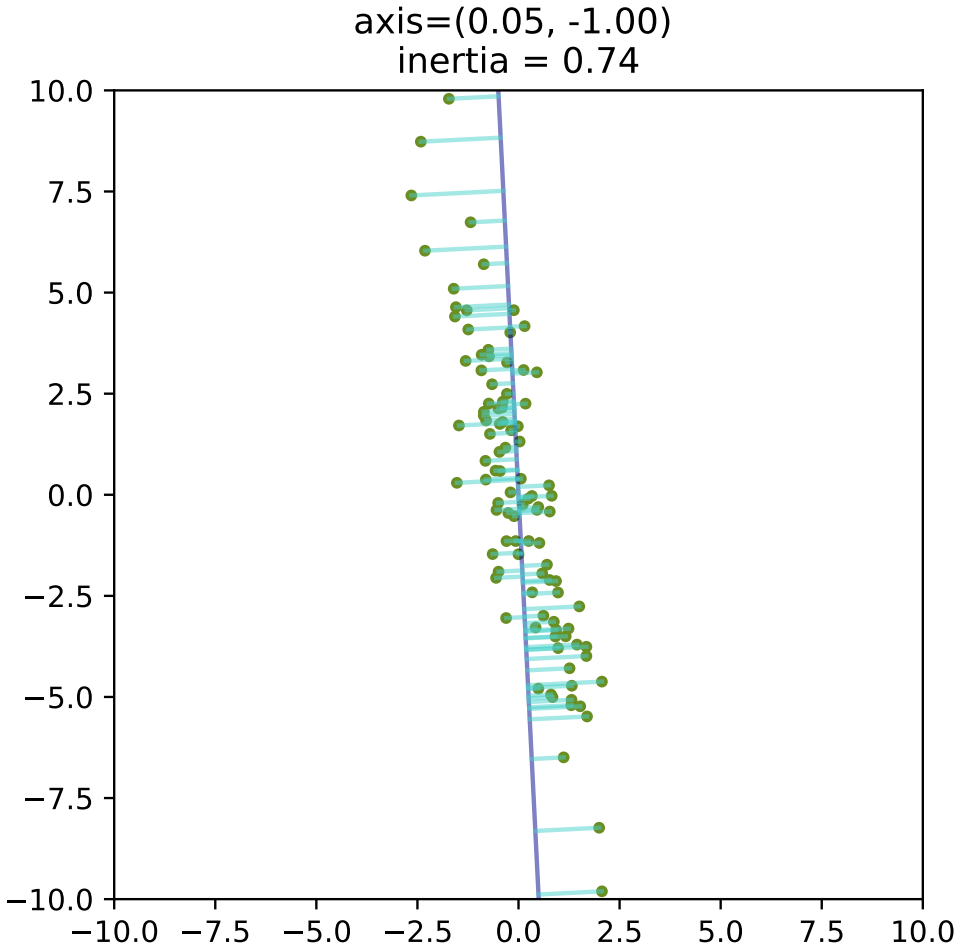
We loof for the subspace that minimized the inertia I_F.

Inertia

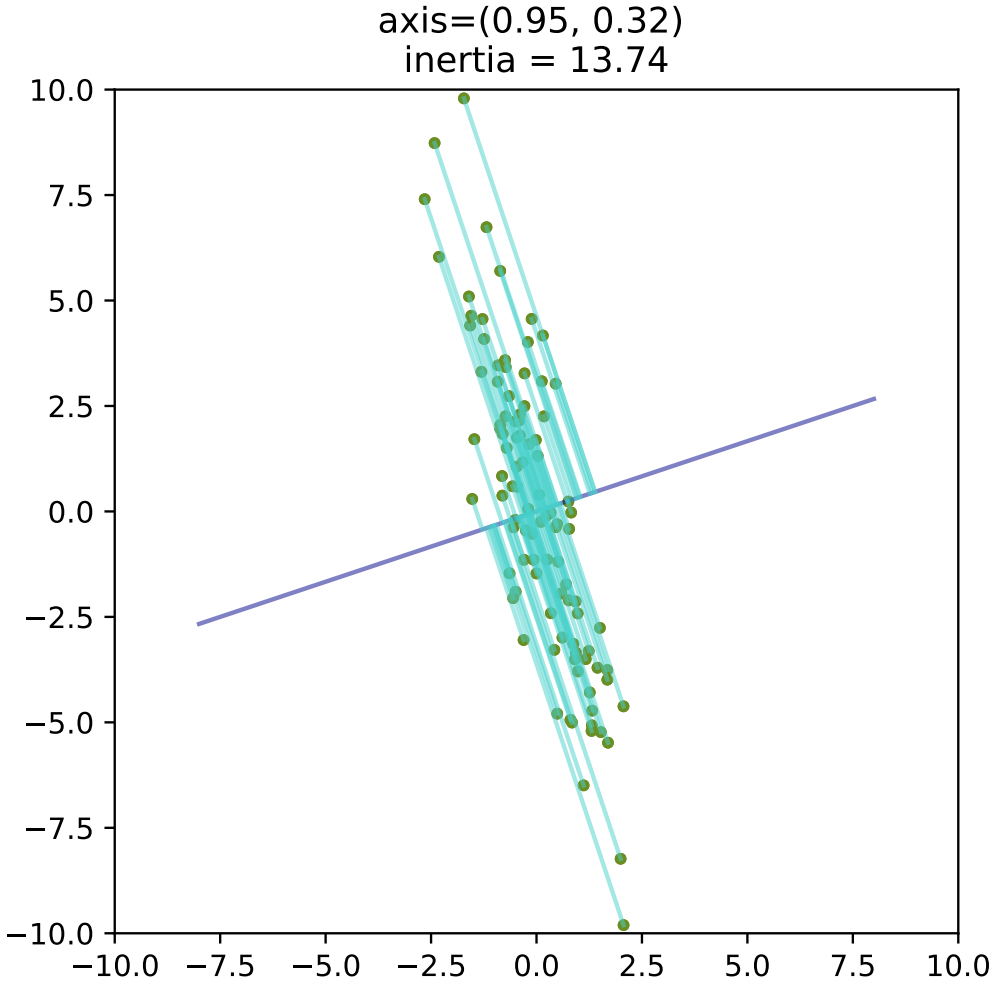
Exercice 6 : **No inertia**

In what situations could we have $I_F = 0$?

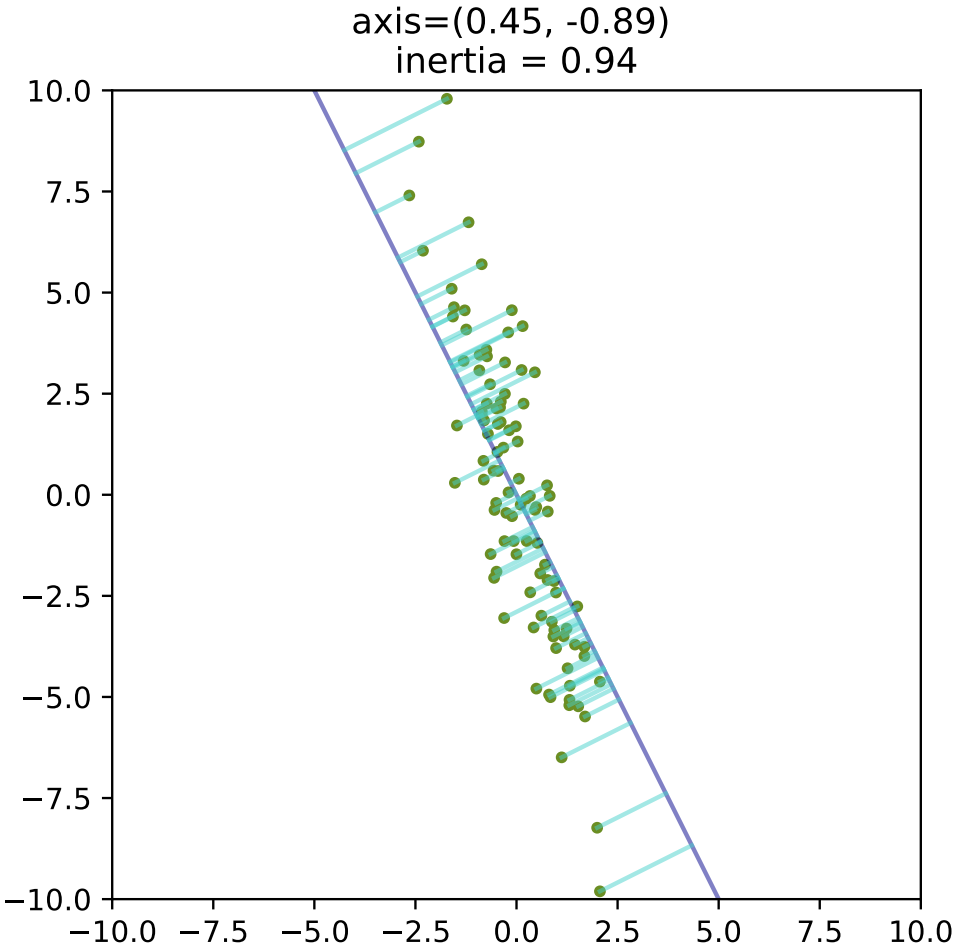
Inertia


axis=(0.05, -1.00)
inertia = 0.74
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

Inertia


axis=(0.95, 0.32)
inertia = 13.74
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

Inertia


axis=(0.45, -0.89)
inertia = 0.94
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

See also

Power iteration algorithm for finding the largest eigenvalue.

Iris dataset

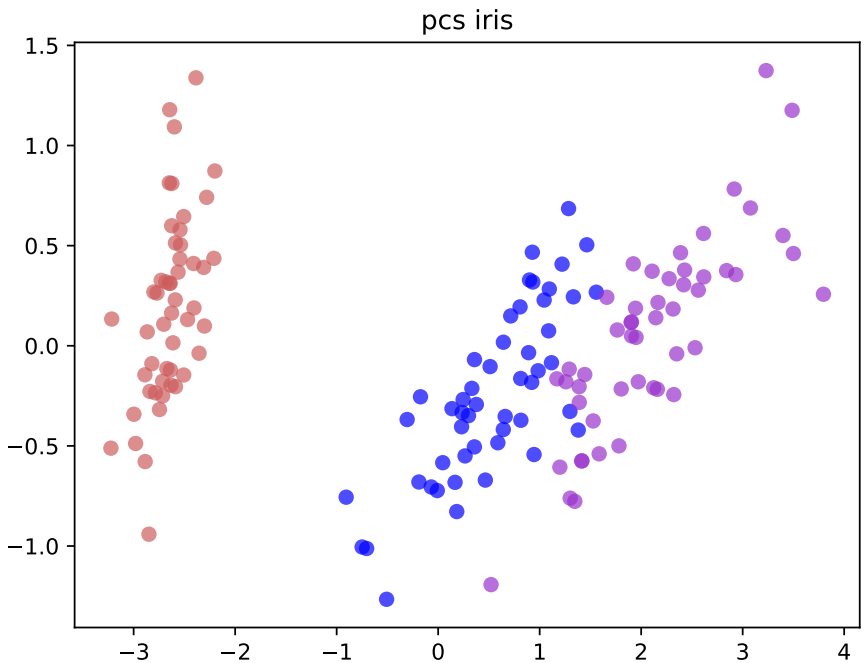


Figure – PCA performed on the iris dataset. We see that the principal components are able to separate the data.

In this paper, astrophysicists use PCA in order to test a new star temperature prediction method [Bermejo et al., 2013]

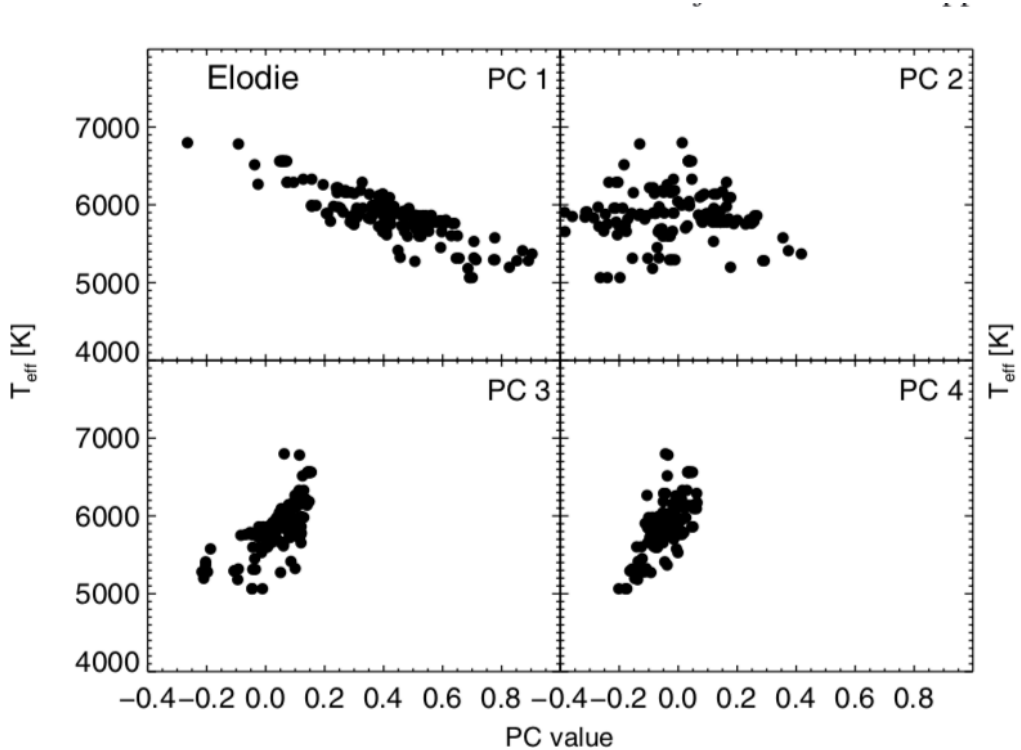
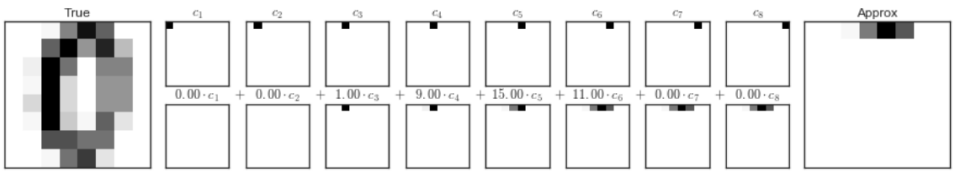


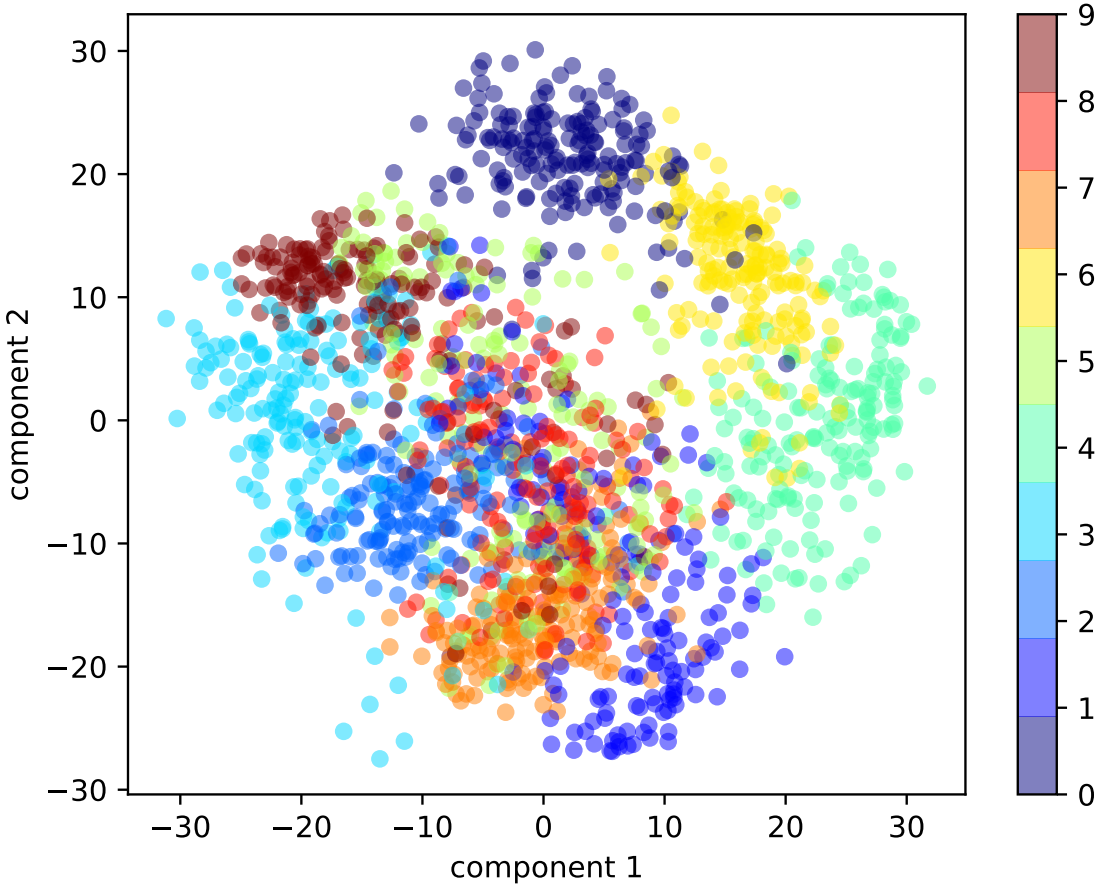
Figure – PCA used in order to predict temperature.

PCA on digits

- We can perform the PCA on a dataset consisting in 8×8 pixels images of digits, in order to see if the PCA allows a visualization of some structure in the data.



PCA on digits



PCA on digits : reconstruction

With 8 principal components, we can monitor the reconstruction of the images (originally in 64 dimensions)

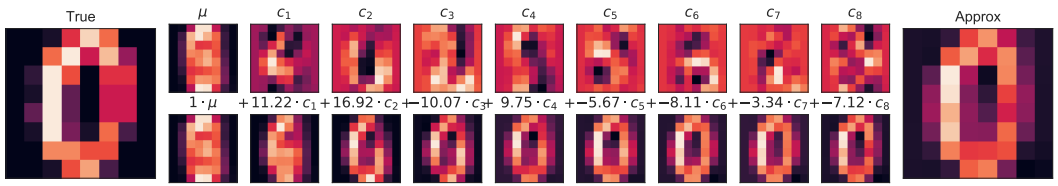


Figure – Reconstruction of 0

`https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html`

PCA on digits : reconstruction

With 8 principal components, we can monitor the reconstruction of the images (originally in 64 dimensions)

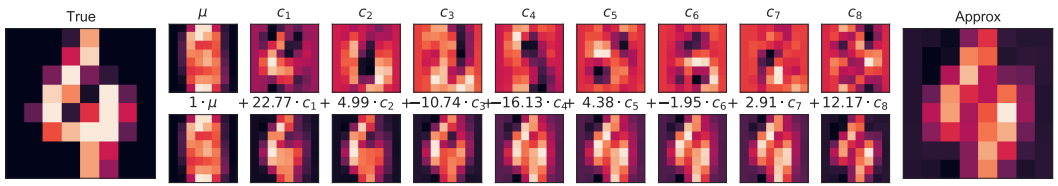


Figure – Reconstruction of 4

`https://jakevdp.github.io/PythonDataScienceHandbook/05.09-principal-component-analysis.html`

Explained variance

As always, a natural question is : what is a relevant number of principal components ?
A common quantity that is used is **explained variance**. Each component w_k carries a percentage of the total variance of the data.

$$\frac{\hat{Var}(w_k^T x)}{\sum_{j=1}^{d} \hat{Var}(x^j)} \tag{34}$$

where $\hat{Var}(x^j)$ is the variance of the component j.

$$\hat{Var}(x^j) = \sum_{i=1}^{n} (x_i^j)^2 \tag{35}$$

Number of components

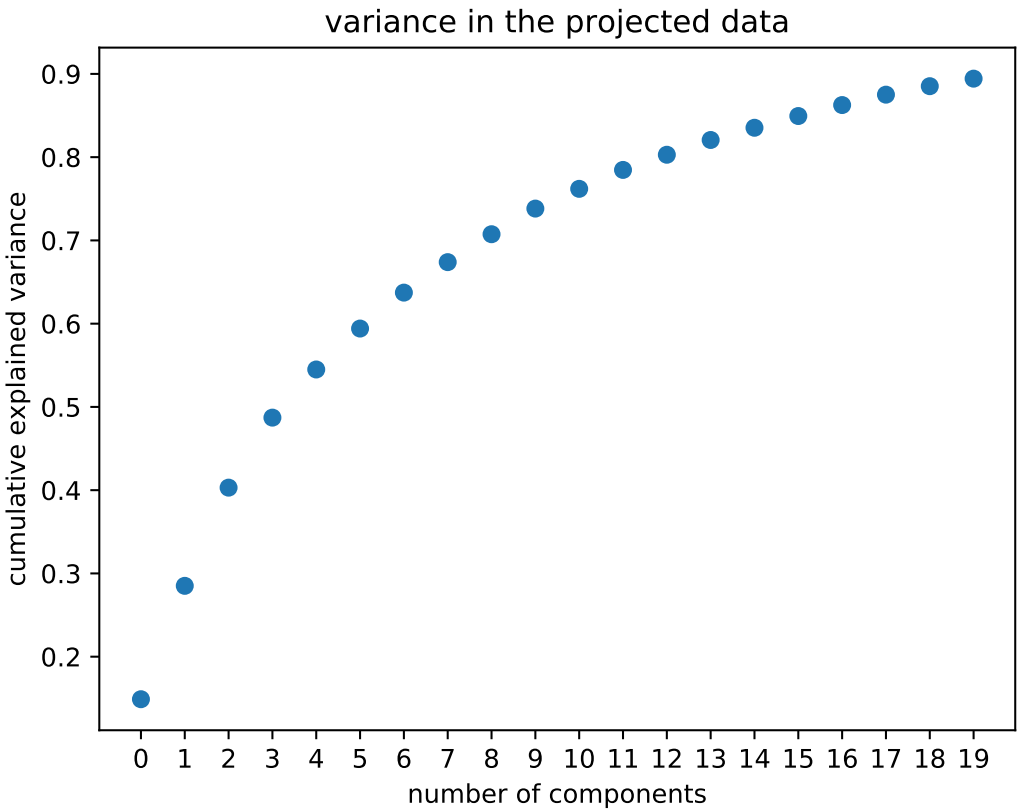
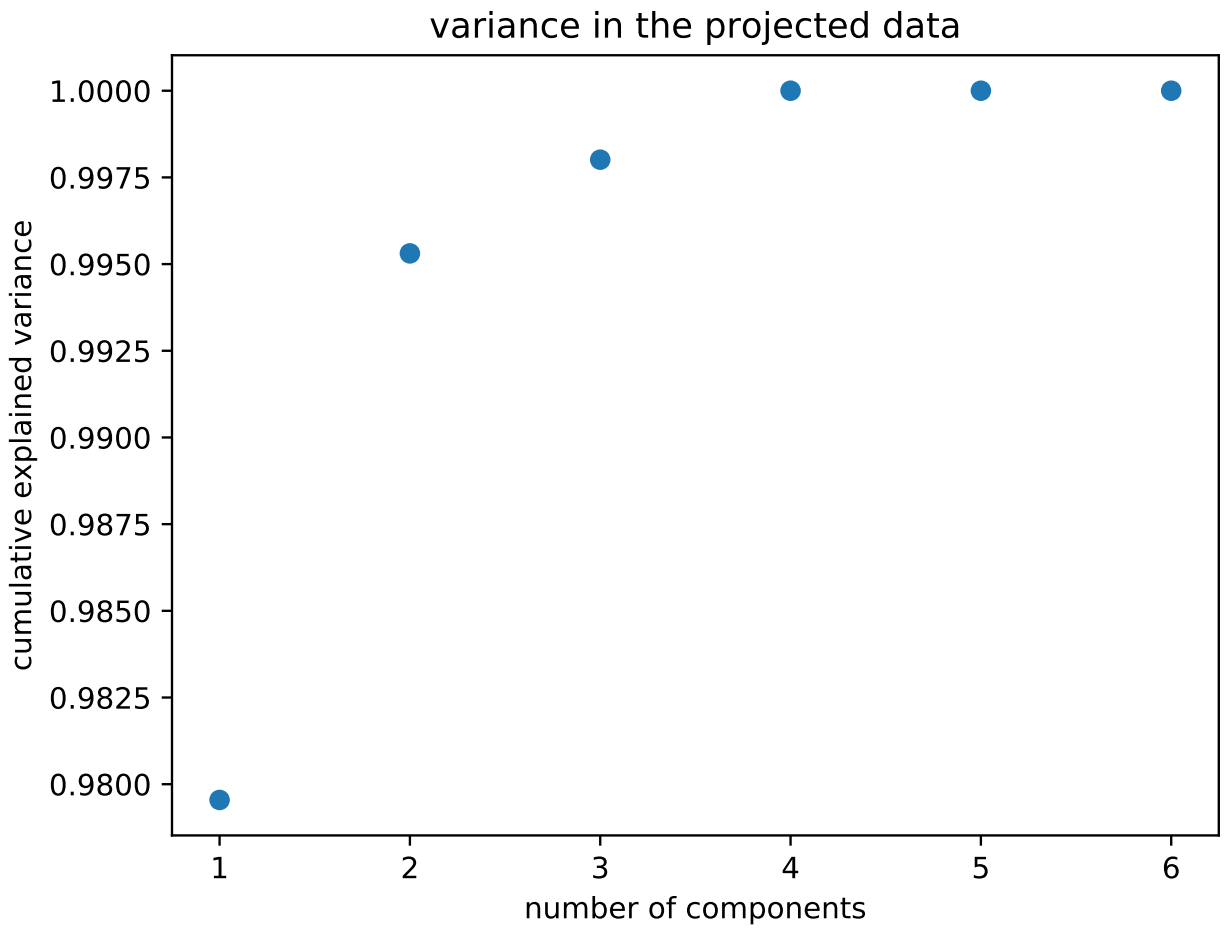


Figure – Variance of the projected data as a function of the number of components (digits dataset)

Number of components

Exercice 7 : What happens with this dataset ?



Number of components

Conclusion : PCA can help detemine whether some components carry no information in the data.

Shortcomings of PCA

PCA is sensitive to :

- outliers
- initial data scaling

Non-linear dimensionality reduction

- In the case of PCA, the manifolds are **linear subspaces** of the ambient space : PCA is a **linear method** and does **linear projections.**
- However, in many situations, the data might lie on **nonlinear** manifolds.
- **Manifold learning** is the unsupervised mearning of these manifolds in order to study the structure of the data.

Comments

Some disadvantages of non linear manifold learning :

- Hard to determine a good output dimension (whereas in PCA we can use explained variance) and it is hard to interpret the embedded dimensions (whereas in PCA we know what they mean).
- Depends on the number of neighbors chosen (if relevant)
- Often compuationally slower.

Density estimation

Applications of density estimation

Kernel density estimation

References I


Bermejo, J. M., Ramos, A. A., and Prieto, C. A. (2013).
Astrophysics A PCA approach to stellar effective temperatures.
95 :1–9.


Fix, J., Frezza-Buet, H., Geist, M., and Pennerath, F.
Machine Learning.pdf.


Hosseinimotlagh, S. and Papalexakis, E. E. (2018).
Unsupervised content-based identification of fake news articles with tensor decomposition ensembles.
Proceedings of the WSDM MIS2 : Misinformation and Misbehavior Mining on the Web Workshop, pages 1–8.

References II


Sharma, A. and Rastogi, V. (2014).
Spam Filtering using K mean Clustering with Local Feature Selection Classifier.
International Journal of Computer Applications, 108(10) :35–39.


Woo, D. M., Park, D. C., Song, Y. S., Nguyen, Q. D., and Tran, Q. D. N. (2007).
Terrain classification using clustering algorithms.
Proceedings - Third International Conference on Natural Computation, ICNC 2007, 1 :315–319.

References III

Zhao, Y. and Karypis, G. (2002).
Evaluation of hierarchical clustering algorithms for document datasets.
International Conference on Information and Knowledge Management, Proceedings, (August 2002) :515–524.