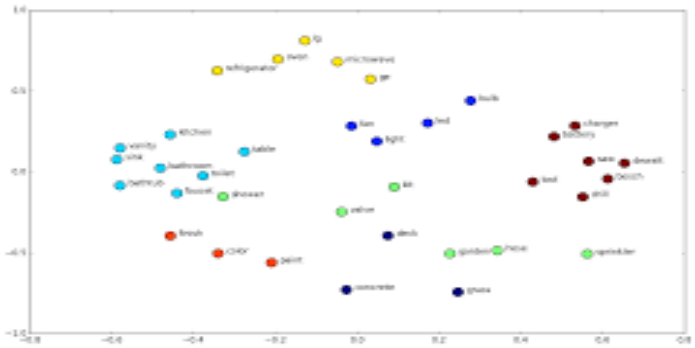
Dimension reduction with PCA and t-SNE

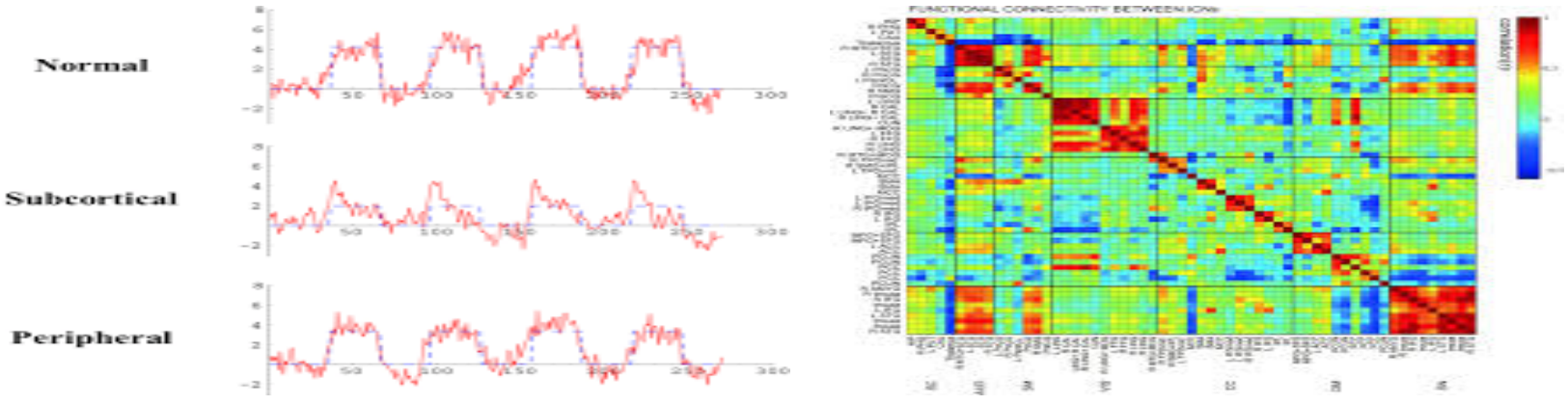
High dimensional data in data analysis?

		1	2	3	4	5	6	7	.....	
1	Apple	0.9898	0.7865	0.5645	0.7509	0.4534	0.5467	0.6498	0.7613	0.8
2	Banana	0.4533	0.8644	0.1538	0.4313	0.3511	0.2422	0.2422	0.3553	0.2
3	Cat	0.8734	0.8363	0.4821	0.1378	0.2341	0.2122	0.6775	0.3432	0.
4	Dog	0.9873	0.4836	0.1342	0.19564	0.2131	0.3433	0.2244	0.7453	0.5
5	Eag	0.9473	0.4836	0.4343	0.9211	0.1221	0.4634	0.7464	0.2424	0.5
6	Google	0.7634	0.4836	0.1313	0.1344	0.1232	0.6222	0.6564	0.3522	0.3
7	Home	0.8463	0.9732	0.4411	0.1333	0.6453	0.3435	0.3535	0.2442	0.3
.....	.	0.8653	0.4835	0.1343	0.4421	0.7567	0.2424	0.5241	0.3221	0.3
)00	Zoo	0.4736	0.9473	0.1453	0.1134	0.6564	0.1749	0.1892	0.1344	0.3



Words embeddings in NLP

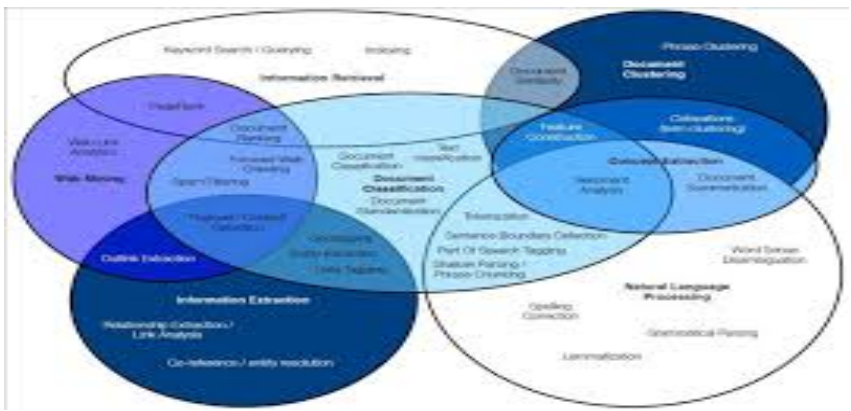
High dimensional data in data analysis?

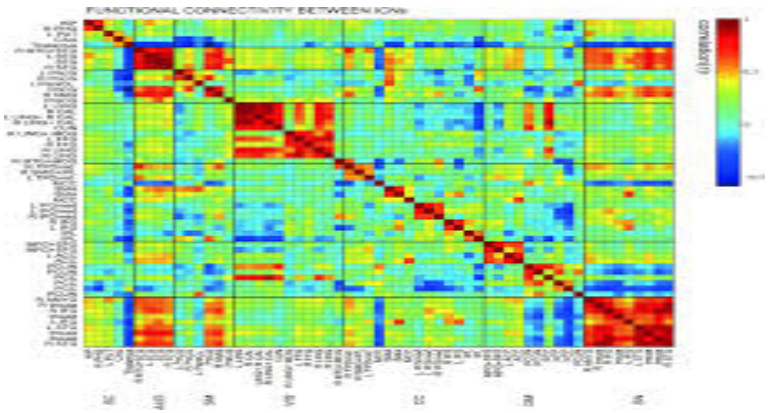


Brain activity

High dimensional data in data analysis?

Challenges ?

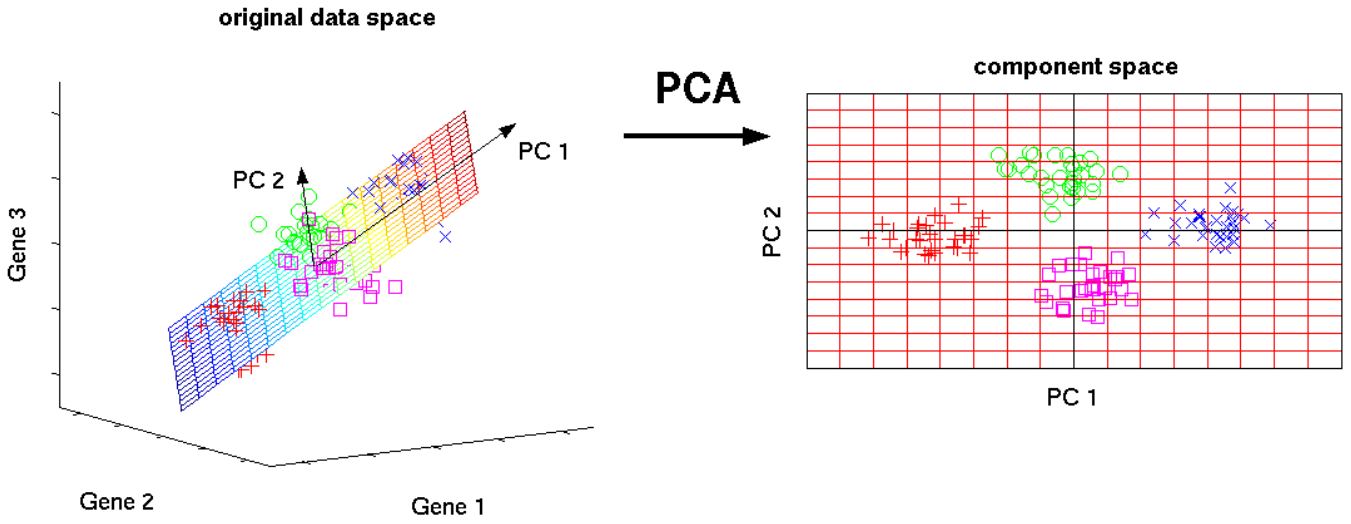




High dimensional data in data analysis?

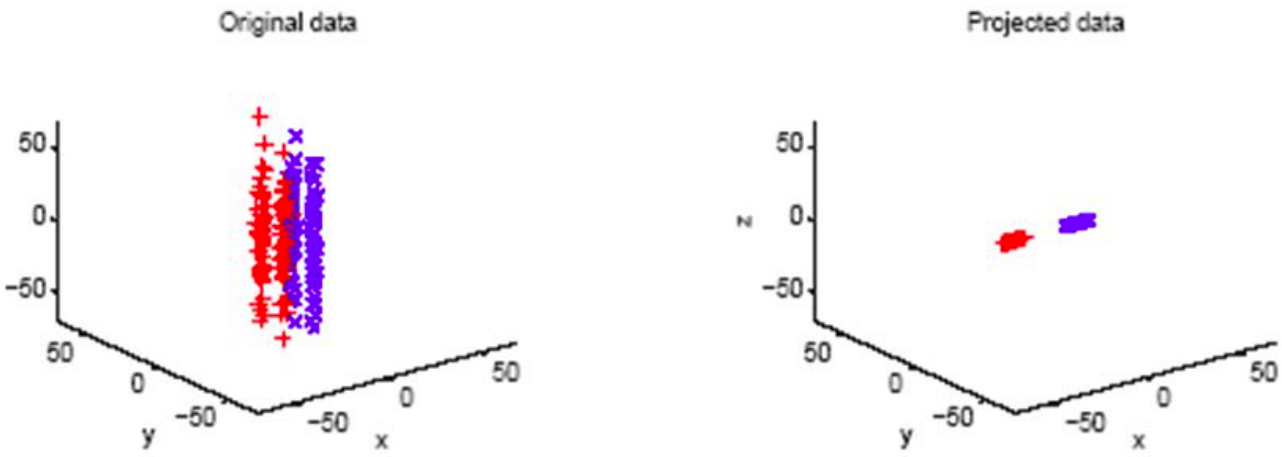
- Challenges ?
  - Visualize
  - Group in relevant clusters
- Difficult with high dimensional data!
- A classical dimension reduction approach Principal Component Analysis

Dimension reduction?



Dimension reduction without loss of information?

Dimension reduction



Dimension reduction without loss of information?

Dimension reduction

Scientific questions

How can we reduce dimension to separate observations

- A linear approach : Principal Component Analysis (PCA)
- A non linear alternative : t-distributed stochastic neighbourhood embedding (t-SNE)

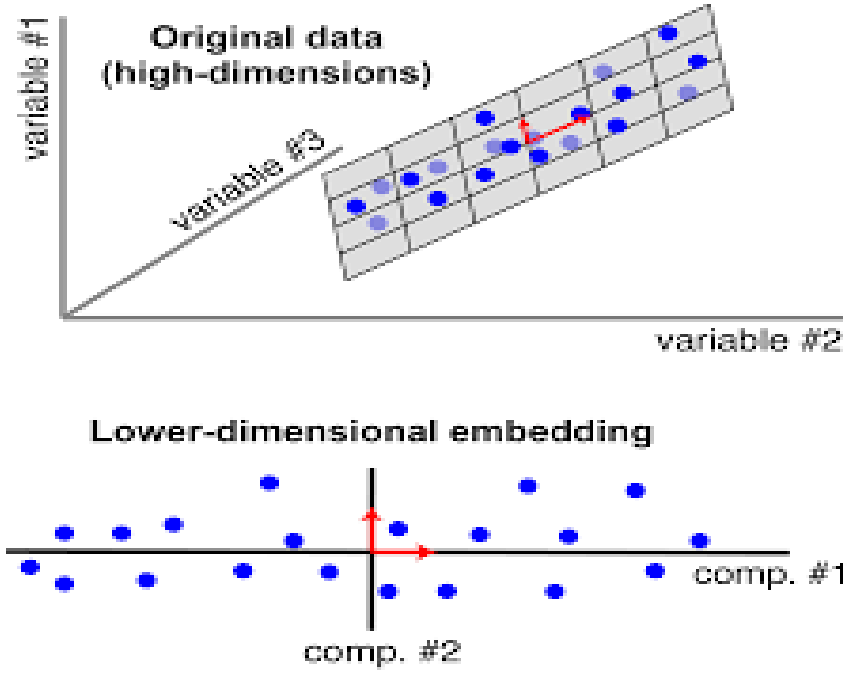
Dimension reduction

PCA vs t-SNE?

- Principal Component analysis (PCA): preserves the global structure of data. Maps all the clusters as a whole
  Potential applications : noise filtering, feature extractions, stock market predictions, and gene data analysis.
- t-distributed stochastic neighbourhood embedding (t-SNE): preserves the local structure of data.
  Potential applications : music analysis, bioinformatics, and biomedical signal processing.

Principal Component Analysis

Principle : find a linear projection on a low-dimensional space



How can we find the low-dimensional space H?

Principal Component Analysis

Practical examples with Python

Let us consider a data set describing tree kinds of leafs coming from the website : `https://archive.ics.uci.edu/ml/datasets/Leaf`

A toy example

- Leaf data set : describe tree kinds of leafs coming from the website : `https://archive.ics.uci.edu/ml/datasets/Leaf` Focus on the two following variables
  - Elongation : maximal normalized distance between a point of the leaf and its boundary
  - Isoperimetric factor : ratio between the area and the square of the perimeter of the leaf

Principal Component Analysis

Practical examples with Python

Leaf Data Set

Download: Data Folder, Data Set Description

Abstract: This dataset consists in a collection of shape and texture features extracted from digital images of leaf specimens originating from a total of 40 different plant species.

Data Set Characteristics:	Multivariate	**Number of Instances:**	340	**Area:**	Computer
Attribute Characteristics:	Real	**Number of Attributes:**	16	**Date Donated**	2014-02-24
Associated Tasks:	Classification	**Missing Values?**	N/A	**Number of Web Hits:**	88647

Source:

This dataset was created by Pedro F. B. Silva and AndrÃ© R. S. MarÃ§al using leaf specimens collected by Rubim Almeida da Silva at the Faculty of Science, University of Porto, Portugal.

Data Set Information:

For further details on this dataset and/or its attributes, please read the 'ReadMe.pdf' file included and/or consult the Master's Thesis 'Development of a System for Automatic Plant Species Recognition' available at [Web Link].

Attribute Information:

1. Class (Species)
2. Specimen Number
3. Eccentricity
4. Aspect Ratio
5. Elongation
6. Solidity
7. Stochastic Convexity
8. Isoperimetric Factor
9. Maximal Indentation Depth
10. Lobedness
11. Average Intensity
12. Average Contrast
13. Smoothness
14. Third moment
15. Uniformity
16. Entropy

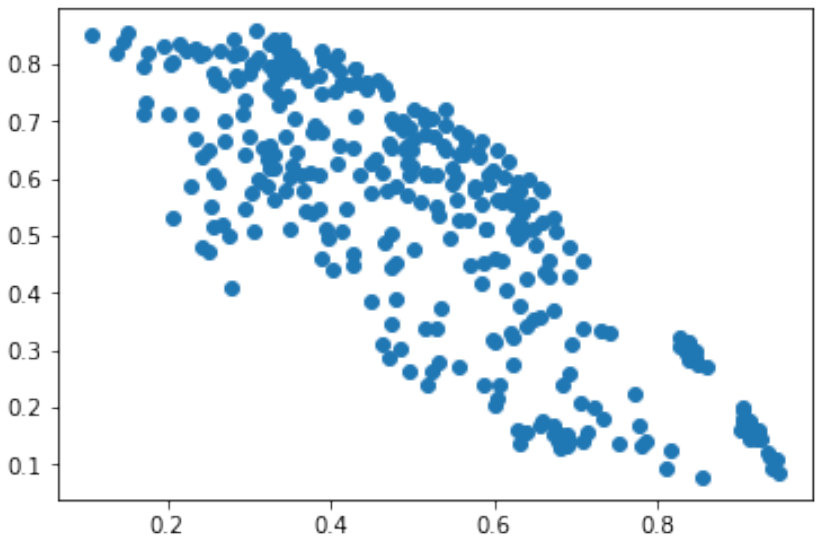
Principal Component Analysis

Practical examples with Python

```
import numpy as np
leaf =
np.loadtxt('/home/marianne/Desktop/Enseignement
/2017-2018/S2/M1-AD/TP1/leaf.csv', delimiter=',')
import matplotlib.pyplot as plt
fig, ax = plt.subplots()
ax.scatter(leaf[:,4],leaf[:,7])
plt.show()
```

Principal Component Analysis

Practical examples with Python



The Leaf data set

Principal Component Analysis

Practical examples with Python

Some questions

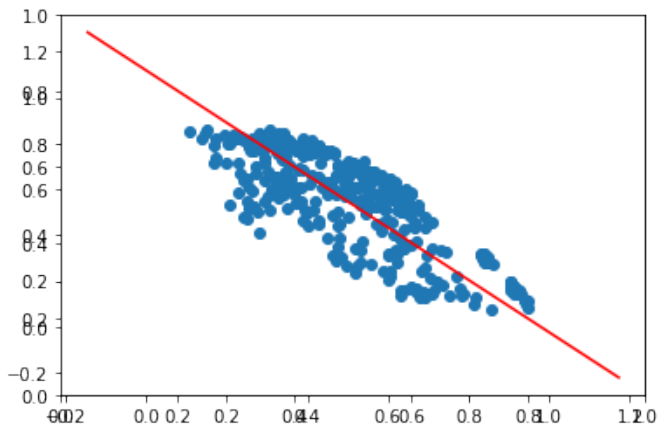
- How can we summarize properly using only one variable the information in this data set?
- Which variable allows to separate in the best possible way the data?
- Can we find an orientation along which the variance of the data is much higher ?

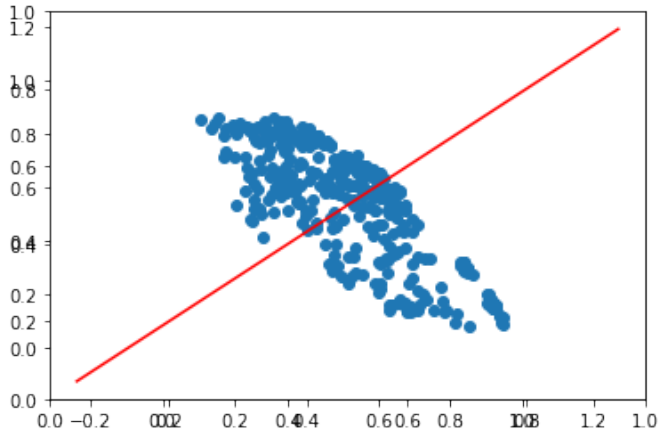
Principal Component Analysis

Practical examples with Python

Several possibilities....

...the axis of the figure on the left seems to be the best one!





Principal Component Analysis

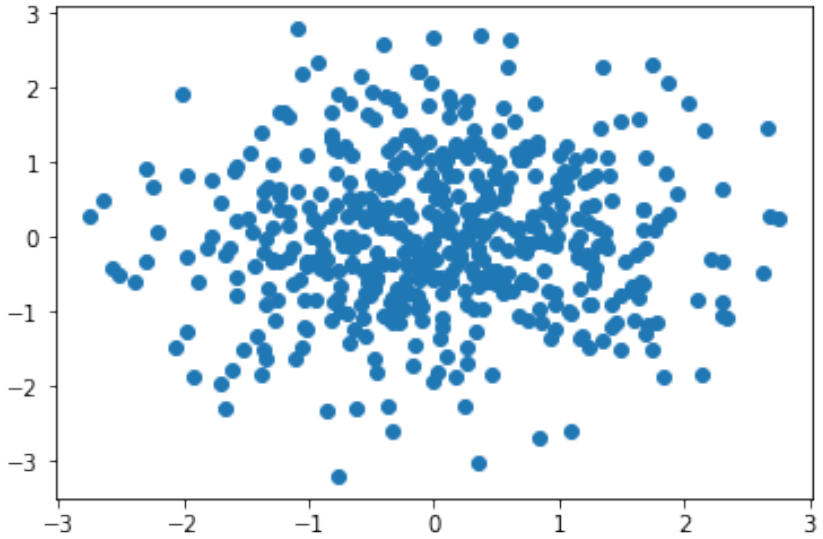
Practical examples with Python

We try with a synthetic dataset!

```
rndn = np.random.randn(500,2)
fig, ax = plt.subplots()
ax.scatter(rndn[:,0],rndn[:,1])
plt.show()
```

Principal Component Analysis

Practical examples with Python



Not always possible to find an axis separating properly the data!

Principal Component Analysis

Principle

Principal component analysis : how does it work?

- We want to find specific directions maximizing the variability of the data which is summarized in the covariance matrix $X^T X$
- The k dimensional space H that we are looking for is generated by the k eigenvectors u_i, $i = 1, \cdots, k$ associated to the k largest eigenvalues λ_i of the matrix $X^T X$

Principal Component Analysis

Principle

Principal component analysis : how does it work?

- The eigenvectors $u_1, \cdots, u_k$ are called the k principal components.
- The eigenvalues $\lambda_1, \cdots, \lambda_k$ are the explained variance ratio corresponding to each principal component
- By definition, the k top principal components contain higher variance from the data.
- PCA can then be used as filtering, by selecting only the top significant PCs

Principal Component Analysis

Principle

Principal component analysis : how does it work?

- We have several possible choices for the matrix X :
  - General PCA : the raw data matrix $X = R$
  - Centered PCA: the centered data matrix. the matrix $X^T X$ is then the matrix of empirical covariances
  - Normed PCA : the normed and centered data matrix. The matrix $X^T X$ is then the matrix of empirical correlations
- Projection of the observation o_i on the axis α

Principal Component Analysis

Principle

Principal component analysis : how does it work?

- In general $n \gg d$ (number of observations $\gg$ number of initial variables)
- It is the reason why we deal with the matrix $X^T X$ with dimension $d \times d$ rather than XX^T with dimension $n \times n$
- Existence of some links between these two analysis

Principal Component Analysis

Pro and cons of PCA

Main advantages of PCA

- Simple to implement, no tuning
- Highly interpretable. We can find decide on how much variance to preserve using eigen values.

Main drawbacks of PCA

- It is a global transform which may not preserve local structure (clusters)
- It is sensitive to outliers

An alternative : t-distributed stochastic neighbourhood embedding (t-SNE)

t-distributed stochastic neighbourhood embedding (t-SNE)

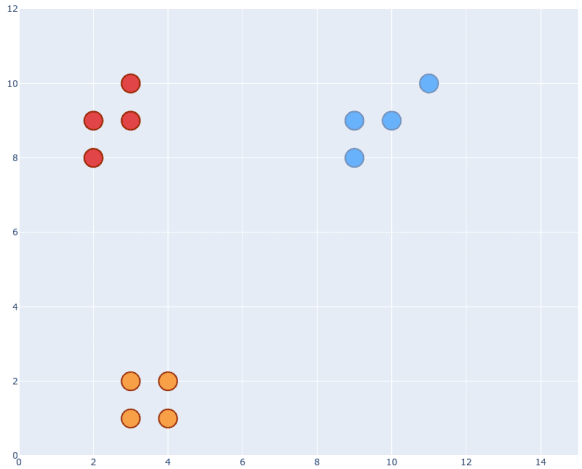
Principle

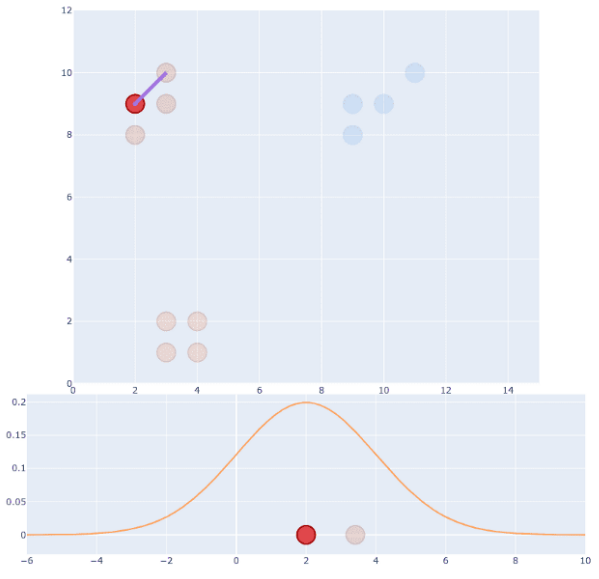
- Core idea : define an embedding from a high dimensional space to a low dimensional one, so that neighborhood identities are preserved.
- Similarity of x_j to x_i is conditional probabilty $p_{j|i}$ that x_i would pick x_j as its nearest neighbor

t-distributed stochastic neighbourhood embedding (t-SNE)

Principle

First stage : definition of a probability associated to each observation





t-distributed stochastic neighbourhood embedding (t-SNE)

Principle

- This conditional probability p is a Gaussian in the high dimensional space and depends on the known position of the observations in the original space
- This conditional probability q is a Student distribution in the low dimensional space. We have to fit it and it depends on the unknown position of the observations in the target space
- To fit q, we solve

$$Argmin \sum_{i,j} D(p_{j|i} || q_{j|i})$$

where D is the KL divergence

t-distributed stochastic neighbourhood embedding (t-SNE)

Pro and cons of t-SNE

Main advantages of t-SNE

- It tries to preserve the local structure(cluster) of data.
- It is one of the best dimensionality reduction technique
- It can handle outliers.

Main drawbacks of t-SNE

- It is not deterministic and iterative so each time it runs, it could produce a different result.
- It is long to run compared to PCA
- It involves Hyperparameters such as perplexity, learning rate and number of steps.