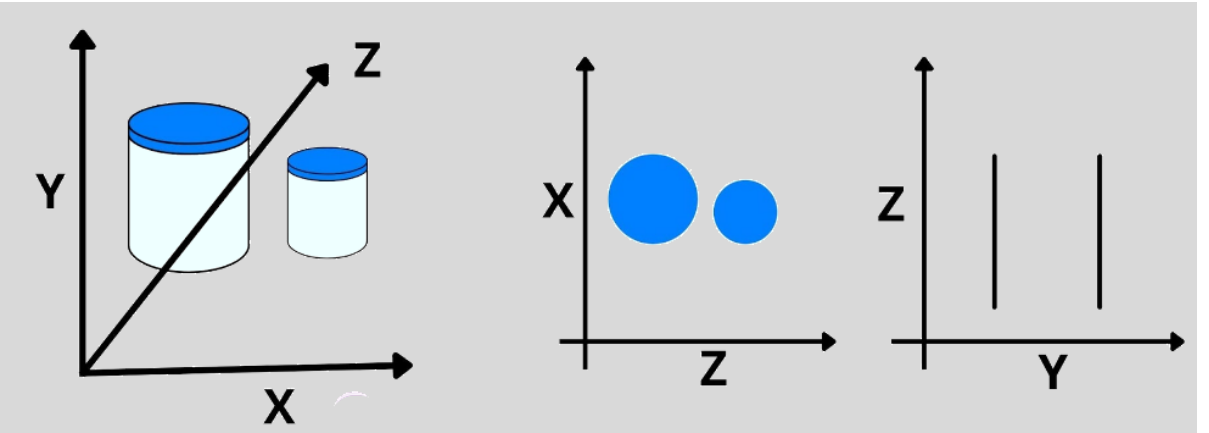
Dimensionality reduction



Lluís Talavera, 2022

UNIVERSITAT POLITÈCNICA DE CATALUNYA BARCELONATECH
ETSEIB

Dimensionality reduction

It is an **unsupervised learning** technique.

Dimensionality: The number of input features for a dataset.

Dimensionality reduction refers to techniques that map the data into a lower dimesional space to reduce the number of features of a dataset while preserving as much relevant information as possible.

Advantages

- Reduces computational time and storage requirements.
- Avoids the *curse of dimensionality*.
- Remove irrelevant features that may decrease the performance of the models.
- Helps avoiding multicollinerarity.
- Makes visualization easier.

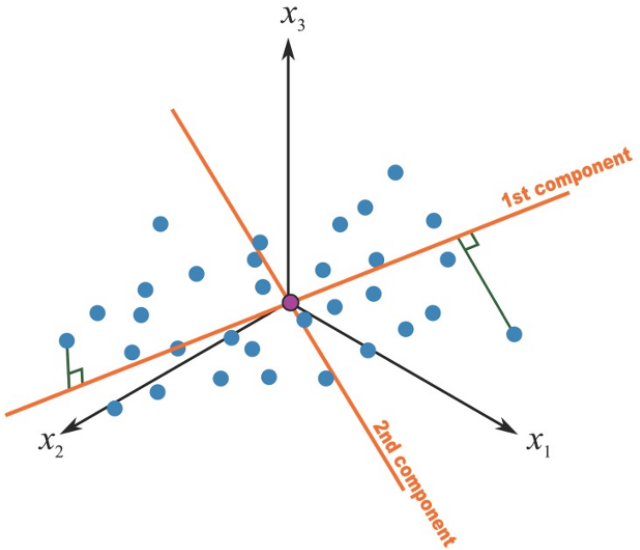
The information preserved depends on the assumptions of each method. Can be measured as variance, data structure (neighborhood),...

PCA (Principal Component Analysis)

A principal component (PC) is the direction in space where there is the most variance (the data is most spread out). A PC is calculated as a linear combination of the original n predictors in the data matrix X, i.e. $T = XW_{n}$

where the weights W_{n} maximize explained variability. Mathematically, these weights are the *eigenvectors* of the original correlation matrix.

The kth component is the variance-maximizing direction *orthogonal* (uncorrelated) to the previous $k – 1$ components.



source

PCA algorithm

Input: **standardized** matrix of examples and variables

1. Compute the covariance matrix C
2. Find the *eigenvectors* and *eigenvalues* of C
3. The eigenvectors (PCs) are the new axes
4. The eigenvalues are the variance explained by each PC

Output: eigenvectors and eigenvalues of the covariance matrix

Covariance is a measure of the joint variability of two variables (different from correlation):

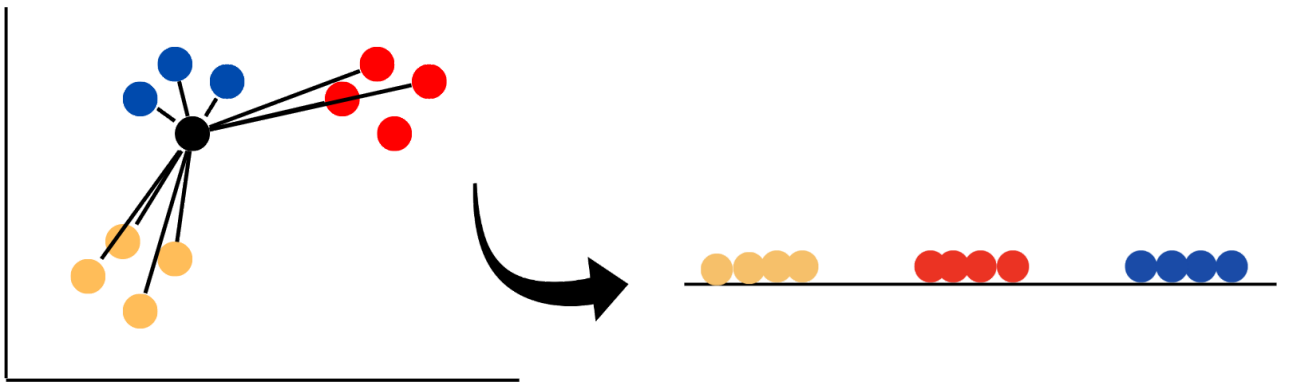
$$cov(X,Y)=E[(X–E[X])(Y–E[Y])]$$

Some remarks about PCA

- We can select only the subset of PCs that explain most of the variance and reduce dimensionality. We can fix a threshold (e.g. explain 80-90% of variability of the original data) but how many PCs are best to choose depends on each particular problem.
- Data must be scaled before applying PCA.
- The PCs are uncorrelated, but they are not "real" variables so results lose interpretability.
- PCA is an unsupervised method, it does not take into account any labeling of the data. It may help in understanding the underlying structure that may or may not match up with any external labeling imposed to the data.

t-SNE

- t-distributed Stochastic Neighbor Embedding
- Tries to preserve *neighborhood*, local structure. Points close to each other in high dimensions will remain close in the projected low dimensions.
- Does not tend to preserve distance between groups.
- It is a non-linear algorithm.
- Non-deterministic, its results can change with different runs.



t-SNE algorithm

Input: A matrix of data points (examples) in n dimensions (features) and the desired number k of dimensions.

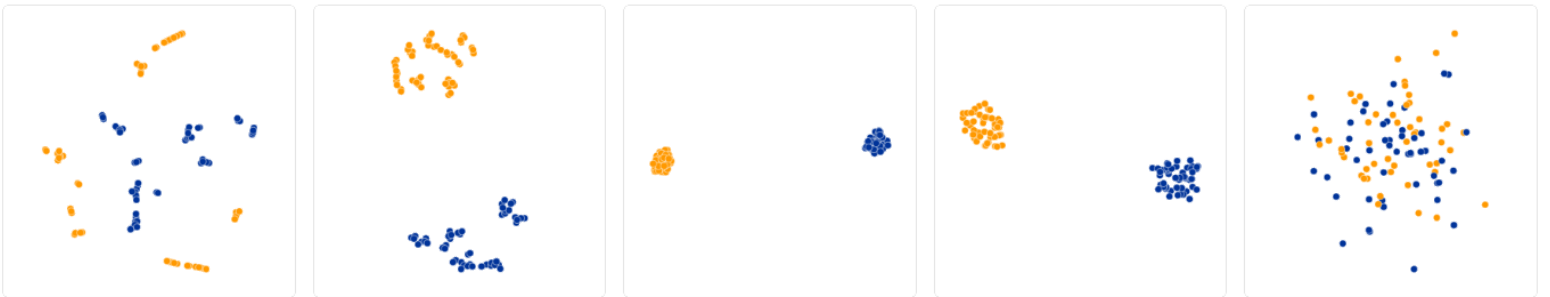
1. Compute a matrix P of distances between each pair of data points. Convert into a conditional Gaussian probability.
2. Generate randomly the same number of data points but in k dimensions. From this new data, compute a matrix Q of distances between each pair of data points and convert into a conditional Student's t-distribution probability.
3. Gradually adjust data points in the k-dimensional space so that the Kullback-Leibler (KL) divergence* between P and Q gets reduced (gradient descent).
4. Continue until a number of iterations has been reached or no improvement is achieved.

Output: A projection of the data points into a k-dimensional space.

*The KL divergence determines how similar two distributions are.

Some remarks about t-SNE

- Perplexity is an important hyperparameter. Can be thought as an analogue to the k nearest neighbours, how many t-SNE will consider to attempt to preserve distances. Lower perplexity values ignore global information favoring more local neighborhoods.



source

- The computational complexity of t-SNE is higher that for PCA.
- No `transform` method for t-SNE in `sklearn`. We cannot learn a transformation and apply it to different data as the training and testing sets.