MATHEMATICAL TOOLS FOR DATA SCIENCE

Leila GHARSALLI (leila.gharsalli@ipsa.fr)

IPSA, AERO 4

2023-2024

GOALS OF THE COURSE

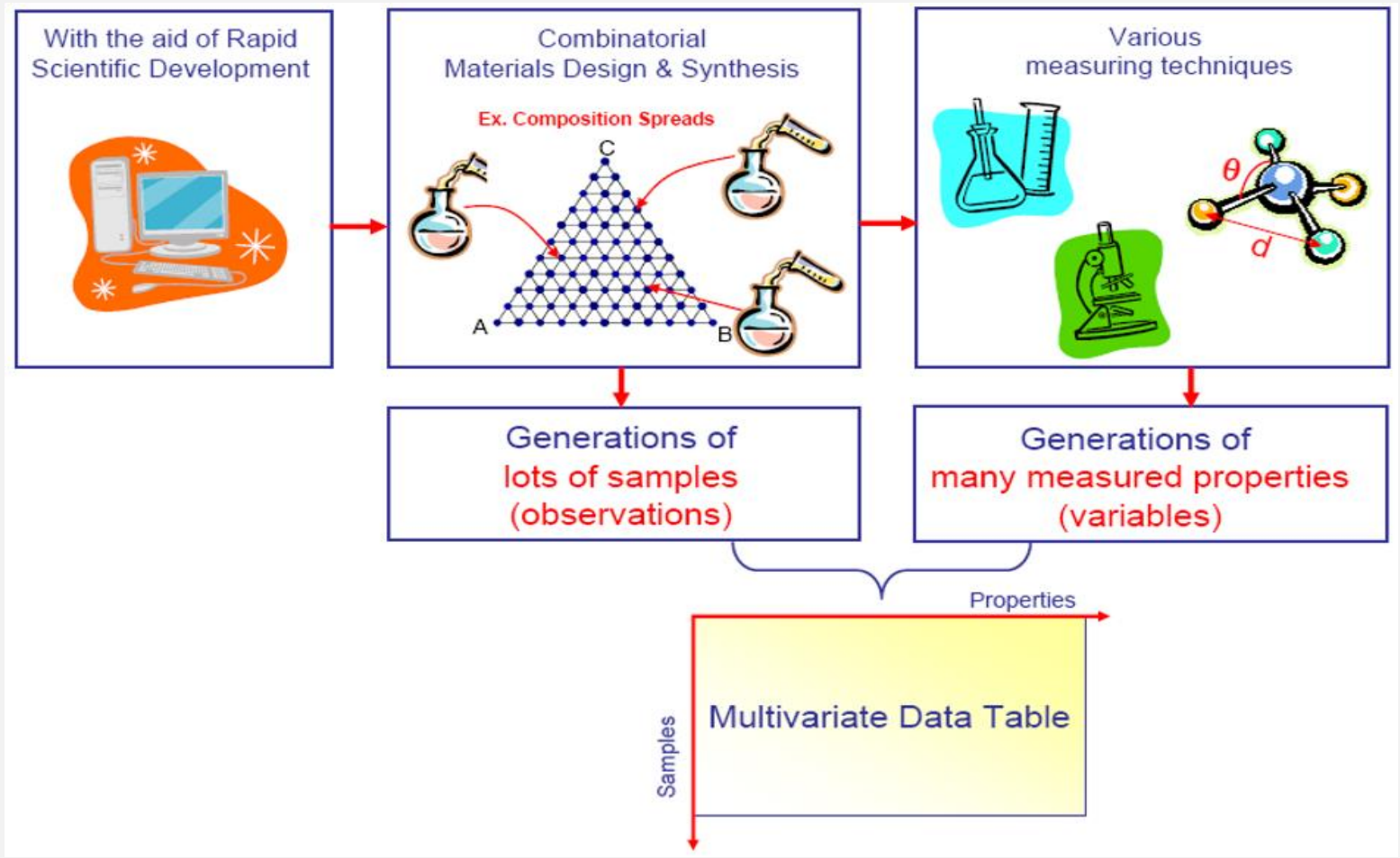
- *Instructor : Leila GHARSALLI*
- *mail to : leila.gharsalli@ipsa.fr*
- Provide a general introduction to data mining and data science.
- Introduce some important tools for solving problems in data science.
- Grading : Participation in class (practical work) as well as a final exam will be graded.
- Main background needed: basic notions in probabilities and statistics, programming skill.

CONTENT OF THE COURSE

1. Introduction
2. Linear regression
3. Sparse regression
4. Classification
5. Principal Component Analysis
6. Clustering
7. Density estimation

INTRODUCTION

INTRODUCTION



INTRODUCTION

Most of the scientific or industrial data is Multivariate data (huge size of data),

- Is all the data useful?

- If not, how do we quickly extract useful information only?

INTRODUCTION

- **Clustering**

- One way to summarize a complex real-valued data point with a single categorical variable.

- **Dimensionality reduction**

- Another way to simplify complex high-dimensional data.

- Summarize data with a lower dimensional real valued vector.

INTRODUCTION

When we use traditional techniques,

- Not easy to extract useful information from the multivariate data,
- Many bivariate plots are needed,
- Bivariate plots, however, mainly represent correlations between variables (not samples).

PRINCIPAL COMPONENTS ANALYSIS (PCA)

PCA

- Consider the following 3D points:

$$\begin{Bmatrix}1\\2\\3\end{Bmatrix}, \begin{Bmatrix}2\\4\\6\end{Bmatrix}, \begin{Bmatrix}4\\8\\12\end{Bmatrix}, \begin{Bmatrix}3\\6\\9\end{Bmatrix}, \begin{Bmatrix}5\\10\\15\end{Bmatrix}, \begin{Bmatrix}6\\12\\18\end{Bmatrix}$$

If each component is stored in a byte, we need $18 = 3 \times 6$ bytes.

PCA

- Looking closer, we can see that all the points are related geometrically: they are all the same point, scaled by a factor:

$$\begin{Bmatrix}1\\2\\3\end{Bmatrix} = 1 \times \begin{Bmatrix}1\\2\\3\end{Bmatrix}, \begin{Bmatrix}2\\4\\6\end{Bmatrix} = 2 \times \begin{Bmatrix}1\\2\\3\end{Bmatrix}, \begin{Bmatrix}4\\8\\12\end{Bmatrix} = 4 \times \begin{Bmatrix}1\\2\\3\end{Bmatrix},$$

$$\begin{Bmatrix}3\\6\\9\end{Bmatrix} = 3 \times \begin{Bmatrix}1\\2\\3\end{Bmatrix}, \begin{Bmatrix}5\\10\\15\end{Bmatrix} = 5 \times \begin{Bmatrix}1\\2\\3\end{Bmatrix}, \begin{Bmatrix}6\\12\\18\end{Bmatrix} = 6 \times \begin{Bmatrix}1\\2\\3\end{Bmatrix}$$

PCA

$$\begin{Bmatrix}1\\2\\3\end{Bmatrix} = 1 \times \begin{Bmatrix}1\\2\\3\end{Bmatrix}, \begin{Bmatrix}2\\4\\6\end{Bmatrix} = 2 \times \begin{Bmatrix}1\\2\\3\end{Bmatrix}, \begin{Bmatrix}4\\8\\12\end{Bmatrix} = 4 \times \begin{Bmatrix}1\\2\\3\end{Bmatrix},$$

$$\begin{Bmatrix}3\\6\\9\end{Bmatrix} = 3 \times \begin{Bmatrix}1\\2\\3\end{Bmatrix}, \begin{Bmatrix}5\\10\\15\end{Bmatrix} = 5 \times \begin{Bmatrix}1\\2\\3\end{Bmatrix}, \begin{Bmatrix}6\\12\\18\end{Bmatrix} = 6 \times \begin{Bmatrix}1\\2\\3\end{Bmatrix}$$

- They can be stored using only 9 bytes (50% savings!): Store one point (3 bytes) + the multiplying constants (6 bytes).

PCA

- Given a set of points, how do we know if they can be compressed like in the previous example?
- The answer is to look into the correlation between the points,
- The tool for doing this is called **Principal Component Analysis**.

PCA

- **Principal Components Analysis** (PCA) is a technique that can be used to simplify a dataset,

- It is **a linear transformation** that chooses a new coordinate system for the data set such that greatest variance by any projection of the dataset comes to lie on the first axis (then called the first principal component), the second greatest variance on the second axis, and so on.

- PCA can be used for reducing dimensionality by eliminating the later principal components.

PCA

- By finding the **eigenvalues** and **eigenvectors** of the covariance matrix, we find that the eigenvectors with the largest eigenvalues correspond to the dimensions that have the **strongest correlation in the dataset**.
- **The covariance** measures the linear connection that may exist between a pair of statistical variables or a pair of quantitative random variables.
- This is **the principal component**.
- PCA is a useful statistical technique that has found application in:

  - fields such as face recognition and image compression.

  - finding patterns in data of high dimension.

PCA

Some reminders: Eigenvector and Eigenvalue:

- Example 1: Find the eigenvalues of $A = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}$
- Example 2: Find the eigenvalues of $A = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$

PCA

Some reminders: Eigenvector and Eigenvalue

- Example 1: Find the eigenvalues of $A = \begin{bmatrix} 2 & -12 \\ 1 & -5 \end{bmatrix}$

$$|\lambda I - A| = \begin{vmatrix} \lambda - 2 & 12 \\ -1 & \lambda + 5 \end{vmatrix} = (\lambda + 1)(\lambda + 2)$$

two eigenvalues: $-1, -2$.

- Example 2: Find the eigenvalues of $A = \begin{bmatrix} 2 & 1 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{bmatrix}$

$$|\lambda I - A| = \begin{bmatrix} \lambda - 2 & -1 & 0 \\ 0 & \lambda - 2 & 0 \\ 0 & 0 & \lambda - 2 \end{bmatrix} = (\lambda - 2)^3$$

An eigenvalue of multiplicity 3: 2

PCA

- Let $x_1, x_2, \dots, x_n$ be a set of $nN \times 1$ vectors and let $\bar{x}$ be their average:

$$x_i = \begin{bmatrix} x_{i1} \\ \vdots \\ x_N \end{bmatrix}, \bar{x} = \frac{1}{n} \sum_{i=1}^{n} \begin{bmatrix} x_{i1} \\ \vdots \\ x_N \end{bmatrix}$$

- Let X be the $N \times 1$ matrix: $X = [x_1 - \bar{x} \;\; x_2 - \bar{x} \;\; \dots \; x_n - \bar{x}]$ (translating the coordinate system to the location of the mean.)

- Let $Q = XX^T$ be the $N \times N$ matrix:

$$Q = XX^T = [x_1 - \bar{x} \;\; x_2 - \bar{x} \;\; \dots \; x_n - \bar{x}] \begin{bmatrix} x_1 - \bar{x} \\ x_2 - \bar{x} \\ \vdots \\ x_n - \bar{x} \end{bmatrix}$$

PCA THEOREM

Theorem:

- Each x_j can be written as $x_j = \bar{x} + \sum_{i=1}^{n} g_{ji} e_i$ where e_i are the n **eigenvectors** of Q with **non-zero eigenvalues**.
- $e_1, e_2, \dots, e_n$ are $N \times 1$ orthonormal vectors (directions in N-Dimensional space)

- The scalars g_{ji} are the coordinates of x_j in the space:

$$g_{ji} = (x_j - \bar{x}) \cdot e_i$$

PCA TO COMPRESS DATA

- Expressing x in terms of e_1,…, e_n has not changed the size of the data.
- However, if the points are highly correlated many of the coordinates of x will be zero or closed to zero.
- Sort the eigenvectors e_i according to their eigenvalue:

$$\lambda_1 \geq \lambda_2 \geq \cdots \lambda_n$$

- Assuming that $\lambda_i \approx 0$, if $i > k$ then

$$x_j \approx \bar{x} + \sum_{i=1}^{i=k} g_{ji} e_i$$

PCA EXAMPLE

- Let's suppose that our data set is 2-dimensional with 2 variables $\boldsymbol{x}, \boldsymbol{y}$ and that the eigenvectors and eigenvalues of the covariance matrix are as follows:

$$v_1 = \begin{bmatrix} 0.6778736 \\ 0.7351785 \end{bmatrix}, \lambda_1 = 1.284028$$

$$v_2 = \begin{bmatrix} -0.7351785 \\ 0.6778736 \end{bmatrix}, \lambda_2 = 0.04908323$$

PCA EXAMPLE

- If we rank the eigenvalues in descending order, we get $\lambda_1 > \lambda_2$, which means that the eigenvector that corresponds to the first principal component (PC1) is v_1 and the one that corresponds to the second component (PC2) is v_2.

- After having the principal components, to compute the percentage of variance (information) accounted for by each component, we divide the eigenvalue of each component by the sum of eigenvalues. If we apply this on the example above, we find that PC1 and PC2 carry respectively 96% and 4% of the variance of the data.

PCA EXAMPLE

- we can either form a feature vector with both of the eigenvectors v_1 and v_2 :

$$\begin{bmatrix} 0.6778736 & -0.7351785 \\ 0.7351785 & 0.6778736 \end{bmatrix}$$

- Or discard the eigenvector v_2 , which is the one of lesser significance, and form a feature vector with v_1 only:

$$\begin{bmatrix} 0.6778736 \\ 0.7351785 \end{bmatrix}$$

- Discarding the eigenvector v_2 will **reduce dimensionality by 1** and will consequently cause a loss of information in the final data set. But given that v_2 was carrying only 4% of the information, the loss will be therefore not important, and we will still have 96% of the information that is carried by v_1 .

PYTHON

```
from sklearn.decomposition import PCA
```

https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html

RECAPITULATION

- PCA is basically a statistical procedure to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables.

- Each of the principal components is chosen in such a way so that it would describe most of them still available variance and all these principal components are orthogonal to each other. In all principal components first, principal component has a maximum variance.

STEPS

- **Step 1:** standardization (range of the continuous initial variables so that each one of them contributes equally to the analysis)
- **Step 2:** covariance matrix computation (see if there is any relationship between variables)
- **Step 3:** compute the eigenvectors and eigenvalues of the covariance matrix to identify the principal components (concepts that we need to compute from the covariance matrix)
- **Step 4:** feature vector (choose whether to keep all the components or discard those of lesser significance)
- **Step 5:** recast the data along the principal component's axes (use the feature vector formed using the eigenvectors of the covariance matrix, to reorient the data from the original axes to the ones represented by the principal components)

EXERCISE

In this exercise, we will perform PCA on breast cancer dataset.

1. Import the python libraries necessary for reading, writing and viewing data.

2. Import breast_cancer from *sklearn.datasets*. the data set has 569 data elements with 30 input attributes. There are two output classes benign and malignant. Due to 30 input features, it is not possible to view this data.

3. Standardize the dataset before PCA. Import PCA from *sklearn.decomposition* then choose the number of main components (we can select it at 3).

4. Check the values of eigen vectors prodeced by principal components using *principal.components_*. What do we remark?

5. Plot principal components for better data visualization 2D then 3D.