

# Principal Component Analysis: dalla teoria alla pratica

Marco Buracchi

Università degli studi di Firenze

18 febbraio 2018



# Sommario

## Principal Component Analysis

Cosa fa?

Come funziona?

## Implementazione Python

Strumenti

Implementazione

## Caso di studio

Attacchi

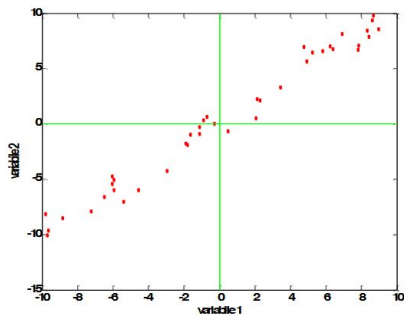
Analisi

Risultati

# PCA

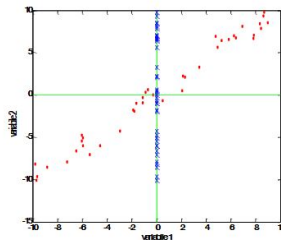
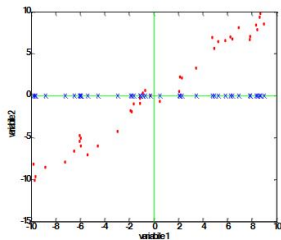
- Trasformazione lineare della matrice dei dati  $\mathcal{X}$
- Misurazione della variazione delle variabili utilizzando un numero minore di "fattori"
- Trasportare il problema in uno spazio  $k$ -variato (generalmente bi-trivariato)
- Semplificazione di visualizzazione e lettura dei dati

# Esempio



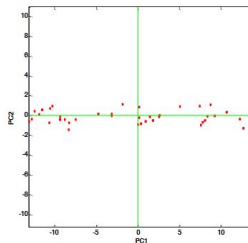
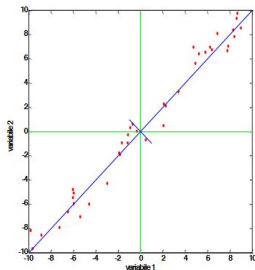
- 40 campioni
- 2 variabili

## Esempio - 2



- Nessuna delle due variabili descrive completamente la variabilità dei dati

# Componenti



- Prendiamo come componenti principali le linee blu
- La prima componente spiega la massima percentuale di variabilità rappresentabile in una dimensione

# Varianza

- Questa percentuale di variabilità può essere calcolata tramite la varianza
- La varianza è un indice della dispersione dei dati lungo una particolare direzione
- La varianza è indipendente dal sistema di riferimento
- Ruotare gli assi mantiene inalterata la varianza totale

## Componenti - 2

- La prima componente cattura quasi tutta la variabilità presente nei dati (99.83%)
- La seconda descrive la rimanente (0.17%)
- Generalizzando, le componenti principali successive spiegano una sempre minore percentuale della variabilità originale
- Le ultime componenti principali descrivono principalmente rumore



# Funzionamento

## 1. Standardizzazione

- Standardizzare i dati (media = 0, varianza = 1)
- Possiamo lavorare con variabili su scale e unità di misure differenti

## 2. Calcolo covarianza/correlazione

- Calcoliamo la matrice S di covarianza

$$S = \frac{1}{n-1} \sum_1^n (x - \mu)(x - \mu)^T$$

- Possiamo usare anche la matrice di correlazione

## 3. Calcolo autovalori/autovettori

- $S \times v = \lambda \times v$

## Funzionamento

### 4. Scelta delle componenti

- Ordiniamo in maniera decrescente gli autovalori ottenuti
- Selezioniamo i primi  $k$
- Costruiamo  $\mathcal{V}$ , la matrice dei rispettivi autovettori

### 5. Rotazione dei dati

- Moltiplichiamo i dati originali per gli autovettori che indicano le direzioni dei nuovi assi (componenti principali)
- I dati ruotati vengono chiamati *score*

$$Sc = \mathcal{X} \times \mathcal{V}$$

## Strumenti utilizzati

- Linguaggio: Python
- Libreria per l'analisi dei dati: *PANDAS*
- Dataset: IRIS

# PANDAS

- Libreria Python, open source, ad alte prestazioni e con licenza BSD
- Strutture dati e strumenti per l'analisi facili da usare (R-like)
  - Serie (unidimensionali)
  - Dataframe (bidimensionali)
- Dati organizzati in maniera *relazionale* o *etichettata*
- Sponsorizzato da NumFocus

A Fiscally Sponsored Project of

**NUMFOCUS**  
OPEN CODE = BETTER SCIENCE

- sviluppo continuo, a livello mondiale e sistema di donazioni a supporto



# Dataset

**Samples**  
(instances, observations)

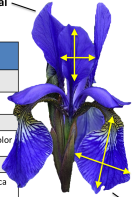
	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...	...	...	...	...	...
50	6.4	3.5	4.5	1.2	Versicolor
...	...	...	...	...	...
150	5.9	3.0	5.0	1.8	Virginica

**Features**  
(attributes, measurements, dimensions)

**Class labels**  
(targets)

**Petal**

**Sepal**



- Dataset IRIS
- 150 misurazioni di fiori iris
- 3 diverse specie

## Caricamento Dataset

```
# download dataset
df = pd.read_csv(
    filepath_or_buffer='https://archive.ics.uci.edu/ml/machine-learning-
    databases/iris/iris.data',
    header=None,
    sep=',')

# scelgo solamente le colonne con i valori di interesse
df.columns=['sepal_len', 'sepal_wid', 'petal_len', 'petal_wid', 'class']
df.dropna(how="all", inplace=True) # Elimina i valori NA
print(df.tail()) #visualizza ultime 5 righe
```

	sepal_len	sepal_wid	petal_len	petal_wid	class
145	6.7	3.0	5.2	2.3	Iris-virginica
146	6.3	2.5	5.0	1.9	Iris-virginica
147	6.5	3.0	5.2	2.0	Iris-virginica
148	6.2	3.4	5.4	2.3	Iris-virginica
149	5.9	3.0	5.1	1.8	Iris-virginica

## Divisione valori

- Matrice valori numerici  $X \in \mathcal{M}^{150 \times 4}$
- Vettore specie  $y \in \mathcal{M}^{150 \times 1}$

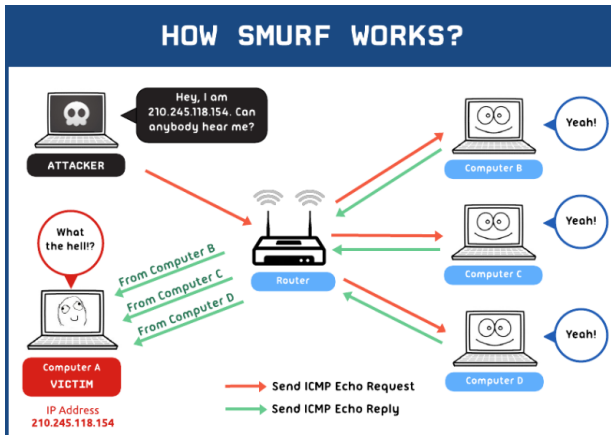
```
# X = tabella con valori , y = etichette  
X = df.ix[:,0:4].values  
y = df.ix[:,4].values
```

# Caso di studio

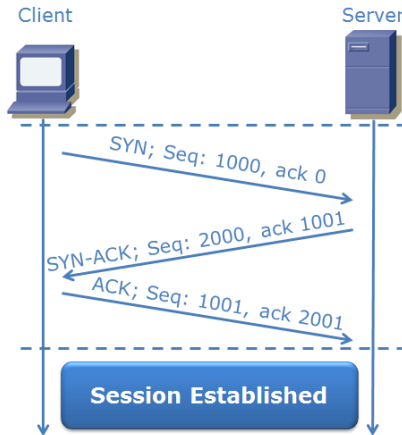
Caso di studio



# Attacco SMURF



# Attacco Neptune



# Attacchi Network Probe

NP

# Analisi

Preprocessing

# Rilevazione

Rilevazione

# Bibliografia



T.W. Anderson.

*An Introduction to Multivariate Statistical Analysis.*

Wiley Series in Probability and Statistics. Wiley, 2003.



K.V. Mardia, J.T. Kent, and J.M. Bibby.

*Multivariate analysis.*

Probability and mathematical statistics. Academic Press, 1979.



Pandas documentation.

<http://pandas.pydata.org/pandas-docs/stable/index.html>.



Python data analysis library.

<http://pandas.pydata.org/>.



Khaled Labib and V. Rao Vemuri.

An application of principal component analysis to the detection and visualization of computer network attacks.

*Annales Des Télécommunications*, 61(1):218–234, Feb 2006.

