Principal Component Analysis (PCA)

Raúl Benítez Iglesias
Universitat Politècnica de Catalunya

raul.benitez@upc.edu

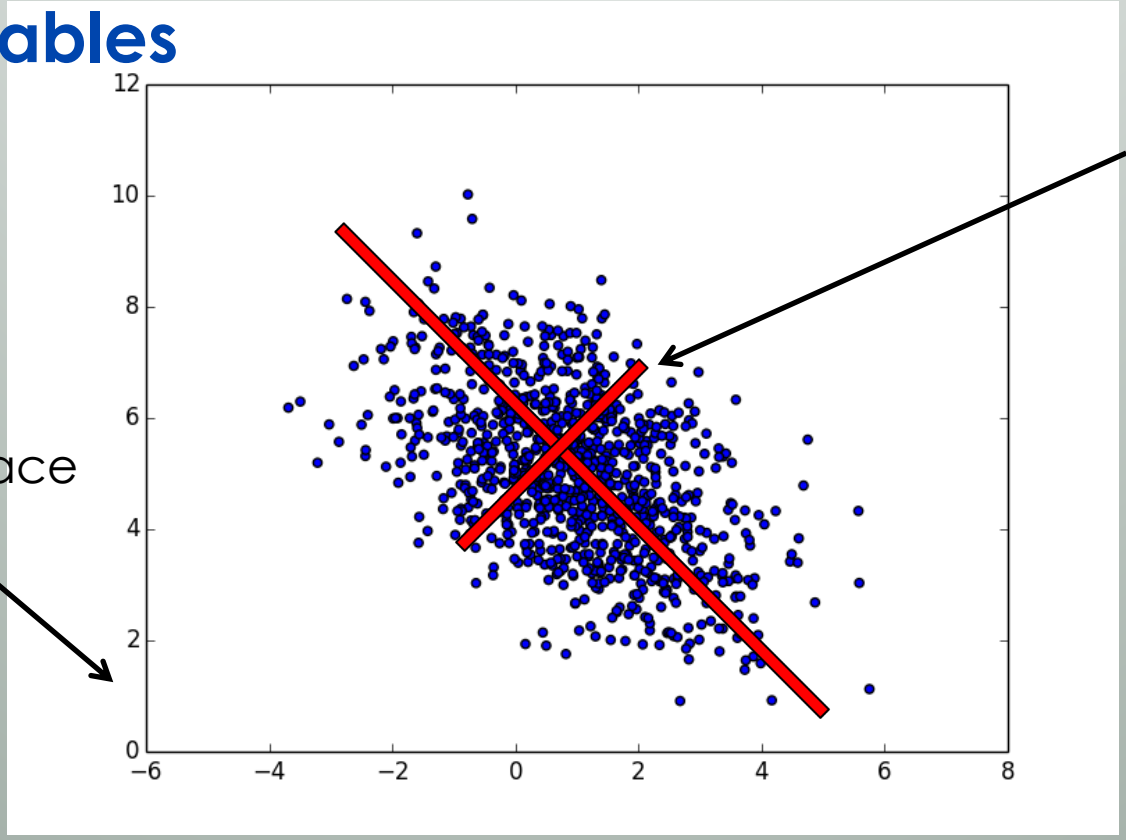
Index

- The idea: What is PCA?
- The math: How is it formulated?
- The code: How to perform PCA in Python?

PCA: The idea

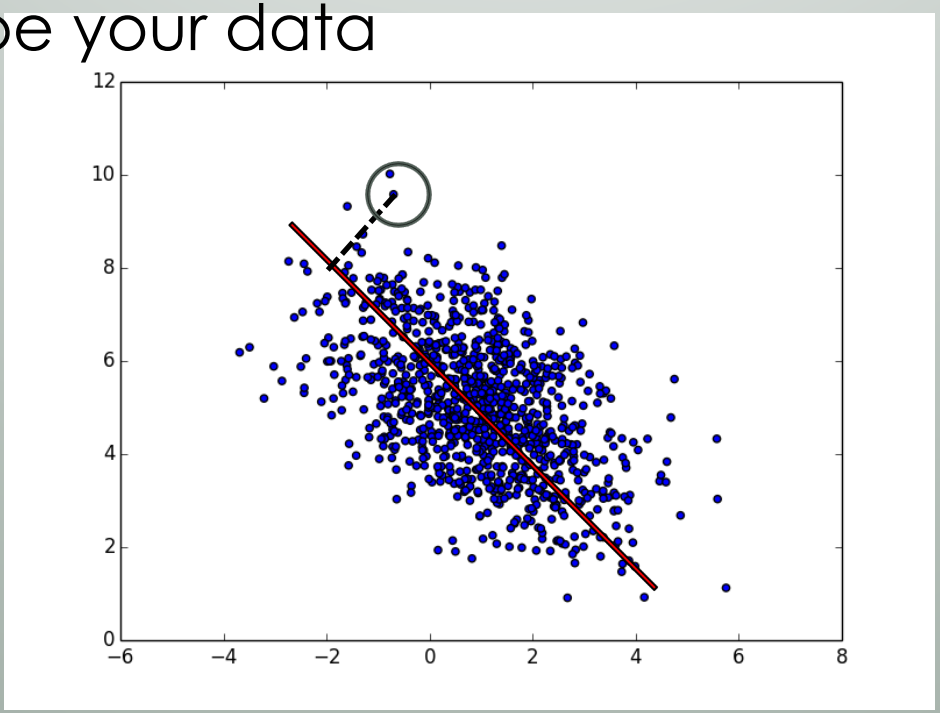
Dimensionality reduction:

- **Simplify the description of your data**
- **Linear transformation of the original variables**



PCA: The idea

- Keep only the most relevant components (How many?)
- Project each data point into the PCA subspace
- Use the projection as a new variable to describe your data



Mathematical formulation:

Data: n variables, m observations

variables

$$A_{m\times n} = \begin{pmatrix} x_1^1 & x_2^1 & \cdots & x_n^1 \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ x_1^3 & x_2^3 & \cdots & x_n^3 \\ \cdots & & & \\ x_1^m & x_2^m & \cdots & x_n^m \end{pmatrix}$$

observations

Covariance matrix of data:

$$C_{n\times n} = (A - \bar{A})^T (A - \bar{A})$$

Procedure

1. Diagonalize covariance matrix:

$$C \cdot \vec{v}_i = \lambda_i \cdot \vec{v}_i, \; i = 1 \ldots n$$

Eigenvectors $\vec{v}_i$: Coordinates of the PCA space

Eigenvalues λ_i : Relevance of each PCA coordinate

2. Keep PCA components by the variance of data explained by each component:

$$100 * \frac{\lambda_i}{\sum_{1=1}^{n} \lambda_i}$$

3. Project data into PCA subspace:

$$P_i = A \cdot v_i$$

Projection of data into the PCA coordinate associated to eigenvector $\vec{v}_i$

Python Implementation

Approach 1: do it manually (scipy)

```
21 # Obtain covariance matrix:
22 A1 = A - A.mean(0)
23 matcov = dot(A1.transpose(),A1)
24
25 # Diagonalization of covariance matrix:
26 valp,vecp = linalg.eig(matcov)
27
28 ind_creciente = argsort(valp) # sort eigenvalues
```

Approach 2: Use sklearn tools

```
46 # Use sklearn rutines to obtain PCA projection:
47
48 from sklearn.decomposition import PCA
49
50 # Projection into 1d PCA space:
51 pca = PCA(n_components=1)
52 #X_r = pca.fit(A).transform(A)
53 X_r = pca.fit_transform(A)
```

More information

Scipy linear algebra-eigenvalues:

http://docs.scipy.org/doc/numpy/reference/generated/numpy.linalg.eig.html

Sklearn tools:

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