Principal Component Analysis

April 16, 2020

Overview

Problem statement

Orthogonal projections and inertia

Optimization

Applications

Issues of PCA

Introduction

- Principal Component Analysis is a method used for some unsupervised learning problems.

Introduction

- Principal Component Analysis is a method used for some unsupervised learning problems.
- It is based on statistical and geometrical considerations .

Introduction

- Principal Component Analysis is a method used for some unsupervised learning problems.
- It is based on statistical and geometrical considerations.
- It was invented by Pearson at the beginning of XXth century but is still used today.

Introduction

- Principal Component Analysis is a method used for some unsupervised learning problems.
- It is based on statistical and geometrical considerations.
- Applications include :
  - dimensionality reduction
  - noise filtering
  - prediction
  - general data visualization and analysis

Example

In this paper, astrophysicists use PCA in order to test a new star temperature prediction method [Bermejo et al., 2013]

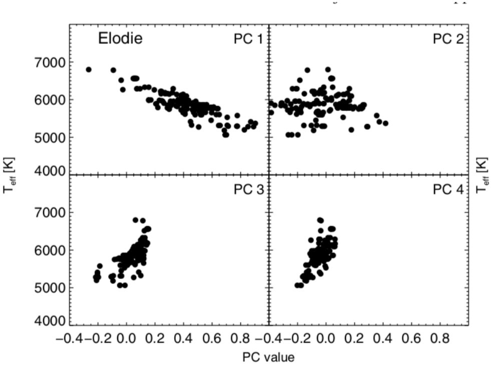
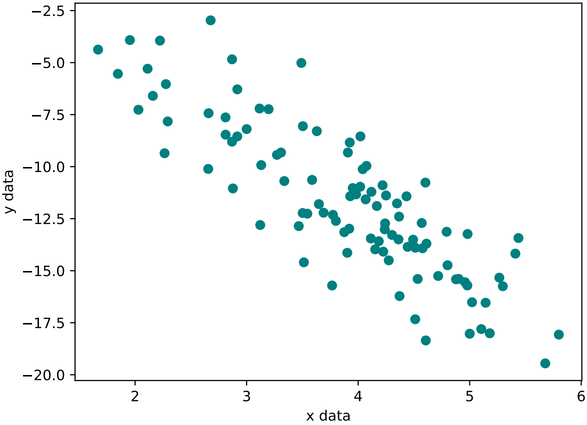


Figure: PCA used in order to predict temperature.

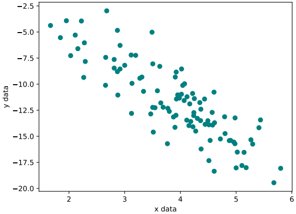
Problem statement

- We have multidimensional data.



Problem statement

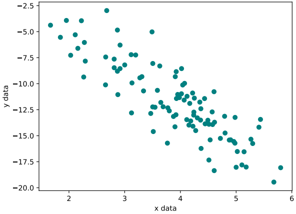
- We have multidimensional data.



We look for the axis that explain or carry the most variations in the data.

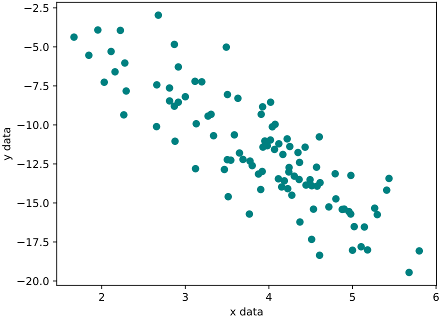
Problem statement

- We have multidimensional data.



We look for the axis that explain or carry the most variations in the data.
Thoses axes will be called the **principal components.**

Visual intuition



When looking at this image, what axis would you suggest in order to explain the biggest variation among the data ?

Formalization

We need a mathematical criterion in order to formalize this intuition.

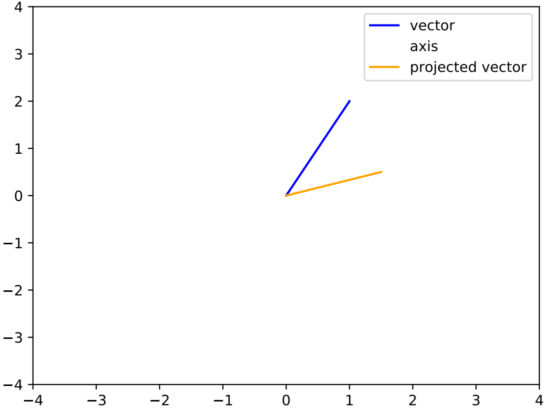
Formalization

We need a **mathematical criterion** in order to formalize this intuition.
In the case of a larger problem with more dimensions, it will not be possible to use visual feedback in order to choose the principal components (the axis).
Furthermore, even in 2D, using only visualization, we can only do a **approximation** of the most relevant axis.

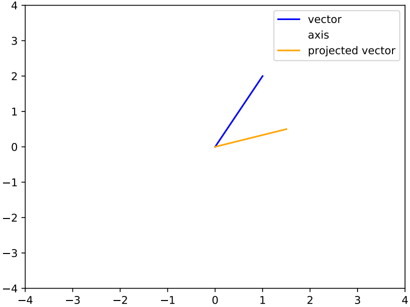
Orthogonal projections

We will restate the problem in a mathematical way, using tools coming from **linear algebra.**
In particular, the concept of **orthogonal projection** is essential.

Orthogonal projection



Orthogonal projection



Exercice 1 : **Mathematical problem** Can you link orthogonal projections to the problem of finding the most relevant axis ?

Inertia

- The **inertia** related to an axis will be a measure of the quality of an axis.
- If x_i is a sample from n samples, and Δ an axis, we call $p_{x_i \to \Delta}$ the orthogonal projection of x_i on the axis Δ.

Inertia

- The **inertia** related to an axis will be a measure of the quality of an axis.
- If x_i is a sample from n samples, and Δ an axis, we call $p_{x_i \rightarrow \Delta}$ the orthogonal projection of x_i on the axis Δ.
- The inertia related to Δ is:

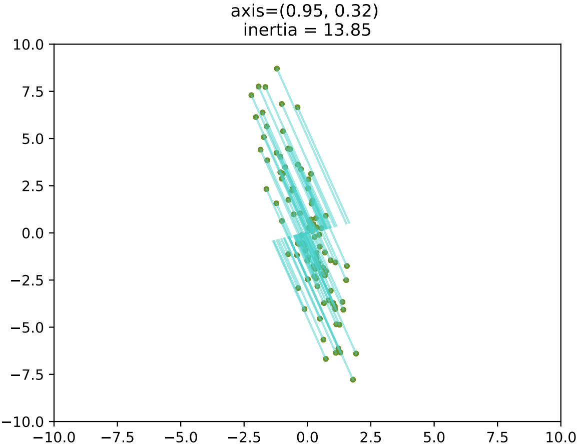
$$I_\Delta = \frac{1}{n} \sum_{i=1}^{n} d^2(x_i, p_{x_i \rightarrow \Delta}) \tag{1}$$

Inertia

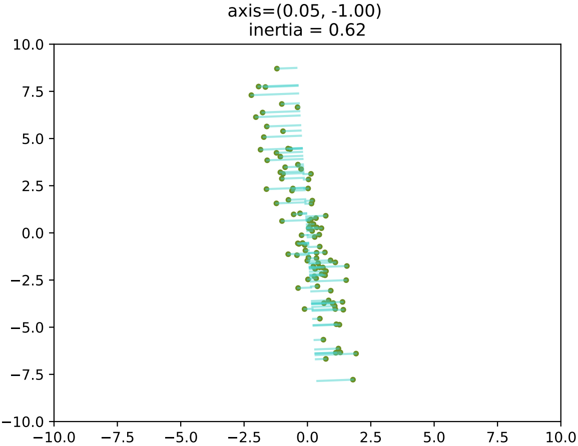
Exercice 2 : **No inertia**

In what situations could we have $I_\Delta = 0$?

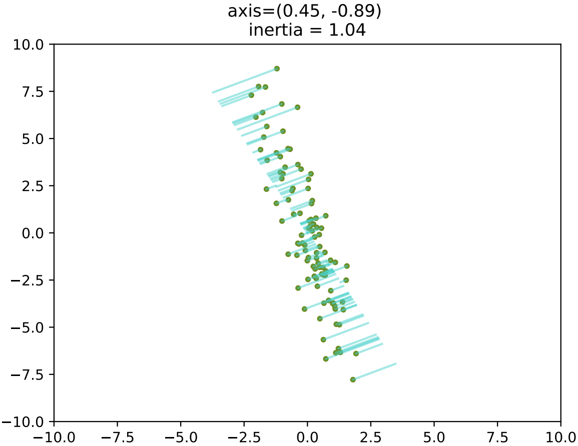
Inertia


axis=(0.95, 0.32)
inertia = 13.85
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

Inertia


axis=(0.05, -1.00)
inertia = 0.62
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

Inertia


axis=(0.45, -0.89)
inertia = 1.04
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
−7.5
−10.0
−10.0
−7.5
−5.0
−2.5
0.0
2.5
5.0
7.5
10.0

Inertia

Exercice 3 : **Computing an inertia**

Please **cd pca/custom data/** and use the file **inertia.py** in order to represent the projections of the data onto a chosen axis and to compute the inertia.

What is your optimal axis ?

Remark

Minimizing I_Δ is the same as maximizing I_{Δ^*} where Δ^* is the supplementary orthogonal space.

Several principal components

- In $2D$ (like in the former example), we can only have 1 or 2 principal component.

Several principal components

- In $2D$ (like in the former example), we can only have 1 or 2 principal component.
- However, when the data have more dimensions, it is possible to have **several principal components.**
- the data are then projected on these components.

Link with expected values

We will connect PCA and the inertia to **expected values and variances.**

Expected value (esprance)

- For a discrete random variable X that takes the values x_i with probability p_i:

$$E(X) = \sum_{i=1}^{n} p_i x_i \tag{2}$$

- For a continuous random variable X with density p(x):

$$E(X) = \int xp(x)dx \tag{3}$$

Expected value (esprance)

Exercice 4 : Computing an expected value

- For a discrete random variable X that takes the values x_i with probability p_i:

$$E(X) = \sum_{i=1}^{n} p_i x_i \tag{4}$$

- For a continuous random variable X with density p(x):

$$E(X) = \int x p(x) dx \tag{5}$$

Compute the expected value of the dice game.

Variance

$$var(X) = E\left(\left(X - E(X)\right)^2\right) \tag{6}$$

Variance and Covariance

$$var(X) = E\left(\left(X - E(X)\right)^2\right) \tag{7}$$

$$cov(X, Y) = E\left(\left(X - E(X)\right)\left(Y - E(Y)\right)\right) \tag{8}$$

Variance and Inertia

Exercice 5 : **Linking variance and inertia**
What is the relationship between variance and inertia ?

Optimization

- In real applications, algorithms or analytic solutions are used in order to find the optimal axes.

Optimization with analytic solution

- Methods include the famous **Lagrange multiplier method**.
- This involves computing the **covariance matrix.**

Optimization with algorithms

- One can also use an approximation method to find the first principal component, called the **power iteration** algorithm.
- It is useful when the dimensionality is high, when the Lagrange multiplier method becomes too slow, since it involves computation of the covariance matrix.

Optimization with algorithms

Exercice 6 : **Complexity of computing the covariance matrix.**
If n is the number of datapoints, and p the number of variables, what is the complexity of this calculation ?

Optimization with algorithms

Exercice 6 : **Complexity of computing the covariance matrix.**
If n is the number of datapoints, and p the number of variables, what is the complexity of this calculation ?
The power iteration algorithm only involves cnp operations, where c is a constant way smaller than p, so it is way faster.

PCA with sklearn

- We will use sklearn in order to perform PCA on our dataset.
- `https://scikit-learn.org/stable/modules/decomposition.html`
- `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html`

PCA with sklearn

Exercice 7 : **Sklearn**

Please use the file **pca_sklearn.py** in order to find the principal components on the dataset of the previous exercise.

Applications

- Let us review some applications of the method.

Iris dataset

- We can perform PCA on the iris dataset as in the file **iris/pca-iris.py**.

Iris dataset

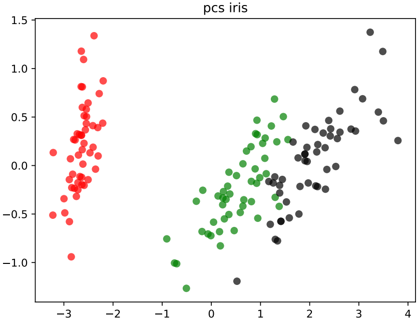
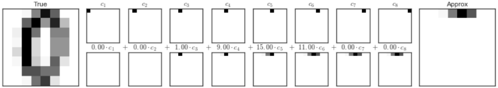


Figure: PCA performed on the iris dataset. We see that the principal components are able to separate the data.

PCA on digits

- We will also perform the PCA on a dataset consisting in 8×8 pixels images of digits.
- The idea is to see if the PCA can help us visualize structure in the data.

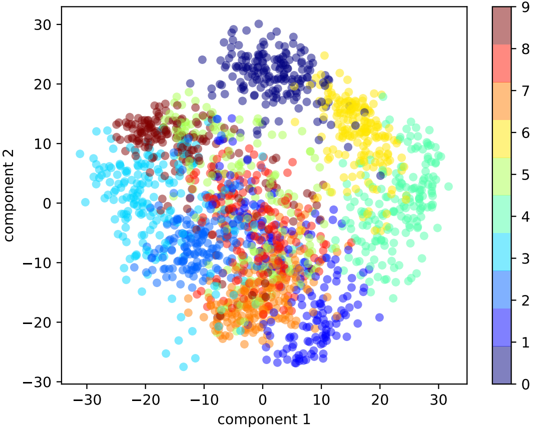

True
c_1
c_2
c_3
c_4
c_5
c_6
c_7
c_8
Approx
$0.00 \cdot c_1 + 0.00 \cdot c_2 + 1.00 \cdot c_3 + 9.00 \cdot c_4 + 15.00 \cdot c_5 + 11.00 \cdot c_6 + 0.00 \cdot c_7 + 0.00 \cdot c_8$

PCA on digits

Exercice 8 : **Performing PCA**

Please use the file **pca_digits.py** in order to apply PCA to this dataset.

PCA on digits



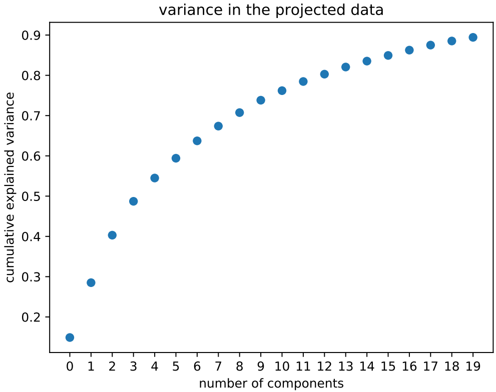
Number of components

What is a relevant number of components for PCA ?

Number of components

Exercice 9 : **Choosing the number of components**
Use the file **pca_digits_variance.py** in order to determine how many components are necessary in order to keep 75% of the variance in the digits dataset.

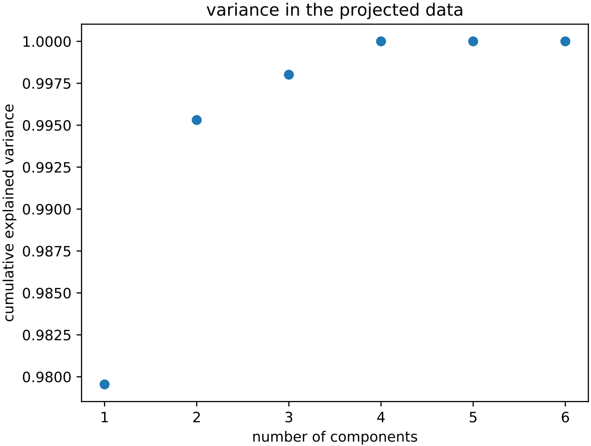
Number of components



Number of components

Exercice 10 : **Redundancy**
What happens with the data contained in **redundancy/redundant_data.npy** ? You can analyze them with the file **redundancy/pca_variance.py**.

Number of components



Number of components

Conclusion: PCA can help detemine whether some components carry no information in the data.

Shortcomings of PCA

PCA is sensitive to :

- outliers
- initial data scaling

Shortcomings of PCA

PCA is sensitive to :

- outliers
- initial data scaling
- Many variants and heuristics exist on this topic.

Asset of PCA

Reducing the dimensionality of the data might be very helpful in a situation where you need to train a classification algorithm on the projected data. The algorithm might be way faster on data that are in a smaller space. However, enought information should be kept during the reduction, hence the necessity of heuristics.

References

Bermejo, J. M., Ramos, A. A., and Prieto, C. A. (2013).
Astrophysics A PCA approach to stellar effective temperatures.

95:1–9.