Mathematic Analysis with Matlab

Lecture 10: Principal Component Analysis

Lecturer: *Dr*. Lian-Sheng Wang

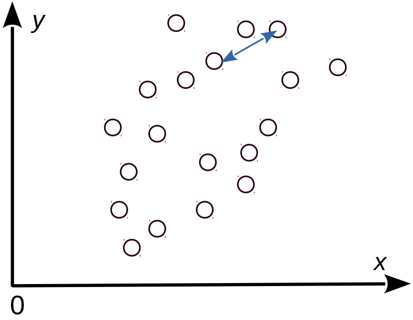
Fall Semester 2022

Contact: lswang@xmu.edu.cn

Outline

PCA: the idea (1)

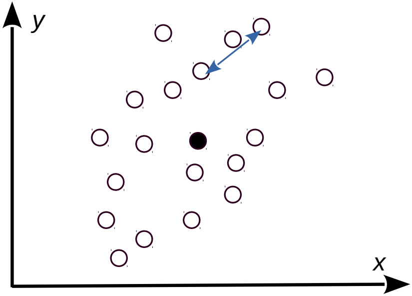
- Let's start with a simple example
- Given a group of 2D data shown in the figure



- We want to reduce their dimension to 1D
- While preserving their relative distances as much as possible

PCA: the idea (2)

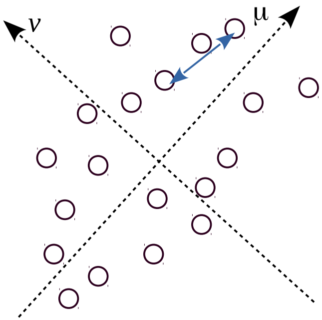
- The distribution of the data is apart from the origin
- Let's first move the axis origin to the center of the distribution
- This is done by



- Notice that translation does not change their distance
- Now we want to find out the way how to project the data from 2D to 1D

PCA: the idea (3)

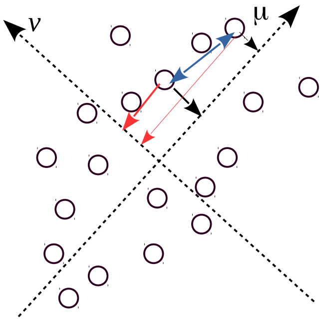
- Suppose we intuitively know they should be projected either along direction μ or v



- Which one we should choose?

PCA: the model (1)

- Observation is that data should be projected on the direction that distribution shows largest variances



- Given projection vector is μ
- We should maximize: $(x^T \mu)^2$

PCA: the model (2)

- Given there are m instances/ data items here
- We should maximize following function

$$\begin{array}{rc} \text{Maximize} & \frac{1}{m}\sum_{i=1}^{m}(x_i^T\mu)^2 \\ \text{s.t.} & \mu^T\mu = 1 \end{array}$$

- Above function is rewritten as

$$\begin{array}{rc} \text{Maximize} & \frac{1}{m}\sum_{i=1}^{m}\mu^T x_i x_i^T \mu \\ \text{s.t.} & \mu^T\mu = 1 \end{array}$$

- This is a typical quadratic optimization problem
- Can be easily solved by **Lagrangian multiplier**

PCA: the model (3)

- Given following problem:

$$\begin{array}{cc} \text{Maximize} & \frac{1}{m}\sum_{i=1}^{m} \mu^T x_i x_i^T \mu \\ \text{s.t.} & \mu^T \mu = 1 \end{array}$$

- We define its Lagrangian function as:

$$L = \mu^T\left(\frac{1}{m}\sum x_i x_i^T\right)\mu + \lambda[\mu^T\mu - 1]$$

- Take partial derivative on μ and λ, we have

$$\frac{\partial L}{\partial \mu} = \frac{1}{m}\sum x_i x_i^T \mu - \lambda\mu = 0 \quad \frac{\partial L}{\partial \lambda} = \mu^T\mu - 1 = 0$$

PCA: the model (4)

- Take partial derivative on μ and λ, we have

$$\frac{\partial L}{\partial \mu} = \frac{1}{m}\sum x_i x_i^T \mu - \lambda\mu = 0 \quad \frac{\partial L}{\partial \lambda} = \mu^T \mu - 1 = 0$$

- With above results, we have

$$\begin{cases} \frac{1}{m}\sum x_i x_i^T \mu = \lambda\mu \\ \mu^T \mu = 1 \end{cases}$$

- Now it is clear, when μ is the eigenvector of matrix $\frac{1}{m}\sum x_i x_i^T$
- The objective function attains maximum
- $\frac{1}{m}\sum x_i x_i^T$ is nothing more the covariance matrix
- It is semi-definite

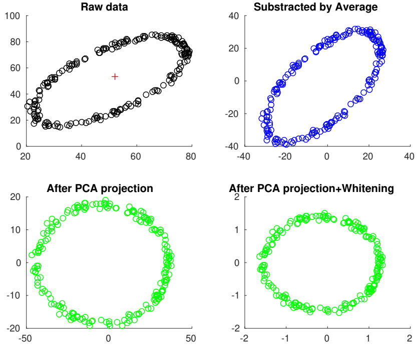
PCA: the procedure

- Now we summarize the PCA learning steps:
  1. Translate the data by substracting mean
  2. Calculate covariance matrix, then take the average
  3. Perform eigenvalue decomposition
  4. Sort the eigenvalues in descending order
  5. Shuffle the eigenvectors accordingly
- Given new data z comes, procedure below performs projection
- Given we reduce the data dimension from D to P
  1. Substract mean from the z'=z-mean
  2. Take inner-product between z' and the first P eigenvectors
  3. The P inner-products are organized as projected vector **w** for input vector **z**

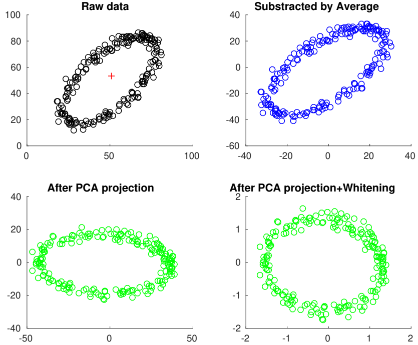
PCA: Matlab commands to be called

1. Plots multiple figures on one panel
   - subplot(2, 2, 1);
   - title('Raw data');
   - subplot(2, 2, 2);
   - title('Substracted by Mean');
   - subplot(2, 2, 3);
   - title('Data after PCA projection');
   - subplot(2, 2, 4);
   - title('Data after PCA projection+Whitening');
2. Plots 2D points
   - scatter(D(:,1), D(:,2));
3. Duplicate a matrix for **N** times
   - repmat(rand(1,2), N, 1);

PCA: results you should achieve



(a) Data-1



(b) Data-2

PCA: outlining the codes

1. avg = mean(Data);
2. Display raw data and its center
3. N = max(size(Data));
4. avg = repmat(avg, N, 1);
5. Calculate covariance matrix
6. Calculate eigenvalues of the matrix
7. V = [v(:,2) v(:,1)];
8. E = [e(2,2),e(1,1)];
9. E = sqrt(E);
11. mData = Data-avg;
12. Display raw data after mean substr.
13. Perform projection;
14. prj(:,1) = prj(:,1)/E(1);
15. prj(:,2) = prj(:,2)/E(2);
16. Display prj;

PCA: the codes (1)

```
function LPCA(rData)
    clf;
    subplot(2, 2, 1);
    avg   = mean(rData)

    scatter(avg(1,1),avg(1,2),'r+');
    hold on;
    scatter(rData(:,1), rData(:,2), 'k');
    title('Raw data');

    N     = max(size(rData));
    avg   = repmat(avg, N, 1);
    covr  = (rData-avg)/sqrt(N-1);
    covr  = covr'*covr;
    [v, e] = eig(covr);
    V     = [v(:,2) v(:,1)];
    E     = [e(2,2),e(1,1)];
    E     = sqrt(E)
    mData = rData-avg;
    subplot(2, 2, 2);
    scatter(mData(:,1), mData(:,2), 'b');
    title('Substracted by Average');
```

PCA: the codes

```
    prjV1   = mData*V;
    prjV2   = (mData*V);
    prjV2(:,1) = prjV2(:,1)/E(1);
    prjV2(:,2) = prjV2(:,2)/E(2);
    subplot(2, 2, 3);
    scatter(prjV1(:,1),prjV1(:,2), 'g');
    title('After␣PCA␣projection');

    subplot(2, 2, 4);
    scatter(prjV2(:,1),prjV2(:,2), 'g');
    title('After␣PCA␣projection+Whitening');
end
```

PCA in another case (1)

$$\begin{cases} \frac{1}{m}\sum x_i x_i^T \mu = \lambda\mu \\ \mu^T \mu = 1 \end{cases}$$

- Given $A = \frac{1}{m}\sum x_i x_i^T$, we are doing eigenvalue decomposition on $A_{d\times d}$
- Given $x_i \in R^d$, we have m samples $(m \geq d)$
- What if $m < d$?

PCA in another case (2)

- Let's rewrite the result we have

$$A\mu = \lambda\mu, \tag{1}$$

$$\text{where } A = \frac{1}{m}\sum x_i x_i^T$$

- A can be written as $A = \frac{1}{m}XX^T$

$$X = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(m)} \\ x_2^{(1)} & x_2^{(2)} & \cdots & x_2^{(m)} \\ \cdots & \cdots & & \\ x_d^{(1)} & x_2^{(2)} & \cdots & x_d^{(m)} \end{bmatrix}.$$

- We are therefore solving $XX^T\mu = \lambda\mu$

PCA in another case (3)

- Let's look at *Gramian* matrix

$$X^T X\mu^* = \lambda\mu^*, \tag{2}$$

where

$$X = \begin{bmatrix} x_1^{(1)} & x_1^{(2)} & \cdots & x_1^{(m)} \\ x_2^{(1)} & x_2^{(2)} & \cdots & x_2^{(m)} \\ \cdots & \cdots & & \\ x_d^{(1)} & x_2^{(2)} & \cdots & x_d^{(m)} \end{bmatrix}.$$

- $X^T X$ is $m \times m$, which is solvable.
- μ^* is the eigenvector of $X^T X$
- <u>How to relate it to XX^T?</u>

Think about it in two minutes...

PCA in another case (4)

- Let's look at *Gramian* matrix

$$X^T X \mu^* = \lambda \mu^* \tag{3}$$

- X is $m \times m$, which is solvable.
- μ^* is the eigenvector of $X^T X$
- How to relate it to XX^T?
- Left-multiplication X to Equation (3), we have

$$(XX^T) X \mu^* = \lambda X \mu^* \tag{4}$$

- That means $X\mu^*$ is the eigenvector of XX^T

Q & A

Thanks for your attention!