Homework 2: PCA & SVD

Leslie O'Bray
leslie.obray@bsse.ethz.ch

Christian Bock
christian.bock@bsse.ethz.ch

Dr. Michael Moor
michael.moor@bsse.ethz.ch

Prof. Dr. Karsten Borgwardt
karsten.borgwardt@bsse.ethz.ch

Submission deadline: 31.03.2021 at 14:00

Objectives

The goals of this homework are:

- to show that a standardised covariance matrix gives the same results as a correlation matrix using Pearson's correlation coefficient.
- to implement PCA using SVD in `Python`.
- to compare PCA using eigen-value decomposition vs. SVD.
- to implement the Moore-Penrose inverse using SVD.

Exercises

Exercise 1

Sometimes, the correlation matrix using Pearson's Correlation coefficient is used to perform a PCA instead of a covariance matrix. Show that the correlation matrix using Pearson's Correlation coefficient is identical to a covariance matrix computed from standardised data.

For this purpose, let's assume you are given two random variables $x \in \mathbb{R}^n$ and $y \in \mathbb{R}^n$, where n is the number of samples. Pearson's correlation coefficient r is defined as:

$$r = \frac{\frac{1}{n-1}\sum_{i=1}^{n}\left[(x_i - \mu_x)(y_i - \mu_y)\right]}{\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu_x)^2}\sqrt{\frac{1}{n-1}\sum_{i=1}^{n}(y_i - \mu_y)^2}},$$

where μ_x and μ_y are the means of the two random variables x and y, respectively:

$$\mu_x = \frac{1}{n}\sum_{i=1}^{n} x_i, \qquad \mu_y = \frac{1}{n}\sum_{i=1}^{n} y_i.$$

The random variables x and y have been standardised before the covariance matrix has been computed, as follows:

$$x\prime = \frac{x - \mu_x}{\sigma_x}, \qquad y\prime = \frac{y - \mu_y}{\sigma_y},$$

where σ_x and σ_y are the standard deviations of x and y, respectively. The covariance is defined as follows:

$$\sigma_{xy} = \frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu_x)(y_i - \mu_y).$$

Show all intermediate steps in your solution for this homework!

Exercise 2

In the following exercise you will compare the PCA implementation from the first Homework to a PCA implementation using singular value decomposition (SVD). This time, you will apply the methods on the Iris dataset, which is a collection of 150 iris flower samples of three closely related species (50 samples per species). For each sample four features have been measured, the length and the width of both the sepals and the petals in centimetres. The dataset is already loaded in your main script file (run_hw2.py). You have to extend the pca.py file from the first homework with a variety of new functions (you can also use the pca.py file attached to this homework). In addition, you also have to modify the main file (run hw2.py) in order to perform the following experiments:

(a) Perform a PCA like we did in the first homework on the new Iris data using the implementation from Homework 1 (plot transformed data, highlight samples according to their class, how much variance can be explained, show results in your solution)!

(b) Write down the pseudo-code for a PCA using SVD!

(c) Implement the following two functions: `zeroMean` and `computePCA_SVD` (Note: You are allowed to use the SVD solver from SciPy)!

(d) Use the SVD based PCA on the Iris data and compare the results to those from part (a). Report and compare all results from both experiments in your solution report (plot transformed data, highlight samples according to their class, how much variance can be explained).

(e) Do you see any advantage in using SVD for PCA instead of an eigen-value decomposition?

Exercise 3

The singular value decomposition for a matrix X is given by:

$$X = L\Delta R^T$$

The Moore-Penrose pseudo-inverse X^+ of a matrix X is defined as

$$X^+ = R\Delta^+ L^T,$$

where the diagonal of the matrix Δ^+ are the reciprocals of the singular values: $\frac{1}{\delta_1}, \ldots, \frac{1}{\delta_r}$ and r is the number of non zero singular values (e.g. all singular values that are above a certain tolerance threshold t, e.g. 10^{-15}).

(a) Implement the Moore-Penrose pseudo-inverse by modifying the function `compute_pinv` in the file `pinv.py`! Update the main file `run_hw2.py` and compute the pseudo inverse using your implemented solution on the original Iris data!

(b) Update the main file `run_hw2.py` to check that the pseudo-inverse X^+ satisfies the following two criteria:

   (a) $XX^+X = X$

   (b) $X^+XX^+ = X^+$

Command-line arguments

Your program should generate all the plots and should output all necessary information in a readable format.
The file `run_hw2.py` must be executable using the following command:

```
$ python run_hw2.py
```

Note: **Zero** points for the **programming part** of this homework if your script is **not** executable with the above shown command line argument!

Grading and submission guidelines

This homework is worth a total of 100 points. Table 1 shows the points assigned to each exercise/question. Follow the submission guidelines posted on the Moodle webpage. Refer

Table 1: Grading key for homework 2

30 pts.	**Exercise 1**	
	30 pts.	Exercise 1
50 pts.	**Exercise 2**	
	5 pts.	Exercise 2.a
	20 pts.	Exercise 2.b
	10 pts.	Exercise 2.c
	10 pts.	Exercise 2.d
	5 pts.	Exercise 2.e
20 pts.	**Exercise 3**	
	15 pts.	Exercise 3.a
	5 pts.	Exercise 3.b

to the document titled "General guidelines for homework sheets" (link named "General guidelines").

Acknowledgments

This exercise was created by Karsten Borgwardt, Dean Bodenham, Dominik Grimm, Xiao He and Damian Roqueiro.