Universität Zürich HS 2021
Institut für Informatik

Student Name: Youngseo Kim
Matrikel-Nr: 20-744-470

# Numerical Methods in Informatics - Exercise 4

Hand out: 15.11.2021 - Due to: 28.11.2021

Please upload your solutions to the Olat system.

# Theory

### 4.1 Eigendecomposition

**a) (10 Min, 10 Points)** Calculating Eigenvalues
Please calculate the Eigenvalues of the following matrix step by step:

$$A = \begin{pmatrix} 4 & 0 & 5 & 0 \\ 0 & \frac{5}{2} & 0 & \frac{1}{2} \\ 5 & 0 & 4 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{5}{2} \end{pmatrix}$$

Find $\lambda \in \mathbb{R}$, such that $(A-\lambda I)x=0$ has a non-trivial solution.

$$A-\lambda I = \begin{bmatrix} 4 & 0 & 5 & 0 \\ 0 & 5/2 & 0 & 1/2 \\ 5 & 0 & 4 & 0 \\ 0 & 1/2 & 0 & 5/2 \end{bmatrix} - \begin{bmatrix} \lambda & 0 & 0 & 0 \\ 0 & \lambda & 0 & 0 \\ 0 & 0 & \lambda & 0 \\ 0 & 0 & 0 & \lambda \end{bmatrix} = \begin{bmatrix} 4-\lambda & 0 & 5 & 0 \\ 0 & 5/2-\lambda & 0 & 1/2 \\ 5 & 0 & 4-\lambda & 0 \\ 0 & 1/2 & 0 & 5/2-\lambda \end{bmatrix}$$

Row reduction

$$\left[\begin{array}{cccc|c} 4-\lambda & 0 & 5 & 0 & 0 \\ 0 & 5/2-\lambda & 0 & 1/2 & 0 \\ 5 & 0 & 4-\lambda & 0 & 0 \\ 0 & 1/2 & 0 & 5/2-\lambda & 0 \end{array}\right] R_3 \to R_3 - 5/(4-\lambda) R_1$$

$$(4-\lambda) - 25/4-\lambda = \frac{(4-\lambda)^2-25}{4-\lambda} = \frac{16-8\lambda+\lambda^2-25}{4-\lambda} = \frac{\lambda^2-8\lambda-9}{4-\lambda} \quad (\lambda \neq 4)$$

$$\Leftrightarrow \left[\begin{array}{cccc|c} 4-\lambda & 0 & 5 & 0 & 0 \\ 0 & 5/2-\lambda & 0 & 1/2 & 0 \\ 0 & 0 & \frac{(\lambda-9)(\lambda+1)}{4-\lambda} & 0 & 0 \\ 0 & 1/2 & 0 & 5/2-\lambda & 0 \end{array}\right] R_4 \to R_4 - \left(\frac{1}{5-2\lambda}\right) R_2$$

$$\Leftrightarrow \left[\begin{array}{cccc|c} 4-\lambda & 0 & 5 & 0 & 0 \\ 0 & 5/2-\lambda & 0 & 1/2 & 0 \\ 0 & 0 & \frac{(\lambda-9)(\lambda+1)}{4-\lambda} & 0 & 0 \\ 0 & 0 & 0 & \frac{2(\lambda-2)(\lambda-3)}{5-2\lambda} & 0 \end{array}\right]$$

$$\frac{5}{2} - \lambda - \frac{1}{2}\left(\frac{1}{5-2\lambda}\right) = \frac{5-2\lambda}{2} - \frac{1}{2(5-2\lambda)} = \frac{(5-2\lambda)^2 - 1}{2(5-2\lambda)}$$

$$= \frac{4\lambda^2 - 20\lambda + 25 - 1}{2(5-2\lambda)} = \frac{2\lambda^2 - 10\lambda + 12}{5-2\lambda} = \frac{2(\lambda^2 - 5\lambda + 6)}{5-2\lambda} = \frac{2(\lambda-2)(\lambda-3)}{5-2\lambda}$$

$$(\lambda \neq 5/2)$$

$$\det(A) = (\cancel{4-\lambda}) \cdot \left(\frac{5}{2} - \lambda\right) \cdot \left(\frac{(\lambda-9)(\lambda+1)}{\cancel{4-\lambda}}\right) \cdot \left(\frac{2(\lambda-2)(\lambda-3)}{5-2\lambda}\right)$$

$$= \left(\cancel{\frac{5}{2} - \lambda}\right)(\lambda-9)(\lambda+1) \cdot \left(\frac{(\lambda-2)(\lambda-3)}{\cancel{5/2 - \lambda}}\right)$$

$$= (\lambda-9)(\lambda+1)(\lambda-2)(\lambda-3) = 0$$

$\therefore$ eigenvalues $= \{-1, 2, 3, 9\}$

# Practice

In many real world scenarios, you get some noisy data, which doesn't perfectly represent the phenomenon, you're observing. E.g. in Figure 1, you can see some datapoints, which are measured from any linear relationship.

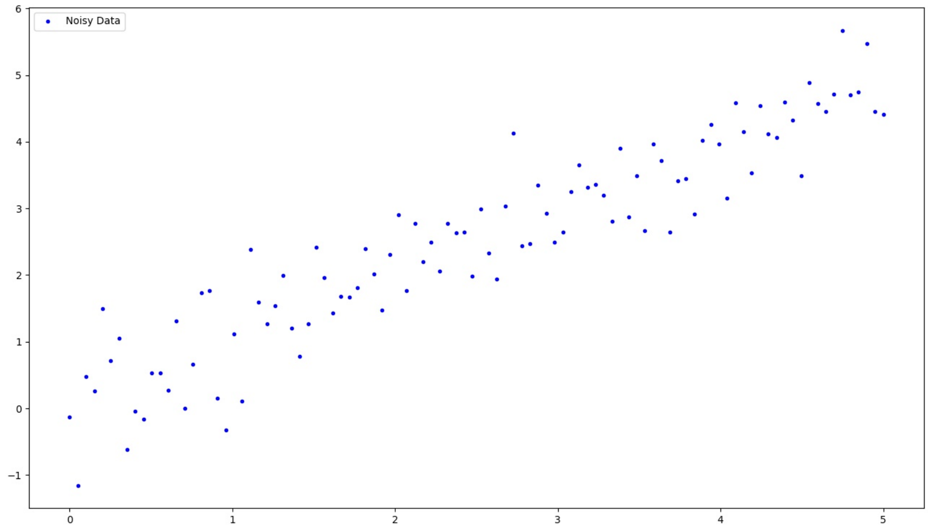


Figure 1: Noisy Data

Fitting a polynom through that data, will result in a highly oscilating courve, as you've seen in an earlier lecture already. Interpolating with some sort of spline will reduce the oscilation, but still result in a curve, passing through all those points, which is undesired.

Here, the "general direction" is interesting, as shown in Figure 2.

This "general direction" can also be formulated as the eigenvector with the largest eigenvalue of the covariance matrix of the given data.

**Shifting Data:**

Vectors are originated in $(0, 0)$. Thus, to calculate the direction within the given data, the data has to be shifted by its mean. That way the data is centered around the origin and a vector calculated from that data can represent an data-internal orientation.

**Covaraince Matrix:**

To set up the covariance matrix, you have to write each (mean shifted) coordinate $(x, y)$ as a line into a matrix $D$:

$$\boxed{D_i = (x_i, y_i)}$$

$D^\top D$ is then called the covariance matrix of $D^\top$.

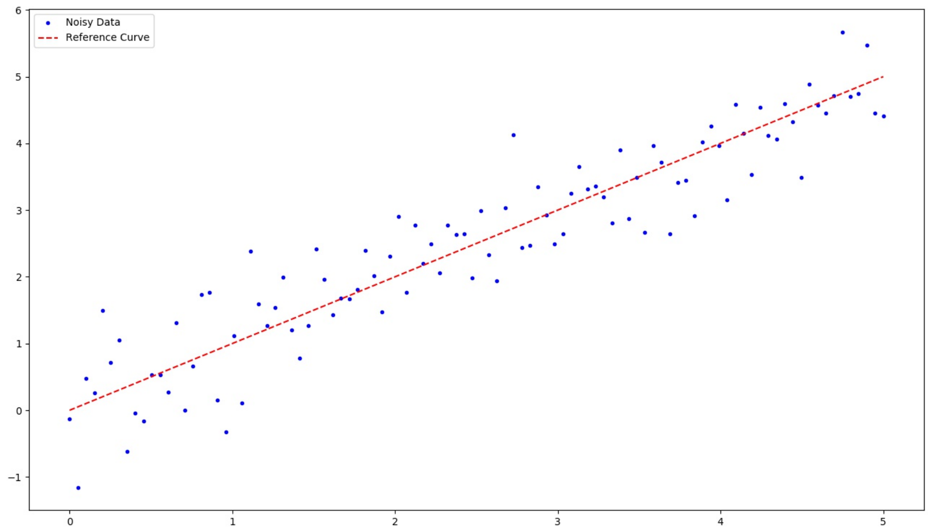


Figure 2: Noisy Data (blue) with a reference curve (red, dashed)

**Eigendecomposition:**
Calculating the eigenvector with the largest eigenvalue will now result in the “prinicpal direction” of the given data. The result is shown in Figure 3.

### 4.2 Eigendecomposition

**a) (40 Min, 2 Points)** Power Method

Please implement the function `powerMethod(A, b)` in `backend.py`. The method is supposed to calculate the largest Eigenvector (with respect to absolute Eigenvalues) of A using b as a starting point.

**b) (40 Min, 2 Points)** Inverse Power Method

Please implement the function `inversePowerMethod(A, b)` in `backend.py`. The method is supposed to calculate the smallest Eigenvector (with respect to absolute Eigenvalues) of A using b as a starting point.

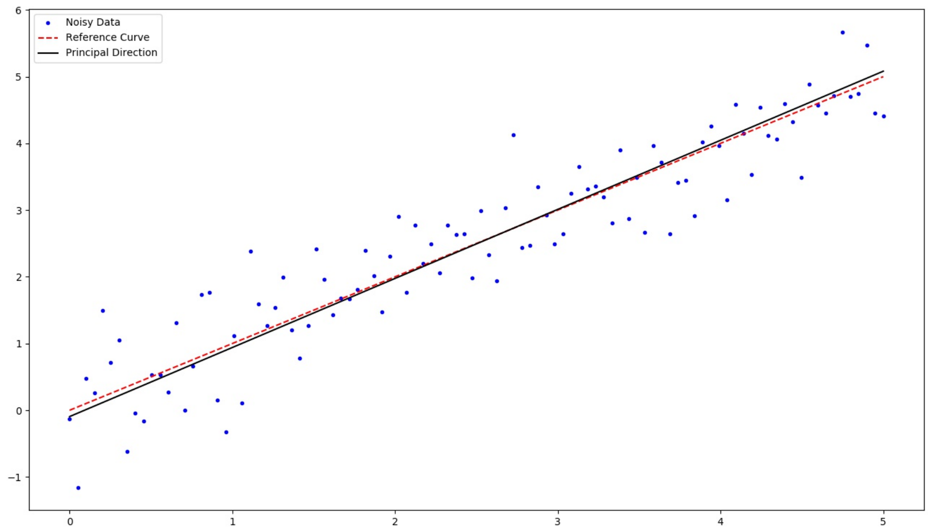


Figure 3: Noisy Data (blue) with a reference curve (red, dashed) and the estimated principal direction (black).

**c) (120 Min, 6 Points)** Linear Approximation via Eigenvectors

Please implement the function `linearPCA(x, y)` in `backend.py`, which performs the above described line fitting algorithm on the given data.

**Hints:**

- If you don't want to use your own implementation of the power method, you can use numpys eigendecomposition. This will always return all eigenvalues and eigenvectors, which may take longer than just calculating the largest one.
- $D$, as constructed from above is psoitive-(semi)-definite, which means, you can use the `np.linalg.eigh` method, instead of `np.linalg.eig`.

**Handing in:**

Please only include your `backend.py` in your hand in.