Problem set 3

Workout 0.1. A matrix has the following SVD:

$$U = \begin{pmatrix} 0 & -\frac{1}{\sqrt{5}} & 0 & 0 & -\frac{2}{\sqrt{5}} \\ 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & -\frac{2}{\sqrt{5}} & 0 & 0 & \frac{1}{\sqrt{5}} \end{pmatrix} \quad \Sigma = \begin{pmatrix} 3 & 0 & 0 & 0 \\ 0 & \sqrt{5} & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad V = \begin{pmatrix} 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

q. Without explicitly form A^TA, write down A^TA's eigenvalues and eigenvectors.

b. Without explicitly form AA^T, write down AA^T's eigenvalues and eigenvectors.

c. Without explicitly form A^TA and AA^T, what is $rank(A^TA)$ and $rank(AA^T)$, respectively?

d. Without explicitly form the matrix A, what is $rank(A)$?

e. Without explicitly form A, what is $cond_2(A)$ in this case?

f. What does the reduced form (or "economy" form) SVD look like? Write it down on paper.

g. Write the matrix A as a series of rank-1 matrices (form the matrices using paper and pen).

h. Form the the rank-2 matrix, A_2 closest to A in 2-norm (meaning the distance $\|A - A_2\|_2$ is as small as possible). What is the value of $\|A - A_2\|_2 =$?

i. Given U, Σ, V^T stored in 2D-arrays in Python, as `U`, `S` and `Vt`, respectively. Write down the Python-command that would create (compute) the rank-2 matrix in in item (h).

Workout 0.2. Given the same SVD as in previous task, and a right-hand-side

$$\boldsymbol{b} = \begin{pmatrix} 1 \\ 4 \\ 3 \\ 0 \\ 2 \end{pmatrix}$$

a. Calculate (paper and pen) and write down the pseudoinverse $A^\dagger$.

b. Find (and write down) all least squares solutions $A\boldsymbol{x} = \boldsymbol{b}$.

c. Find the norm minimal least squares solution $\boldsymbol{x}^\dagger = A^\dagger \boldsymbol{b}$.

Workout 0.3. Assume that the SVD of A is given by $A = U\Sigma V^T$.

a. Write down the SVD of A^T. Show a schematic for a 4×2 matrix A.

b. Write down the SVD of αA where α is a real number.

c. If A is square and non-singular, write down the SVD of A^{-1}.

d. Show that $(A^\dagger)^T = (A^T)^\dagger$

Workout 0.4. The Power method is an iterative method for finding eigenvectors and eigenvalues. Use the matrix A^TA (that you formed in workout 1 part (b)), and use the power method for finding its dominant eigenpair. Follow the algorithm and perform three iterations to find $\boldsymbol{v}^{(3)}$ and $\lambda^{(3)}$. Use $\boldsymbol{v}^{(0)} = \begin{pmatrix} 1 & 1 & 1 & 1 \end{pmatrix}^T$ as the initial vector.
You can use Python as a pocket calculator, but write down all the results $\boldsymbol{v}^{(1)}, \boldsymbol{v}^{(2)}, \boldsymbol{v}^{(3)}$ and $\lambda^{(1)}, \lambda^{(2)}, \lambda^{(3)}$ on paper.
In this particular case we already have the correct eigenvalues/eigenvectors. Compare the result with these.

Workout 0.5. The Power method does not always work, though. Try the method on the matrix

$$A = \begin{pmatrix} 5 & 0 & 0 \\ 0 & -5 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

with eigenvectors

$$\boldsymbol{v}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \quad \boldsymbol{v}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \quad \boldsymbol{v}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}$$

Calculate for example $\boldsymbol{v}^{(10)}$ with initial vector $\boldsymbol{v}^{(0)} = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}^T$. Try to explain the result theoretically and what the problem is here.

Non-mandatory

Workout 0.6.

The Eckard-Young theorem can be expressed "if a matrix B has $k = rank(B)$ then $\|A - B\| \geq \|A - A_k\|$", where $A_k = U\Sigma_k V^T$. The theorem is the basis for many applications in big data. Use compression of images as an example, and explain briefly how this application can be solved using the SVD, and how the Eckard-Young theorem comes in.

Workout 0.7. One possibility to compute the second dominant eigenpair is to use the following known theorem: If A has eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ and corresponding eigenvectors $\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_n$ then the **deflation matrix**

$$B = A - \lambda_1 \boldsymbol{v}_1 \boldsymbol{x}^T$$

for any vector $\boldsymbol{x}$ with property $\boldsymbol{v}_1^T \boldsymbol{x} = 1$ has eigenvalues $0, \lambda_2, \lambda_3, \ldots, \lambda_n$ and eigenvectors $\boldsymbol{v}_1, \boldsymbol{u}_2, \boldsymbol{u}_3, \ldots, \boldsymbol{u}_n$. Only $\boldsymbol{v}_1$ is the same but $\boldsymbol{v}_j$ can be obtained from $\boldsymbol{u}_j$:

$$\boldsymbol{u}_j = \boldsymbol{v}_j - \left(\frac{\lambda_1}{\lambda_j} \boldsymbol{x}^T \boldsymbol{v}_j \right) \boldsymbol{v}_1, \quad j = 2, \ldots, n.$$

Based in this theorem:

a. Find a simple candidate for $\boldsymbol{x}$.

b. Assume that eigenvalues are ordered as $|\lambda_1| > |\lambda_2| > |\lambda_3| \geqslant \cdots \geqslant |\lambda_n|$. How the power method can be applied to obtain λ_2?

c. Finally find a way to compute $\boldsymbol{v}_2$.

d. Can this idea be extended to compute other eigenvalues/eigenvectors?

Workout 0.8. The QR-method is a method that is used to find eigenvalues (and eigenvectors).

a. Is the QR-method an iterative method or not? Motivate your answer.

b. The basis for the method is similarity transformations and the Schur decomposition $A = Q\widetilde{T}Q^T$. Explain briefly how these two concepts are used in the QR-method to find the eigenvalues.

c. Show that the computational cost of the basic QR-iteration is $k\mathcal{O}(n^3)$, where n is the matrix size and k is number of iteration. Explain a technique to reduce the cost to $\mathcal{O}(n^3) + k\mathcal{O}(n^2)$.