Homework 1

Information about homework:

The problems below are mandatory and can give bonus points if they are handed in before the bonus deadline. If solutions are handed in before the date are not correct, they have to be redone, but the second time without yielding bonus points for the exam. Solutions must be clearly written, and easy to follow. If not, they will not generate bonus points, and must be redone. It is highly recommended that the answers are done on a computer (for instance LaTeX or word or a PDF-file from a Jupiter notebook). Please submit:

- written solutions (in the form of a PDF-file)
- your matlab or julia programs for the task

The files should be uploaded in
CANVAS → SF2526 → Assignments → Homework 1
You need to create a homework group in order to submit the assignment.

---

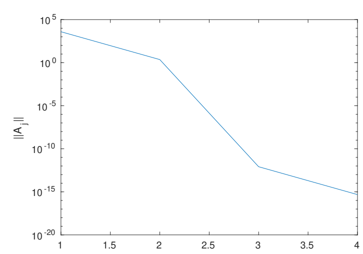
1. We shall investigate Gram-Schmidt by column elimination (Algorithm 1 in EJ1).

   (a) Consider the matrix

$$A = \begin{bmatrix} 1 & 2 & 2003 & 2005 \\ 2 & 2 & 2002 & 2004 \\ 3 & 2 & 2001 & 2003 \\ 4 & 7 & 7005 & 7012 \end{bmatrix}$$

   Show by hand that the matrix has rank three. (Hint: What is the sum of the two middle columns?)

   (b) Implement and apply Algorithm 1 (standard) and apply it to the matrix in (a). Verify that after $i = 3$, the "error" $\|A_j\|$ is small. Plot $\|A_j\|$ against iteration j. Compare with figure in margin.



(c) Carry out the same simulation as in (b) but with the larger matrix: `load_mat_hw1(1000,100)` where `load_mat_hw1.m` can be downloaded from course web page. Provide the error plot.

Warning: (c) will not be as smooth as convergence as (b)

(d) Carry out the same simulation as in (c) but with Algorithm 1 (greedy). Provide the error plot. Why is this algorithm better, if we want an approximation of the span of the columns?

2. Consider the following matrix

$$A = \begin{bmatrix} 5 & -1 \\ 5 & 7 \end{bmatrix}$$

Let $A = USV^T$ be the singular value decomposition of A.

(a) Given that

$$S = \begin{bmatrix} \sqrt{80} & 0 \\ 0 & \sqrt{20} \end{bmatrix}, V = \beta \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

determine β and U with a closed formula (no computer solution). Verify that U is an orthogonal matrix (you may use computer).

(b) Determine the matrix $X = uv^T \in \mathbb{R}^{2\times 2}$ where u and v are vectors, such that

$$X = \underset{\mathrm{rank}(X)=1}{\operatorname{argmin}} \|A - X\|$$

How big is the error $\|A - X\|$ according to theory? Compare with your computed solution.

3. Use the A matrix generated by `load_mat_hw1.m` as in Task 1c-d, which we now wish to approximate with an error 10^{-10} with as low rank as possible.

(a) What is the lowest rank of the matrix that approximates of the matrix that approximates A with an error (Frobenius or 2-norm) 10^{-10}. Use any matlab command to determine this.

(b) Use the approximation stemming from Algorithm 1 and the SVD of a small matrix as described in "SVD via Algorithm 1" in EJ1. What is the corresponding rank? You may use the matlab command `svd` for the small matrix.

The section called "SVD via Algorithm 1" is also described in PGM Section 1.5.4.

4. Summarize in your own words (2-5 sentences for each activity):

   - Video quiz 1: Low-rank approximation example
   - Video quiz 2a: QR-factorization from Gram-Schmidt
   - Video quiz 2b: Solving linear least squares problem via QR-factorization
   - Video quiz 3a: SVD introduction and properties (compare SVD and QR)
   - Video quiz 3b: SVD and the Eckart-Young theorem
   - Video quiz 4a: Index vectors
   - Video quiz 4b: Randomized SVD
   - Video quiz 5: Introduction to interpolatory decompositions

In task 5 you need to use image loading in MATLAB. A short introduction to the main tools you need can be found on Canvas.

5. Follow these steps to analyze the testbild video. Only written answers to c-f need to be handed it.

   (a) Download `testbild_snapshots.zip`. Familiarize yourself with plotting through julia/matlab by plotting the first snapshot, and compare with the `testbild.avi` video.

   (b) Write a program which puts all the snapshots into one matrix $A \in \mathbb{R}^{n\times m}$, where m=number of columns=number of snapshots. Note that by design all the video frames are almost (but not exactly) the same. What rank should one expect for A then?

   (c) Form a rank-one matrix as follows. Let $u = a_1 \in \mathbb{R}^n$ be the first column of A, and let $v^T = [1, \dots, 1] \in \mathbb{R}^m$. Compute $A - uv^T$. How is this related to how the images are different?

   (d) Compute an approximate SVD of the A matrix using your program in the previous task. Use $tol = ||A - uv^T||$ with u, v as in (c). What numerical rank do you get with this tolerance?

   (e) How does your implementation in (d) compare with MATLAB's `svd(A,0)`? Which is faster? This will heavily depend on your computer, and if the SVD-call fails on your computer report the error message and some interpretation.

   (f) Extract the best rank one approximation $\tilde{A}_1 = u_1\sigma_1 v_1^T$ from the SVD in (e) and compare $||A - \tilde{A}_1||$ with the corresponding results in (c) and (d). Discuss the differences.

6. We will now do an analysis similar to task 5 but for the video corresponding to `roundabout.mkv` and `roundabout_snapshots.zip`.

   (a) Carry out the greedy version of Algorithm 1 until $p = 20$, and plot $\|A_j\|$ for $j = 1, \ldots, p$.

   (b) What is the corresponding error for the result from (a)? Plot the error between the video snapshots and the rank k approximation.

   (c) Visualize the snapshots (image of the roundabout) for rank $k = 5$, $k = 10$ and $k = 20$.

   (d) Finally, recreate a video based on your low-rank approximations using the `makevideo` program (see Canvas). Try with rank $k = 1$ and some small $k > 1$.

   Extra/fun: Test the algorithm on the more difficult video `market.mkv`, which is also available on Canvas. You may need to restrict the analysis to a subset of snapshots for that case.

7. Use "basic randomized SVD" on the matrix in task 6, with target rank $k = 5$.

   (a) Carry out randomized SVD, and visualize the error as a function of oversampling parameter: Provide a semilogy image where the x-axis is the oversampling parameter up to $s = 25$, and the y-axis the approximation error introduced by randomized SVD.

   (b) Provide a CPU-time comparison for the method in the previous task, with $s = 15$. *Note: This highly depends on implementation and different student groups may get different results.*

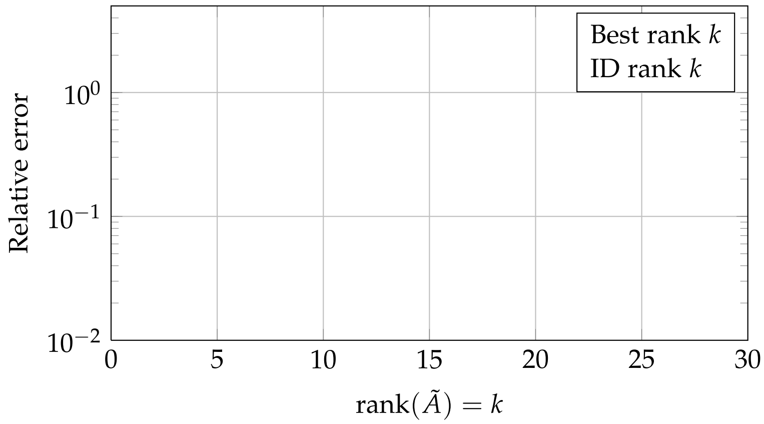
It is difficult to make correct CPU-timing in MATLAB. The reliability can be increased if you follow some techniques described in Canvas, under Systematic CPU time comparisons

8. Download and familiarize yourself with the function `zalando_plot.m` and the data file `zalando_items.mat`. Each `item0`,...,`item9` are matrices where every column corresponds to an image, which can be plotted with the `zalando_plot(item3(:,10))`. Compute ID by completing the skeleton code `ID_col.m` from the course files.

   a) Let A be the matrix where every column is a sandal. Plot the error for the best rank k approximation of A in a semilogy plot, as well as the error in the ID-approximation. On the y-axis use the relative error:

$$\frac{\|A - \tilde{A}\|}{\|A\|}$$

   where $\tilde{A}$ is the low-rank approximation. At what rank do we reach 25% accuracy of approximation for SVD? for ID?



b) For your favorite item, plot the images corresponding to the first 3 singular vectors, and first 3 interpolatory decomposition vectors. What are advantages/disadvantages of ID versus SVD?

Note: You are expected to plot the singular vectors (u_1, u_2, u_3) and ID-vectors c_1, c_2, c_3, not the SVD-approximation

Read about the Zalando MNIST dataset here `https://github.com/zalandoresearch/fashion-mnist`