Numerical Linear Algebra
A.A. 2024/2025

Lecturer: Prof. P. F. Antonietti TA: Dott. M. Botti

Challenge 2 - Singular Value Decomposition and image compression

Goal: Apply the singular value decomposition to image compression and noise reduction.
Data: Download the file `einstein.jpg` (256px) and move it to your working directory.

Overview

Any $m \times n$ matrix A possesses a **singular value decomposition** (SVD) of the form:

$$A = U\Sigma V^{\mathrm{T}};$$

where U is an orthogonal $m \times m$ matrix satisfying the condition $U^{\mathrm{T}}U = I_m$, Σ is an $m \times n$ rectangular diagonal matrix with nonnegative values $\sigma_1, \sigma_2, \ldots, \sigma_p$, with $p = \min(m, n)$, on the main diagonal, and V is an orthogonal $n \times n$ matrix satisfying the condition $V^{\mathrm{T}}V = I_n$. The singular values are arranged in the decreasing order: $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_p \geq 0$. The singular value decomposition of a matrix is **not unique**.

Singular value decomposition is important for the implementation of many numerical algorithms. In this project we will look at two applications related to image processing and noise filtering. We remark that other more efficient image compression methods exist. The examples in this project are used mainly to show the reduction in the amount of data that can be accomplished with SVD.

- The SVD can be used to express A in the form

  $$A = \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^{\mathrm{T}} + \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^{\mathrm{T}} + \cdots + \sigma_r \boldsymbol{u}_r \boldsymbol{v}_r^{\mathrm{T}},$$

  where r is the rank of A and $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ are the ith columns of U and V respectively. Observe that since the singular values are arranged in the decreasing order, the first terms in this sum provide a larger contribution then the subsequent terms.

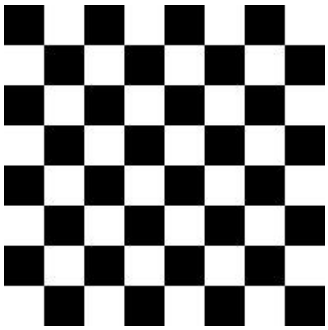
- The truncated SVD considering the first k terms, with $k < r$ is the best approximation of the matrix A among the matrices of the rank at most k in the sense of Frobenius norm.This can be used for image compression as follows: instead of storing the whole $m \times n$ matrix A, we can instead store the $m \times k$ and $n \times k$ matrices

  $$C = [\boldsymbol{u}_1\ \boldsymbol{u}_2\ \ldots\ \boldsymbol{u}_k], \qquad D = [\sigma_1 \boldsymbol{v}_1\ \sigma_2 \boldsymbol{v}_2\ \ldots\ \sigma_r \boldsymbol{v}_k]. \tag{1}$$

  If k is much smaller than p, then storing C and D will take much less space than storing the full matrix A. The compressed image can be simply computed as $\tilde{A} = CD^{\mathrm{T}}$.

Tasks

- Load the image as an `Eigen` matrix A with size $m \times n$. Each entry in the matrix corresponds to a pixel on the screen and takes a value somewhere between 0 (black) and 255 (white). Compute the matrix product $A^{\mathrm{T}}A$ and report the euclidean norm of $A^{\mathrm{T}}A$.
- Solve the eigenvalue problem $A^{\mathrm{T}}A\boldsymbol{x} = \lambda\boldsymbol{x}$ using the proper solver provided by the `Eigen` library. Report the two largest computed singular values of A.
- Export matrix $A^{\mathrm{T}}A$ in the matrix market format and move it to the `lis-2.1.6/test` folder. Using the proper iterative solver available in the LIS library compute the largest eigenvalue of $A^{\mathrm{T}}A$ up to a tolerance of 10^{-8}. Report the computed eigenvalue. Is the result in agreement with the one obtained in the previous point?
- Find a `shift` $\mu \in \mathbb{R}$ yielding an acceleration of the previous eigensolver. Report μ and the number of iterations required to achieve a tolerance of 10^{-8}.
- Using the SVD module of the `Eigen` library, perform a singular value decomposition of the matrix A. Report the Euclidean norm of the diagonal matrix Σ of the singular values.
- Compute the matrices C and D described in (1) assuming $k = 40$ and $k = 80$. Report the number of nonzero entries in the matrices C and D.
- Compute the compressed images as the matrix product CD^{T} (again for $k = 40$ and $k = 80$). Export and upload the resulting images in `.png`.
- Using `Eigen` create a black and white checkerboard image (as the one depicted below) with height and width equal to 200 pixels. Report the Euclidean norm of the matrix corresponding to the image.



- Introduce a noise into the checkerboard image by adding random fluctuations of color ranging between $[-50, 50]$ to each pixel. Export the resulting image in `.png` and upload it.
- Using the SVD module of the `Eigen` library, perform a singular value decomposition of the matrix corresponding to the noisy image. Report the two largest computed singular values.
- Starting from the previously computed SVD, create the matrices C and D defined in (1) assuming $k = 5$ and $k = 10$. Report the size of the matrices C and D.
- Compute the compressed images as the matrix product CD^{T} (again for $k = 5$ and $k = 10$). Export and upload the resulting images in `.png`.
- Compare the compressed images with the original and noisy images. Comment the results.