Math 504, Fall 2018 - Homework 1

October 4, 2018

1. Consider the matrix A and its full singular value decomposition given below.

$$A = \begin{bmatrix} 11 & 8 & 5 & 8 \\ -10 & -12 & -14 & -12 \end{bmatrix}$$

$$= \begin{bmatrix} 2/\sqrt{13} & 3/\sqrt{13} \\ -3/\sqrt{13} & 2/\sqrt{13} \end{bmatrix} \begin{bmatrix} 8\sqrt{13} & 0 & 0 & 0 \\ 0 & 2\sqrt{13} & 0 & 0 \end{bmatrix} \begin{bmatrix} 1/2 & 1/2 & 1/2 & 1/2 \\ 1/2 & 0 & -1/2 & 0 \\ 0 & 1/2 & 0 & -1/2 \\ 1/2 & -1/2 & 1/2 & -1/2 \end{bmatrix}$$

(a) Write down the singular values, and corresponding left and right singular vectors for A.

(b) Express A as a sum of two outer products.

(c) Describe the set $\{Av \mid v \in \mathbb{R}^2 \text{ s.t. } \|v\|_2 = 1\}$ geometrically.

2. Calculate a reduced singular value decomposition for the matrix

$$A = \frac{1}{2} \begin{bmatrix} 7 & -1 \\ 1 & -7 \\ 7 & -1 \\ 1 & -7 \end{bmatrix},$$

and find an orthonormal basis for its column space.

3.

(a) Let $A \in \mathbb{R}^{n \times n}$. Prove that $A_S := (A + A^T)/2$ satisfies the following:

$$\|A_S - A\|_F \quad \leq \quad \|B - A\| \quad \forall B \in \mathbb{S}^{n \times n},$$

where $\mathbb{S}^{n \times n}$ denotes the set of $n \times n$ real symmetric matrices, and $\| \cdot \|_F$ is the standard Euclidean norm on $\mathbb{R}^{n \times n}$ (commonly referred also as the Frobenius norm) defined by

$$\|C\|_F \quad := \quad \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} |c_{ij}|^2}.$$

(b) Specifically, for

$$A \quad = \quad \begin{bmatrix} -4 & 6 \\ -2 & 4 \end{bmatrix}$$

find ***(i)*** a symmetric matrix B_* so that $\|B_* - A\|_F$ is as small as possible, ***(ii)*** a singular matrix C_* so that $\|C_* - A\|_2$ is as small as possible.

4. Letting $B = \begin{bmatrix} 1 & 7 \\ 3 & 5 \end{bmatrix}$, find Schur factorizations for B and $B + B^T$.

5.

(a) Let $B \in \mathbb{C}^{m \times n}$ with $m \geq n$. Show that

$$\sigma_n(B) = \min_{v \in \mathbb{C}^n, \ \|v\|_2 = 1} \|Bv\|_2,$$

where $\sigma_n(B)$ denotes the smallest singular value of B.

(b) Let $A, B \in \mathbb{C}^{n \times n}$. Show that

$$\|AB\|_2 \quad \leq \|A\|_2 \|B\|_2.$$

6. A symmetric matrix $A \in \mathbb{R}^{n \times n}$ is called positive definite if $v^T A v > 0$ for all nonzero $v \in \mathbb{R}^n$.

(a) Let $A \in \mathbb{R}^{n \times n}$ be symmetric. Prove that A is positive definite if and only if all eigenvalues of A are positive.

(b) Let $A \in \mathbb{R}^{n \times n}$ be a symmetric and positive definite matrix. Show that

$$\langle x, y \rangle_A := x^T A y, \quad \forall x, y \in \mathbb{R}^n$$

is an inner product on $\mathbb{R}^n$.

7. Given a set of linearly independent vectors $\{a_1, \dots, a_p\}$ in $\mathbb{C}^n$, recall that the Gram-Schmidt procedure generates an orthonormal basis $\{q_1, \dots, q_p\}$ for $\mathcal{S} = \mathrm{span}\{a_1, \dots, a_p\}$ defined by

$$\begin{aligned} &q_1 := a_1 / \|a_1\|_2, \\ &q_j := \widetilde{q}_j / \|\widetilde{q}_j\|_2, \quad \text{where } \widetilde{q}_j := a_j - \sum_{k=1}^{j-1} (q_k^* a_j) q_k \quad j = 2, \dots, p. \end{aligned} \tag{0.1}$$

Now observe that, letting $r_{kj} := q_k^* a_j$ for $j > k$ and $r_{jj} := \|\widetilde{q}_j\|_2$, defining equations (0.1) for the vectors $q_1, \dots, q_p$ can be combined into

$$\underbrace{\begin{bmatrix} a_1 & a_2 & \dots & a_p \end{bmatrix}}_{A \in \mathbb{C}^{n \times p}} = \underbrace{\begin{bmatrix} q_1 & q_2 & \dots & q_p \end{bmatrix}}_{Q \in \mathbb{C}^{n \times p}} \underbrace{\begin{bmatrix} r_{11} & r_{12} & \dots & & r_{1p} \\ 0 & r_{22} & \dots & & r_{2p} \\ & & \ddots & & \vdots \\ & & & r_{(p-1)(p-1)} & \\ 0 & & & 0 & r_{pp} \end{bmatrix}}_{R \in \mathbb{C}^{p \times p}} \tag{0.2}$$

where Q has orthonormal columns, and R is upper triangular. This is known as the (reduced) QR factorization of A.

Write down a Matlab routine based on the Gram-Schmidt procedure (0.1) and the observation (0.2) that computes the (reduced) QR factorization of a given matrix $A \in \mathbb{C}^{n \times p}$ with $n \geq p$ and with linearly independent columns. You can use the Matlab implementation of the Gram-Schmidt procedure that is made available on the course webpage.

8. In class we have seen a Matlab routine that computes the reduced SVD of a given matrix $A \in \mathbb{C}^{m\times n}$ with $m \geq n$ based on the eigenvalues and eigenvectors of A^*A. This Matlab routine is made available on the course webpage.

Now implement a Matlab routine that computes the reduced SVD of a given matrix $A \in \mathbb{C}^{m\times n}$ with $m < n$. Apply your implementation to compress the image below, or any image that you may like with more columns than rows, by replacing the image with its best rank r approximation, where r is a prescribed positive integer. Recall that in Matlab you

can open this image by typing

```
H = imread('sunrise-zermatt-gray.jpg');
H = double(H);
```

These commands will store the image in the matrix H in double precision. Furthermore, you can record the matrix H back in the image `sunrise-zermatt-gray.jpg` by typing

```
imwrite(uint8(H),'sunrise_zermatt-gray.jpg');
```

9. Write down a Matlab routine that computes a full SVD of a given matrix $A \in \mathbb{C}^{m \times n}$. Once again you can use the routine on the course webpage that computes a reduced SVD.

The remaining two questions will not be collected or graded.

10. Let $A \in \mathbb{C}^{m \times n}$ with the full SVD

$$A = \underbrace{\begin{bmatrix} u_1 & u_2 & \dots & u_m \end{bmatrix}}_{U \in \mathbb{C}^{m \times m}} \underbrace{\Sigma}_{\in \mathbb{R}^{m \times n}} \underbrace{\begin{bmatrix} v_1^* \\ v_2^* \\ \vdots \\ v_n^* \end{bmatrix}}_{V^* \in \mathbb{C}^{n \times n}}.$$

Suppose that the diagonal entries $\sigma_1, \dots, \sigma_p$ of Σ are as follows: $\sigma_1, \dots, \sigma_r > 0$, whereas $\sigma_{r+1}, \dots, \sigma_p = 0$, where $p := \min\{m, n\}$. Show that $\{v_{r+1}, \dots, v_n\}$ is an orthonormal basis for Null(A).

11. Write down a routine that computes a Schur factorization of a given matrix $A \in \mathbb{C}^{n \times n}$ based on the proof discussed in the class. Since the proof is by induction, you can implement this recursively. At every iteration, you need to compute an eigenvalue and a corresponding unit eigenvetor. For this purpose, you could use the following standard command in Matlab.

```
>>  [v,d] = eigs(A,1,'LM');
```

This command returns an eigenvalue of A in `d` and a corresponding eigenvector in `v`.