Homework II

Due on Feb 21, 2024

1. **(5 marks)** Let A be an $m \times n$ matrix. Suppose you are given an SVD of A. Write down an SVD of the following matrices:

$$\begin{bmatrix} 0 & A \\ A^T & 0 \end{bmatrix}, \quad \begin{bmatrix} I & A \\ A^T & 0 \end{bmatrix}.$$

Here 0 and I denote the all-zero and the identity matrices of suitable dimensions respectively.

2. Recall that the Frobenius norm of an $m \times n$ matrix A, written $||A||_F$ is defined as $\left(\sum_{i,j} |A_{ij}|^2\right)^{1/2}$.

   (a) **(2 marks)** Show that for any unitary matrices U and V of suitable dimensions, $||AV||_F = ||UA||_F = ||A||_F$. You may want to use the facts that for any matrix X, $||X||_F^2 = \mathsf{Trace}(XX^T)$, where Trace of a matrix is the sum of its diagonal entries.

   (b) **(1 marks)** Let A have the SVD $U\Sigma V^T$ and A_k denote the matrix $U\Sigma_k V^T$, where Σ_k is the matrix obtained from Σ be zeroing the entries $\sigma_{k+1}, \ldots, \sigma_{\min(m,n)}$. Show that $||A - A_k||_F^2 = \sigma_{k+1}^2 + \ldots + \sigma_r^2$, where r denotes the rank of A.

   (c) **(5 marks)** Show that for any rank k matrix B, $||A - B||_F^2 \geq \sigma_{k+1}^2 + \ldots + \sigma_n^2$.

   (d) **(5 marks)** Show that if $\sigma_{k+1} < \sigma_k$ and B is a rank k matrix such that

$$||A - B||_F^2 = \sigma_{k+1}^2 + \ldots + \sigma_n^2,$$

   then B must be equal to the matrix A_k defined above.

   (e) **(2 marks)** Is it true that if $\sigma_{k+1} < \sigma_k$ and B is a rank k matrix such that $||A - B||_2 = \sigma_{k+1}$, then B must be equal to A_k ? Give reasons.

3. **(5 marks)** Let B be an $m \times n$ matrix and consider the problem of finding an $n \times m$ matrix such that

$$||BX - I||_F$$

is minimized where I is the $m \times m$ identity matrix. Show that X as the pseudo-inverse of B minimizes this objective function. What is the value of the minimum value of this objective function?

4. Let A, B, C be $m \times n, p \times r$ and $m \times r$ matrices. Consider the problem of finding an $n \times p$ matrix X such that $||AXB - C||_F$ norm is minimized.

   - **(5 marks)** Show that $X_0 := A^\dagger C b^\dagger$ minimizes this objective function, where $A^\dagger$ and $B^\dagger$ are the pseudo-inverses of A and B respectively.

- **(5 marks)** Show that if X is any other matrix minimizing $||AXB - C||_F$, then $||X_0||_F \leq ||X||_F$.

5. **(5 marks)** Write a MATLAB program `[W, R] = house(A)` that computes an implicit representation of a full QR factorization $A = QR$ of an $m \times n$ matrix A with $m \geq n$ using Householder reflections. The output variables are a lower traingular matrix W (which is $m \times n$) whose column vectors define the successive Householder reflections, and a triangular $m \times n$ matrix R. Write a MATLAB program `Q = formQ(W)` that takes the matrix W above and generates the corresponding $m \times m$ orthogonal matrix Q.

6. **(5 marks)** Write a MATLAB program `x = leastSquare(A,b)` that solves the least squares problem $Ax = b$. You should use the the program for QR factorization described above. You will use this implementation of least squares in the problem below.

7. **(10 marks)** To demonstrate how results from the normal equations method and QR factorization can differ numerically, we need a least squares problem that is ill-conditioned and also has a small residual. We can generate such a problem as follows. We will fit a polynomial of degree $n-1$,

$$p_{n-1}(t) = x_1 + x_2 t + x_3 t^2 + \cdots + x_n t^{n-1},$$

to m data points (t_i, y_i), $m > n$. We choose $t_i = \frac{i-1}{m-1}$, $i = 1, \ldots, m$, so that the data points are evenly spaced on the interval $[0, 1]$. We will generate the corresponding values y_i by first choosing values for the x_j, say $x_j = 1$, $j = 1, \ldots, n$, and evaluating the polynomial to obtain $y_i = p_{n-1}(t), i = 1, \ldots, m$. We could now see whether we can recover the x_j that we used to generate the y_i, but to make it more interesting, we first randomly perturb the y_i values to simulate the data error. Specifically, we take $y_i = y_i + (2u_i - 1) * \epsilon$, $i = 1, \ldots, m$, where each u_i is a random number uniformly generated from $[0, 1]$, and ϵ is a small positive number. For double precision, use $m = 21, n = 12, \epsilon = 10^{-10}$.

   Having generated the data set, we will now compare the two methods for least squares. First, form the system of normal equations, and solve it using Cholesky factorization (you can use MATLAB built-in function). Next solve the least squares using QR factorization implemented in the question above. Compare the two resulting solutions. For which method is the solution more sensitive to the perturbation we introduced in the data ? Which method comes closer to recovering the x that we used to generate the data ?

8. **(15 marks)** You are given coordinates of 6 points as below, but these coordinates have some errors in them. For each pair of distinct points, you are given (approximate) distance between them. Now, you have to replace these 6 points by more precise coordinates such that the discrepancy between the measured distances and the actual reported distances between the points in minimized. Here is how we shall rephrase this as a least-squares problem. For each point (x_i, y_i), we want to perturb them by

$(x_i + \delta_i, y_i + \epsilon_i)$, where δ_i and ϵ_i are the unknowns, but are assumed to be small. Now, given two points, the square of the Euclidean distance between them can be written as

$$(x_i + \delta_i - x_j - \delta_j)^2 + (y_i + \epsilon_i - y_j - \epsilon_j)^2.$$

Now expand the above terms and remove any term involving product or square of the δ or ϵ. Thus, we get a linear expression in the unknowns. Use this idea to frame the problem as a least squares problem. Explain clearly the formulation obtained. Use this to find the corrections to the following points:

$$(1, 2), (2, 4), (3, 5), (4, 6), (1.5, 5.5), (3.4, 2.7).$$

The matrix showing the measured pair-wise distances is as follows:

$$\begin{pmatrix} 0 & 4.674 & 11.349 & 25.632 & 13.075 & 5.806 \\ & 0 & 1.981 & 9.360 & 3.009 & 4.036 \\ & & 0 & 2.955 & 4.064 & 4.105 \\ & & & 0 & 8.596 & 11.614 \\ & & & & 0 & 12.845 \\ & & & & & 0 \end{pmatrix}$$