

AM205: Assignment 2

Question 1 [10 points]

(a) [4 points] For $p \geq 1$, the p -norm for a vector $x \in \mathbb{R}^n$ is defined as:

$$\|x\|_p \equiv \left(\sum_{i=1}^n |x_i|^p \right)^{1/p}. \quad (*)$$

This definition is in fact meaningful for $p < 1$ as well, although it does not define a norm in this case because it does not satisfy the triangle inequality. The MATLAB function `norm(x,p)` evaluates the function (*) for any real value p , even when it is not technically a norm.

On a single figure, plot the p -norm “unit circle” in \mathbb{R}^2 (i.e. $\{v \in \mathbb{R}^2 : \|v\|_p = 1\}$), for $p = 1, 1.5, 2, 5, 10$. Also include the infinity norm unit circle and the “ $p = 0.5$ norm” unit circle. What do you see happening to the p -norm unit circles as p gets larger?

Provide a specific counter-example that shows that the $p = 0.5$ “norm” does not satisfy the triangle inequality.

(b) [2 points] For $p = 2$, plot the image of the p -norm “unit circle” under the action of the matrix

$$A = \begin{bmatrix} 3 & -1 \\ 1 & 0 \end{bmatrix}.$$

(Note that the image of a set \mathcal{C} under the action of A is $\{Ax : x \in \mathcal{C}\}$.) Indicate a point on your plot such that the p -norm of the point is equal to the p -norm of A (the point need not be unique). Repeat the problem with $p = \infty$.

(c) [4 points] (No code required for this part) Let $A \in \mathbb{R}^{m \times n}$. Prove that for $v \in \mathbb{R}^n$ we have $\|v\|_\infty \leq \|v\|_2 \leq \sqrt{n}\|v\|_\infty$, and hence show that $\|A\|_2 \leq \sqrt{m}\|A\|_\infty$.

Question 2 [20 points]

(a) [4 points] (No code required for this part) Solve $Ax = b$ for

$$A = \begin{bmatrix} 2 & 1 & 1 & 0 \\ 4 & 3 & 3 & 1 \\ 8 & 7 & 9 & 5 \\ 6 & 7 & 9 & 8 \end{bmatrix}, \quad b = \begin{bmatrix} 0 \\ 1 \\ 3 \\ 2 \end{bmatrix},$$

by first forming the LU factorization *without pivoting* and then applying forward and back substitution. You should show all the steps in this process in your write-up.

(b) [10 points] Write three functions:

- `forwardsub` which takes a lower-triangular matrix L and a right-hand-side vector b as arguments and returns the solution for the system $Lx = b$,

- `backsub` which takes an upper-triangular matrix U and a right-hand-side vector b and returns the solution for the system $Ux = b$,
- `myLU` which takes a matrix A as an argument, performs LU factorization with partial pivoting and return matrices L , U and P . (Do not use MATLAB's built-in LU functions in `myLU`.)

Show that your `myLU` function accurately factorizes a set of ten randomly generated matrices $A_n \in \mathbb{R}^{n \times n}$ for $n = 10 : 10 : 100$, by plotting the relative 2-norm error $\|PA_n - LU\|_2 / \|A_n\|_2$ as a function of n . Also, verify your answer from (a) using `myLU`, `forwardsub` and `backsub`.

(c) [6 points] Use the MATLAB function `hilb` to generate the 12×12 Hilbert matrix (Hilbert matrices are famously ill-conditioned). Let $x = [1, 1, \dots, 1]^T \in \mathbb{R}^{12}$ and let $b = Ax$. Use `myLU`, `forwardsub` and `backsub` from (b) to solve $Ax = b$, and let \hat{x} denote solution you obtain. Compute the 2-norm relative residual $\left(\frac{\|r(\hat{x})\|_2}{\|A\|_2 \|\hat{x}\|_2}\right)$ and relative error $\left(\frac{\|x - \hat{x}\|_2}{\|\hat{x}\|_2}\right)$ and discuss the relationship between these two values.

Question 3 [10 points]

Matrices of the form

$$G_n \equiv \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 1 \\ -1 & 1 & 0 & 0 & \dots & 1 \\ -1 & -1 & 1 & 0 & \dots & 1 \\ -1 & -1 & -1 & 1 & \dots & 1 \\ \vdots & & & & \ddots & \vdots \\ -1 & -1 & -1 & -1 & \dots & 1 \end{bmatrix} \in \mathbb{R}^{n \times n}$$

are examples of the very rare situation in which Gaussian elimination is numerically unstable.

Write a function `generateG` that returns G_n . Then, for $n = 10 : 10 : 200$, set $x = [1, 1, \dots, 1]^T \in \mathbb{R}^n$ and construct a right-hand side vector $b = G_n x$. Solve $G_n x = b$ using “backslash” (which uses Gaussian elimination with partial pivoting), and let \hat{x} denote the computed solution. Plot the 2-norm relative error as a function of n and also include a plot that shows that the inequality:

$$\frac{\|x - \hat{x}\|_2}{\|\hat{x}\|_2} \leq \text{cond}(A) \frac{\|r(\hat{x})\|_2}{\|A\|_2 \|\hat{x}\|_2}$$

is satisfied. Based on your results, explain why Gaussian elimination with partial pivoting applied to this case is considered to be numerically unstable.

Question 4 [24 points]

(a) [2 points] (Written question, no code required) Determine a Householder transformation matrix, Q , that annihilates all but the first entry of the vector $a = [2, 1, 0, 2]^T$ (of the two possibilities, choose the Householder matrix Q recommended for numerical stability). What is the first entry of Qa ?

(b) [7 points] (Written question, no code required) An alternative method for constructing a QR factorization that we did not discuss in lectures is the method of Givens rotations. Givens rotations are generally less efficient than Householder reflections for factorization of dense matrices, but can have advantages for sparse matrices.

A Givens rotation matrix is an orthogonal matrix which annihilates a single component of a vector, and

it is represented by a matrix of the form (where $i \neq j$):

$$G(i, j, \theta) = \begin{cases} g_{ii} = c \equiv \cos(\theta) \\ g_{jj} = c \\ g_{ij} = s \equiv \sin(\theta) \\ g_{ji} = -s \\ g_{kk} = 1 & k \neq i, j \\ g_{mn} = 0 & \text{otherwise} \end{cases}$$

$$= \begin{bmatrix} 1 & \cdots & 0 & \cdots & 0 & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & c & \cdots & s & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & -s & \cdots & c & \cdots & 0 \\ \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & \cdots & 0 & \cdots & 1 \end{bmatrix}.$$

When acting from the left $G(i, j, \theta)$ will only modify the i and j rows of a matrix A , and we choose c and s in order to annihilate an entry of row j of A . In particular, let a_1 and a_2 denote entries in a column of A , then we can introduce a zero the column as follows:

$$\begin{bmatrix} c & s \\ -s & c \end{bmatrix} \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix}.$$

Since the matrix in the equation above is orthogonal, we must have $\alpha = \sqrt{a_1^2 + a_2^2}$, and then it also follows that $c = a_1/\sqrt{a_1^2 + a_2^2}$, and $s = a_2/\sqrt{a_1^2 + a_2^2}$.

The formulas above work, but they are more susceptible to underflow or overflow than necessary, due to the $a_1^2 + a_2^2$ factor. Hence it is better to formulate the matrix differently to avoid calculating this factor explicitly. To achieve this, c and s can be calculated in terms of the tangent or cotangent of the rotation angle (denoted t and τ below, respectively) as follows:

- If $|a_1| > |a_2|$, set $t \equiv \tan(\theta) = a_2/a_1$, and hence $c = 1/\sqrt{1+t^2}$, $s = ct$,
- If $|a_2| > |a_1|$, set $\tau \equiv \cot(\theta) = a_1/a_2$, and hence $s = 1/\sqrt{1+\tau^2}$, $c = s\tau$.

With these definitions, $|t| < 1$ (resp. $|\tau| < 1$) and hence there is no danger of overflow or underflow in the term $1+t^2$ (resp. $1+\tau^2$).

Use Givens rotation matrices to compute the QR factorization by hand for the following matrix:

$$A = \begin{bmatrix} 5 & -3 & 0 \\ -3 & 3 & 9 \\ 4 & 1 & 3 \end{bmatrix}$$

Show each G_i explicitly as well as the intermediate products, G_1A and G_2G_1A . Also, write out the Q and R factors that you obtain from this process.

(c) [15 points] Now you will implement and compare several QR factorization algorithms:

- Write a MATLAB function `[Q_hat,R_hat] = mgs(A)` that returns the reduced QR factorization of a matrix $A \in \mathbb{R}^{m \times n}$, $m \geq n$, using the modified Gram-Schmidt method.

- Write a MATLAB function `[W,R] = householder(A)` that computes an implicit representation of a full QR factorization of $A \in \mathbb{R}^{m \times n}$, $m \geq n$ using Householder reflections. The output variables are a lower triangular matrix $W \in \mathbb{R}^{m \times n}$ whose columns are the vectors v_k defining successive Householder reflections, and a triangular matrix $R \in \mathbb{R}^{n \times n}$. Write a second function `Q = formQ(W)` that takes the matrix W produced by `householder` and returns the corresponding orthogonal matrix Q .
- Write a Matlab function `[Q,R] = givens(A)` that computes the full QR factorization of a matrix $A \in \mathbb{R}^{m \times n}$, $m \geq n$, using Givens rotations. This function should use the \tan/\cotan formulation of a Givens matrix.

Use the command `vander(linspace(-1,1,n))` to generate a set of $n \times n$ Vandermonde matrices, for $n = 10 : 10 : 100$ and plot the error $\|A - QR\|_2 / \|A\|_2$ for the QR factorizations obtained from `gsvs`, `householder/formQ` and `givens`. Also, plot $\|Q^T Q - I\|_2$, and hence discuss how well each method preserves orthogonality of Q in this case.

Question 5 [16 points]

Suppose we are given a dataset represented by a matrix $X \in \mathbb{R}^{m \times n}$. One popular way to analyze such a dataset is to perform Principal Component Analysis (PCA), which consists of computing the eigenvalue decomposition of the matrix XX^T (in statistics this matrix is related to correlations between the data in the rows of X). The eigenvectors of XX^T are referred to as principal components, and the eigenvalues relate to the variance in the dataset associated with the principal components.

(a) [1 point] (Written question, no code required) We discussed the relationship between the eigenvalue decomposition of XX^T and the SVD of X in lectures. Based on this relationship, show that the principal components are the same as the left singular vectors of X .

(b) [3 points] It can be less accurate to perform PCA by computing eigenvalues of XX^T as opposed to using the SVD. For example, consider the matrix:

$$X = \begin{bmatrix} 1 & 10^{-8} & & & \\ 1 & & 10^{-8} & & \\ 1 & & & 10^{-8} & \\ 1 & & & & 10^{-8} \end{bmatrix}.$$

What are the exact eigenvalues of XX^T (you may find these by hand or use a symbolic computing environment like Mathematic or Maple)? Also, using MATLAB (or another numerical environment of your choosing), compute:

- the eigenvalues of XX^T ,
- the squares of the singular values of X .

Which method, (i) or (ii), is better in this case?

(c) [12 points] In this question we will use PCA (or, equivalently, the SVD) to do some analysis of data provided in the file `q5_c.mat`, which provides a matrix $X \in \mathbb{R}^{10 \times 20}$.

- Recall from linear algebra that the orthogonal projection of a vector $x \in \mathbb{R}^m$ onto a subspace W of \mathbb{R}^m is defined as follows:

$$v^T(Px - x) = 0, \quad \forall v \in W, \quad (1)$$

where $Px \in W$ denotes the projection of x onto W . (This formula states that the projection error $Px - x$ is orthogonal to W .)

The principal components of X provide an orthogonal basis that we can use to form a low-dimensional approximation to X via orthogonal projection. Compute the orthogonal projection of each column of X onto the first two principal components, and form a 2D scatter plot of the projected data. In your plot, the coefficients of the first and second principal components should correspond to the horizontal and vertical axes, respectively.

(ii) The total projection error from (i) is given by:

$$E_P = \left(\sum_{j=1}^{20} \|Px_j - x_j\|_2^2 \right)^{\frac{1}{2}}, \quad (2)$$

where x_j denotes the j^{th} column of X .

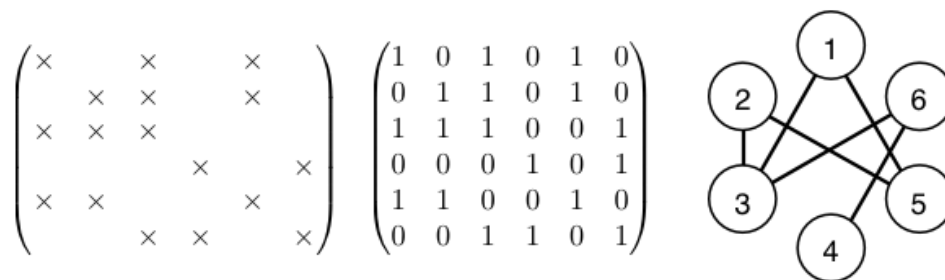
Give an analytical formula for E_P that involves the singular values of X . (Hint: For P denoting the orthogonal projection operator onto the span of the first two principal components, it can be shown that the matrix $[Px_1 \mid Px_2 \mid \dots \mid Px_n] \in \mathbb{R}^{10 \times 20}$ minimizes the total projection error, as defined in (2), among all rank 2 matrices.)

Also, calculate E_P directly based on your results for Px_j , $j = 1, \dots, 20$ from (i). Verify that your calculated value agrees with the result of evaluating your formula (*i.e.* the formula from above that involves singular values) for E_P .

Question 6 [20 points]

As discussed in lectures, it is highly desirable to minimize “fill-in” when performing sparse matrix factorizations. The amount of fill-in that occurs when applying a factorization is very sensitive to the order in which the rows or columns of the matrix are processed. By re-ordering the matrix beforehand it is often possible to greatly reduce the amount of fill-in. The reverse Cuthill-McKee algorithm (RCM)¹ is a popular method for finding a re-ordering of a sparse symmetric matrix that will reduce the matrix’s bandwidth.² This bandwidth reduction leads to reduced fill-in when computing a matrix factorization.

Let $A \in \mathbb{R}^{n \times n}$ denote a sparse symmetric matrix. The RCM algorithm constructs a re-ordering of A by considering a graph with n vertices which represents the sparsity pattern of A .³ For example, in the figure below we have a graph (right-most figure) with an “adjacency matrix” (middle figure) that matches the sparsity pattern of a matrix (left-most figure):



¹E. Cuthill and J. McKee. “Reducing the bandwidth of sparse symmetric matrices.” In Proceedings of the 1969 24th national conference, pages 157-172, New York, NY, USA, 1969. ACM Press.

²Sometimes this is also called the bandwidth minimization problem. Exact minimization of bandwidth turns out to be an NP-complete problem, but heuristic approaches can find pretty good solutions in a reasonable amount of time.

³The re-ordering provided by the RCM algorithm is not necessarily unique. Also note that if the graph corresponding to A has any disconnected sub-graphs then the RCM algorithm must be applied to each of them individually.

(b) [8 points] Load the matrix `q6_b.mat` (provided with the assignment), which contains a symmetric positive-definite sparse matrix. Compute the bandwidth of A , and plot the sparsity pattern of its Cholesky factor L , where $A = LL^T$.

Next, use MATLAB's implementation of RCM (the built-in function `symrcm`) to determine a re-ordering of A . Let \hat{A} denote the re-ordered matrix, and note that if \mathbf{r} is a reordering vector, we can use the syntax `A_hat = A(r,r)` to obtain \hat{A} in MATLAB. Compute the bandwidth of \hat{A} , and plot the sparsity pattern of its Cholesky factor L . How many non-zero entries are there in the Cholesky factors of A and \hat{A} ? Did RCM reduce fill-in for the Cholesky factorization?