

NUMERICAL ANALYSIS: PROJECT 01

Students: Cedric Orton-Urbina (ceo29), Aditya Syam (as2839), Pratyush Sudhakar (ps2245)

Due Date: 11th March, 2024

Question 1.

Show that if z solves

$$\min_z \|A(:, 1 : k)z - b\|_2^2$$

then

$$\|A \begin{bmatrix} z \\ 0 \end{bmatrix} - b\|_2 \leq \|R_{22}\|_2 \|x_{\text{LS}}\|_2 + \|b - Ax_{\text{LS}}\|_2,$$

where x_{LS} solves

$$\min_x \|Ax - b\|_2^2.$$

In other words, the residual from the sparse solution cannot be much larger than the optimal residual.

Answer 1.

Let $x_{\text{LS}}(1 : k) = x_k$ and $x_{\text{LS}}(k + 1 : n) = x_n$. We begin by rewriting A in terms of its partitioned QR factorization:

$$A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix} = \begin{bmatrix} Q_1 R_{11} & Q_1 R_{12} + Q_2 R_{22} \end{bmatrix}$$

We can then write Ax_{LS} and $A \begin{bmatrix} z \\ 0 \end{bmatrix}$ as:

$$Ax_{\text{LS}} = \begin{bmatrix} Q_1 R_{11} & Q_1 R_{12} + Q_2 R_{22} \end{bmatrix} \begin{bmatrix} x_k \\ x_n \end{bmatrix} = Q_1 R_{11} x_k + Q_2 R_{12} x_n + Q_2 R_{22} x_n$$

$$A \begin{bmatrix} z \\ 0 \end{bmatrix} = Q_1 R_{11} z$$

Note that $A \begin{bmatrix} z \\ 0 \end{bmatrix} = A(:, 1 : k)z$. We now observe the right side of the inequality:

$$\|R_{22}\|_2 \|x_{\text{LS}}\|_2 + \|b - Ax_{\text{LS}}\|_2$$

We first note that since x_n is a sparse x_{LS} (namely $\begin{bmatrix} \vec{0} \\ x_n \end{bmatrix}$), we see $\|R_{22}\|_2\|x_n\|_2 \leq \|R_{22}\|_2\|x_{\text{LS}}\|_2$. Thus, we have the inequality:

$$\|R_{22}\|_2\|x_n\|_2 + \|b - Ax_{\text{LS}}\|_2 \leq \|R_{22}\|_2\|x_{\text{LS}}\|_2 + \|b - Ax_{\text{LS}}\|_2$$

We use the norm identity $\|Ax\|_2 \leq \|A\|_2 \cdot \|x\|_2$ to write:

$$\|R_{22}x_n\|_2 + \|b - Ax_{\text{LS}}\|_2 \leq \|R_{22}\|_2\|x_{\text{LS}}\|_2 + \|b - Ax_{\text{LS}}\|_2$$

Since Q_2 is orthogonal, we can insert it inside of a 2-norm without changing it's value, thus we have:

$$\|Q_2R_{22}x_n\|_2 + \|b - Ax_{\text{LS}}\|_2 \leq \|R_{22}\|_2\|x_{\text{LS}}\|_2 + \|b - Ax_{\text{LS}}\|_2$$

Using the triangle inequality and plugging in our value of Ax_{LS} :

$$\begin{aligned} \|Q_2R_{22}x_n + b - Ax_{\text{LS}}\|_2 &\leq \|R_{22}\|_2\|x_{\text{LS}}\|_2 + \|b - Ax_{\text{LS}}\|_2 \\ \|Q_2R_{22}x_n + b - Q_1R_{11}x_k - Q_2R_{12}x_n - Q_2R_{22}x_n\|_2 &\leq \|R_{22}\|_2\|x_{\text{LS}}\|_2 + \|b - Ax_{\text{LS}}\|_2 \\ \|b - Q_1R_{11}x_k - Q_2R_{12}x_n\|_2 &\leq \|R_{22}\|_2\|x_{\text{LS}}\|_2 + \|b - Ax_{\text{LS}}\|_2 \\ \|b - Q_1R_{11}(x_k - (Q_1R_{11})^{-1}Q_2R_{12}x_n)\|_2 &\leq \|R_{22}\|_2\|x_{\text{LS}}\|_2 + \|b - Ax_{\text{LS}}\|_2 \end{aligned}$$

Note that $(Q_1R_{11})^{-1}Q_1R_{11} = I$, and that $(x_k - (Q_1R_{11})^{-1}Q_2R_{12}x_n)$ is simply equal to some vector $v \in \mathbb{R}^m$. We rewrite the inequality as:

$$\|Q_1R_{11}y - b\|_2 \leq \|R_{22}\|_2\|x_{\text{LS}}\|_2 + \|b - Ax_{\text{LS}}\|_2$$

Note $\|b - Q_1R_{11}y\|_2 = \|Q_1R_{11}y - b\|_2$. Now, observe the left side of the original quantity. We rewrite it as:

$$\|A \begin{bmatrix} z \\ 0 \end{bmatrix} - b\|_2 = \|A(:, 1:k)z - b\|_2 = \|Q_1R_{11}z - b\|_2$$

Since z is the minimum of the least squares problem, we find the following inequalities for any vector $y \in \mathbb{R}^m$:

$$\|Q_1R_{11}z - b\|_2^2 \leq \|Q_1R_{11}y - b\|_2^2$$

$$\|Q_1R_{11}z - b\|_2 \leq \|Q_1R_{11}y - b\|_2$$

Combining all these inequalities, we have:

$$\|Q_1 R_{11} z - b\|_2 \leq \|R_{22}\|_2 \|x_{\text{LS}}\|_2 + \|b - Ax_{\text{LS}}\|_2,$$

$$\left\| A \begin{bmatrix} z \\ 0 \end{bmatrix} - b \right\|_2 \leq \|R_{22}\|_2 \|x_{\text{LS}}\|_2 + \|b - Ax_{\text{LS}}\|_2,$$

Question 2.

Show that there exists a matrix B such that

$$\|A - A(:, 1 : k)B\|_2 = \|R_{22}\|_2$$

and

$$\|A(:, 1 : k) - A(:, 1 : k)B(:, 1 : k)\|_2 = 0.$$

The second condition ensures that k columns of A are exactly represented by the rank- k factorization $A(:, 1 : k)B$.

Answer 2.

For this question, we are asked to prove two conditions. We need to prove that there exists some matrix B such that the two conditions above are satisfied. We are given matrix A of the form

$$A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix}$$

Here, $Q_1 \in \mathbb{R}^{m \times k}$, $Q_2 \in \mathbb{R}^{m \times \ell - k}$, $R_{11} \in \mathbb{R}^{k \times k}$, $R_{12} \in \mathbb{R}^{k \times n - k}$, and $R_{22} \in \mathbb{R}^{\ell - k \times n - k}$. Multiplying out the expression for A , we get

$$A = \begin{bmatrix} Q_1 R_{11} & Q_1 R_{12} + Q_2 R_{22} \end{bmatrix}$$

Now, $A(:, 1 : k)$ comprises all the rows of A and the first k columns of A . Hence, $A(:, 1 : k) =$

$$\begin{bmatrix} Q_1 R_{11} \end{bmatrix}$$

Now, we need to define a B that satisfies both conditions. Let $B =$

$$\begin{bmatrix} I_k & R_{11}^{-1} R_{12} \end{bmatrix}$$

Here, I_k is the identity matrix of dimensions $k \times k$. Further, $R_{11}^{-1} R_{12}$ is of dimension $k \times (n - k)$. As a result, B is of size $k \times n$. Now, we check the first condition.

$$A(:, 1 : k)B = \begin{bmatrix} Q_1 R_{11} \end{bmatrix} \begin{bmatrix} I_k & R_{11}^{-1} R_{12} \end{bmatrix} = \begin{bmatrix} Q_1 R_{11} & Q_1 R_{12} \end{bmatrix}$$

However, we previously saw that

$$A = \begin{bmatrix} Q_1 R_{11} & Q_1 R_{12} + Q_2 R_{22} \end{bmatrix}$$

Thus,

$$A - A(:, 1 : k)B = \begin{bmatrix} Q_1 R_{11} & Q_1 R_{12} + Q_2 R_{22} \end{bmatrix} - \begin{bmatrix} Q_1 R_{11} & Q_1 R_{12} \end{bmatrix} = \begin{bmatrix} 0 & Q_2 R_{22} \end{bmatrix}$$

Finally, we calculate the 2-norm of this result. $\|A - A(:, 1 : k)B\|_2 = \left\| \begin{bmatrix} 0 & Q_2 R_{22} \end{bmatrix} \right\|_2$ The zero block does not contribute to the 2-norm, so this equals $\|Q_2 R_{22}\|_2$. Now, Q_2 is a orthogonal matrix and thus does not affect the 2-norm of any matrix when multiplied by it. Hence, $\|Q_2 R_{22}\|_2 = \|R_{22}\|_2$, and our first condition is satisfied since $\|A - A(:, 1 : k)B\|_2 = \|R_{22}\|_2$

We now proceed to the second condition. First, we see that $B(:, 1 : k)$ equals all the rows and the first k columns of B and so

$$B(:, 1 : k) = [I_k]$$

Now,

$$A(:, 1 : k)B(:, 1 : k) = [Q_1 R_{11}] [I_k] = [Q_1 R_{11}]$$

But, $A(:, 1 : k) =$

$$[Q_1 R_{11}]$$

Hence,

$$A(:, 1 : k) - A(:, 1 : k)B(:, 1 : k) = Q_1 R_{11} - Q_1 R_{11} = 0$$

and $\|A(:, 1 : k) - A(:, 1 : k)B(:, 1 : k)\|_2 = \|0\|_2 = 0$. As a result, the second condition is also satisfied for the given value of B .

Question 3.

Show that

$$\|R_{22}\|_2 \leq \sigma_1(A)$$

and that for any m, n , and k with $m \geq n$ and $k < \min(m, n)$ there exists a matrix A such that

$$\|R_{22}\|_2 = \sigma_1(A).$$

This shows that while R_{22} is bounded in size by the scale of A , there are examples where it is that large.

Answer 3.

We begin by noting that $\sigma_1(A)$ is the largest singular value of A , and is also equal to the 2-norm of A .

Since R_{22} is a partition of the QR factorization of A , and QR factorization does not increase any spectral properties of A (magnitude of singular values), $\|R_{22}\|_2$ can at most be σ_1 . This would be the case if the singular value decomposition of R_{22} contained the singular value σ_1 , otherwise, $\|R_{22}\|_2$ will be less than σ_1 . Thus,

$$\|R_{22}\|_2 \leq \sigma_1(A)$$

To show there exists A such that $\|R_{22}\|_2 = \sigma_1(A)$, consider a matrix $A \in \mathbb{R}^{m \times n}$ with entries only along the diagonal (A_{ii} for $i \leq \min(m, n)$). We then place the largest of these entries (σ_1 in the case of a diagonal matrix) in the $(k+1)^{th}$ diagonal of A . Since the QR decomposition of a diagonal matrix A with excess rows or columns is just the identity matrix times A , we know that the largest singular value σ_1 will be in the R_{22} partition of the QR factorization of A . The 2-norm of R_{22} will be equal to its largest singular value, which in this case, is the same largest singular value of A . Thus, there exists $A \in \mathbb{R}^{m \times n}$ such that

$$\|R_{22}\|_2 = \sigma_1(A)$$

Question 4.

$$A \begin{bmatrix} \Pi_1 & \Pi_2 \end{bmatrix} = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix} \quad (1)$$

Show that if we have a factorization of the form eq. (1) then there exists a matrix B such that

$$\|A - A(:, \mathcal{C})B\|_2 = \|R_{22}\|_2$$

and

$$\|A(:, \mathcal{C}) - A(:, \mathcal{C})B(:, \mathcal{C})\|_2 = 0.$$

Here we are just rewriting the earlier result using the permutations. A similar thing can be done for the least-squares problem.

Answer 4.

For this question, we are asked to prove two conditions. We need to prove that there exists some matrix B such that the two conditions above are satisfied. Given the expression in the question, we do not directly have an expression for A so we first use algebra to derive this. We are given

$$A \begin{bmatrix} \Pi_1 & \Pi_2 \end{bmatrix} = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix}$$

To obtain an equivalence for A , we need to multiply both sides by

$$\begin{bmatrix} \Pi_1 & \Pi_2 \end{bmatrix}^T$$

This gives us

$$A \begin{bmatrix} \Pi_1 & \Pi_2 \end{bmatrix} \begin{bmatrix} \Pi_1 & \Pi_2 \end{bmatrix}^T = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix} \begin{bmatrix} \Pi_1 & \Pi_2 \end{bmatrix}^T$$

The product of a permutation matrix with its transpose gives the identity matrix, so we end up with

$$A = \begin{bmatrix} Q_1 & Q_2 \end{bmatrix} \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix} \begin{bmatrix} \Pi_1 & \Pi_2 \end{bmatrix}^T = \begin{bmatrix} Q_1 R_{11} & Q_1 R_{12} + Q_2 R_{22} \end{bmatrix} \begin{bmatrix} \Pi_1 & \Pi_2 \end{bmatrix}^T$$

$$A = [Q_1 R_{11} \Pi_1^T + Q_1 R_{12} \Pi_2^T + Q_2 R_{22} \Pi_2^T]$$

Now, $A(:, C)$ would equal $A \Pi_1$ since C denotes the set of indices for the columns chosen by Π_1 . As a result, $A(:, C) = Q_1 R_{11}$. We need to select a B that satisfies both the conditions outlined previously. To this end, let

$$B = [\Pi_1^T + R_{11}^{-1} R_{12} \Pi_2^T]$$

Let us first investigate the dimensions here. Π_1 is of size $n \times k$, so Π_1^T is of size $k \times n$. Similarly, Π_2 is of size $n \times (n - k)$, so Π_2^T is of size $(n - k) \times n$. We already know the dimensions of the block matrices from R , so the size of B is $k \times n$. Now, we look at the first condition.

$$A(:, C)B = Q_1 R_{11} [\Pi_1^T + R_{11}^{-1} R_{12} \Pi_2^T] = [Q_1 R_{11} \Pi_1^T + Q_1 R_{11} R_{11}^{-1} R_{12} \Pi_2^T]$$

$$A(:, C)B = [Q_1 R_{11} \Pi_1^T + Q_1 R_{12} \Pi_2^T]$$

We already know

$$A = [Q_1 R_{11} \Pi_1^T + Q_1 R_{12} \Pi_2^T + Q_2 R_{22} \Pi_2^T]$$

Therefore,

$$A - A(:, C)B = [Q_1 R_{11} \Pi_1^T + Q_1 R_{12} \Pi_2^T + Q_2 R_{22} \Pi_2^T] - [Q_1 R_{11} \Pi_1^T + Q_1 R_{12} \Pi_2^T]$$

$$A - A(:, C)B = [Q_2 R_{22} \Pi_2^T]$$

Now, $\|Q_2 R_{22} \Pi_2^T\|_2 = \|R_{22} \Pi_2^T\|_2 = \|R_{22}\|_2$. This is because Q_2 is an orthogonal matrix and does not affect the 2-norm of a matrix when multiplied by it. Similarly, Π_2^T is a permutation matrix and does not affect the 2-norm either. This satisfies our first condition because we have $\|A - A(:, C)B\|_2 = \|Q_2 R_{22} \Pi_2^T\|_2 = \|R_{22}\|_2$, as required. Now, we evaluate the second condition. For $B(:, C)$, we take columns of B corresponding to the indices in C . Since Π_1 selects the columns corresponding to C , Π_1^T would select the corresponding rows. Hence, $B(:, C)$ ends up simply being the identity matrix of size $k \times k$. That is,

$$B(:, C) = I_k$$

Now, $A(:, C)$ is of size $m \times k$ and, as seen above, $B(:, C)$ is the identity matrix of size $k \times k$. As such, $A(:, C)B(:, C)$ yields back $A(:, C)$ as the output. We know

$$A(:, C) = Q_1 R_{11}$$

and so

$$A(:, C)B(:, C) = [Q_1 R_{11}] I_k = Q_1 R_{11} = A(:, C)$$

Finally, we have

$$A(:, C) - A(:, C)B(:, C) = 0$$

and $\|A(:, C) - A(:, C)B(:, C)\|_2 = \|0\|_2 = 0$, as required. This concludes the proof.

Question 5.

The greedy heuristic we will follow is to remove the largest column from R_{22} at each step. Specifically, let's say we have completed k steps of our process have have the partial factorization

$$Q^{(k)} \dots Q^{(1)} A \Pi^{(1)} \dots \Pi^{(k)} = \begin{bmatrix} R_{11} & R_{12} \\ & M \end{bmatrix}, \quad (2)$$

where $R_{11} \in \mathbb{R}^{k \times k}$ is upper triangular and $M \in \mathbb{R}^{(m-k) \times (n-k)}$ is dense (because we have not yet reduced it to upper triangular form).

Show that if we continue the factorization in eq. (2) without any additional permutations then $\|R_{22}\|_2 = \|M\|_2$.

Answer 5.

For this question, we essentially want to prove that the 2-norm of M , i.e., the 2-norm of R_{22} after k steps of the process, remains the same as the 2-norm of R_{22} after the factorization is completed, under the assumption that no further pivoting is performed after k steps are completed. The full factorization is of the form $Q^{(n)} \dots Q^{(1)} A \Pi^{(1)} \dots \Pi^{(n-1)} = R$, where we alternate the computations of $\Pi^{(i)}$ and $Q^{(i)}$. Following the greedy heuristic, we are given that k steps of the column-pivoted factorization have been completed. At this stage of the factorization, we have

$$Q^{(k)} \dots Q^{(1)} A \Pi^{(1)} \dots \Pi^{(k)} = \begin{bmatrix} R_{11} & R_{12} \\ & M \end{bmatrix}$$

Consider the 2-norm of the block M at this stage. We know that the 2-norm is the largest singular value. Now, we continue factorization without column pivoting. This means that we only compute $Q^{(i)}$ for the remaining steps, but do not calculate additional $\Pi^{(i)}$. As a result, we end up with a final factorization of the form

$$Q^{(n)} Q^{(n-1)} \dots Q^{(k)} \dots Q^{(1)} A \Pi^{(1)} \dots \Pi^{(k)} = \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix}$$

which is equivalent to

$$A \Pi = Q \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix}$$

where

$$\Pi = \Pi^{(1)} \dots \Pi^{(k)}$$

and

$$Q = Q^{(1)} \dots Q^{(n)}$$

Essentially, after the k^{th} step, there are only additional orthogonal $Q^{(i)}$ multiplied and no additional permutation $\Pi^{(i)}$ multiplied. After the remaining $n-k$ steps are completed, the only change would be the additional

$$Q^{(k+1)} \dots Q^{(n-1)} Q^{(n)}$$

which will make up the remainder of Q in the form

$$Q = Q^{(1)} \dots Q^{(k)} Q^{(k+1)} \dots Q^{(n-1)} Q^{(n)}$$

To express the incompletely factored matrix in terms of the entirety of the $Q^{(i)}$ s, we multiply the additional $Q^{(i)}$ s to both sides of (2) and obtain

$$Q^{(n)} Q^{(n-1)} \dots Q^{(k+1)} Q^{(k)} \dots Q^{(1)} A \Pi^{(1)} \dots \Pi^{(k)} = Q^{(n)} Q^{(n-1)} \dots Q^{(k+1)} \begin{bmatrix} R_{11} & R_{12} \\ & M \end{bmatrix}$$

Setting $Q^{(n)} Q^{(n-1)} \dots Q^{(k+1)}$ as Q_{diff} , we can see that

$$Q_{diff} \begin{bmatrix} R_{11} & R_{12} \\ & M \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} \\ & R_{22} \end{bmatrix} = R$$

Restricting ourselves to only the right block of R , we see that

$$Q_{diff} \begin{bmatrix} R_{12} \\ M \end{bmatrix} = \begin{bmatrix} R_{12} \\ R_{22} \end{bmatrix}$$

Now, let us consider the 2-norms of these two matrices. Since the matrices are equal, the 2-norms must also be equal. Hence, $\|Q_{diff} \begin{bmatrix} R_{12} \\ M \end{bmatrix}\|_2 = \|\begin{bmatrix} R_{12} \\ R_{22} \end{bmatrix}\|_2$. Q_{diff} is a product of orthogonal matrices and hence is itself an orthogonal matrix. We also know that multiplication by an orthogonal matrix does not affect the 2-norm of a matrix. Hence, we can say $\|Q_{diff} \begin{bmatrix} R_{12} \\ M \end{bmatrix}\|_2 = \|\begin{bmatrix} R_{12} \\ M \end{bmatrix}\|_2$ and further $\|\begin{bmatrix} R_{12} \\ M \end{bmatrix}\|_2 = \|\begin{bmatrix} R_{12} \\ R_{22} \end{bmatrix}\|_2$. Now that we know the

2-norms of these two matrices are equal, we can square both sides to obtain $\|\begin{bmatrix} R_{12} \\ R_{22} \end{bmatrix}\|_2^2 =$

$\|\begin{bmatrix} R_{12} \\ M \end{bmatrix}\|_2^2$. We know that the square of the 2-norm can be defined as the square of all its

entries. As such, $\|\begin{bmatrix} R_{12} \\ R_{22} \end{bmatrix}\|_2^2$ is the square of all the entries in R_{12} and R_{22} whereas $\|\begin{bmatrix} R_{12} \\ M \end{bmatrix}\|_2^2$ is the square of all entries in R_{12} and M . Evidently, the square of the entries in R_{12} is common on both sides, and nullifying this sum gives us that the square of the entries in R_{22} equals those in M . Taking the square root on both sides of the equation, we get the equality of the 2-norms of R_{22} and M . Hence, $\|R_{22}\|_2 = \|M\|_2$ and the proof is complete.

Question 6.

Show that Π_1 in eq. (1) only depends on $\Pi^{(1)}$ through $\Pi^{(k)}$. In other words, if we just want/need \mathcal{C} we can stop the factorization after k steps. We can also recover the matrix B in our low-rank factorization without proceeding further.

Answer 6.

Based on previous definitions, we have:

$$\Pi = [\Pi_1 \quad \Pi_2] = \Pi^{(1)} \dots \Pi^{(n-1)}$$

The greedy heuristic employed in the given QR factorization utilizes the fact that $\Pi^{(i)}$ will only affect the i^{th} and subsequent columns of whatever matrix it is acts on. Thus, $\Pi^{(k+1)}$ will only impact the $(k+1)^{th}$ and subsequent columns. This implies that $\Pi^{(1)} \dots \Pi^{(k)}$ will only uniquely affect the first k columns of the matrix it is applied to. By definition, Π_1 represents the first k columns of Π , so it follows trivially that the individual permutation matrices $\Pi^{(1)} \dots \Pi^{(k)}$ will be the only matrices that affect the first k columns. This can be seen as subsequent $\Pi^{(k+1)} \dots \Pi^{(n-1)}$ will contribute to columns in Π after k , in other words, Π_2

1 Goal 1.

Write pseudocode for the described algorithm, and show that its arithmetic complexity is $\mathcal{O}(mnk)$ if we stop at step k .

Answer Goal 1.

Algorithm 1 QR Factorization Using Householder Reflectors and Column Pivoting

```
 $R \leftarrow A$ 
 $Q \leftarrow I$ 
 $P \leftarrow I$ 
for  $j = 1, 2, \dots, k$  do
     $idx = \arg \max_{i=j, \dots, n} \|R(j, j : n)\|_2$ 

    if  $idx \neq j$  then
         $R(:, j : idx) \leftarrow R(:, idx : j)$ 
         $P(:, j : idx) \leftarrow P(:, idx : j)$ 
    end if

     $w \leftarrow R(j : m, j)$ 
     $s \leftarrow \text{sign}(w(1))$ 
     $u \leftarrow s \times \|w\|_2 \times \vec{e}_1 + w$ 
     $u \leftarrow \frac{u}{\|u\|_2}$ 

     $R(j : m, j : n) \leftarrow R(j : m, j : n) - 2 \times (u \times (u^T R(j : m, j : n)))$ 
     $R(j + 1 : m, j) \leftarrow 0$ 
     $Q(j : m, j : m) \leftarrow Q(j : m, j : m) - 2 \times (u \times (u^T Q(j : m, j : n)))$ 
end for
```

We analyze the time complexity of each step:

1. $R \leftarrow A, Q \leftarrow I, P \leftarrow I$ These time complexities are $\mathcal{O}(mn)$ as we assign m rows and n columns.

2. $idx = \arg \max_{i=j, \dots, n} \|R(j, j : n)\|_2$

This has time complexity of $\mathcal{O}(mn)$ as we search through a maximum of n columns, computing the norm of each. Since we have m elements in each column, we approximate the number of computations as $\mathcal{O}(mn)$

3. $R(:, j : idx) \leftarrow R(:, idx : j), P(:, j : idx) \leftarrow P(:, idx : j)$

Depending on the implementation, this swap may take up to $\mathcal{O}(n)$ time, but for now, we assume this to be constant time.

4. $w \leftarrow R(j : m, j), s \leftarrow \text{sign}(w(1)), u \leftarrow s \times \|w\|_2 \times \vec{e}_1 + w$

All of these steps simply perform arithmetic on a single column of R or w itself. These operations all take $\mathcal{O}(m)$ time.

5. $R(j : m, j : n) \leftarrow R(j : m, j : n) - 2 \times (u \times (u^T R(j : m, j : n))), Q(j : m, j : m) \leftarrow Q(j : m, j : m) - 2 \times (u \times (u^T Q(j : m, j : n)))$

Since both these steps take the same form, we analyze them together. The product of u^T and $R(j : m, j : n)$ computes on the order of n dot products. Since each dot product will take $\mathcal{O}(m)$, these steps combined will take $\mathcal{O}(mn)$ time complexity.

Next, the scaling of u with the previously resultant performs m computations for n elements, thus again having time complexity $\mathcal{O}(mn)$.

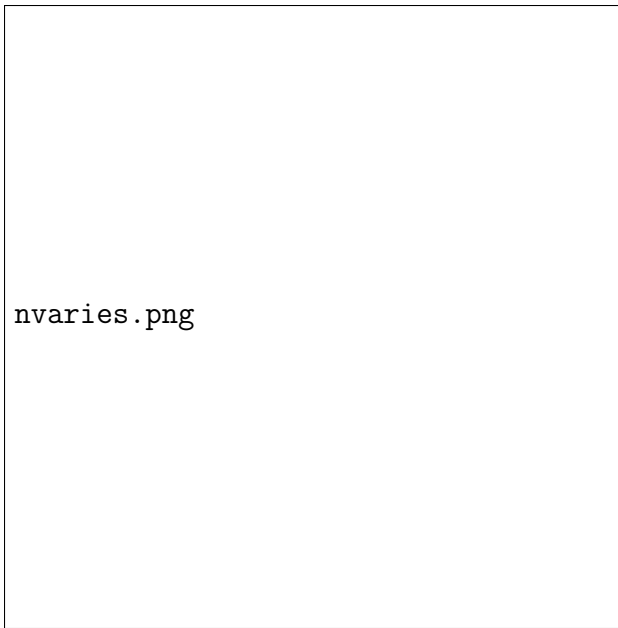
Finally, we have an element-wise matrix subtraction, which will perform mn operations, which is again $\mathcal{O}(mn)$.

If we combine the time complexity of each iteration (2 - 5), we find the total complexity to be $\mathcal{O}(mn)$. Since we are performing k total iterations, this implies the total time complexity of our algorithm is $\mathcal{O}(mnk)$

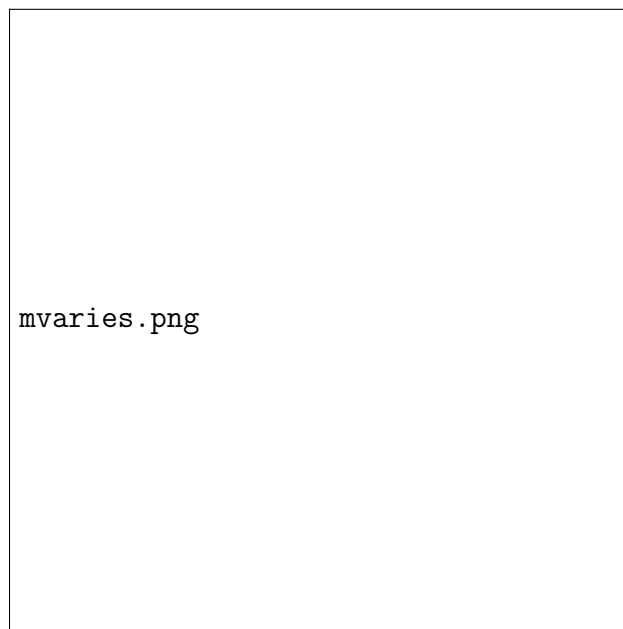
Goal 2.

We are asked to implement the column pivoted QR factorization and show that it meets the three scaling requirements.

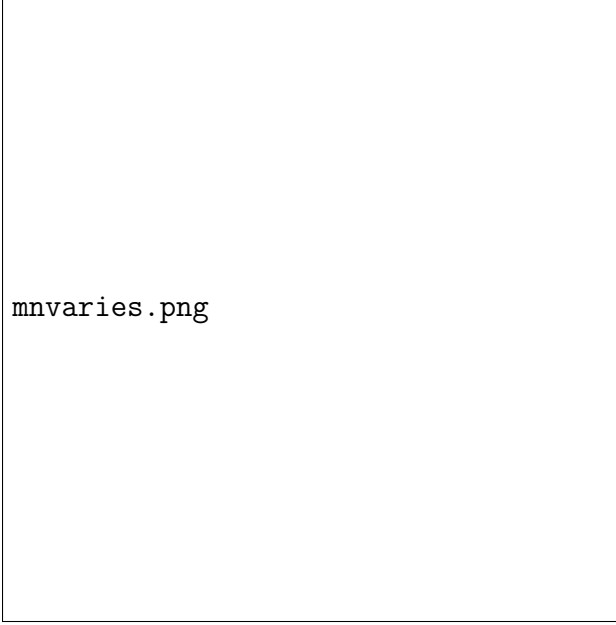
Our primary function `compute_QR` achieves this implementation by using Householder reflections and column pivoting. We now explore the execution times for our algorithm.



For the variation in solely n , we see that the slope is 0.51 for the graph, and hence we can say that it scales linearly, i.e., in $O(n)$ time.



For the variation in solely m , we see that the slope is 1.20 for the graph, and hence we can say that it scales linearly, i.e., in $O(n)$ time.




For the variation in m and n simultaneously, where $m = n$, we see that the slope is 1.74 for the graph, and hence we can say that it scales quadratically, i.e., in $O(n^2)$ time.

Goal 3.

For this goal, we are asked to build a matrix G of size 200×200 where $G_{i,j} = e^{-2\|x_i - x_j\|_2^2}$ and x_i are i.i.d points from the uniform distribution.

Our *generate_matrix_G* function builds a matrix of the appropriate size. We also define a function *approximation_error* that helps us compute the error using `linalg.norm` when needed. Our primary focus here is to use the QR factorization implemented previously to compute a low-rank approximation of G and compare the error to the optimal error for a rank- k approximation. Our function *low_rank_approximations* performs the major computation for this part. The singular value decomposition (SVD) is computed for the optimal case, whereas we call our own function *compute_QR* to compute the low-rank approximation. Finally, we compute the errors between the two approximations and the original matrix G .

In the given graph, it is evident that the approximation error for our QR factorization is higher than the optimal error for all cases. The optimal error is steadily decreasing with increasing k , whereas the pattern for our QR error decrease is more uneven. In fact, as k gets larger (closer and closer to 100), the gap in the approximation errors widens. As a result, we lose more with increasing k by constraining our low-rank optimization.



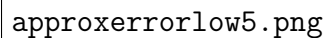
approxerrorlow3.png

Goal 4.

We are asked to implement the model given $M(x) = s_m(f_m(f_{m-1}(\dots f_1(x^T)))) W^{(s)}$. We built our model using a combination of ReLU and Softmax and then compute its accuracy. We have found the accuracy to equal 0.70 or 70%.

Goal 5.

Here, we want to create the matrix of the form $\sigma(X_c^T W^{(1)})$ and compare the error of a rank-k approximation using either column-pivoted QR or singular value decomposition.




approxerrorlow5.png

In the given graph, we notice that the error using QR factorization is once again always higher than that for the optimal (SVD) case. The approximation error decreases as the value of k increases, but in this case, the lines of best fit for both the methods seem to follow an almost identical pattern of descent (a vertical shift of sorts). Thus, the error using QR factorization moves in a similar way to the error using SVD.

Goal 6.

We want to compress the first layer of the model implemented. Essentially, we want to reduce the number of columns in $W^{(1)}$ and see how the accuracy varies. Intuitively, as we compress the first layer of the model, we expect the accuracy to be drastically affected.



accuracies6.png

In the given graph, we observe that the accuracy seems to be rather low when only a fraction of the columns in $W^{(1)}$ remain. The appropriate range of the fractional size of $W^{(1)}$ remaining is shown from about 30% to 50%. We know that the first layer is 1024*2048. For 1024 data points, we never need more than 1024 neurons to exactly match the layers output. As such, we preserve at most 50% of the neurons and so the maximum fraction of columns of $W^{(1)}$ that remain would be 50%. The change in the accuracy with the fraction of columns remaining first increases sharply with the number of columns decreasing from the 50% mark and then drops gradually as we get closer to the 35% mark. Following this, we see another sharp increase in accuracy as we hit the 30% mark for the fraction remaining. The fluctuation in the accuracy probably reflects a recurring pattern of unimportant (and perhaps error-inducing) data being lost, followed by important data being lost, as we lose columns.

Goal 7.

Now, we compress the whole model and report the accuracy of the compressed model versus the fraction of total columns remaining in $\hat{W}^{(1)}$ through $\hat{W}^{(7)}$.

Goal 8.

For the final goal, we try to conceptualize a different scheme of compressing the model and optimizing it to achieve a better trade-off between accuracy and model parameters.

We will check the accuracy when different layers are compressed and then select the layer that has the most impact on the accuracy.

Once, we find the layer that gives the highest accuracy, we will iterate through the next layers to find the next best accuracy improving layer. We will continue this process until we have compressed the model to the desired level.