MATH-6800
Vignesh Ramakrishnan
RIN: 662028006
Due: Thursday September 29, 2022

Problem Set 3

1. Let $P \in \mathbb{C}^{m\times m}$ be a nonzero projector. Show that $||P||_2 \geq 1$, with equality if and only if P is an orthogonal projector.

Since, P is a projector, $P^2 = P$.

$$\begin{aligned} \left|\left|P^2\right|\right|_2 &\leq \; ||P||_2 \, ||P||_2 \\ ||P||_2 &\leq \; ||P||_2^2 \\ 1 &\leq \; ||P||_2 \end{aligned}$$

Now, let P be an orthogonal projector. For an orthogonal projector, the singular values are $1, 0$. $||P||_2 = \max \sigma_i = 1$. This means, the equality comes if P is orthogonal. Coming to the "only if" part, Let P be any non-orthogonal projector. If $y \in \text{range}\,(P)$, then $\exists\; x : Px = y$ and $Py = y$.

$$\begin{aligned} ||Px||_2^2 &= x^*P^*Px \\ &= x^*P^*y \\ &\leq \; ||(Px)^*||_2 \, ||y||_2 \, , \text{ Cauchy-Schwartz inequality} \\ ||P||_2^2 &\leq \frac{||y||_2^2}{||x||_2^2} \end{aligned}$$

If $y = x$, the inequality becomes $||P||_2 \leq 1$ but we know from before that $||P||_2 \geq 1$. This means, the only possibility is $||P||_2 = 1$. Lets see when this can come up. If $P = P^*$, then $x^*P^*y = x^*Py = x^*y = x^*x = ||x||_2^2$. Therefore, its only possible if P is orthogonal.

2. Consider again the matrices A and B of Exercise 6.4.

   (a) Using any method you like, determine (on paper) a reduced QR factorization $A = \hat{Q}\hat{R}$ and a full QR factorization $A = QR$.

$$A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} = \hat{Q}\hat{R}$$

We know that, $a_1 = r_{11}q_1$ and $|r_{11}| = ||a||_2$. So, let $r_{11} = \sqrt{2}$. This means $q_1 = \left(1/\sqrt{2}a_1\right)$. $a_2 = r_{12}q_1 + r_{22}q_2$. Multiplying by q_1^*, we get, $r_{12} = q_1^*a_2 = 0$. This means $a_2 = r_{22}q_2$ and similarly, $|r_{22}| = ||a_2||_2$. I choose $r_{22} = 1$. Therefore, I choose $q_2 = a_2$. The reduced QR factorization is as follows:

$$A = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 \\ 0 & 1 \\ \frac{1}{\sqrt{2}} & 0 \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & 1 \end{bmatrix} = \hat{Q}\hat{R}$$

For a full-factorization we need to choose q_3 which is orthogonal to both q_1 and q_2. I choose it as $q_3 = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ 0 \\ \frac{1}{\sqrt{2}} \end{bmatrix}$. This makes the last row of $R = 0$. Hence full factorization looks like,

$$A = \begin{bmatrix} \frac{1}{\sqrt{2}} & 0 & -\frac{1}{\sqrt{2}} \\ 0 & 1 & 0 \\ \frac{1}{\sqrt{2}} & 0 & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \sqrt{2} & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix} = QR$$

(b) Again using any method you like, determine reduced and full QR factorizations $B = \hat{Q}\hat{R}$ and $B = QR$.

$$B = \begin{bmatrix} 1 & 2 \\ 0 & 1 \\ 1 & 0 \end{bmatrix}$$

We know that $b_1 = r_{11}q_1$ and by taking 2-norm on both sides, we get $|r_{11}| = ||b_1||_2 = \sqrt{2}$. I choose $r_{11} = \sqrt{2}$. This means $q_1 = \frac{1}{\sqrt{2}}b_1$. Then, we also know that, $b_2 = r_{12}q_1 + r_{22}q_2$. Multiplying by q_1^*, yields $q_1^*b_2 = r_{12} = \sqrt{2}$. $||b_2 - r_{12}q_1||_2 = ||r_{22}q_2||_2 = |r_{22}| = \sqrt{3}$. I choose $r_{22} = \sqrt{3}$. This lets me find q_2 as $\begin{bmatrix} \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{3}} \\ -\frac{1}{\sqrt{3}} \end{bmatrix}$. Therefore,

$$B = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} \\ 0 & \frac{1}{\sqrt{3}} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{3}} \end{bmatrix} \begin{bmatrix} \sqrt{2} & \sqrt{2} \\ 0 & \sqrt{3} \end{bmatrix} = \hat{Q}\hat{R}$$

To find a vector that is orthogonal to both q_1 and q_2, I try to satisfy the face that $q_3^*q_1 = 0$ and $q_3^*q_2 = 0$ and that leads to the following,

$$q_3^1 = -q_3^3$$
$$q_3^2 = 2q_3^3$$

And setting, $q_3^3 = \frac{1}{\sqrt{2}}$, we get $q_3 = \begin{bmatrix} -\frac{1}{\sqrt{2}} \\ \sqrt{2} \\ \frac{1}{\sqrt{2}} \end{bmatrix}$ Therefore,

$$B = \begin{bmatrix} \frac{1}{\sqrt{2}} & \frac{1}{\sqrt{3}} & -\frac{1}{\sqrt{2}} \\ 0 & \frac{1}{\sqrt{3}} & \sqrt{2} \\ \frac{1}{\sqrt{2}} & -\frac{1}{\sqrt{3}} & \frac{1}{\sqrt{2}} \end{bmatrix} \begin{bmatrix} \sqrt{2} & \sqrt{2} \\ 0 & \sqrt{3} \\ 0 & 0 \end{bmatrix} = QR$$

3. Let A be an $m \times n$ matrix ($m \geq n$), and let $A = \hat{Q}\hat{R}$ be a reduced QR factorization.

(a) Show that A has rank n if and only if all the diagonal entries of $\hat{R}$ are nonzero.

Let us start the proof by saying that $\hat{R}$ has diagonal entries to be zero and yet we prove that A can have full rank n.

$$\begin{aligned} a_1 &= r_{11}q_1 \\ a_2 &= r_{12}q_1 + r_{22}q_2 \\ &\vdots \\ a_i &= \sum_{k=1}^{i} r_{ki}q_k \end{aligned}$$

From here can write $|r_{ii}| = \left|\left|a_i - \sum_{k=1}^{i-1} r_{ki}q_k\right|\right|_2$. Since $\hat{R}$ has zero diagonal entries,

$$a_i = \sum_{k=1}^{i-1} r_{ki}q_k$$

For A to be full rank n, we need $< a_i >$ and $< q_i >$ to span the same subspaces and all columns of A to be linearly independent. But from above we see that a_i is a linear combination of columns of Q all the way from $< q_1, q_2, \ldots q_{i-1} >$. This means, A cant be full rank as there is linear dependence. Conversely, we can definitely say that if A is full rank n, then it has n linearly independent columns and that $\left|\left|a_i - \sum_{k=1}^{i-1} r_{ki}q_k\right|\right|_2 \neq 0$ which means r_{ii} is non-zero.

(b) Suppose $\hat{R}$ has k nonzero diagonal entries for some k with $0 \leq k \leq n$. What does this imply about the rank of A? Exactly k? At least k? At most k? Give a precise answer, and prove it.

If $\hat{R}$ has k non-zero entries, then

$$\begin{aligned} a_i &= \left(\sum_{k=1}^{i-1} r_{ki}q_k\right) + r_{ii}q_i, \text{ if } r_{ii} \neq 0, \ 0 \leq i \leq k \leq n \\ a_i &= \left(\sum_{k=1}^{i-1} r_{ki}q_k\right), \text{ if } r_{ii} = 0 \end{aligned}$$

Lets say that at $i = k-2$ we have $r_{ii} = 0$, then for $a_i, i = k+2$, we still need contributions of q_{k-2} and the linear combinations will range from utmost k to greater than k but still less than n. This means, A can have at either k linearly independent columns or more than k linearly independent columns (still less than n).

4. The MATLAB codes are listed in Listing 1, 2, and 3.

```
function [Qh, Rh] = clgs(A)

[m,n] = size(A);
Qh = zeros(m,n);
Rh = zeros(n);

for j=1:n
    vj = A(:,j);
    if(j~=1)
```

```
        for i=1:j-1
            Rh(i,j) = Qh(:,i)'*A(:,j);
            vj = vj - Rh(i,j)*Qh(:,i);
        end
    end
    Rh(j,j) = norm(vj);
    Qh(:,j) = vj/Rh(j,j);
end

end
```

Listing 1: clgs algorithm

```
function [Qh, Rh] = mgs(A)

[m,n] = size(A);
Qh = zeros(m,n);
Rh = zeros(n);

V = A;

for i=1:n
    Rh(i,i) = norm(V(:,i));
    Qh(:,i) = V(:,i)/Rh(i,i);
    for j=i+1:n
        Rh(i,j) = Qh(:,i)'*V(:,j);
        V(:,j) = V(:,j) - Rh(i,j)*Qh(:,i);
    end
end

end
```

Listing 2: mgs algorithm

```
function [] = QR_func(m)
A = zeros(m);
x = zeros(m,1);

for i=1:m
    x(i,1) = (i-1)/(m-1);
end

for i=1:m
    A(:,i) = x.^(i-1);
end

[Qc, Rc] = clgs(A);
[Qm, Rm] = mgs(A);
[Q, R] = qr(A);

for i=1:m
    if(R(i,i)<0)
        Q(:,i) = -1*Q(:,i);
        R(i,:) = -1*R(i,:);
    end
end

disp("norm2(A-QR) = ");
disp(norm(A-Q*R));
```

```
disp("norm2(Qc-Q) = ");
disp(norm(Qc-Q));

disp("norm2(Qm-Q) = ");
disp(norm(Qm-Q));

disp("norm2(Rc-R) = ");
disp(norm(Rc-R));

disp("norm2(Rm-R) = ");
disp(norm(Rm-R));

disp("norm2(Qc^*Qc - I)");
disp(norm(Qc'*Qc - eye(m)));

disp("norm2(Qm^*Qm - I)");
disp(norm(Qm'*Qm - eye(m)));

disp("norm2(Q^*Q - I)");
disp(norm(Q'*Q - eye(m)));
end
```

Listing 3: QR function algorithm

It can be noted in Part B, when $m = 100$, that the modified gs algorithm performs much better than the classical gs algorithm as it has significantly lower error norms. Yet, it wasn't remotely even close to the QR factorization that is in-built in Matlab.

```
Part A. results
norm2(A-QR) =
   1.0144e-15

norm2(Qc-Q) =
   2.6987e-14

norm2(Qm-Q) =
   2.6317e-14

norm2(Rc-R) =
   4.5084e-16

norm2(Rm-R) =
   5.1851e-16

norm2(Qc^*Qc - I)
   2.6683e-14

norm2(Qm^*Qm - I)
   2.6493e-14

norm2(Q^*Q - I)
   7.7604e-16

Part B. results
norm2(A-QR) =
   3.5916e-15

norm2(Qc-Q) =
    9.5344

norm2(Qm-Q) =
    1.9839

norm2(Rc-R) =
   20.5795

norm2(Rm-R) =
    0.0051

norm2(Qc^*Qc - I)
   89.9271

norm2(Qm^*Qm - I)
    1.0009

norm2(Q^*Q - I)
   2.2023e-15

>>
```