Set 3.aux Set 3.aux Set 3.aux

# MATH-6800, Problem Set 3, 10-1-15

**Name: Michael Hennessey**

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.

   $\Rightarrow$ If $P$ is orthogonal, we can write it in the form $P = Q^*\Sigma Q$ where $Q$ is unitary and $\Sigma$ is diagonal with entries $\sigma_{ii} = 1$ or 0. Then, by the properties of the 2-norm:

   $$P_2 = Q^*\Sigma Q_2 = \Sigma_2 = 1.$$

   $\Leftarrow$ We begin by showing that $P_2 > 1$ for oblique $P$.
   Let $v = x + \lambda a$ with $x = Pv, \lambda a = (I - P)v$. We choose $x$ and $a$ such that $x_2 = a_2 = 1$. Therefore, we are choosing $Pv_2 = 1$. Now, we attempt to minimize $v_2$ in order to maximize the 2-norm of $P$.

   $$v_2^2 = x + \lambda a_2^2 = (x+\lambda a)^*(x+\lambda a) = x^*x + 2\lambda x^*a + \lambda^2 a^*a$$
   $$= x_2^2 + \lambda^2 a_2^2 + 2\lambda x^*a = 1 + \lambda^2 + 2\lambda x^*a$$

   If we choose $\lambda = -x^*a$, we get $v_2 < 1$:

   $$v_2^2 = 1 + |x^*a|^2 - 2|x^*a|^2 = 1 - |x^*a|^2.$$

   Since $P$ is oblique, $x^*a \neq 0$, and

   $$v_2^2 = 1 - |x^*a|^2 < 1 \implies v_2 < 1.$$

   Thus
   $$P_2 = max_{y\neq 0}\frac{Py_2}{y_2} \geq \frac{Pv_2}{v_2} > 1.$$
   However, if $P$ is orthogonal, we know $x^*a = 0$, then $v_2 = 1$. Then,
   $$P_2 = max_{y_2\neq 0}\frac{Py_2}{y_2} \geq \frac{Pv_2}{v_2} = 1.$$
   Therefore, if $P_2 = 1$, $P$ must be orthogonal.

2. Consider the matrices

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

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

$$r_{11} = \sqrt{1^2 + 0^2 + 1^2} = \sqrt{2} \implies q_1 = \begin{bmatrix} 1/\sqrt{2} \\ 0 \\ 1/\sqrt{2} \end{bmatrix}$$

$$r_{12} = \begin{bmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = 0$$

$$r_{22} = a_2 - r_{12}q_{12} = a_{22} = 1 \implies q_2 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

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

To find $A = QR$ we must find $q_3 = [q_x, q_y, q_z]^T$ orthogonal to $q_1, q_2$:

$$q_1^* q_3 = 0 \implies q_x = -q_z$$

$$q_2^* q_3 = 0 \implies q_y = 0$$

Then,

$$q_3 = \begin{bmatrix} 1/\sqrt{2} \\ 0 \\ -1/\sqrt{2} \end{bmatrix}$$

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

(b) Determine reduced and full QR factorizations $B = \hat{Q}\hat{R}$ and $B = QR$.

$$r_{11} = \sqrt{1^2 + 0^2 + 1^2} = \sqrt{2} \implies q_1 = \begin{bmatrix} 1/\sqrt{2} \\ 0 \\ 1/\sqrt{2} \end{bmatrix}$$

$$r_{12} = \begin{bmatrix} 1/\sqrt{2} & 0 & 1/\sqrt{2} \end{bmatrix} \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} = \sqrt{2}$$

$$r_{22} = \left\| \begin{bmatrix} 2 \\ 1 \\ 0 \end{bmatrix} - \sqrt{2} \begin{bmatrix} 1/\sqrt{2} \\ 0 \\ 1/\sqrt{2} \end{bmatrix} \right\|_2 = \left\| \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} \right\|_2 = \sqrt{3}$$

$$\implies q_2 = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} = \begin{bmatrix} 1/\sqrt{3} \\ 1/\sqrt{3} \\ -1/\sqrt{3} \end{bmatrix}$$

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

To find $B = QR$ we must find $q_3 = [q_x, q_y q_z]^T$ orthogonal to $q_1, q_2$. We solve the matrix equation

$$\begin{bmatrix} 1 & 0 & 1 \\ 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} q_x \\ q_y \\ q_z \end{bmatrix} = 0$$

We then get

$$\tilde{q_3} = \begin{bmatrix} -1 \\ 2 \\ 1 \end{bmatrix} \implies q_3 = \begin{bmatrix} -1/\sqrt{6} \\ 2/\sqrt{6} \\ 1/\sqrt{6} \end{bmatrix}$$

Then

$$B = QR = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{3} & -1/\sqrt{6} \\ 0 & 1/\sqrt{3} & 2/\sqrt{6} \\ 1/\sqrt{2} & -1/\sqrt{3} & 1/\sqrt{6} \end{bmatrix}$$

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.

$\Rightarrow$ If $A$ has rank $n$, $r_{jj} > 0$ by Theorem 7.2.
$\Leftarrow$ We begin by looking at the matrix equation $Ax = 0$ to determine the dimension of the null space of $A$. We then factor $A$ to get

$$QRx = 0.$$

We then multiply both sides of the equation by $Q^*$ to get

$$Rx = 0.$$

We can then solve this equation by back substitution. Since all diagonal elements of $R$ are nonzero, we know that the dimension of the null space of $A$ is zero. Therefore, the rank of $A$ is $n$.

(b) Suppose $\hat{R}$ has $k$ nonzero diagonal entries for some $k$ with $0 \leq k < n$. What does this imply about the rank of $A$?

$rank(A) \geq k$
Following the same logic as the proof above, the number of independent columns of $A$ can be determined by the null space of $A$.

$$Ax = 0 \implies QRx = 0 \implies Rx = 0$$

Since $R$ has $k$ nonzero diagonal entries, we know that the dimension of the null space of $A$ can be equal to $n - k$. Thus A would have $n - (n - k) = k$ independent columns. We look at the matrix

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

. We know $rank(A) = 1$, but $r_{11} = r_{22} = 0$. Therefore, $rank(A) \geq k$.

4. Write matlab functions $[Qc, Rc] = clgs(A)$ and $[Qm, Rm] = mgs(A)$ that implement the reduced QR factorization using the classical Gram-Schmidt and modified Gram-Schmidt algorithms, respectively. Test the implementations by computing the QR factorization for the $m \times m$ Vandermonde matrix for points $x_i \equiv (i - 1)/(m - 1)$, and compare the results from the built-in Matlab function $[Q, R] = qr(A)$.

(a) For $m = 5$, compute $A - QR_2$ for each approximation. Also compute the 2-norm differences $Q_i - Q_2, R_i - R_2$ for $i = c$ and $i = m$, and also compute the error $Q^*Q - I_2$ for each approximation to $Q$.

(b) Repeat (a), but with $m = 100$.

In order to compute the norms we use the algorithms below: function [Vm]=Vandermonde(m) for i=1:m x(i)=(i-1)/(m-1); for j=1:m Vm(i,j)=x(i)$^(j-1); endend$

function [Qm,Rm]=mgs(A)

[m,n]=size(A); Rm=zeros(n,n); Qm=zeros(m,n); I=1:m; for i=1:n v(I,i)=A(I,i); end for i=1:n Rm(i,i)=norm(v(I,i),2); Qm(I,i)=v(I,i)/Rm(i,i); for j=(i+1):n Rm(i,j)=dot(Qm(I,i),v(I,j)); v(I,j)=v(I,j)-Rm(i,j)*Qm(I,i); end end

function [Qc,Rc]=clgs(A)

[m,n]=size(A); Rc=zeros(n,n); Qc=zeros(m,n); I=1:m; for j=1:n vj=A(I,j); for i=1;(j-1); Rc(i,j)=dot(Qc(I,i),A(I,j)); vj=vj-Rc(i,j)*Qc(I,i); end Rc(j,j)=norm(vj,2); Qc(I,j)=vj/Rc(j,j); end function [Qp,Rp]=qr$_plus(Q,R)$

[m,m]=size(R); I=1:m; for i=1:m if R(i,i)¡0 Rp(i,I)=-1*R(i,I); Qp(I,i)=-1*Q(I,i); else Rp(i,I)=R(i,I); Qp(I,i)=Q(I,i); end end See attached printouts for computations.