Gram–Schmidt Process and QR Decomposition

Recall from class that the Gram–Schmidt process takes a basis $\{x_1, \ldots, x_p\}$ for a subspace W of $\mathbb{R}^n$ and produces an orthogonal basis $\{v_1, \ldots, v_p\}$ for the same subspace W, where

$$\begin{aligned}
v_1 &= x_1\\
v_2 &= x_2 - \frac{x_2 \cdot v_1}{v_1 \cdot v_1} v_1\\
v_3 &= x_3 - \frac{x_3 \cdot v_1}{v_1 \cdot v_1} v_1 - \frac{x_3 \cdot v_2}{v_2 \cdot v_2} v_2\\
&\vdots\\
v_p &= x_p - \frac{x_p \cdot v_1}{v_1 \cdot v_1} v_1 - \frac{x_p \cdot v_2}{v_2 \cdot v_2} v_2 - \cdots - \frac{x_p \cdot v_{p-1}}{v_{p-1} \cdot v_{p-1}} v_{p-1}
\end{aligned}$$

1. The first example in class was applying the Gram–Schmidt process (before it was called that) to the following basis for $\mathbb{R}^2$

$$\left\{ \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 3 \\ 1 \end{bmatrix} \right\}$$

to get the orthogonal basis for $\mathbb{R}^2$

$$\left\{ \begin{bmatrix} 2 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \end{bmatrix} \right\}$$

Apply the Gram–Schmidt process again, but with the order of the basis vectors reversed, i.e. apply it to the basis

$$\left\{ x_1 = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, x_2 = \begin{bmatrix} 2 \\ 0 \end{bmatrix} \right\}$$

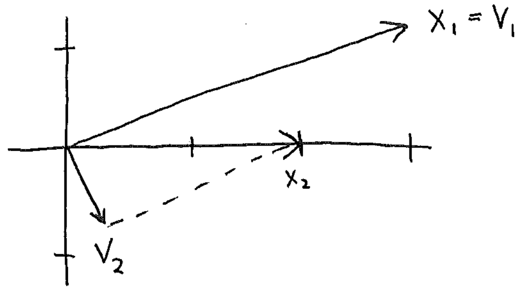
$$V_1 = X_1 = \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

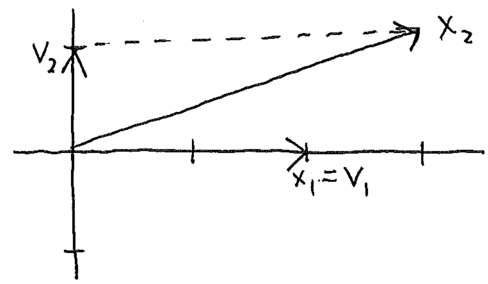
$$V_2 = X_2 - \frac{X_2 \cdot V_1}{V_1 \cdot V_1} V_1 = X_2 - \frac{6}{10} V_1 = \begin{bmatrix} 2 \\ 0 \end{bmatrix} - \frac{3}{5} \begin{bmatrix} 3 \\ 1 \end{bmatrix} = \begin{bmatrix} 1/5 \\ -3/5 \end{bmatrix}$$

$$\left\{ V_1 = \begin{bmatrix} 3 \\ 1 \end{bmatrix}, V_2 = \begin{bmatrix} 1/5 \\ -3/5 \end{bmatrix} \right\}$$

check that it's orthogonal basis: $V_1 \cdot V_2 = \frac{3}{5} - \frac{3}{5} = 0$

Check that your result is indeed an orthogonal basis. On the same graph, plot both the original basis and the basis produced by the Gram–Schmidt process. Compare this to the graph from the example in class.





from class

2. The following is a basis for a two dimensional subspace W of $\mathbb{R}^3$

$$\left\{ \overset{x_1}{\overset{\parallel}{\begin{bmatrix} 0 \\ 4 \\ 2 \end{bmatrix}}}, \overset{x_2}{\overset{\parallel}{\begin{bmatrix} 5 \\ 6 \\ -7 \end{bmatrix}}} \right\}$$

Use the Gram–Schmidt process to produce an orthogonal basis for W.

$$V_1 = \begin{bmatrix} 0 \\ 4 \\ 2 \end{bmatrix}$$

$$V_2 = X_2 - \frac{X_2 \cdot V_1}{V_1 \cdot V_1} V_1 = X_2 - \frac{10}{20} V_1 = \begin{bmatrix} 5 \\ 6 \\ -7 \end{bmatrix} - \frac{1}{2} \begin{bmatrix} 0 \\ 4 \\ 2 \end{bmatrix} = \begin{bmatrix} 5 \\ 4 \\ -8 \end{bmatrix}$$

orthogonal basis for W is $\left\{ V_1 = \begin{bmatrix} 0 \\ 4 \\ 2 \end{bmatrix}, V_2 = \begin{bmatrix} 5 \\ 4 \\ -8 \end{bmatrix} \right\}$

check that it's orthogonal basis : $V_1 \cdot V_2 = 0 + (4)(4) + (2)(-8)$
$= 16 - 16$
$= 0$

Recall that a QR factorization of a matrix A, is a factorization $A = QR$ where Q is an orthogonal matrix (i.e. the columns are an orthonormal set) and R is an upper triangular matrix. We can use Gram–Schmidt process on the columns of an invertible matrix A to produce a QR factorization of A.

3. This problem will use your answer from question 1 to produce a QR factorization of

$$A = \begin{bmatrix} 3 & 2 \\ 1 & 0 \end{bmatrix}$$

First, let $q_i = v_i/||v_i||$ be the vectors obtained in question 1 scaled to have unit length. Let Q be the 2×2 matrix with the q_i as columns: $Q = [q_1\ q_2]$. A simple way to check that a matrix U is orthogonal is to verify that $U^TU = I$ (see Theorem 6 on page 343). Verify that Q is orthogonal.

$$\|v_1\| = (9+1)^{1/2} = \sqrt{10} \qquad \|v_2\| = \left(\frac{1}{25} + \frac{9}{25}\right)^{1/2} = \sqrt{\frac{2}{5}}$$

$$q_1 = \frac{v_1}{\|v_1\|} = \begin{bmatrix} 3/\sqrt{10} \\ 1/\sqrt{10} \end{bmatrix} \qquad q_2 = \frac{v_2}{\|v_2\|} = \frac{\sqrt{5}}{\sqrt{2}} \begin{bmatrix} 1/5 \\ -3/5 \end{bmatrix} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1/\sqrt{5} \\ -3/\sqrt{5} \end{bmatrix} = \begin{bmatrix} 1/\sqrt{10} \\ -3/\sqrt{10} \end{bmatrix}$$

$$Q = [q_1\ q_2] = \begin{bmatrix} 3/\sqrt{10} & 1/\sqrt{10} \\ 1/\sqrt{10} & -3/\sqrt{10} \end{bmatrix} \qquad Q^TQ = \begin{bmatrix} 3/\sqrt{10} & 1/\sqrt{10} \\ 1/\sqrt{10} & -3/\sqrt{10} \end{bmatrix} \begin{bmatrix} 3/\sqrt{10} & 1/\sqrt{10} \\ 1/\sqrt{10} & -3/\sqrt{10} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = I$$

Next, find an upper triangular matrix R, such that $A = QR$.

$$\begin{bmatrix} 3 & 2 \\ 1 & 0 \end{bmatrix} = Q \begin{bmatrix} r_{11} & r_{12} \\ 0 & r_{22} \end{bmatrix}$$

There are a number of different ways to do this, such as solving the two systems of linear equations. But consider using the fact that Q is orthogonal, and it's easy to compute the inverse of an orthogonal matrix.

Because $Q^TQ = I$, we can multiply both sides of $A=QR$ on the left to get

$$Q^TA = Q^TQR = (Q^TQ)R = IR = R$$

So

$$R = Q^TA = \begin{bmatrix} 3/\sqrt{10} & 1/\sqrt{10} \\ 1/\sqrt{10} & -3/\sqrt{10} \end{bmatrix} \begin{bmatrix} 3 & 2 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 10/\sqrt{10} & 6/\sqrt{10} \\ 0 & 2/\sqrt{10} \end{bmatrix} = \begin{bmatrix} \sqrt{10} & 6/\sqrt{10} \\ 0 & 2/\sqrt{10} \end{bmatrix}$$

The QR factorization is ubiquitous in scientific computing because orthogonal and upper triangular matrices have very nice properties, and the Gram–Schmidt process is an algorithm that a computer can use to produce QR factorizations. In fact, MATLAB has a function that does exactly this.

4. The **qr** function in MATLAB takes a matrix as an argument and returns a QR factorization of that matrix. For example

$$[\mathbf{Q}\ \mathbf{R}] = \mathbf{qr}(\mathbf{A})$$

will store an orthogonal matrix in the variable **Q**, and an upper triangular matrix in the variable **R**, such that **A** = **QR**.

(a) Use the **qr** function to get a QR factorization for the matrix A in question 3.

(b) Verify that Q is orthogonal by calculating Q^TQ in MATLAB. To get the transpose of a matrix **M** use the function **transpose(M)**.

(c) Verify that $A = QR$.

(d) Compare the factorization MATLAB gave you to the one you obtained in question 3. Note that QR factorizations are not unique, and they may differ by a negative sign.

One of the primary uses of the QR factorization in scientific computing is to find the eigenvalues of a matrix. To following shows how this is done.

5. Suppose that A is an $n \times n$ matrix. Let $A = Q_0R_0$ be a QR factorization of A and create a new matrix $A_1 = R_0Q_0$ by multiplying the factors in the opposite order. Let $A_1 = Q_1R_1$ be a QR factorization of A_1 and create a new matrix $A_2 = R_1Q_1$ by again multiplying the factors in the opposite order.

(a) Show that $A = Q_0A_1Q_0^T$.

Because Q_0 is orthogonal $A_1Q_0^T = (R_0Q_0)Q_0^T = R_0(Q_0Q_0^T)$
$= R_0I = R_0$

So $A = Q_0R_0 = Q_0(A_1Q_0^T) = Q_0A_1Q_0^T$

(b) Show that $A = (Q_0Q_1)A_2(Q_0Q_1)^T$.

Using the same argument as part (a), $A_2Q_1^T = R_1$,
so $A_1 = Q_1R_1 = Q_1A_2Q_1^T$.

Using fact that $(CD)^T = D^TC^T$, for any matrices C and D, and part (a)

$A = Q_0A_1Q_0^T = Q_0(Q_1A_2Q_1^T)Q_0^T = (Q_0Q_1)A_2(Q_1^TQ_0^T)$
$= (Q_0Q_1)A_2(Q_0Q_1)^T$

(c) Show that Q_0Q_1 is an orthogonal matrix.

A matrix U is orthogonal if $U^TU = I$.

$$(Q_0Q_1)^T(Q_0Q_1) = (Q_1^TQ_0^T)(Q_0Q_1) = Q_1^T\underbrace{(Q_0^TQ_0)}_{I}Q_1$$

$$= Q_1^TQ_1 = I$$

(d) Show that A, A_1, A_2 all have the same eigenvalues. (HINT: Suppose λ is an eigenvalue of A. Write down what that means in terms of determinants. Then use the properties of determinants and the fact that $Q^TQ = QQ^T = I$ for orthogonal matrices.)

Let λ be an eigenvalue of A. Then

$$0 = \det(A - \lambda I) = \det(Q_0A_1Q_0^T - \lambda(Q_0Q_0^T)) = \det(Q_0(A_1 - \lambda I)Q_0^T)$$

$$= \det(Q_0)\det(A_1 - \lambda I)\det(Q_0^T)$$

because $\det(Q_0^T) = \det(Q_0^{-1}) = \frac{1}{\det(Q_0)}$

$$= \det(Q_0)\det(A_1 - \lambda I)\frac{1}{\det(Q_0)}$$

$$= \det(A_1 - \lambda I)$$

The same argument shows that A_1 and A_2 have the same eigenvalues.

The QR method for finding eigenvalues of an $n \times n$ matrix A is the following:

Step 1: Let $A = Q_0R_0$ be a QR factorization of A; create $A_1 = R_0Q_0$.

Step 2: Let $A_1 = Q_1R_1$ be a QR factorization of A_1; create $A_2 = R_1Q_1$.

Step 3: Continue this process. Once A_m has been created, let $A_m = Q_mR_m$ be a QR factorization of A_m and create $A_{m+1} = R_mQ_m$.

Step 4: Stop the process when the entries below the main diagonal of A_m are sufficiently small, or stop if it appears that convergence will not happen. As you showed in question 5, the eigenvalues of A_m are the same as A. Since A_m is getting closer to an upper triangular matrix, you can read off the approximate eigenvalues from the diagonal of A_m.

A version of the preceding algorithm was the standard method for computing all the eigenvalues of a matrix since its development in the 1960s until a competing algorithm emerged in the 1990s.

6. To see the process described above on MATLAB, enter the matrix A and then type the following line:

[Q R] = qr(A); A=R*Q

Then use the arrow up key to repeat the command. Using MATLAB, do enough steps of the QR method to make each entry below the main diagonal of the matrix smaller than 0.1. Record estimates for the for the eigenvalues. Find the actual eigenvalues by using the **eig** MATLAB function and compare to the estimates.

(a)

$$B = \begin{bmatrix} 1 & -2 & 8 \\ 7 & -7 & 6 \\ 5 & 7 & -8 \end{bmatrix}$$

(b)

$$C = \begin{bmatrix} 4 & -2 & 3 & -7 \\ 1 & 2 & 6 & 8 \\ 8 & 5 & 1 & -5 \\ -5 & 8 & -5 & 3 \end{bmatrix}$$

(c)

$$D = \begin{bmatrix} 2 & 6 & -3 & 4 & -9 \\ -1 & 7 & -4 & -3 & -7 \\ -6 & -6 & -1 & 6 & 5 \\ 9 & 2 & 6 & 2 & -8 \\ -7 & -8 & 6 & -9 & -1 \end{bmatrix}$$