Classical Gram-Schmidt Orthogonalization

Example 1. Let $Q = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{bmatrix}$.

1. Are the columns of Q orthogonal? linearly independent?

   Since Q has two columns, we only have to compute one dot product to check orthogonality: $\mathbf{q}_1 \cdot \mathbf{q}_2 = (\frac{\sqrt{2}}{2})^2 - (\frac{\sqrt{2}}{2})^2 = 0$, so yes, they are orthogonal. Thus the columns form an orthogonal set and an orthogonal set is always linearly independent.

2. Compute the 2-norm of each column of Q.

   $||\mathbf{q}_1||_2 = \sqrt{(\frac{\sqrt{2}}{2})^2 + (\frac{\sqrt{2}}{2})^2} = 1$. The 2-norm of $\mathbf{q}_2$ is the same. Vectors with a 2-norm of one are called *unit vectors.*

3. Compute Q^TQ.

   Since $Q^T = Q$, we have:

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

   So $Q^T = Q^{-1}$.

Q is an example of an *orthogonal matrix.*

Definition 1. An orthogonal matrix is a square matrix whose columns are unit vectors and form an orthogonal set.

Theorem 2. *If Q is orthogonal, then $Q^{-1} = Q^T$.*

Thus, orthogonal matrices are super nice because they are easily inverted.

Theorem 3. *If Q is an orthogonal $(n \times n)$ matrix, then for all vectors $\boldsymbol{x} \in \mathbb{R}^n$, $||Q\boldsymbol{x}||_2 = ||\boldsymbol{x}||_2$.*

Proof. $||Q\mathbf{x}||_2^2 = (Q\mathbf{x})^T(Q\mathbf{x}) = \mathbf{x}^TQ^TQ\mathbf{x} = \mathbf{x}^T\mathbf{x} = ||\mathbf{x}||_2^2$, and if $||Q\mathbf{x}||_2^2 = ||\mathbf{x}||_2^2$, then $||Q\mathbf{x}||_2 = ||\mathbf{x}||_2$ by the nonnegativity of the norm. □

If you're thinking about Q as a transformation of a vector $\mathbf{x}$, Q is nice because it only rotates the vector; it does not stretch or compress it. Numerically, Q will not magnify errors.

A matrix Q with fewer columns than rows can have orthogonal unit vectors for columns. Sometimes such a Q is also called orthogonal, sometimes it is called orthonormal, and sometimes neither - no standard terminology. The only thing that's different is that $Q^TQ = I_n$ while $QQ^T = I_m$, so they result in identity matrices of different sizes.

The QR Factorization

Every $m \times n$ matrix A that has linearly independent columns has a QR factorization. There are two versions of this factorization: "reduced" and "full." In the reduced version, Q is also $m \times n$ and R is $m \times m$ upper triangular. In the full version, Q is extended to be an orthogonal $m \times m$ matrix and R is $m \times n$ upper triangular (we just add zeros below the reduced version). For solving the least squares problem, reduced QR is sufficient. When A is square, there is no difference between reduced and full QR.

Suppose $A = QR$. Then in least squares, we're trying to minimize $||\mathbf{b} - A\mathbf{x}||_2$, which equals $||\mathbf{b} - QR\mathbf{x}||_2$. Then by Theorem 3, we are minimizing $||Q^T(\mathbf{b} - QR\mathbf{x})||_2 = ||Q^T\mathbf{b} - Q^TQR\mathbf{x})||_2 = ||Q^T\mathbf{b} - R\mathbf{x})||_2$. So we can solve the equation

$$R\mathbf{x} = Q^T\mathbf{b}$$

for a least squares solution using back substitution.

Note that this is just another matrix factorization, like LU and Cholesky. So it can be used to solve $A\mathbf{x} = \mathbf{b}$ when the system is consistent, too, though the method of computing QR that we're about to cover is about three times as much work as solving using LU. So in practice it's only used for least squares problems, and later, eigenvalue problems.

Classical Gram-Schmidt Orthogonalization

Much like the LU factorization is a convenient way of storing the steps in Gaussian Elimination, the QR factorization stores the steps of the algorithm called Classical Gram-Schmidt orthogonalization (CGS). CGS is a method for orthogonalizing a set of linearly independent vectors.

Let $A_1, \ldots, A_n$ be linearly independent vectors in $\mathbb{R}^m$ (so necessarily $n \leq m$). Typically these are the columns of A, the matrix to be factored. Then CGS is the following process.

for $j = 1, \ldots, n$ **do** ▷ For each column
  $\mathbf{y} = A_j$ ▷ Let y be the new column to be orthogonalized
  for $i = 1, \ldots, j-1$ **do** ▷ For each row above the current diagonal element
    $r_{ij} = \mathbf{q}_i^T A_j$ ▷ Compute the above-diagonal elements in R
    $\mathbf{y} = \mathbf{y} - r_{ij}\mathbf{q}_i$ ▷ Update the column, now orthogonal to the preceding columns
  $r_{jj} = ||\mathbf{y}||_2$ ▷ Compute the diagonal element in R
  $\mathbf{q}_j = \frac{1}{r_{jj}}\mathbf{y}$ ▷ Scale the result to be a unit vector

To summarize, we proceed through each column. The inner loop that computes the above diagonal elements of R and updates y is removing the parts of the current column that aren't orthogonal to the previous columns. The last two steps turn the remaining vector into a unit vector.

Just to be clear, the direction of the columns of A are changing in this process. What's being preserved is the span of the columns, that is, col(A). In reduced QR, col$(Q) =$ col(A). In full QR, col$(R) =$ col(A).

Example 2. Find the QR factorization of $A = \begin{bmatrix} 4 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$.

On the first step, there are no above diagonal elements to compute so we go to the step where we compute the first element of R. $r_{11} = ||\mathbf{y}||_2 = || \begin{bmatrix} 4 \\ 0 \\ 0 \end{bmatrix} ||_2 = 4$. Then the first column of Q is $\mathbf{q}_1 = \frac{1}{4}\mathbf{y} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$.

So thus far we have:

$$Q = \begin{bmatrix} 1 & & \\ 0 & & \\ 0 & & \end{bmatrix} \qquad R = \begin{bmatrix} 4 & & \\ 0 & & \\ 0 & & \end{bmatrix}.$$

Next step, set $\mathbf{y} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$. Then compute $r_{12} = \mathbf{q}_1^T A_2 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} = 1$. Which means we have

$$Q = \begin{bmatrix} 1 & & \\ 0 & & \\ 0 & & \end{bmatrix} \qquad R = \begin{bmatrix} 4 & 1 & \\ 0 & & \\ 0 & & \end{bmatrix}.$$

Then update the vector $\mathbf{y}$ as $\mathbf{y} = \mathbf{y} - r_{12}\mathbf{q}_1 = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix} - (1) \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$. Also compute $r_{22} = ||\mathbf{y}||_2 = 1$, so that $\mathbf{q}_2 = 1\mathbf{y} = \mathbf{y}$, and we have:

$$Q = \begin{bmatrix} 1 & 0 & \\ 0 & 1 & \\ 0 & 0 & \end{bmatrix} \qquad R = \begin{bmatrix} 4 & 1 & \\ 0 & 1 & \\ 0 & 0 & \end{bmatrix}$$

Now the last column. Set $\mathbf{y} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$. We need r_{13} and r_{23}. First, $r_{13} = \mathbf{q}_1^T A_3 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = 1$. Update $\mathbf{y}$ as $\mathbf{y} = \mathbf{y} - (1)\mathbf{q}_1 = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix}$. Then $r_{23} = \mathbf{q}_2^T A_3 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = 1$, so $\mathbf{y} = \mathbf{y} - (1)\mathbf{q}_2 = \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$.

Finishing, $r_{33} = ||\mathbf{y}||_2 = 1$ so $\mathbf{q} = \frac{1}{1}\mathbf{y} = \mathbf{y}$, giving

$$Q = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad R = \begin{bmatrix} 4 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

Starting with an upper triangular matrix is a little bit of a silly example, as it's already R. But this gives us a good demonstration of the algorithm without the basically inevitable ugly fractions. Note that the columns of Q are the standard basis of $\mathbb{R}^3$.

Example 3. Find the reduced QR factorization of $A = \begin{bmatrix} 1 & 1 \\ 1 & 1 \\ 0 & 2 \end{bmatrix}$. Then use the factorization to solve the least squares problem for $\mathbf{b} = \begin{bmatrix} 2 \\ 4 \\ 4 \end{bmatrix}$.

We start with $\mathbf{y} = \begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$. Since $||\mathbf{y}||_2 = \sqrt{2}$, we have:

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

Then setting $\mathbf{y} = \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix}$, we compute $r_{12} = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 2 \end{bmatrix} = \sqrt{2}$. Then update $\mathbf{y}$ as $\mathbf{y} = \mathbf{y} - \sqrt{2} \begin{bmatrix} \frac{\sqrt{2}}{2} \\ \frac{\sqrt{2}}{2} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$. So we have:

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

Make $\mathbf{y}$ a unit vector, so that $r_{22} = ||\mathbf{y}||_2 = 2$ and $\frac{1}{||\mathbf{y}||_2}\mathbf{y} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$. Final QR is :

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

Then to solve the least squares problem, we first compute

$$Q^T\mathbf{b} = \begin{bmatrix} \frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 2 \\ 4 \\ 4 \end{bmatrix} = \begin{bmatrix} 3\sqrt{2} \\ 4 \end{bmatrix}.$$

Then solve $R\mathbf{x} = Q^T\mathbf{b}$ by back substitution.

$$\begin{bmatrix} \sqrt{2} & \sqrt{2} & 3\sqrt{2} \\ 0 & 2 & 4 \end{bmatrix} \sim \begin{bmatrix} 1 & 1 & 3 \\ 0 & 1 & 2 \end{bmatrix} \sim \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

So $\hat{\mathbf{x}} = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$.

Example 4. We previously looked at the Van der Monde matrix for a degree 6 polynomial, and found that using the normal equations we lost almost all of the accuracy. Using QR instead, the solution (which should be all 1's) comes out to

$$\begin{bmatrix} 0.999999997906637 \\ 1.000000004074936 \\ 0.999999996735643 \\ 1.000000001378255 \\ 0.999999999676420 \\ 1.000000000040063 \\ 0.999999999997955 \end{bmatrix}$$

It appears we have about 8 digits of accuracy, which while not perfect is certainly better than none!