Math 108B	Professor: Padraic Bartlett

Lecture 4: Applications of Orthogonality: QR Decompositions

Week 4 *UCSB 2014*

In our last class, we described the following method for creating orthonormal bases, known as the **Gram-Schmidt** method:

Theorem. Suppose that V is a k-dimensional space with a basis $B = \{\vec{b_1}, \ldots \vec{b_k}\}$.

The following process (called the Gram-Schmidt process) creates an orthonormal basis for V:

1. First, create the following vectors $\{\vec{v_1}, \ldots \vec{v_k}\}$:
   - $\vec{u_1} = \vec{b_1}$.
   - $\vec{u_2} = \vec{b_2} - \text{proj}(\vec{b_2} \text{ onto } \vec{u_1})$.
   - $\vec{u_3} = \vec{b_3} - \text{proj}(\vec{b_3} \text{ onto } \vec{u_1}) - \text{proj}(\vec{b_3} \text{ onto } \vec{u_2})$.
   - $\vec{u_4} = \vec{b_4} - \text{proj}(\vec{b_4} \text{ onto } \vec{u_1}) - \text{proj}(\vec{b_4} \text{ onto } \vec{u_2}) - \text{proj}(\vec{b_4} \text{ onto } \vec{u_3})$.

   $\vdots$

   - $\vec{u_k} = \vec{b_k} - \text{proj}(\vec{b_k} \text{ onto } \vec{u_1}) - \ldots - \text{proj}(\vec{b_k} \text{ onto } \vec{u_{k-1}})$.
2. Now, take each of the vectors $\vec{u_i}$, and rescale them so that they are unit length: i.e. redefine each $\vec{u_i}$ as the rescaled vector $\frac{\vec{u_i}}{||\vec{u_i}||}$.

In this class, I want to talk about a useful application of the Gram-Schmidt method: the QR decomposition! We define this here:

1 The QR decomposition

Definition. We say that an $n \times n$ matrix Q is **orthogonal** if its columns form an orthonormal basis for $\mathbb{R}^n$.

As a side note: we've studied these matrices before! In class on Friday, we proved that for any such matrix, the relation

$$Q^T \cdot Q = I$$

held; to see this, we just looked at the (i, j)-th entry of the product $Q^T \cdot Q$, which by definition was the i-th row of Q^T dotted with the j-th column of A. The fact that Q's columns formed an orthonormal basis was enough to tell us that this product must be the identity matrix, as claimed.

Definition. A **QR-decomposition** of an $n \times n$ matrix A is an orthogonal matrix Q and an upper-triangular[1] matrix R, such that

$$A = QR.$$

Theorem. Every invertible matrix has a QR-decomposition, where R is invertible.

Proof. We prove this using the Gram-Schmidt process! Specifically, consider the following process: take the columns $\vec{a_{c_1}}, \ldots \vec{a_{c_n}}$ of A. Because A is invertible, its columns are linearly independent, and thus form a basis for $\mathbb{R}^n$. Therefore, running the Gram-Schmidt process on them will create an orthonormal basis for $\mathbb{R}^n$! Do this here: i.e. set

- $\vec{u_1} = \vec{a_{c_1}}$.
- $\vec{u_2} = \vec{a_{c_2}} - \text{proj}(\vec{a_{c_2}} \text{ onto } \vec{u_1})$.
- $\vec{u_3} = \vec{a_{c_3}} - \text{proj}(\vec{a_{c_3}} \text{ onto } \vec{u_1}) - \text{proj}(\vec{a_{c_3}} \text{ onto } \vec{u_2})$.
- $\vec{u_4} = \vec{a_{c_4}} - \text{proj}(\vec{a_{c_4}} \text{ onto } \vec{u_1}) - \text{proj}(\vec{a_{c_4}} \text{ onto } \vec{u_2}) - \text{proj}(\vec{a_{c_4}} \text{ onto } \vec{u_3})$.

$\vdots$

- $\vec{u_n} = \vec{a_{c_n}} - \text{proj}(\vec{a_{c_n}} \text{ onto } \vec{u_1}) - \ldots - \text{proj}(\vec{a_{c_n}} \text{ onto } \vec{u_{n-1}})$.

Skip the rescaling step for a second. If we take these equations and solve them for the columns $\vec{a_{c_i}}$ of A, we get

- $\vec{a_{c_1}} = \vec{u_1}$.
- $\vec{a_{c_2}} = \vec{u_2} + \text{proj}(\vec{a_{c_2}} \text{ onto } \vec{u_1})$.
- $\vec{a_{c_3}} = \vec{u_3} + \text{proj}(\vec{a_{c_3}} \text{ onto } \vec{u_1}) + \text{proj}(\vec{a_{c_3}} \text{ onto } \vec{u_2})$.
- $\vec{a_{c_4}} = \vec{u_4} + \text{proj}(\vec{a_{c_4}} \text{ onto } \vec{u_1}) + \text{proj}(\vec{a_{c_4}} \text{ onto } \vec{u_2}) + \text{proj}(\vec{a_{c_4}} \text{ onto } \vec{u_3})$.

$\vdots$

- $\vec{a_{c_n}} = \vec{u_n} + \text{proj}(\vec{a_{c_n}} \text{ onto } \vec{u_1}) + \ldots + \text{proj}(\vec{a_{c_n}} \text{ onto } \vec{u_{n-1}})$.

Now, notice that all of the $\text{proj}(\vec{a_{c_i}} \text{ onto } \vec{u_j})$-terms are actually, by defintion, just multiples of the vector $\vec{u_j}$. To make this more obvious, we could replace each of these terms with what they are precisely defined to be, i.e. $\frac{\vec{a_{c_i}} \cdot \vec{u_j}}{||\vec{u_1}||^2}\vec{u_j}$. However, that takes up a lot of space! Instead, for shorthand's sake, denote the constant $\frac{\vec{a_{c_i}} \cdot \vec{u_j}}{||\vec{u_1}||^2}$ as $p_{c_i,j}$. Then, we have the following:

- $\vec{a_{c_1}} = \vec{u_1}$.

[1] A matrix is called **upper-triangular** if all of its entries below the main diagonal are 0. For example, $\begin{bmatrix} 1 & 2 & 3 \\ 0 & 3 & 2 \\ 0 & 0 & 1 \end{bmatrix}$ is upper-triangular.

- $\vec{a_{c_2}} = \vec{u_2} + p_{c_2,1}\vec{u_1}$.
- $\vec{a_{c_3}} = \vec{u_3} + p_{c_3,1}\vec{u_1} + p_{c_3,2}\vec{u_2}$.
- $\vec{a_{c_4}} = \vec{u_4} + p_{c_4,1}\vec{u_1} + p_{c_4,2}\vec{u_2} + p_{c_4,3}\vec{u_3}$.

$\vdots$

- $\vec{a_{c_n}} = \vec{u_n} + p_{c_2,1}\vec{u_1} + \ldots + p_{c_2,n-1}\vec{u_{n-1}}$.

If we do this, then it is not too hard to see that we actually have the following identity:

$$\begin{bmatrix} \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ \vec{u_1} & \vec{u_2} & \vec{u_3} & \ldots & \vec{u_n} \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \end{bmatrix} \cdot \begin{bmatrix} 1 & p_{c_2,1} & p_{c_3,1} & \ldots & p_{c_n,1} \\ 0 & 1 & p_{c_3,2} & \ldots & p_{c_n,2} \\ 0 & 0 & 1 & \ldots & p_{c_n,3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & 1 \end{bmatrix}$$

$$= \begin{bmatrix} \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ \vec{u_1} & (\vec{u_2} + p_{c_2,1}\vec{u_1}) & (\vec{u_3} + p_{c_3,2}\vec{u_2} + p_{c_3,1}\vec{u_1}) & \ldots & \left(\vec{u_n} + \sum_{i=1}^{n-1} p_{c_n,i}\vec{u_i}\right) \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \end{bmatrix} = A.$$

In other words, we have a QR-decomposition!

Well, almost. The left-hand matrix's rows form an orthogonal basis for $\mathbb{R}^n$, but they are not yet all length 1. To fix this, simply scale the left matrix's columns so that they're all length 1, and then increase the scaling on the right-hand matrix's rows so that it cancels out: i.e.

$$\begin{bmatrix} \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ \frac{\vec{u_1}}{||\vec{u_1}||} & \frac{\vec{u_2}}{||\vec{u_2}||} & \frac{\vec{u_3}}{||\vec{u_3}||} & \ldots & \frac{\vec{u_n}}{||\vec{u_n}||} \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \end{bmatrix} \cdot \begin{bmatrix} ||\vec{u_1}|| & p_{c_2,1} \cdot ||\vec{u_1}|| & p_{c_3,1} \cdot ||\vec{u_1}|| & \ldots & p_{c_n,1} \cdot ||\vec{u_1}|| \\ 0 & ||\vec{u_2}|| & p_{c_3,2} \cdot ||\vec{u_2}|| & \ldots & p_{c_n,2} \cdot ||\vec{u_2}|| \\ 0 & 0 & ||\vec{u_3}|| & \ldots & p_{c_n,3} \cdot ||\vec{u_3}|| \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \ldots & ||\vec{u_n}|| \end{bmatrix}$$

$$= \begin{bmatrix} \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \\ \vec{u_1} & (\vec{u_2} + p_{c_2,1}\vec{u_1}) & (\vec{u_3} + p_{c_3,2}\vec{u_2} + p_{c_3,1}\vec{u_1}) & \ldots & \left(\vec{u_n} + \sum_{i=1}^{n-1} p_{c_n,i}\vec{u_i}\right) \\ \vdots & \vdots & \vdots & & \vdots \\ \vdots & \vdots & \vdots & & \vdots \end{bmatrix} = A.$$

A QR-decomposition! $\square$

As a side note, it bears mentioning that this result holds even if the matrix is not invertible:

Theorem. Every matrix has a QR-decomposition, though R may not always be invertible.

The proof is pretty much exactly the same as above, except you have to be careful when dealing with the $||\vec{u_i}||$'s, as you might be dividing by zero in the situations where A's columns were linearly dependent. On your own, try to think about what you'd need to change in the above proof to make it work for a general matrix!

Instead, I want to focus on why this decomposition is nice: solving systems of linear equations!

2 Applications of QR decompositions: Solving Systems of Linear Equations

Suppose you have an invertible matrix A and vector $\vec{b}$. Consider the task of a vector $\vec{v}$ such that

$$A\vec{v} = \vec{b};$$

in other words, solving the system of n linear equations $a_{r_i} \cdot \vec{v} = b_i, i = 1 \ldots n$. This is typically a doable if slightly tedious task, via Gaussian elimination (i.e. pivoting on entries in A.) However it takes time, and (from the perspective of implementing on a computer) can be fairly sensitive to small changes or errors: i.e. if you're pivoting on an entry in A that is nearly zero, it is easy for small rounding errors in a computer program to suddenly cause very big changes in what the entries in your matrix should be!

However, suppose that we have a QR decomposition for A: i.e. we can write $A = QR$, for some upper-triangular R and orthogonal Q. Then, solving

$$QR\vec{v} = \vec{b}$$

is the same task as solving

$$Q^TQR\vec{v} = R\vec{v} = Q^T\vec{b};$$

and this is suddenly much easier!

In particular, because R is upper-triangular, $R\vec{v}$ is just

$$\begin{bmatrix} r_{1,1}v_1 + r_{1,2}v_2 + r_{1,3}v_3 + \ldots + r_{1,n}v_n \\ r_{2,2}v_2 + r_{2,3}v_3 + \ldots + r_{2,n}v_n \\ \vdots \\ r_{n-1,n-1}v_{n-1} + r_{n-1,n}v_n \\ r_{n,n}v_n \end{bmatrix}.$$

And finding values of this to set equal to some fixed vector $Q^T\vec{b}$ is **really easy**! In particular, the last coordinate of $R\vec{v}$ just has one variable v_n, so it's easy to solve for that variable. From here, the second coordinate of $R\vec{v}$ has just two variables, v_{n-1}, v_n, one of which we know now! So it's equally easy to solve for v_{n-1}. Working our way up, we have the same situation for each variable: we never have to do any "work" to solve for the variables v_i!

To illustrate this, we work an example:

Example. Consider the matrix

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

First, find its QR decomposition. Then, use that QR decomposition to find a vector A such that

$$A \cdot \vec{v} = \begin{bmatrix} 1 \\ 2 \\ 0 \\ 1 \end{bmatrix}.$$

Answer. We start by performing Gram-Schmidt on the columns of A:

$$\begin{aligned}
\vec{u_1} &= \vec{a_{c_1}} = (2,2,2,2). \\
\vec{u_2} &= \vec{a_{c_2}} - \text{proj}(\vec{a_{c_2}} \text{ onto } \vec{u_1}) \\
&= (1,1,-1,-1) - \frac{(1,1,-1,-1)\cdot(2,2,2,2)}{(2,2,2,2)\cdot(2,2,2,2)}(2,2,2,2) \\
&= (1,1,-1,-1) - 0 \\
&= (1,1,-1,-1). \\
\vec{u_3} &= \vec{a_{c_3}} - \text{proj}(\vec{a_{c_3}} \text{ onto } \vec{u_1}) - \text{proj}(\vec{a_{c_3}} \text{ onto } \vec{u_2}) \\
&= (3,-1,3,-1) - \frac{(3,-1,3,-1)\cdot(2,2,2,2)}{(2,2,2,2)\cdot(2,2,2,2)}(2,2,2,2) - \frac{(3,-1,3,-1)\cdot(1,1,-1,-1)}{(1,1,-1,-1)\cdot(1,1,-1,-1)}(1,1,-1,-1) \\
&= (3,-1,3,-1) - \frac{8}{16}(2,2,2,2) - 0 \\
&= (2,-2,2,-2). \\
\vec{u_4} &= \vec{a_{c_4}} - \text{proj}(\vec{a_{c_4}} \text{ onto } \vec{u_1}) - \text{proj}(\vec{a_{c_4}} \text{ onto } \vec{u_2}) - \text{proj}(\vec{a_{c_4}} \text{ onto } \vec{u_3}) \\
&= (3,1,-3,-1) - \frac{(3,1,-3,-1)\cdot(2,2,2,2)}{(2,2,2,2)\cdot(2,2,2,2)}(2,2,2,2) - \frac{(3,1,-3,-1)\cdot(1,1,-1,-1)}{(1,1,-1,-1)\cdot(1,1,-1,-1)}(1,1,-1,-1) \\
&\qquad - \frac{(3,1,-3,-1)\cdot(2,-2,2,-2)}{(2,-2,2,-2)\cdot(2,-2,2,-2)}(2,-2,2,-2) \\
&= (3,1,-3,-1) - 0 - \frac{8}{4}(1,1,-1,-1) - 0 \\
&= (1,-1,-1,1).
\end{aligned}$$

Using these four vectors, we construct a QR-decomposition as described earlier. Notice that we've already calculated the $\frac{\vec{a_{c_i}} \cdot \vec{u_j}}{||\vec{u_1}||^2}\vec{u_j} = p_{c_i,j}$'s above, and so we don't need to repeat our work here:

$$\begin{bmatrix} \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\vec{u_1}}{||\vec{u_1}||} & \frac{\vec{u_2}}{||\vec{u_1}||} & \frac{\vec{u_3}}{||\vec{u_1}||} & \frac{\vec{u_4}}{||\vec{u_4}||} \\ \vdots & \vdots & \vdots & \vdots \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix} \cdot \begin{bmatrix} ||\vec{u_1}|| & p_{c_2,1} \cdot ||\vec{u_1}|| & p_{c_3,1} \cdot ||\vec{u_1}|| & p_{c_4,1} \cdot ||\vec{u_1}|| \\ 0 & ||\vec{u_2}|| & p_{c_3,2} \cdot ||\vec{u_2}|| & p_{c_4,2} \cdot ||\vec{u_2}|| \\ 0 & 0 & ||\vec{u_3}|| & p_{c_4,3} \cdot ||\vec{u_3}|| \\ 0 & 0 & 0 & ||\vec{u_4}|| \end{bmatrix}$$

$$= \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 4 & 0 & 2 & 0 \\ 0 & 2 & 0 & 4 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix}.$$

From here, to solve the equation $A\vec{v} = (1,2,0,1)$, we can use our QR-decomposition to write $QR\vec{v} = (1,2,0,1)$, or equivalently $R\vec{v} = Q^T(1,2,0,1)$. We find the right-hand-side here:

$$\frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}^T \cdot \begin{bmatrix} 1 \\ 2 \\ 0 \\ 1 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 2 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ -1 \\ 0 \end{bmatrix}.$$

So: we have the simple problem of finding $\vec{v} = (v_1, v_2, v_3, v_4)$ such that

$$\begin{bmatrix} 4 & 0 & 2 & 0 \\ 0 & 2 & 0 & 4 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 2 \end{bmatrix} \cdot \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \\ -1 \\ 0 \end{bmatrix}.$$

Ordering from bottom to top, this problem is just the task of solving the four linear equations

$$\begin{aligned} 2v_4 &= 0 \\ 4v_3 &= -1 \\ 2v_2 + 4v_4 &= 1 \\ 4v_1 + 2v_3 &= 2. \end{aligned}$$

If we solve from the top down, this is trivial: we get $v_4 = 0, v_3 = -1/4, v_2 = 1/2$, and $v_1 = \frac{5}{8}$. Thus, we've found a solution to our system of linear equations, and we're done!