Lecture Notes for Math 104: Fall 2010 (Week 6)

Jacob Bernstein

CHAPTER 1

Sixteenth and Eighteenth Lectures

We discuss alternate methods of computing the QR factorization. These are better suited for implementation on a computer.

1. Modified Gram-Schmidt

We have seen how to compute the QR factorizations using the Gram-Schmidt algorithm and this is perfectly fine from a theoretical point of view. Practically however, the algorithm handles rounding errors very poorly (mainly an issue when the initial basis contains vectors that are nearly parallel). To see how to get around this we first give a variant of Gram-Schmidt that is better behaved.

Lets think for a moment about what Gram-Schmidt itself does. Let $\{\mathbf{a}_1, \ldots, \mathbf{a}_n\}$ be a of linearly independent vectors in $\mathbb{C}^m$. The Gram-Schmidt algorithm produces,

$$\mathbf{q}_1 = \frac{P_1\mathbf{a}_1}{||P_1\mathbf{a}_1||_2}, \mathbf{q}_2 = \frac{P_2\mathbf{a}_2}{||P_2\mathbf{a}_2||_2}, \ldots, \mathbf{q}_k = \frac{P_k\mathbf{a}_k}{||P_k\mathbf{a}_k||_2}$$

where here P_1 is the identity matrix and for $j \geq 2$ each $P_j \in \mathbb{C}^{m\times m}$ is orthogonal projection onto $span(\mathbf{q}_1, \ldots, \mathbf{q}_{j-1})^{\perp}$. As the $\mathbf{q}_i$ are an orthonormal set of vectors we can write:

$$\hat{Q}_{j-1} = \begin{bmatrix}\mathbf{q}_1| & \cdots & |\mathbf{q}_k\end{bmatrix}$$

and then $\hat{Q}_{j-1}\hat{Q}_{j-1}^*$ gives projection onto $span(\mathbf{q}_1, \ldots, \mathbf{q}_{j-1})$. Thus P_j projection onto the orthogonal complement is given by

$$P_j = I - \hat{Q}_{j-1}\hat{Q}_{j-1}^*$$

Notice that each P_j is of rank $m-(j-1)$. An important observation is that one can factorize the P_j in terms of rank $m-1$ orthogonal projectors:

$$P_j = P_{\perp\mathbf{q}_{j-1}} \cdots P_{\perp\mathbf{q}_2} P_{\perp\mathbf{q}_1} I$$

This follows by noting that $P_j = I - \sum_{i=1}^{j} \mathbf{q}_i\mathbf{q}_i^*$ and $P_{\perp\mathbf{q}_i} = I - \mathbf{q}_i\mathbf{q}_i^*$ and taking a product. I leave the details as an exercise. This gives another algorithm which is less sensitive to rounding errors.

Idea of the algorithm: Start with $\mathbf{a}_j$ set $\mathbf{v}_j^{(1)} = \mathbf{a}_j$. One iteratively computes as follows: At the ith step, set $\mathbf{v}_j = \mathbf{v}_j^{(j)}$, $r_{ii} = ||\mathbf{v}_i||_2$ and $\mathbf{q}_i = r_{ii}^{-1}\mathbf{v}_i$ and set $\mathbf{v}_j^{(i+1)} = P_{\perp\mathbf{q}_i}\mathbf{v}_j^{(i)}$.

Another way to think about this is that there are upper triangular matrices $R_1, R_2, \ldots, R_n$ (in $\mathbb{C}^{n\times n}$) so that

$$\begin{bmatrix}\mathbf{q}_1 & \cdots & |\mathbf{q}_{j-1} & |\mathbf{v}_j^{(j)}| & \cdots & |\mathbf{v}_n^{(j)}\end{bmatrix} R_j = \begin{bmatrix}\mathbf{q}_1 & \cdots & |\mathbf{q}_j & |\mathbf{v}_{j+1}^{(j+1)}| & \cdots & |\mathbf{v}_n^{(j+1)}\end{bmatrix}$$

We refer to Trefethen and Bau Lecture 8 for the exact form of the R_j. Then one has

$$AR_1R_2\dots R_n = \hat{Q}$$

One can check that the product of upper triangular matrices is still upper triangular and one of your homework exercises was to show the inverse was upper triangular so with

$$\hat{R} = (R_1R_2\dots R_n)^{-1}$$

one obtains a reduced QR factorization:

$$A = \hat{Q}\hat{R}$$

2. Householder Reflections

So we saw how to determine a (reduced) QR factorization by repeated multiplications by upper triangular matrices that in the end produces a unitary matrix. Another approach is to use unitary matrices to produce an upper triangular matrix. This is known as the Householder algorithm.

Basic idea is to write

$$A = \begin{bmatrix}\mathbf{a}_1| & \cdots & |\mathbf{a}_n\end{bmatrix}$$

we want to find a $Q_1 \in \mathbb{C}^{m\times m}$ that is unitary and so that

$$Q_1A = \begin{bmatrix} r_{11}\mathbf{e}_1 & |\mathbf{a}_2^{(2)}| & \cdots & |\mathbf{a}_n^{(2)}\end{bmatrix}$$

Then find a $Q_2 \in \mathbb{C}^{m\times m}$ that is unitary and so that

$$Q_2Q_1A = \begin{bmatrix} r_{11}\mathbf{e}_1 & |r_{12}\mathbf{e}_1 + r_{22}\mathbf{e}_2 & |\mathbf{a}_3^{(3)}| & \cdots & |\mathbf{a}_n^{(3)}\end{bmatrix}$$

and so on. The end result will be $Q_1, \dots, Q_n$ all unitary so that

$$Q_nQ_{n-1}\dots Q_1A = R$$

where $R \in \mathbb{C}^{m\times n}$ is upper triangular. We get the QR factorization by setting

$$Q = (Q_nQ_{n-1}\dots Q_1)^*$$

where we use that the product of unitary matrices is still unitary as is the adjoint of a unitary matrix. We leave this fact to you to check.

We now discuss how to find such unitary matrices. The first property that we want is for the Q_k to preserve the first $k-1$ columns of $Q_{k-1}\cdots Q_1A$. To do so we may take Q_k to be of the form

$$Q_k = \begin{bmatrix} I_{k-1} & 0 \\ 0 & F \end{bmatrix}$$

where $I_{k-1} \in \mathbb{C}^{(k-1)\times(k-1)}$ is the identity and $F \in \mathbb{C}^{(m-k+1)\times(m-k+1)}$. This works as the first $k-1$ columns of $Q_{k-1}\cdots Q_1A$ are upper triangular so the F term doesn't effect those columns. For Q_k to be unitary it must have orthonormal columns and hence F must have orthonormal columns and so also be unitary.

Let $\mathbf{x} \in \mathbb{C}^(m-k+1)$ denote the vector obtained from $\mathbf{a}_k^{(k)} \in \mathbb{C}^m$ by omitting the first $k-1$ entries. One has

$$Q_k\mathbf{a}_k^{(k)} = \begin{bmatrix} * \\ F\mathbf{x} \end{bmatrix}$$

where $*$ represents $m-k-1$ entries. In particular, to to find F (and hence Q_k it suffices to ensure $F\mathbf{x} = r_{kk}\mathbf{e}_1$.

As unitary matrices preserve distance, one needs $|r_{kk}| = ||\mathbf{x}||_2$ and so we start by taking $r_{kk} = ||\mathbf{x}||_2$. One geometric way to do this would be by rotation. This is not optimal from a pratical point of view. Instead, reflection is a better choice. Namely, let $\mathbf{v}_+ = ||\mathbf{x}||_2\mathbf{e}_1 - \mathbf{x}$ and let $E_+ = span(\mathbf{v}_+)$ and $H_+ = E_+^\perp$. We can then take F to be reflection across H_+. As it will then be a unitary matrix with the desired mapping property. (Note: this geometric intrepretation works best over the reals).

Lets figure out the matrix for the reflection. Let $P_{\mathbf{v}_+}$ denote orthogonal projection onto E_+ and P_{H_+} denote orthgonal projection onto H_+, i.e they are complementary orthogonal projectors. One verifes then that $F = I - 2P_{E_+}$ is unitary and has the desired behavior. In other words:

$$F = I - 2\frac{\mathbf{v}_+\mathbf{v}_+^*}{||\mathbf{v}_+||_2^2}$$

Notice there are other choices. For instance one could try and make $F\mathbf{x} = -||\mathbf{x}||_2\mathbf{e}_1$. Here we let $\mathbf{v}_- = -||\mathbf{x}||_2\mathbf{e}_1 - \mathbf{x}$ and work as above then we are reflecting across the hyperplane H_- which is orthgonal to $\mathbf{v}_-$. Mathematically both choices are equivalent (i.e. lead to the same answer) however, numerically it turns out to be better to choose the sign so that $F\mathbf{x}$ is as far as possible from $\mathbf{x}$. It is easy to see that this is equivalent to choosing $r_{kk} = -sign(x_1)||\mathbf{x}||_2$ where x_1 is first component of $\mathbf{x}$ and $sign(x_1) = 1$ if $x_1 \geq 0$ and $= -1$ if $x_1 < 0$. One way to see why this might be the case is to consider the real case when the dimension is 3 or larger, i.e. where $F \in \mathbb{R}^{n\times n}$ for $n \geq 3$. When this is the case, if $\mathbf{x}$ is near $||\mathbf{x}||_2\mathbf{e}_1$ and one trys to reflect $\mathbf{x}$ to $||\mathbf{x}||_2\mathbf{e}_1$ a very small perturbation of $\mathbf{x}$ could cause F to change a lot (think what happens if you rotation $\mathbf{x}$ around the $\mathbf{e}_1$ axis by 90°–the reflecting plane also rotates by 90°). On the other hand in this case reflecting to $-||\mathbf{x}||_2\mathbf{e}_1$ is not very sensitive to small perturbations (the plane will always be near the one perpendicular to $\mathbf{e}_1$.