L24 QR factorization (cont.), QR algorithm

CONSTRUCTING GIVENS ROTATIONS

The general scheme for constructing the Givens rotation that zeros element (i,j) in an $n \times n$ matrix is:

$$G(i,j) = \begin{array}{c c} & \begin{array}{ccccccc} & & & j & & & \end{array} \\ \begin{array}{c} \\ \\ \\ i \\ \\ \end{array} & \begin{bmatrix} 1 & & \vdots & & & & \\ & \ddots & \vdots & & & 0 & \\ & & c & \cdots & -s & & \\ & & \vdots & \ddots & \vdots & & \\ \cdots & \cdots & s & \cdots & c & & \\ & 0 & & & & \ddots & \\ & & & & & & 1 \end{bmatrix} \end{array} \qquad (i > j)$$

The reason is that:

$$b_{ij} = \left[G(i,j)A\right]_{ij} = \begin{array}{c} \begin{array}{ccccccc} & & j & & i & & \end{array} \\ (0 \cdots s \cdots c \cdots 0) \end{array} \begin{array}{c c} \begin{bmatrix} a_{1j} \\ \vdots \\ a_{jj} \\ \vdots \\ a_{ij} \\ \vdots \\ a_{nj} \end{bmatrix} & \begin{array}{c} \\ \\ j \\ \\ i \\ \\ \\ \end{array} \end{array}$$

$$= s a_{jj} + c a_{ij}$$

while

$$b_{jj} = [G(ij)A]_{jj} = \begin{array}{c} \quad\ \ j \qquad\ i \\ (0 \cdots c \cdots -s \cdots 0) \end{array} \begin{array}{cc} \begin{pmatrix} a_{1j} \\ \vdots \\ a_{jj} \\ \vdots \\ a_{ij} \\ \vdots \\ a_{nj} \end{pmatrix} & \begin{array}{c} \\ \\ j \\ \\ i \\ \\ \\ \end{array} \end{array}$$

$$= c a_{jj} - s a_{ij}$$

ie. $$\begin{bmatrix} c & -s \\ s & c \end{bmatrix} \begin{bmatrix} a_{jj} \\ a_{ij} \end{bmatrix} = \begin{bmatrix} b_{jj} \\ b_{ij} \end{bmatrix}$$

We can then use the boxed formulae on p3 to choose c, s such that $b_{ij} = 0$. Concretely:

$$c = a_{jj}/r$$

$$s = -a_{ij}/r$$

$$r = \sqrt{a_{jj}^2 + a_{ij}^2}$$

Now we turn to the second factor, Q, in the factorization, $A = QR$.

Def: $Q = G_1^T G_2^T$

Then

$$QR = G_1^T G_2^T G_2 G_1 A$$

Now recall that a Given matrix is orthogonal. In particular that mean that

$$G^T = G^{-1}$$

Thus

$$QR = G_1^T G_2^{-1} G_2 G_1 A$$

$$= G_1^T G_1 A = G_1^{-1} G_1 = A.$$

We have factorized A as

$$A = QR$$

where R is upper triangular, and Q is orthogonal:

$$Q^T Q = (G_1^T G_2^T)^T (G_1^T G_2^T)$$

$$= G_2 G_1 G_1^T G_2^T$$

$$= I$$

... implying that Q is orthogonal.

The factorization $A = QR$ is known as QR factorization

QR ALGORITHM

The QR Algorithm is a procedure to compute the eigenvalues of a matrix, A.

The basic idea is to perform a QR factorization,

$$A = QR,$$

multiply the factors in the reverse order,

$$RQ,$$

and iterate, ie

$$A_0 = A$$

$$A_0 = Q_0 R_0$$

$$A_1 = R_0 Q_0$$

$$A_1 = Q_1 R_1$$

$$A_2 = R_1 Q_1$$

$$\vdots$$

What does this have to do with the eigenvalues of A? Well, at the k^{th} step we have:

$$A_{k+1} = R_k Q_k$$

$$= (Q_k^{-1} Q_k) R_k Q_k$$

$$= Q_k^{-1} (Q_k R_k) Q_k$$

$$= Q_k^{-1} A_k Q_k \qquad (*)$$

So all the A_k's are similar, and hence they have the same eigenvalues.

Why is this useful? Because, under certain conditions, the A_k converge to a simple form from which one can easily "pick off" the evalues. In particular, for A symmetric & tridiagonal, $A_k \xrightarrow{k \to \infty} D$, where D is a diagonal matrix. Of course, the evalues of a diagonal matrix are the diagonal elements.

Also, note that:

$$A_{k+1}^T = (Q_k^{-1} A_k Q_k)^T \qquad \text{by } (*)$$

$$= (Q_k^T A_k Q_k)^T$$

$$= Q_k^T A_k^T Q_k$$

$$= Q_k^{-1} A_k^T Q_k$$

$$= A_{k+1} \quad \text{if } A_k \text{ is symmetric}$$

Thus if A is symmetric, then so are all the A_k's. As $k \to \infty$ the off-diagonal elements approach zero in a symmetric fashion.

EXAMPLE Recall our example from L23:

$$A = \begin{bmatrix} 1 & 3 & 0 \\ 3 & 1 & 3 \\ 0 & 3 & 1 \end{bmatrix} = A_0.$$

We found:

$$A_0 = Q_0 R_0$$

$$= \begin{bmatrix} .94 & -.29 & .12 \\ .31 & .88 & -.34 \\ 0 & .36 & .92 \end{bmatrix} \begin{bmatrix} \sqrt{10} & 6/\sqrt{10} & 3/\sqrt{10} \\ 0 & 2.720 & 1.985 \\ 0 & 0 & 2.441 \end{bmatrix}$$

Turns out that:

$$A_1 = R_0 Q_0$$

$$= \begin{bmatrix} 3.6 & .86 & 0 \\ .86 & 3.12 & .89 \\ 0 & .89 & 2.3 \end{bmatrix}$$

Comparing A_1 with A_0, we see that one iteration of the QR algorithm has reduced considerably the off-diagonal elements.

Also, the new matrix A_1, is symmetric and tridiagonal, just like the old one A_0.

If we repeat the process 13 times we find:

$$A^{(13)} = \begin{bmatrix} 4.41 & .01 & 0 \\ .01 & 3.00 & .00095 \\ 0 & .00095 & 1.58 \end{bmatrix}$$

$$\approx \begin{bmatrix} 4.41 & .01 & 0 \\ .01 & 3.00 & 0 \\ 0 & 0 & 1.58 \end{bmatrix}$$

for which one of the evalues, 1.58, can be easily "picked off".

We may then apply the QR algorithm to the matrix

$$\begin{bmatrix} 4.41 & .01 \\ .01 & 3.00 \end{bmatrix}$$

to find the remaining evalues!