Gene Golub/Persi Diaconis
CS 339
Winter Quarter 2001-02
The Decompositional Approach to Matrix Computations, cont'd

We will examine various methods for computing the QR factorization of an $m \times n$ matrix A,

$$A = QR, \quad \text{rank}(A) = r, \quad Q^T Q = I \tag{1}$$

where R is upper triangular. Let $A = [a_1 a_2 \cdots a_n]$ and $Q = [q_1 q_2 \cdots q_n]$ be column partitionings of A and Q, respectively. Setting columns of A and QR equal to one another yields the classical *Gram-Schmidt* process:

$$\begin{aligned} a_j &= r_{1j}q_1 + \cdots + r_{jj}q_j \\ q_i^T q_k &= 0, \quad i \neq k, \quad i,k = 1,\ldots,j-1 \\ r_{ij} &= q_i^T a_j \\ z_j &= a_j - r_{1j}q_1 - \cdots - r_{1,j-1}q_{j-1} \\ \|z\|_2 = r_{jj} & \\ q_j = z_j/r_{jj} & \end{aligned}$$

While this procedure is elegant as a mathematical algorithm, it has poor numerical qualities in a practical implementation. This is due to catastrophic cancellation that occurs if the columns of A are nearly parallel. The *Modified Gram-Schmidt* algorithm is more stable.

A second method relies on *Householder reflections.* Suppose that we define a matrix P by

$$P = I - 2uu^T, \quad u^T u = 1$$

for some vector u. Then

- P is symmetric, i.e. $P = P^T$
- P is orthogonal, i.e. $PP^T = I$.

Given a vector a, we would like to choose u so that $Pa = \alpha e_1$, where α is a scalar and $e_1^T = [10 \cdots 0]$. The orthogonality of P determines α:

$$\|Pa\|_2 = \|a\|_2 = \|\alpha e_1\|_2 = |\alpha|\|e_1\|_2 = |\alpha|$$

which yields $\alpha = \pm\|a\|_2$. Using the definition of P, we obtain

$$a = \alpha P^T e_1 = \alpha P e_1 = \alpha \begin{bmatrix} 1 - 2u_1^2 \\ -2u_1 u_2 \\ \vdots \\ -2u_1 u_m \end{bmatrix}.$$

This yields

$$u_1 = \pm\frac{1}{2}\left(1 - \frac{a_1}{\alpha}\right)^{1/2}, \quad u_j = \frac{a_j}{2\alpha u_1}, \quad j = 2, \ldots, m.$$

From this method of choosing u to zero selected elements, we can now develop the algorithm for computing the QR factorization using Householder transformations. Proceeding as above with $a = a_1$, we obtain

$$P_1 A = A^{(1)} = \begin{bmatrix} r_{11} & r_{12} & \cdots & r_{1n} \\ 0 & & & \\ \vdots & & A_1 & \\ 0 & & & \end{bmatrix}$$

where A_1 is $(m-1) \times (n-1)$ and $P_1 a_j = a_j^{(1)}$ for $j = 2, \ldots, n$. Note for $j = 2, \ldots, n$ we compute

$$\begin{aligned} P_1 a_j &= (1 - 2u_1 u_1^T) a_j \\ &= a_j - \gamma u_1, \quad \gamma = 2u_1^T a_j \\ &= a_j^{(1)} \end{aligned}$$

from which it follows that we can compute each column of the updated A in parallel. Now

$$P_2 = \begin{bmatrix} 1 & 0^T \\ 0 & I - 2u_2 u_2^T \end{bmatrix}, \quad P_2 A^{(1)} = A^{(2)} = \begin{bmatrix} \times & \times & \times & \cdots & \times \\ 0 & r_{22} & r_{32} & \cdots & r_{2n} \\ 0 & 0 & r_{33} & \cdots & r_{3n} \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & r_{m3} & \cdots & r_{mn} \end{bmatrix}$$

and in general we have $P_j A^{(j-1)} = A^{(j)}$ for $j = 2, \ldots, n$ resulting in a matrix $A^{(n)}$ which is upper triangular. The number of operations required for this process is

$$
\begin{aligned}
2\sum_{j=1}^{n-1}(n-j)(mj) &= 2\left(\sum_{j=1}^{n-1} nm - jm - jn + j^2\right) \\
&\sim 2\left[n(n-1)m - \frac{n(n-1)(m+n)}{2} + \frac{n^3}{3}\right] \\
&\sim 2\left[\frac{n^2 m}{2} + \frac{n^3}{3} - \frac{n^3}{2}\right] \\
&\sim n^2 m - \frac{n^3}{3}
\end{aligned}
$$

or $2n^3/3$ when $m = n$.

So, since we have

$$P_n \cdots P_2 P_1 A = \begin{bmatrix} R \\ 0 \end{bmatrix}$$

it follows that

$$A = (P_n \cdots P_2 P_1)^T \begin{bmatrix} R \\ 0 \end{bmatrix}$$

and therefore $Q = P_1 \cdots P_n$. We do not need to store the matrices P_j but only the vectors u_j. The collection $\{u_1, u_2, \ldots, u_n\}$ can be stored in the lower triangular portion of an $m \times n$ matrix, so we can store the entire QR factorization in a compact form by storing the nonzero elements of the vectors u_1, …, u_n in the lower triangular portion of A, and the nonzero elements of R in the upper triangular portion.

If A is itself an orthogonal matrix, then R must be a diagonal matrix, with each diagonal entry equal to ± 1, since R would be both orthogonal and upper triangular. It follows from our compact representation of the QR factorization that every orthogonal matrix can be expressed as $n(n+1)/2$ parameters. This could be a good way to generate a random orthogonal matrix: choose that many parameters randomly, and apply Householder transformations based on the resulting vectors $u_1, \ldots, u_n$ to the identity matrix.

Finally, we discuss an algorithm due to Jacobi, whose work from the 19th century was later popularized by Givens. Consider the matrix Z_{ij} defined by

$$
Z_{ij} = \begin{bmatrix}
1 & & & & & & & & & & \\
& \ddots & & & & & & & & & \\
& & 1 & & & & & & & & \\
& & & \cos\theta_{ij} & & & & \sin\theta_{ij} & & & \\
& & & & 1 & & & & & & \\
& & & & & \ddots & & & & & \\
& & & & & & 1 & & & & \\
& & & -\sin\theta_{ij} & & & & \cos\theta_{ij} & & & \\
& & & & & & & & 1 & & \\
& & & & & & & & & \ddots & \\
& & & & & & & & & & 1
\end{bmatrix}.
$$

It can be shown that $Z_{ij}^T Z_{ij} = I_n$. Given a vector $[ab]^T$ we would like to choose θ so that

$$
\begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} a \\ b \end{bmatrix} = \begin{bmatrix} \sqrt{a^2+b^2} \\ 0 \end{bmatrix}.
$$

From this system of linear equations we obtain

$$
\tan\theta = \frac{b}{a}, \quad \cos\theta = \pm\frac{a}{\sqrt{a^2+b^2}}, \quad \sin\theta = \frac{b}{\pm\sqrt{a^2+b^2}}.
$$

Choosing θ_{ij} in this manner for select i and j, with a and b chosen from a matrix that we are trying to reduce to upper triangular form, we can compute the QR factorization of A as follows:

$$
Z_{21}A = \begin{bmatrix}
a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\
0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\
a_{31} & & & \\
\vdots & & & \\
a_{m1} & & &
\end{bmatrix},
$$

$$
Z_{m1}\cdots Z_{31}A = \begin{bmatrix}
a_{11}^{(1)} & a_{12}^{(1)} & \cdots & a_{1n}^{(1)} \\
0 & a_{22}^{(1)} & \cdots & a_{2n}^{(1)} \\
\vdots & \vdots & & \vdots \\
0 & a_{m2}^{(1)} & \cdots & a_{mn}^{(2)}
\end{bmatrix} = A^{(1)},
$$

$$Z_{m2}\cdots Z_{32}A^{(1)} = \begin{bmatrix} a_{11}^{(1)} & \cdots & \cdots & a_{1n}^{(1)} \\ 0 & a_{22}^{(2)} & \cdots & a_{2n}^{(2)} \\ \vdots & 0 & & \\ & \vdots & & \end{bmatrix},$$

$$Z_{mn}\cdots Z_{n+1,n}A^{(n-1)} = A^{(n)} = \begin{bmatrix} r_{11} & \cdots & r_{1n} \\ & \ddots & \vdots \\ & & r_{nn} \\ & 0 & \end{bmatrix}.$$

This process can be shown to be very stable. It is also well-suited for parallelization, since each rotation only affects two rows. For example, one could zero elements starting from the lower left corner, proceeding by diagonal.

We can use these different techniques for different problems, but why are we interested in the QR factorization? It is very useful in linear least-squares problems, where given an $m \times n$ matrix A, with $m \geq n$ and $\text{rank}(A) = n$, along with a right side b, our goal is to find the linear least squares solution x that minimizes $\|b - Ax\|_2$. We can proceed as follows:

$$\begin{aligned} \|b - Ax\|_2^2 &= \|Q^T b - Q^T Ax\|_2^2 \\ &= \left\| \begin{bmatrix} c \\ d \end{bmatrix} - \begin{bmatrix} R \\ 0 \end{bmatrix} x \right\|_2^2 \\ &= \|c - Rx\|_2^2 + \|d\|_2^2 \end{aligned}$$

It follows that the solution is easily obtained by solving the upper triangular system $Rx = c$. We can summarize the procedure as follows:

1. Find the QR factorization of A
2. Compute $Q^T b = \begin{bmatrix} c \\ d \end{bmatrix}$.
3. Solve $Rx = c$.

Note that if we change the right-hand side, we can easily solve the modified problem when the QR factorization is known. So, given A and g, along with the QR factorization of A, we can find y that minimizes $\|g - Ay\|_2$ as follows:

1. Compute $\begin{bmatrix} p \\ q \end{bmatrix} = Q^T g$

2. Solve $Ry = p$.

We now illustrate another use of the QR factorization. Suppose $b - Ax = z$ where z is the solution to the least squares problem. Then

$$\begin{aligned} z &= b - Q\begin{bmatrix} R \\ 0 \end{bmatrix}\begin{bmatrix} R^{-1} & Q^T \end{bmatrix}\begin{bmatrix} c \\ d \end{bmatrix} \\ &= \left(I - Q\begin{bmatrix} I & 0 \\ 0 & 0 \end{bmatrix}Q^T\right) b \\ &= Q\begin{bmatrix} 0 & 0 \\ 0 & I \end{bmatrix}Q^T b \end{aligned}$$

The matrix multiplying b in the last step is called the *projection matrix*, which projects b onto the orthogonal complement of the column space of A.

The QR factorization is also useful for updating and downdating. Suppose A_1 is $m_1 \times n$, A_2 is $m_2 \times n$, and the QR factorization of A_1, $A_1 = Q_1 R_1$, is known. Then

$$\begin{bmatrix} Q_1^T & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = \begin{bmatrix} R_1 \\ 0 \\ A_2 \end{bmatrix}.$$

Rearranging rows on the right side, we can then complete the QR factorization of the updated matrix:

$$\begin{bmatrix} Q_2^T & 0 \\ 0 & I \end{bmatrix}\begin{bmatrix} A_2 \\ R_1 \\ 0 \end{bmatrix} = \begin{bmatrix} R_2 \\ 0 \end{bmatrix}.$$

Sometimes $m_2 = 1$ and in this case we apply Jacobi rotations to zero the appropriate elements of the updated matrix.

What happens if A is $m \times n$ and $\text{rank}(A) = r < n$? We can choose a permutation matrix Π so that

$$Q^T A \Pi = \begin{bmatrix} R_{r\times r} & S \\ 0 & 0 \end{bmatrix}$$

where S is $r \times (n - r)$. Computing the QR factorization of $\Pi^T A^T Q$, we obtain

$$W^T(\Pi^T A^T Q) = \begin{bmatrix} U & 0 \\ 0 & 0 \end{bmatrix}$$

where U is an $r \times r$ upper triangular matrix. Defining $Z = \Pi W$, we obtain the *complete orthogonal decomposition* of A,

$$A = Q \begin{bmatrix} L & 0 \\ 0 & 0 \end{bmatrix} Z^T.$$

Another useful decomposition is the *Cholesky decomposition*, which factors a symmetric positive definite matrix A into $A = F^T F$, where F is upper triangular. Equating $A = F^T F$, we have

$$\begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & & \vdots & \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{bmatrix} = \begin{bmatrix} f_{11} & & & \\ f_{12} & f_{22} & & \\ \vdots & & \ddots & \\ f_{1n} & & & f_{nn} \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & \cdots & f_{1n} \\ & f_{22} & & f_{2n} \\ & & \ddots & \vdots \\ & & & f_{nn} \end{bmatrix}$$

which yields the following formulas for the elements of F:

$$\begin{aligned} f_{11} &= \sqrt{a_{11}} \\ f_{1j} &= a_{1j}/f_{11} \\ f_{22} &= \sqrt{a_{22} - f_{12}^2} \\ f_{2j} &= (a_{2j} - f_{12} f_{1j})/f_{22} \end{aligned}$$

and so on. Under mild assumptions, this procedure is very stable provided that A is positive definite. Furthermore, F is unique when the positive square roots are chosen for the diagonal elements, and no pivoting is necessary. Therefore, *any* pivoting is permissible, and can be used to preserve sparsity. In any case, band structure is preserved: if A is tridiagonal, for example, then F is bidiagonal. Finally, note that if $A = QR$, then $A^T A = R^T R$, so R is the Cholesky factor of A.

Finally, we consider the singular value decomposition (SVD) of A,

$$A = U \Sigma V^T$$

where A is $m \times n$ with $m \geq n$, $U^T U = I_m$, $V^T V = I_n$, and

$$\Sigma = \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_n \\ & 0 & \end{bmatrix}$$

where $\sigma_1 \geq \cdots \geq \sigma_n$ are the *singular values* of A. We see that

$$A^T A = V\Sigma^T \Sigma V^T = V \begin{bmatrix} \sigma_1^2 & & \\ & \ddots & \\ & & \sigma_n^2 \end{bmatrix} V^T$$

so the columns of V are the eigenvectors of $A^T A$, with corresponding eigenvalues $\sigma_1^2, \ldots, \sigma_n^2$. Similarly, the columns of U are the eigenvectors of AA^T, with corresponding eigenvalues $\sigma_1^2, \ldots, \sigma_n^2$, along with $(m-n)$ zero eigenvalues.

We now discuss some applications of the SVD. Suppose X is an $n \times n$ matrix that is intended to be an orthogonal matrix but isn't, due to roundoff errors. What is the closest orthogonal matrix to X? And what do we mean by closest?

If Q is orthogonal, then the singular values are equal to 1 since $Q^T Q = I$. Now, the norm that we use to describe distance from an orthogonal matrix is the *Frobenius norm*

$$\|A\|_F = \left(\sum_{i,j} |a_{ij}|^2 \right)^{1/2}.$$

This norm has the properties that $\|PA\|_F^2 = \|A\|_F^2$ and $\|AQ^T\|_F^2 = \|A\|_F^2$ when P and Q are orthogonal. So we want to find $\hat{Q}$ so that

$$\|X - \hat{Q}\|_F \leq \|X - Q\|_F$$

where Q and $\hat{Q}$ belong to the set of all $n \times n$ orthogonal matrices. The solution is $\hat{Q} = UV^T$ where $X = U\Sigma V^T$ is the SVD of X. To measure X's departure from orthogonality, we proceed as follows:

$$\begin{aligned} \|X - \hat{Q}\|_F^2 &= \|U\Sigma V^T - UV^T\|_F^2 \\ &= \|\Sigma - I\|_F^2 \\ &= \sum_{i=1}^{n} (\sigma_i - 1)^2 \end{aligned}$$