QR Decomposition

Stephen D. Butalla & Valerie Kobzarenko

September 10, 2019

CSE-5400
Florida Institute of Technology
QR Decomposition

- QR decomposition uses about twice as many computations as other methods.
- Works on rectangular matrices (only square will be discussed here).
- Matrix must be symmetric.
- Pivoting is not necessary unless the coefficient matrix is (approximately) singular.
• QR decomposition takes the coefficient matrix and breaks it into two matrices [1, 2]:

\[ A = Q \cdot R \]

where \( Q \) is an orthogonal matrix, i.e.,

\[ Q \cdot Q^T = I \]

and \( R \) is an upper triangular matrix:

\[
R = \begin{pmatrix}
    r_{00} & \cdots & r_{0n} \\
    \vdots & \ddots & \vdots \\
    0 & \cdots & r_{nn}
\end{pmatrix}
\]

• Substituting:

\[ Q \cdot R \cdot x = b \]

\[ I \cdot R \cdot x = R \cdot x = Q^T \cdot b \]
• First, let

$$Q^T \cdot b = y$$

Then solve

$$R \cdot x = y$$

by back substitution.

• To find the matrices $Q$ and $R$, we use Householder transformations (or reflections) [2, 3]

• This method reduces a symmetric square matrix to tridiagonal form by successive orthogonal transformations which zeroes the proper elements in the corresponding column/row [2]
A Householder transformation takes a vector $\mathbf{x}$ and reflects it through a (hyper)plane with respect to the normal vector of the (hyper)plane $\mathbf{v}$, whose norm is of unit length [4]:

$$\langle \mathbf{v} | \mathbf{v} \rangle = |\mathbf{v}|^2 = \mathbf{v}^T \cdot \mathbf{v} = 1$$

The Householder matrix is given by [2]

$$\mathbf{P} = \mathbb{I} - 2\mathbf{v} \otimes \mathbf{v}$$

where $\otimes$ is the outer product, i.e.,

$$\mathbf{v} \otimes \mathbf{v} = |\mathbf{v}\rangle \langle \mathbf{v}| = \mathbf{v} \cdot \mathbf{v}^T$$

Properties [5]:

1. Involutary: $\mathbf{P} \cdot \mathbf{P} = \mathbb{I}$
2. Hermitian: $\mathbf{P} = \mathbf{P}^\dagger \Rightarrow$ symmetric
3. Unitary: $\mathbf{P}^{-1} = \mathbf{P}^\dagger \Rightarrow$ orthogonal
4. Determinant: $|\mathbf{P}| = -1$
• Applying the correct Householder matrix zeroes all non-diagonal elements in a column. First, operate on the first column and get [3]

\[
A' = P_0 A = \begin{pmatrix}
a_{00} & \cdots & a_{0n} \\
0 & a_{ii} & a_{in} \\
\vdots & \ddots & \vdots \\
0 & \cdots & a_{nn}
\end{pmatrix}
\]

• This is achieved using the following prescription [2]:
1. \[ P = I - 2v \otimes v = I - \frac{v \cdot v^T}{H} \]
   where \( H = \frac{1}{2} |v|^2 \)
2. Let \( v \) be
   \[ v = x_0 \mp |x_0| |0\rangle \]
   where \( |0\rangle \) is the unit vector \( \langle 0 | = [1 \ 0 \ \cdots \ 0] \) and \( x_0 \) is the first column vector of some matrix \( X \), i.e.,
   \[ x_0 = [x_{00} \ x_{10} \ \cdots \ x_{n0}] \]
3. Operate on the vector \( x_0 \) with \( P \):
   \[ P \cdot x = x_0 - \frac{v}{H} \left( x_0 \mp |x_0| |0\rangle \right)^T \cdot x_0 = x_0 - v = \pm |x_0| |0\rangle \]
Applying this procedure to $A$ [2], we choose the vector $x_0$ to be the first $n - 1$ elements of $a_0$, i.e.,

$$a_0^T = [a_{10}, a_{20}, \ldots, a_{n0}]$$

So, the lower $n - 2$ elements will be zeroed, leaving:

$$P_0 \cdot A = \begin{bmatrix} a_{00} & a_{01} & \cdots & a_{0,n-1} \\ c & * & \cdots & * \\ \vdots & * & \ddots & * \\ 0 & * & \cdots & * \end{bmatrix}$$

where $c = |a_0|$

Applying the first Householder transformation twice (sandwiching $A$), we zero both the 0th column and row:

$$P_0 \cdot A P_0 = \begin{bmatrix} a_{00} & c & \cdots & 0 \\ c & * & \cdots & * \\ \vdots & * & \ddots & * \\ 0 & * & \cdots & * \end{bmatrix}$$

which is possible because $P_0^T = P$
Householder Transformations

- Now, continue choosing the next vector for the Householder transformation as the $n - 2$ elements from column 1 and repeat.
- To save time/memory used in performing matrix multiplication, we can compute the vector:

$$p = \frac{A \cdot v}{H}$$

and use the following procedure [2]:

$$A' = PAP$$

$$= A - p \cdot v^T - v \cdot p^T + 2Kv \cdot v^T$$

where $K = \frac{v \cdot p}{2H}$

- Simplifying, let $q = p - Kv$
- Finally,

$$A' = A - q \cdot v^T - v \cdot p^T$$
Householder Transformations

- At any stage $k$ in the algorithm, the vector $\mathbf{v}$ takes the form

$$\mathbf{v}^T = [a_{k0}, a_{k1}, a_{k,k-1} \pm \sqrt{\sigma}, \cdots, 0]$$

where $\pm \sqrt{\sigma} = |a_k|$, where the sign is chosen to reduce roundoff error [2]

- If [2]

$$\sigma < \frac{\text{smallest (+) number representable}}{\text{machine precision}}$$

Define $\tau = \sum_{k=0}^{i-1} |a_{ik}|$ If $\tau = 0$ compared to the machine precision, we can skip the Householder transformation, else we set

$$a_{ik} \rightarrow \frac{a_{ik}}{\tau}$$
Decomposition Code Example

Adapted from [2]:

```c
for (k=0;k<n-1;k++) {
    scale=0.0;
    for (i=k;i<n;i++) scale=MAX(scale,abs(r[i][k]));
    // If singular, skip the transformation:
    if (scale == 0.0) {
        sing=true;
        c[k]=d[k]=0.0;
    } else {
        // Use scaled vectors for Householder transformation
        for (i=k;i<n;i++) r[i][k] /= scale;
        for (sum=0.0,i=k;i<n;i++) sum += SQR(r[i][k]);
        sigma=SIGN(sqrt(sum),r[k][k]);
        r[k][k] += sigma;
        c[k]=sigma*r[k][k];
        d[k] = -scale*sigma;
        for (j=k+1;j<n;j++) {
            for (sum=0.0,i=k;i<n;i++) sum += r[i][k]*r[i][j];
            tau=sum/c[k];
            for (i=k;i<n;i++) r[i][j] -= tau*r[i][k];
        }
    }
}
```
Once the transformations are complete, we have the $Q$ and $R$ matrices:

\[
Q = P_0 P_1 \cdots P_m
\]
\[
R = P_m P_{m-1} \cdots P_0 A
\]

Finally, we can solve the original set of linear equations,

\[
A \cdot x = b
\]

∴ $A = QR$, we can substitute:

\[
QR \cdot x = b
\]
\[
R \cdot x = Q^T b
\]

First, find $y$ by multiplying:

\[
Q^T b = y
\]

```c
for (i=0;i<n;i++) {
    sum = 0.;
    for (j=0;j<n;j++) sum += qt[i][j]*b[j];
    y[i] = sum;
}
```
Using backsubstitution, solve

\[ Rx = y \]

```c
for (i=n-1; i>=0; i--) {
    sum = b[i];
    for (j=i+1; j<n; j++) sum -= r[i][j]*x[j];
    x[i] = sum / r[i][i];
}
```
References


