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 Algorithm

- QR decomposition takes the coefficient matrix and breaks it into two matrices [1, 2]:

$$\mathbb{A} = \mathbb{Q} \cdot \mathbb{R}$$

where $\mathbb{Q}$ is an orthogonal matrix, i.e.,

$$\mathbb{Q} \cdot \mathbb{Q}^T = \mathbb{I}$$

and $\mathbb{R}$ is an upper triangular matrix:

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

- Substituting:

$$\mathbb{Q} \cdot \mathbb{R} \cdot \mathbf{x} = \mathbf{b}$$
$$\mathbb{I} \cdot \mathbb{R} \cdot \mathbf{x} = \mathbb{R} \cdot \mathbf{x} = \mathbb{Q}^T \cdot \mathbf{b}$$

QR Decomposition Algorithm

- First, let

$$\mathbb{Q}^T \cdot \mathbf{b} = \mathbf{y}$$

  Then solve

$$\mathbb{R} \cdot \mathbf{x} = \mathbf{y}$$

  by back substitution.
- To find the matrices $\mathbb{Q}$ and $\mathbb{R}$, we use Housholder 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]

Householder Transformations

- 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]

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

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: $\mathbb{P} \cdot \mathbb{P} = \mathbb{I}$
  2. Hermitian: $\mathbb{P} = \mathbb{P}^\dagger \quad \Rightarrow \quad$ symmetric
  3. Unitary: $\mathbb{P}^{-1} = \mathbb{P}^\dagger \quad \Rightarrow \quad$ orthogonal
  4. Determinant: $|\mathbb{P}| = -1$

Householder Transformations

- Applying the correct Householder matrix zeroes all non-diagnonal elements in a column. First, operate on the first column and get [3]

$$\mathbb{A}' = \mathbb{P}_0\mathbb{A} = \begin{pmatrix} a_{00} & \cdots & a_{0n} \\ 0 & a_{ii} & a_{in} \\ \vdots & \ddots & \vdots \\ 0 & \cdots & a_{nn} \end{pmatrix}$$

- This is achieved using the following prescription [2]:
  1. $$\mathbb{P} = \mathbb{I} - 2\mathbf{v} \otimes \mathbf{v} = \mathbb{I} - \frac{\mathbf{v} \cdot \mathbf{v}^T}{H}$$

     where $H = \frac{1}{2}|\mathbf{v}|^2$
  2. Let $\mathbf{v}$ be

     $$\mathbf{v} = \mathbf{x}_0 \mp |\mathbf{x}_0|\,|0\rangle$$

     where $|0\rangle$ is the unit vector $\langle 0| = [1\ 0\ \cdots\ 0]$ and $\mathbf{x}_0$ is the first column vector of some matrix $\mathbf{X}$, i.e.,

     $$\mathbf{x}_0 = [x_{00}\ x_{10}\ \cdots\ x_{n0}]$$
  3. Operate on the vector $\mathbf{x}_0$ with $\mathbb{P}$:

     $$\mathbb{P} \cdot \mathbf{x} = \mathbf{x}_0 - \frac{\mathbf{v}}{H}\big(\mathbf{x}_0 \mp |\mathbf{x}_0|\,|0\rangle\big)^T \cdot \mathbf{x}_0 = \mathbf{x}_0 - \mathbf{v} = \pm|\mathbf{x}_0|\,|0\rangle$$

Householder Transformations

- Applying this procedure to **A** [2], we choose the vector $\mathbf{x}_0$ to be the first $n-1$ elements of $\mathbf{a}_0$, i.e.,

$$\mathbf{a}_0^T = \begin{bmatrix} a_{10}, & a_{20}, & \cdots & a_{n0} \end{bmatrix}$$

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

$$\mathbb{P}_0 \cdot \mathbb{A} = \begin{bmatrix} a_{00} & a_{01} & \cdots & a_{0,n-1} \\ c & * & \cdots & * \\ \vdots & * & \ddots & * \\ 0 & * & \cdots & * \end{bmatrix}$$

  where $c = |\mathbf{a}_0|$

- Applying the first Householder transformation twice (sandwiching $\mathbb{A}$), we zero both the 0^{th} column and row:

$$\mathbb{P}_0 \cdot \mathbb{A}\mathbb{P}_0 = \begin{bmatrix} a_{00} & c & \cdots & 0 \\ c & * & \cdots & * \\ \vdots & * & \ddots & * \\ 0 & * & \cdots & * \end{bmatrix}$$

  which is possible because $\mathbb{P}_0^T = \mathbb{P}$

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:

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

and use the following procedure [2]:

$$\begin{aligned} \mathbb{A}' &= \mathbb{P}\mathbb{A}\mathbb{P} \\ &= \mathbb{A} - \mathbf{p} \cdot \mathbf{v}^T - \mathbf{v} \cdot \mathbf{p}^T + 2K\mathbf{v} \cdot \mathbf{v}^T \end{aligned}$$

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

- Simplifying, let $\mathbf{q} = \mathbf{p} - K\mathbf{v}$
- Finally,

$$\mathbb{A}' = \mathbb{A} - \mathbf{q} \cdot \mathbf{v}^T - \mathbf{v} \cdot \mathbf{p}^T$$

Householder Transformations

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

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

where $\pm\sqrt{\sigma} = |\mathbf{a}_k|$, 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]:

```
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];
                }
        }
}
```

QR Decomposition

- Once the transformations are complete, we have the $\mathbb{Q}$ and $\mathbb{R}$ matrices:

$$\mathbb{Q} = \mathbb{P}_0 \mathbb{P}_1 \cdots \mathbb{P}_m$$
$$\mathbb{R} = \mathbb{P}_m \mathbb{P}_{m-1} \cdots \mathbb{P}_0 \mathbb{A}$$

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

$$\mathbb{A} \cdot \mathbf{x} = \mathbf{b}$$

- $\because \mathbb{A} = \mathbb{Q}\mathbb{R}$, we can substitute:

$$\mathbb{Q}\mathbb{R} \cdot \mathbf{x} = \mathbf{b}$$
$$\mathbb{R} \cdot \mathbf{x} = \mathbb{Q}^T \mathbf{b}$$

- First, find **y** by multiplying:

$$\mathbb{Q}^T \mathbf{b} = \mathbf{y}$$

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

QR Decomposition

- Using backsubstitution, solve

$$\mathbb{R}\mathbf{x} = \mathbf{y}$$

```
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

[1] M. Parker, *Digital Signal Processing 101: Everything You Need to Know to Get Started*, 2017, Elsevier Inc.

[2] W. Press, S. Teukolsky, W. Vetterling, & B. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 2007, Cambridge University Press.

[3] "QR Decomposition," *RosettaCode*, https://rosettacode.org/wiki/QR_decomposition#C.2B.2B.

[4] R. Wang, "Householder transformation and QR decomposition," 2015, http://fourier.eng.hmc.edu/e176/lectures/NM/node10.html.

[5] J. Kerl, "The Householder transformation in numerical linear algebra," 2008, https://johnkerl.org/doc/hh.pdf.