Rank revealing QR decomposition

Given any matrix $A \in \mathbb{R}^{n\times m}$, $n \geq m$, with $\operatorname{rank} A = r(\leq m)$, there exists a permutation matrix P, an orthogonal matrix $Q \in \mathbb{R}^{n\times n}$ and a matrix $R = \begin{array}{cc} \begin{array}{cc} r & m-r \end{array} & \\ \left[\begin{array}{c|c} R_1 & R_2 \\ \hline 0 & 0 \end{array}\right] & \begin{array}{l} r \\ n-r \end{array} \end{array}$ where $R_1 \in \mathbb{R}^{r\times r}$ is nonsingular and upper triangular and $R_2 \in \mathbb{R}^{r\times(m-r)}$ such that

$$AP = QR. \tag{3}$$

Such a decomposition is called a *column pivoted* or *rank revealing* decomposition of A as the size of R_1 ‘reveals’ the rank of A.

If A is a complex matrix, then the above decomposition exists for a unitary matrix Q with $\mathbb{R}$ replaced by $\mathbb{C}$ throughout.

Computing rank revealing QR

Find an $m \times m$ permutation matrix P_1 such that the first column of AP_1 has the largest 2-norm among all the columns of A.

Computing rank revealing QR

Find an $m \times m$ permutation matrix P_1 such that the first column of AP_1 has the largest 2-norm among all the columns of A.
[Costs 2nm + m flops]

Computing rank revealing QR

Find an $m \times m$ permutation matrix P_1 such that the first column of AP_1 has the largest 2-norm among all the columns of A.
[Costs 2nm + m flops]

Then choose an $n \times n$ reflector Q_1 such that

$$Q_1 A P_1 = \underbrace{\begin{bmatrix} \pm\|AP_1(:,1)\|_2 & a_{12}^{(1)} & \cdots & a_{1m}^{(1)} \\ 0 & a_{22}^{(1)} & \cdots & a_{2m}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nm}^{(1)} \end{bmatrix}}_{=:A_1}$$

Computing rank revealing QR

Find an $m \times m$ permutation matrix P_1 such that the first column of AP_1 has the largest 2-norm among all the columns of A.
[Costs 2nm + m flops]

Then choose an $n \times n$ reflector Q_1 such that

$$Q_1 A P_1 = \underbrace{\begin{bmatrix} \pm\|AP_1(:,1)\|_2 & a_{12}^{(1)} & \cdots & a_{1m}^{(1)} \\ 0 & a_{22}^{(1)} & \cdots & a_{2m}^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 0 & a_{n2}^{(1)} & \cdots & a_{nm}^{(1)} \end{bmatrix}}_{=:A_1}$$

[Costs 4n(m-1) flops]

Computing rank revealing QR

Find an $m \times m$ permutation P_2 such that $A_1P_2(2:n,2)$ has the largest 2-norms among the columns $A_1(2:n,k), k = 2,\ldots,m$.

Computing rank revealing QR

Find an $m \times m$ permutation P_2 such that $A_1 P_2(2:n, 2)$ has the largest 2-norms among the columns $A_1(2:n, k), k = 2, \ldots, m$.
[Costs 2(n-1)(m-1) + m-1 flops]

Computing rank revealing QR

Find an $m \times m$ permutation P_2 such that $A_1 P_2(2:n,2)$ has the largest 2-norms among the columns $A_1(2:n,k), k = 2,\ldots,m$.
[Costs 2(n-1)(m-1) + m-1 flops]

Then choose an $n \times n$ reflector $Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{Q}_2 \end{bmatrix}$ where $\tilde{Q}_2$ is an $n-1 \times n-1$ reflector such that

$$Q_2 Q_1 A P_1 P_2 = Q_2 A_1 P_2 = \underbrace{\begin{bmatrix} \pm\|A(:,1)\|_2 & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1m}^{(1)} \\ 0 & \pm\|A_1 P_2(2:n,2)\|_2 & a_{23}^{(2)} & \cdots & a_{2m}^{(2)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3m}^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & a_{n3}^{(2)} & \cdots & a_{nm}^{(2)} \end{bmatrix}}_{=:A_2}$$

Computing rank revealing QR

Find an $m \times m$ permutation P_2 such that $A_1 P_2(2:n,2)$ has the largest 2-norms among the columns $A_1(2:n,k), k = 2,\dots,m$.
[Costs 2(n-1)(m-1) + m-1 flops]

Then choose an $n \times n$ reflector $Q_2 = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{Q}_2 \end{bmatrix}$ where $\tilde{Q}_2$ is an $n-1 \times n-1$ reflector such that

$$Q_2 Q_1 A P_1 P_2 = Q_2 A_1 P_2 = \underbrace{\begin{bmatrix} \pm\|A(:,1)\|_2 & a_{12}^{(1)} & a_{13}^{(1)} & \cdots & a_{1m}^{(1)} \\ 0 & \pm\|A_1 P_2(2:n,2)\|_2 & a_{23}^{(2)} & \cdots & a_{2m}^{(2)} \\ 0 & 0 & a_{33}^{(2)} & \cdots & a_{3m}^{(2)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & a_{n3}^{(2)} & \cdots & a_{nm}^{(2)} \end{bmatrix}}_{=:A_2}$$

[Costs 4(n-1)(m-2) flops]

Computing rank revealing QR

Continuing in this way,

$$Q_r Q_{r-1} \cdots Q_1 A P_1 P_2 \cdots P_r = \begin{bmatrix} R_1 & R_2 \\ 0 & 0 \end{bmatrix}$$

where R_1 is $r \times r$ an upper triangular and R_2 is an $r \times (m-r)$ matrix.

Computing rank revealing QR

Continuing in this way,

$Q^T A P = R$

$\Rightarrow AP = QR$

$$\underbrace{Q_r Q_{r-1} \cdots Q_1}_{=:Q^T} A \underbrace{P_1 P_2 \cdots P_r}_{P} = \underbrace{\begin{bmatrix} R_1 & R_2 \\ 0 & 0 \end{bmatrix}}_{R}$$

where R_1 is $r \times r$ an upper triangular and R_2 is an $r \times (m-r)$ matrix.

The total flop count is

$$\underbrace{\Sigma_{k=1}^{r}(2n-2k+3)(m-k+1)}_{\text{pivoting}} + \underbrace{4\Sigma_{k=1}^{r}(n-k+1)(m-k)}_{\text{applying the reflectors}}$$

Computing rank revealing QR

Continuing in this way,

$$Q_r Q_{r-1} \cdots Q_1 A P_1 P_2 \cdots P_r = \begin{bmatrix} R_1 & R_2 \\ 0 & 0 \end{bmatrix}$$

where R_1 is $r \times r$ an upper triangular and R_2 is an $r \times (m - r)$ matrix.

The total flop count is

$$\underbrace{\Sigma_{k=1}^{r}(2n - 2k + 3)(m - k + 1)}_{\text{pivoting}} + \underbrace{4\Sigma_{k=1}^{r}(n - k + 1)(m - k)}_{\text{applying the reflectors}}$$

The cost of pivoting is comparable in order to the cost of applying the reflectors. If $r = m$, then the total cost is more than finding the R of the QR decomposition without pivoting.

Computing rank revealing QR

The following strategy can reduce the cost of pivoting.

Find the norms of the columns $A_1(2:n,k)$ for $k=2,\ldots,m$ by noticing that

$$\|A_1(2:n,k)\|_2^2 = \begin{cases} \|A(:,k)\|_2^2 - |A(1,k)|^2 & \text{if } P_1(:,k) = e_k \\ \|A(:,1)\|_2^2 - |A(1,1)|^2 & \text{otherwise.} \end{cases}$$

[This costs 2(m-1) flops.]

Computing rank revealing QR

The following strategy can reduce the cost of pivoting.

Find the norms of the columns $A_1(2:n,k)$ for $k=2,\ldots,m$ by noticing that

$$\|A_1(2:n,k)\|_2^2 = \begin{cases} \|A(:,k)\|_2^2 - |A(1,k)|^2 & \text{if } P_1(:,k)=e_k \\ \|A(:,1)\|_2^2 - |A(1,1)|^2 & \text{otherwise.} \end{cases}$$

[This costs 2(m-1) flops.]

The remaining norms of columns may be calculated in a similar way. This reduces the cost of pivoting to $2nm +$ $O(m^2)$ flops and the cost of finding the R and the P of the rank revealing QR decomposition $AP=QR$ of an $n\times m$ matrix A becomes $2nm^2 - \frac{2}{3}m^3 + O(nm) + O(m^2)$ flops.

Exercise: Show that in the QR decomposition with column pivoting of any $A\in\mathbb{F}^{n\times m}$, with $\mathbb{F}=\mathbb{R}$ or $\mathbb{C}$, and $n>m$, the diagonal entries of the $r\times r$ upper triangular matrix R_1 are real numbers arranged in decreasing order of magnitude, i.e., $|R(1,1)|\geq\cdots\geq|R(r,r)|$.

Computing rank revealing QR: numerical issues

In practice, the rank r of A will not be known and if $r < m$, then at some stage $p < m$, the computed R is of the form

$$\begin{bmatrix} R_1 & R_2 \\ 0 & R_3 \end{bmatrix}$$

where R_1 is upper triangular of size $p \times p$ and R_3 is an $n - p \times m - p$ matrix with entries that have very small absolute values.

Computing rank revealing QR: numerical issues

In practice, the rank r of A will not be known and if $r < m$, then at some stage $p < m$, the computed R is of the form

$$\begin{bmatrix} R_1 & R_2 \\ 0 & R_3 \end{bmatrix}$$

where R_1 is upper triangular of size $p \times p$ and R_3 is an $n - p \times m - p$ matrix with entries that have very small absolute values.

Here p may or may not be equal to the exact rank of A.

Computing rank revealing QR: numerical issues

In practice, the rank r of A will not be known and if $r < m$, then at some stage $p < m$, the computed R is of the form

$$\begin{bmatrix} R_1 & R_2 \\ 0 & R_3 \end{bmatrix}$$

where R_1 is upper triangular of size $p \times p$ and R_3 is an $n - p \times m - p$ matrix with entries that have very small absolute values.

Here p may or may not be equal to the exact rank of A.

In such a case if all entries of R_3 have absolute values less than a predetermined tolerance level, then those may be set to 0 and the process is over. Else, the next column pivoted QR steps may be performed.

Computing rank revealing QR: numerical issues

In practice, the rank r of A will not be known and if $r < m$, then at some stage $p < m$, the computed R is of the form

$$\begin{bmatrix} R_1 & R_2 \\ 0 & R_3 \end{bmatrix}$$

where R_1 is upper triangular of size $p \times p$ and R_3 is an $n - p \times m - p$ matrix with entries that have very small absolute values.

Here p may or may not be equal to the exact rank of A.

In such a case if all entries of R_3 have absolute values less than a predetermined tolerance level, then those may be set to 0 and the process is over. Else, the next column pivoted QR steps may be performed.

The first nonzero rows of the resulting R are called the *numerical rank* of R.

Computing rank revealing QR: numerical issues

In practice, the rank r of A will not be known and if $r < m$, then at some stage $p < m$, the computed R is of the form

$$\begin{bmatrix} R_1 & R_2 \\ 0 & R_3 \end{bmatrix}$$

where R_1 is upper triangular of size $p \times p$ and R_3 is an $n - p \times m - p$ matrix with entries that have very small absolute values.

Here p may or may not be equal to the exact rank of A.

In such a case if all entries of R_3 have absolute values less than a predetermined tolerance level, then those may be set to 0 and the process is over. Else, the next column pivoted QR steps may be performed.

The first nonzero rows of the resulting R are called the *numerical rank* of R.

In MATLAB the default tolerance level is $\epsilon n |R(1,1)|$. This method of computing the numerical rank is however less reliable than the SVD method.