Triantafyllou Dimitrios

Professor
Department of Mathematics and Engineering Sciences
Hellenic Military Academy
GR-16673, Vari, Greece
e-mail: dtriant@math.uoa.gr, dtriant@sse.gr

2025

Householder Transformations

Definition

Let $u \in \mathbb{R}^n$ be a nonzero vector. A matrix of the form

$$H = I - 2\frac{uu^T}{u^T u}$$

is called Householder.

Properties

i. H is a symmetric matrix.

ii. H is orthogonal matrix, thus $H^T H = I$.

iii. $\|Hx\|_2 = \|x\|_2, \ \forall x \in \mathbb{R}^n$.

iv. $H^2 = I$.

v. $det(H) = -1$.

vi. Matrix H has an eigenvalue equal to -1 with multiplicity 1 and an eigenvalue equal to 1 with multiplicity $n-1$.

Householder Transformations

Lemma

Let x be a nonzero vector, $x \neq e_1$, where e_1 the first column of the identity matrix I. Then there exists a Householder matrix H such that Hx is a multiple of e_1. More precisely, $H = I - 2\frac{uu^T}{u^T u}$, with $u = x + sign(x_1)\|x\|_2 e_1$.

$$Hx = H \begin{pmatrix} * \\ * \\ \vdots \\ * \end{pmatrix} = \begin{pmatrix} \sigma \\ 0 \\ \vdots \\ 0 \end{pmatrix},$$

where $\sigma \in \mathbb{R}^*$.

Algorithm Householder: Creating zeros in a vector

Input: A nonzero vector $x \in \mathbb{R}^n$.

Output: A vector which is multiple of e_1.

Step 1: $m = max_{1 \le i \le n} |x_i|$
Step 2: $x_i \equiv u_i = \frac{x_i}{m}$
Step 3: $\sigma = sign(u_1)\sqrt{u_1^2 + u_2^2 + \ldots + u_n^2}$
Step 4: $x_1 \equiv u_1 = u_1 + \sigma$
Step 5: $\sigma = -m \cdot \sigma$

Complexity

The required flops for the previous algorithm are $2(n+1)$.

QR factorization with Column Pivoting (QRCP)

Theorem

Let $A \in \mathbb{R}^{m\times n}$, $m \geqslant n$. There exists an orthogonal matrix $Q \in \mathbb{R}^{m\times m}$, a permutation matrix $P \in \mathbb{Z}^{n\times n}$ an upper triangular matrix $R_{11} \in \mathbb{R}^{r\times r}$ with $r \leqslant n$ and a matrix $R_{12} \in \mathbb{R}^{r\times(n-r)}$ such that

$$AP = QR \text{ or } Q^T AP = R = \begin{bmatrix} R_{11} & R_{12} \\ \mathbb{O}_{21} & \mathbb{O}_{22} \end{bmatrix}$$

where the matrix Q is the product of Householder matrices H_i, $i = 1, 2, \ldots, r$: $Q = H_1 H_2 \ldots H_r$ and $\mathbb{O}_{21}$, $\mathbb{O}_{22}$ are zero matrices of size $(m-r)\times r$ and $(m-r)\times(n-r)$ respectively.

Product of Householder matrix H with a matrix A

Let $H = I - 2\frac{uu^T}{u^T u} \in \mathbb{R}^{m\times m}$, $u \in \mathbb{R}^{m\times 1}$ be a Householder matrix and vector respectively, and $A \in \mathbb{R}^{m\times n}$. The QR factorization requires the products of Householder matrices with A. If we perform the matrix-matrix product with the classical way, the floating point operations are of order of $O(m^2 n)$. Because of the special structure of H, this computation can be evaluated in less flops, as follows.

Let $\beta = \frac{2}{u^T u}$. Then the (i, j)-th entry of $A - HA = (A - \beta uu^T A)$ is equal to

$$a_{ij} = \beta(u_1 a_{1j} + u_2 a_{2j} + \ldots + u_m a_{mj})u_i, \quad i = 1, \ldots, m, \quad j = 1, \ldots, n.$$

Algorithm: Product of Householder matrix H with a matrix A

Input: Householder matrix $H \in \mathbb{R}^{m\times m}$ and a matrix $A \in \mathbb{R}^{m\times n}$.

Output: $A - HA$.

```
for j = 1, 2, ..., n
    c = 0
    for k = 1, 2, ..., m
        c = c + u_k · a_{k,j}
    end for
    c = β · c
    for i = 1, 2, ..., m
        a_{i,j} = c · u_i
    end for
end for
```

Complexity

The required flops for the previous algorithm are $2mn = O(mn)$, significantly less than those of the classical way.

QRCP factorization

Let $A \in \mathbb{R}^{m\times n}$, $m > n$.
Step 1:

Find column of A having max norm, let c be the one, and permute columns 1 and c of A : $A \leftarrow AP_1$. Use Householder algorithm to construct a Householder matrix H_1 such that the first column of H_1A is a multiple of e_1: $A^{(1)} = H_1AP_1 = \begin{pmatrix} * & * & \dots & * \\ 0 & * & \dots & * \\ \vdots & \vdots & \ddots & \vdots \\ 0 & * & \dots & * \end{pmatrix}$, where

$$H_1 = I_m - 2\frac{u_m u_m^T}{u_m^T u_m} \text{ and } H_1 \begin{pmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{m1} \end{pmatrix} = \begin{pmatrix} * \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

$$A^{(1)} = H_1 \cdot A \cdot P_1 = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ 0 & a_{22}^{(1)} & \dots & a_{2n}^{(1)} \\ \vdots & \vdots & \vdots & \vdots \\ 0 & a_{m2}^{(1)} & \dots & a_{mn}^{(1)} \end{pmatrix}$$

QRCP factorization

Step 2:

Find column of $\hat{A}^{(1)}$ having max norm, where $\hat{A}^{(1)}$ is obtained from $A^{(1)}$ by deleting its 1st row and column. Let c be the one, and permute columns 1 and c of $\hat{A}^{(1)}$. Set $\tilde{A}^{(2)} \leftarrow A^{(1)}\hat{P}_2$. Use Householder algorithm to construct a Householder matrix H_2 such

that : $A^{(2)} = H_2 A^{(1)} P_2 = \begin{pmatrix} * & * & * & \dots & * \\ 0 & * & * & \dots & * \\ 0 & 0 & * & \dots & * \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & * & \dots & * \end{pmatrix}$,

where $\hat{H}_2 = I_{m-1} - 2\frac{u_{m-1}u_{m-1}^T}{u_{m-1}^T u_{m-1}}$, $\hat{H}_2 \begin{pmatrix} a_{22} \\ a_{32} \\ \vdots \\ a_{m2} \end{pmatrix} = \begin{pmatrix} * \\ 0 \\ \vdots \\ 0 \end{pmatrix}$

and $H_2 = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & \hat{H}_2 & \\ 0 & & & \end{pmatrix}$, $P_2 = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & \hat{P}_2 & \\ 0 & & & \end{pmatrix}$.

QRCP factorization

In order to save memory, overwrite A with $A^{(2)}$

$$A^{(2)} = H_2 \cdot A^{(1)} P_2 = \begin{pmatrix} a_{11} & a_{12} & a_{13} & \dots & a_{1n} \\ 0 & a_{22}^{(1)} & a_{23}^{(1)} & \dots & a_{2n}^{(1)} \\ 0 & 0 & a_{33}^{(2)} & \dots & a_{3n}^{(2)} \\ \vdots & \vdots & \vdots & \vdots & \\ 0 & 0 & a_{m3}^{(2)} & \dots & a_{mn}^{(2)} \end{pmatrix}$$

QRCP factorization

$\vdots$

Step r:

$$A^{(r)} = H_r A^{(r-1)} P_r = \left(\begin{array}{ccccc|ccc} * & * & \dots & * & * & * & \dots & * \\ 0 & * & \dots & * & * & * & \dots & * \\ \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & * & * & * & \dots & * \\ 0 & 0 & \dots & 0 & * & * & \dots & * \\ \hline 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & 0 & \dots & 0 \end{array}\right),$$

where $\hat{H}_r = I_{m-r+1} - 2\frac{u_{m-r+1}u_{m-r+1}^T}{u_{m-r+1}^T u_{m-r+1}}$, $\hat{H}_r \begin{pmatrix} a_{rn} \\ a_{r+1,n} \\ \vdots \\ a_{mn} \end{pmatrix} = \begin{pmatrix} * \\ 0 \\ \vdots \\ 0 \end{pmatrix}$

and $H_r = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & \hat{H}_{m-r+1} & \\ 0 & & & \end{pmatrix}$, $P_r = \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & & & \\ \vdots & & \hat{P}_{m-r+1} & \\ 0 & & & \end{pmatrix}$.

Home Page

Title Page

QRCP factorization

Thus,

$$R \equiv A^{(r)} = H_r \cdot A^{(r-1)} P_r = \left(\begin{array}{cc} R_{11} & R_{12} \\ \hline \mathbb{O}_{21} & \mathbb{O}_{22} \end{array} \right).$$

Concluding,

$R = A^{(r)} = H_r A^{(r-1)} P_r = H_r H_{r-1} A^{(r-2)} P_{r-1} P_r = \dots$

$= H_r H_{r-1} \dots H_1 A P_1 \dots P_{r-1} P_r.$

We set $Q^T = H_r H_{r-1} \dots H_1,\ P = P_1 P_2 \dots P_r$ and thus,

$$R = Q^T A P \Leftrightarrow AP = QR$$

Algorithm: QRCP factorization

for $j = 1, \ldots, n$

$\quad c_j = A^T_{1:m,j} A_{1:m,j}$

end

$r = 0;\ \ t = max\{c_1, \ldots, c_n\}$

while $t > 0$ and $r < n$

$\quad r = r + 1$

$\quad$ **Find** smallest $k,\ \ r \le k \le n\ \ : c_k = t$

$\quad piv(r) = k;\ \ A_{1:m,r} \leftrightarrow A_{1:m,k};\ \ c_r \leftrightarrow c_k$

$\quad [u, \beta] =$ **house**$(A_{r:m,r})$

$\quad A_{r:m,r:n} = (I_{m-r+1} - \beta u u^T) A_{r:m,r:n}$

$\quad A_{r+1:m,r} = u_{2:m-r+1}$

$\quad$ **for** $\ i = r + 1 : n$

$\qquad c_i = c_i - A^2_{r,i}$

$\quad$ **end**

$\quad t = max\{c_{r+1}, \ldots, c_n\}$

end

Algorithm: QRCP factorization

Remark

The column norms do not have to be computed at each step because of the property:

$$Q^T z = \begin{bmatrix} a \\ w \end{bmatrix} \begin{array}{c} 1 \\ k-1 \end{array} \Rightarrow \left\|Q^T z\right\|_2 = \left\| \begin{bmatrix} a \\ w \end{bmatrix} \right\|_2$$

$$\Rightarrow \left\|Q^T\right\|_2^2 \|z\|_2^2 = a^2 + \|w\|_2^2 \Rightarrow \|w\|_2^2 = \|z\|_2^2 - a^2.$$

Numerical Rank

$$A^{(k)} = \begin{bmatrix} R_{11}^{(k)} & R_{12}^{(k)} \\ \mathbb{O}_{21}^{(k)} & R_{22}^{(k)} \end{bmatrix}$$

If $\left\|R_{22}^{(k)}\right\|_2 \leqslant \epsilon \|A\|_2$, with $\epsilon = O(u)$, where u the unit round off error, then the algorithm terminates and $Rank(A) = k$.

Algorithm: QRCP factorization

Complexity

The computational complexity for the QRCP factorization is:

$$O\left(2mnr - r^2(m+n) + \frac{2r^3}{3}\right) \text{ flops.}$$