Dense QR factorization and its error analysis

Laura Grigori

INRIA and LJLL, UPMC

February 2016

Plan

Plan

Direct methods of factorization

- QR factorization
- Block QR factorization

The QR factorization

Given a matrix $A \in \mathbb{R}^{m \times n}$, $m \geq n$, its QR factorization is

$$A = QR = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_1 \\ 0 \end{pmatrix} = Q_1 R_1$$

where $Q \in \mathbb{R}^{m \times m}$ is orthogonal and $R \in \mathbb{R}^{m \times n}$ is upper triangular.

If A has full rank, the factorization $Q_1 R_1$ is essentialy unique (modulo signs of diagonal elements of R).

- $A^T A = R_1^T R_1$ is a Cholesky factorization and $A = A R_1^{-1} R_1$ is a QR factorization.
- $A = Q_1 D \cdot D R_1$, $D = diag(\pm 1)$ is a QR factorization.

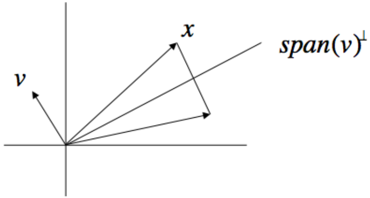
Householder transformation

The Householder matrix

$$P = I - \frac{2}{v^T v} v v^T$$

has the following properties:

- is symmetric and orthogonal, $P^2 = I$,
- is independent of the scaling of v,
- it reflects x about the hyperplane $span(v)^{\perp}$



$$Px = x - \frac{2v^T x}{v^T v} v = x - \alpha v$$

Presentation of Householder transformations and stability analysis from [N.J.Higham, 2002].

Householder for the QR factorization

We look for a Householder matrix that allows to annihilate the elements of a vector x, except first one.

$$Px = y, \quad \|x\|_2 = \|y\|_2, \quad y = \sigma e_1, \quad \sigma = \pm\|x\|_2$$

With the choice of sign made to avoid cancellation when computing $v_1 = x_1 - \sigma$, we have

$$\begin{aligned} v &= x - y = x - \sigma e_1, \\ \sigma &= -sign(x_1)\|x\|_2, v = x - \sigma e_1, \\ P &= I - \beta v v^T, \beta = \frac{2}{v^T v} \end{aligned}$$

Householder based QR factorization

$$A = \begin{pmatrix} x & x & x \\ x & x & x \\ x & x & x \end{pmatrix} = P_1 \begin{pmatrix} x & x & x \\ 0 & x & x \\ 0 & x & x \end{pmatrix} = P_1 \begin{pmatrix} 1 & \\ & \tilde{P}_2 \end{pmatrix} \begin{pmatrix} x & x & x \\ 0 & x & x \\ 0 & 0 & x \end{pmatrix} = R$$

So we have

$$\begin{aligned} Q^T A &= P_n P_{n-1} \dots P_1 A = R, \\ Q &= (I - \beta_1 v_1 v_1^T) \dots (I - \beta_{n-1} v_{n-1} v_{n-1}^T)(I - \beta_n v_n v_n^T) \end{aligned}$$

$\#flops = 2n^2(m - n/3)$

Error analysis of Householder transformations

Lemma (Lemma 19.1 in [N.J.Higham, 2002])

Consider the computation of $P = I - \beta v v^T$, *where* $Px = \sigma e_1$, $v \in \mathbb{R}^m$, *as*

1: $v = x$
2: $s = sign(x_1)\|x\|_2, \quad \%\sigma = -s$
3: $v_1 = v_1 + s$
4: $\beta = 1/(s v_1)$

Then we have

$$\begin{aligned} \hat{v}(2:n) &= v(2:n) \\ \hat{\beta} &= \beta(1+\tilde{\theta}_m), \quad \hat{v}_1 = v_1(1+\tilde{\theta}_m), \ \textit{where } |\tilde{\theta}_m| \le \tilde{\gamma}_m \end{aligned}$$

Proof based on the fact that $fl(x^T x) = (1+\theta_m) x^T x$. The result can be re-written as

$$\hat{v} = v + \Delta v, \quad |\Delta v| \le \tilde{\gamma}_m |v|$$

In the following results, $v = \sqrt{\beta} v$, $\beta = 1$, and so $\|v\|_2 = \sqrt{2}$.

Error analysis of Householder transformations

Lemma (Lemma 19.2 in [N.J.Higham, 2002])

Consider the computation $y = \hat{P}b = (I - \hat{v}\hat{v}^T)b$*, where* $b, \hat{v} \in \mathbb{R}^m$*. Then*

$$\hat{y} = (P + \Delta P)b, \quad \|\Delta P\|_F \le \tilde{\gamma}_m. \tag{1}$$

Proof.

$$\begin{aligned} \hat{w} &= fl(\hat{v}(\hat{v}^T b)) = (\hat{v} + \Delta\hat{v})(\hat{v}^T(b + \Delta b)), \quad |\Delta\hat{v}| \le u|\hat{v}| \text{ and } |\Delta b| \le \gamma_m |b| \\ &= (v + \Delta v + \Delta\hat{v})(v + \Delta v)^T(b + \Delta b) \end{aligned}$$

Hence

$$\hat{w} = v(v^T b) + \Delta w, \quad \text{where } |\Delta w| \le \tilde{\gamma}_m |v||v^T||b|$$

□

Error analysis of Householder transformations

Continued proof of the previos lemma. We obtain

$$\hat{y} = fl(b - \hat{w}) = b - v(v^T b) - \Delta w + \Delta y_1, \qquad |\Delta y_1| \leq u|b - \hat{w}|$$

Since

$$|-\Delta w + \Delta y_1| \leq u|b| + \tilde{\gamma}_m |v||v^T||b|$$

we obtain

$$\hat{y} = Pb + \Delta y, \qquad \|\Delta y\|_2 \leq \tilde{\gamma}_m \|b\|_2$$

Finally, with $\Delta P = \Delta y b^T / b^T b$, we have

$$\hat{y} = (P + \Delta P)b, \qquad \|\Delta P\|_F = \|\Delta y\|_2 / \|b\|_2 \leq \tilde{\gamma}_m$$

Error analysis of a sequence of transformations

Lemma ([N.J.Higham, 2002])

Let $Q = P_r P_{r-1} \dots P_1$ *and let* $A_{r+1} = Q^T A$, $A \in \mathbb{R}^{m \times n}$. *We have*

$$\hat{A}_{r+1} = Q^T(A + \Delta A), \quad \|\Delta a_j\|_2 \leq r\tilde{\gamma}_m \|a_j\|_2, \; j = 1:n$$

Sketch of the proof: Let a_j be the j-th column of A.

$$\hat{a}_j^{(r+1)} = (P_r + \Delta P_r) \dots (P_1 + \Delta P_1) a_j, \quad \|\Delta P_k\|_F \leq \tilde{\gamma}_m, \; k = 1:r$$

We obtain

$$\begin{aligned} \hat{a}_j^{(r+1)} &= Q^T(a_j + \Delta a_j), \\ \|\Delta a_j\|_2 &\leq ((1 + \tilde{\gamma}_m)^r - 1)\|a_j\|_2 \leq \frac{r\tilde{\gamma}_m}{1 - r\tilde{\gamma}_m}\|a_j\|_2 = r\tilde{\gamma}_m^{'}\|a_j\|_2 \end{aligned}$$

Error analysis of the QR factorization

The following result follows

Theorem ([N.J.Higham, 2002])

Let $\hat{R} \in \mathbb{R}^{m \times n}$ be the computed factor of $A \in \mathbb{R}^{m \times n}$ obtained by using Householder transformations. Then there is an orthogonal $Q \in \mathbb{R}^{m \times m}$ such that

$$A + \Delta A = Q\hat{R}, \text{ where } \|\Delta a_j\|_2 \leq \tilde{\gamma}_{mn} \|a_j\|_2, \quad j = 1 : n$$

Householder-QR factorization

Require: $A \in \mathbb{R}^{m \times n}$

Let $R \in \mathbb{R}^{n \times n}$ be initialized with zero matrix
for k = 1 to n **do**
    ▷ Compute Householder matrix $P_k = I - \beta_k v_k v_k^T$ s.t. $P_k A(k:m,k) = \pm\|A(k:m,k)\|_2 e_1$. Store v_k in $Y()$ and β_k in $\mathcal{T}(k)$
    $R(k,k) = -sgn(A(k,k)) \cdot \|A(k:m,k)\|_2$
    $\mathcal{T}(k) = \frac{R(k,k)-A(k,k)}{R(k,k)}$
    $Y(k+1:m,k) = \frac{1}{R(k,k)-A(k,k)} \cdot A(k+1:m,k)$
    ▷ Update trailing matrix
    $A(k:m,k+1:n) = (I - Y(k+1:m,k)\mathcal{T}(k)Y(k+1:m,k)^T) \cdot A(k:m,k+1:n)$
    $R(k,k+1:n) = A(k,k+1:n)$
end for

Assert: $A = QR$, where $Q = P_1 \dots P_n = (I - \beta_1 v_1 v_1^T)\dots(I - \beta_n v_n v_n^T)$, the Householder vectors v_k are stored in Y and $\mathcal{T}$ is an array of size n.

Computational complexity

- Flops per iterations
  - Dot product $w = v_k^T A(k:m, k+1:n)$: $2(m-k)(n-k)$
  - Outer product $v_k w$: $(m-k)(n-k)$
  - Subtraction $A(k:m, k+1:n) - \ldots$: $(m-k)(n-k)$
- Flops of Householder-QR

$$\sum_{k=1}^{n} 4(m-k)(n-k) = 4\sum_{k=1}^{n}(mn - k(m+n) + k^2)$$
$$\approx 4mn^2 - 4(m+n)n^2/2 + 4n^3/3 = 2mn^2 - 2n^3/3$$

Algebra of block QR

Storage efficient representation for Q [Schreiber and Loan, 1989]

$$Q = Q_1 Q_2 \dots Q_k = (I - \beta_1 v_1 v_1^T) \dots (I - \beta_k v_k v_k^T) = I - YTY^T$$

Example for $k = 2$

$$Y = (v_1 | v_2), \quad T = \begin{pmatrix} \beta_1 & -\beta_1 v_1^T v_2 \beta_2 \\ 0 & \beta_2 \end{pmatrix}$$

Example for combining two compact representations

$$\begin{aligned} Q &= (I - Y_1 T_1 Y_1^T)(I - Y_2 T_2 Y_2^T) \\ T &= \begin{pmatrix} T_1 & -T_1 Y_1^T Y_2 T_2 \\ 0 & T_2 \end{pmatrix} \end{aligned}$$

Block algorithm for computing the QR factorization

Partitioning of matrix A of size $m \times n$

$$A = \left[\begin{array}{cc} A_{11} & A_{12} \\ A_{21} & A_{22} \end{array} \right]$$

where A_{11} is of size $b \times b$, A_{21} is of size $(m-b) \times b$, A_{12} is of size $b \times (n-b)$ and A_{22} is of size $(m-b) \times (n-b)$.

Block QR algebra

The first step of the block QR factorization algorithm computes:

$$Q_1^T A = \begin{pmatrix} R_{11} & R_{12} \\ & A^1 \end{pmatrix}$$

The algorithm continues recursively on the trailing matrix A^1.

Algebra of block QR factorization

$$A = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix} = Q_1 \begin{pmatrix} R_{11} & R_{12} \\ & A^1 \end{pmatrix}$$

Block QR algebra

1. Compute the factorization

$$\begin{pmatrix} A_{11} \\ A_{12} \end{pmatrix} = Q_1 R_{11}$$

2. Compute the compact representation $Q_1 = I - YTY^T$
3. Apply Q_1^T on the trailing matrix

$$(I - YT^TY^T) \begin{pmatrix} A_{12} \\ A_{22} \end{pmatrix} = \begin{pmatrix} A_{12} \\ A_{22} \end{pmatrix} - Y \left(T^T \left(Y^T \begin{pmatrix} A_{12} \\ A_{22} \end{pmatrix} \right) \right)$$

4. The algorithm continues recursively on the trailing matrix A^1.

Parallel implementation of the QR factorization

QR factorization on a $P = P_r \times P_c$ grid of processors

For ib = 1 to n-1 step b

1. Compute panel factorization on P_r processors

$$\begin{pmatrix} A_{11} \\ A_{12} \end{pmatrix} = Q_1 R_{11} = (I - YTY^T) R_{11}$$

2. The P_r processors broadcast along the rows their parts of Y and T
3. Apply Q_1^T on the trailing matrix:
   - All processors compute their local part of

     $$W_l = Y_l^T (A_{21l}; A_{22l})$$

   - The processors owning block row *ib* compute the sum over W_l, that is

     $$W = Y^T (A_{21}; A_{22})$$

     and then compute $W' = T^T W$
   - The processors owning block row *ib* broadcast along the columns their part of W'
4. All processors compute

$$(A^1_{21}; A^1_{22}) = (A_{21}; A_{22}) - (A_{21}; A_{22}) * W'$$

Cost of parallel QR factorization

$$
\begin{aligned}
& \gamma \cdot \left(\frac{6mnb - 3n^2 b}{2p_r} + \frac{n^2 b}{2p_c} + \frac{2mn^2 - 2n^3/3}{p} \right) \\
+ \quad & \beta \cdot \left(nb \log p_r + \frac{2mn - n^2}{p_r} + \frac{n^2}{p_c} \right) \\
+ \quad & \alpha \cdot \left(2n \log p_r + \frac{2n}{b} \log p_c \right).
\end{aligned}
$$

Solving least squares problems

Given matrix $A \in \mathbb{R}^{m\times n}$, $rank(A) = n$, vector $b \in \mathbb{R}^{m\times 1}$,
the unique solution to $\min_x \|Ax - b\|_2$ is

$$x = A^+ b, \qquad A^+ = (A^T A)^{-1} A^T$$

Using the QR factorization of A

$$A = QR = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_1 \\ 0 \end{pmatrix} \tag{2}$$

We obtain

$$\begin{aligned} ||r||_2^2 &= ||b - Ax||_2^2 = ||b - \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_1 \\ 0 \end{pmatrix} x||_2^2 \\ &= || \begin{pmatrix} Q_1^T \\ Q_2^T \end{pmatrix} b - \begin{pmatrix} R_1 \\ 0 \end{pmatrix} x||_2^2 = || \begin{pmatrix} Q_1^T b - R_1 x \\ Q_2^T b \end{pmatrix} ||_2^2 \\ &= ||Q_1^T b - R_1 x||_2^2 + ||Q_2^T b||_2^2 \end{aligned}$$

Solve $R_1 x = Q_1^T b$ to minimize $||r||_2$.

Acknowledgement

- Stability analysis results presented from [N.J.Higham, 2002]
- Some of the examples taken from [Golub and Van Loan, 1996]

References (1)

Golub, G. H. and Van Loan, C. F. (1996).
Matrix Computations (3rd Ed.).
Johns Hopkins University Press, Baltimore, MD, USA.

N.J.Higham (2002).
Accuracy and Stability of Numerical Algorithms.
SIAM, second edition.

Schreiber, R. and Loan, C. V. (1989).
A storage efficient *WY* representation for products of Householder transformations.
SIAM J. Sci. Stat. Comput., 10(1):53–57.