QR and least square

Zecheng Zhang

March 13, 2023

1 QR factorization

We study $A \in \mathbb{R}^{m\times n}$ matrix with linearly independent columns. QR algorithm is a key algorithm in numerical linear algebra. We want to study the column space of A.

Recall the Gram–Schmidt process for producing an orthogonal or an orthonormal basis for any nonzero subspace of $\mathbb{R}^n$. Given a basis $\{x_1, ..., x_p\}$ for a nonzero subspace W, define

$$
\begin{aligned}
q_1 &= a_1/r_{11} \\
q_2 &= a_2/r_{22} - \frac{r_{12}}{r_{22}} q_1 \\
q_3 &= a_3/r_{33} - \frac{r_{13}}{r_{33}} q_1 - \frac{r_{23}}{r_{33}} q_2 \\
&\cdots \\
q_p &= a_p/r_{pp} - \frac{r_{1p}}{r_{pp}} a_1 - \frac{r_{2p}}{r_{pp}} q_2 - \frac{r_{(p-1)p}}{r_{pp}} q_{p-1},
\end{aligned}
$$

where $r_{ij} = q_i^T a_j$ and $r_{jj} = \|a_j - \sum_{i=1}^{j} r_{ij} q_i\|$. Then $\{q_1, ..., q_p\}$ is an orthonormal basis for W, i.e., $span\{a_1, a_2, ..., a_p\} = span\{q_1, q_2, ..., q_p\}$.

Theorem 1.1. If A is an $m \times n$ matrix with linearly independent columns, then A can be factored as $A = QR$, where Q is an $m \times n$ matrix whose columns form an orthonormal basis for Col A and R is an $n \times n$ upper triangular invertible matrix with positive entries on its diagonal.

Proof. Let $a_1, ..., a_n$ be columns of A. Perform Gram-Schmidt, we obtain $Q = [q_1, ..., q_n]$, which is an orthonormal set of vectors whose span is $col(A)$. For a_k, a_k is in $span\{a_1, ..., a_k\} = span\{q_1, ..., q_k\}$. That is there exists $r_{1k}, ..., r_{kk}$ such that $a_k = r_{1k}q_1 + ... + r_{kk}q_k + 0q_{k+1}...0q_n$. Without loss of generality, we assume $r_{kk} > 0$, otherwise multiply r_{kk} and q_k by -1 simultaneously. Denote $Q = [q_1, q_2, ..., q_n]$, $R = [r_1, ..., r_n]$ where $r_k = [r_{1k}, ..., r_{kk}, 0, ..., 0]^t \in \mathbb{R}^n$, recall the matrix multiplication we have $A = QR$. We now claim that R is upper triangular with a positive diagonal (easy to verify) and invertible. Recall $rank(QR) \leq min(rank(Q), rank(R))$. Since $rank(A) = n = rank(Q)$, this implies that $rank(R) = n$. □

When $m > n$, we can append $m - n$ columns to Q to make it a $m \times m$ unitary matrix $\tilde{Q}$. In this process, we will append $m - n$ 0 rows to R to obtain $\tilde{R}$. We call $A = \tilde{Q}\tilde{R}$ full QR of A.

2 Modified QR

The GS-QR algorithm is not numerically stable. For the moment, a stable algorithm is one that is not too sensitive to the effects of rounding off errors. The modified GS is the way to improve

Mar 13 (MON)

$A \in \mathbb{R}^{m \cdot n}$, $\quad \text{rank}(A) = n = \#\text{cols of } A$.

$A = QR$, $\quad Q$ is unitary

R is upper triangular

invertible

all diagonal entries of R are non-negative.

projector.

$$\begin{cases} q_1 = a_1 / r_{11} = \dfrac{I a_1}{\| I a_1 \|} \\ q_2 = a_2 / r_{22} - \dfrac{r_{12}}{r_{22}} q_1 = \dfrac{a_2 - \text{Proj}_{q_1} a_2}{\| \cdots \|} = \dfrac{\boxed{P_2} a_2}{\| P_2 a_2 \|} \\ q_p = a_p / r_{pp} - \dfrac{r_{1p}}{r_{pp}} q_1 - \cdots - \dfrac{r_{1(p-1)}}{r_{pp}} q_{p-1} = a_p - \text{Proj}_{\text{span}(q_1 \ldots q_{p-1})} a_p \end{cases}$$

$$= \frac{P_j a_j}{\| P_j a_j \|}$$

$r_{ij} = \langle q_i, a_j \rangle \qquad r_{jj} = \| a_j - \sum_{i=1}^{j-1} r_{ij} q_i \|$

$Q = [q_1 \cdots q_n]$, $\quad R = [r_{ij}]_{ij}$.

all q_i are orthonormal

Pseudo - Code of GS for QR

$$
\boxed{
\begin{array}{l}
\text{for } j = 1, \dots, n. \\
\quad v_j = a_j \\
\quad \text{for } i = 1, \dots, j-1 \\
\qquad r_{ij} = \langle q_i, a_j \rangle \\
\qquad v_j = v_j - r_{ij} q_i \\
\quad r_{jj} = \| v_j \|_2 \\
\quad q_j = v_j / r_{jj}
\end{array}
}
$$

Thm (w/o proof)

$$P_j = \underbrace{P_{\perp q_{j-1}}}_{\downarrow} \cdots\cdots \underbrace{P_{\perp q_1}}_{\downarrow} \underbrace{P_{\perp q_0}}_{I}$$

$$(I - q_{j-1} q_{j-1}^t) \qquad (I - q_1 q_1^+)$$

jth step

G.S.

$$P_j a_j = I a_j - \mathrm{Proj}_{\{q_1 \cdots q_{j-1}\}} a_j = (I - Q) a_j$$

$$\mathrm{rank}(Q) + \mathrm{rank}(P_j) = m \quad \Rightarrow \mathrm{rank}(P_j) = m - (j-1)$$

at jth step,

$$q_j = \frac{P_j a_j}{\|P_j a_j\|} = \frac{P_{\perp q_{j-1}} P_{\perp q_{j-2}} \cdots P_{\perp q_1} \boxed{P_{\perp q_0}}^{\;= I}}{\| \cdots \cdots \cdots \|} a_j$$

$$v_j^1 = a_j = P_{\perp q_0} a_j = I a_j = a_j$$

$$v_j^2 = P_{\perp q_1} \underbrace{P_{\perp q_0} a_j} = P_{\perp q_1} v_j^1 = (I - q_1 q_1^t) v_j^1$$

$$v_j^3 = P_{\perp q_2} P_{\perp q_1} P_{\perp q_0} a_j = P_{\perp q_2} v_j^2 = (I - q_2 q_2^t) v_j^2$$

output w/o normalization $\longleftarrow$ $v_j = P_{\perp q_{j-1}} v_j^{j-1} = (I - q_{j-1} q_{j-1}^t) v_j^{j-1}$

---

at the 3rd step

$$v_3^1 = a_3$$

$$v_3^2 = (I - q_1 q_1^t) v_3^1$$

$$v_3 = (I - q_2 q_2^t) v_3^2$$

at the 4th step

$$v_4^1 = a_4$$

$$v_4^2 = (I - q_1 q_1^t) v_4^1$$

$$v_4^3 = (I - q_2 q_2^t) v_4^2$$

$$v_4 = (I - q_3 q_3^t) v_4^3$$

Modified G.S.

for $i = 1, \ldots, n$

$\quad v_j = a_j$

for $i = 1, \ldots, n$

$\quad r_{ii} = \|v_i\|$

$\quad q_i = \dfrac{v_i}{r_{ii}}$

$\quad$ for $j = i+1, \ldots, n$

$\quad\quad r_{ij} = q_i^{+} v_j$

$\quad\quad v_j = v_j - r_{ij} q_i$

Each "$+$", "$-$", "$\times$", "$\div$", "$\sqrt{\ }$" will be counted as one flop.

Operation count:

all flops of an algorithm.

$a \in \mathbb{R}^m, \quad b \in \mathbb{R}^m$

$$a \cdot b = a_1 b_1 + a_2 b_2 + \cdots + a_m b_m \simeq 2m - 1$$

m "$\times$" $+$ $(m-1)$ "$+$"

$a \cdot b$ operation count u $2m$

$$\lim_{m \to \infty} \frac{2m-1}{2m} = 1 \simeq \lim_{m \to \infty} \frac{2m}{2m} = 1$$

$$\|a\| = \sqrt{a_1^2 + a_2^2 + \ldots + a_m^2}$$

$$\left.\begin{array}{ll} m & \text{" } \times \text{ "} \\ m-1 & \text{" } + \text{ "} \\ 1 & \text{" } \sqrt{\ } \text{ "} \end{array}\right\} \text{ 2m flops , operation count } \|\ \| \simeq 2m.$$

Algorithm 1: Gram Schmidt

Data: $n \geq 0$

for $j = 1$ *to* n **do**
  $v_j = a_j$
  for $i = 1$ *to* $j - 1$ **do**
    $r_{ij} = q_i^t a_j$
    $v_j = v_j - r_{ij} q_i$
  $r_{jj} = \|v_j\|_2$
  $q_j = v_j / r_{jj}$

the stability of the QR algorithm. GS can be expressed as an orthogonal projection:

$$q_1 = \frac{P_1 a_1}{\|P_1 a_1\|}, q_2 = \frac{P_2 a_2}{\|P_2 a_2\|}, ..., q_n = \frac{P_n a_n}{\|P_n a_n\|}, \tag{1}$$

where $P_j \in \mathbb{R}^{m \times m}$ denotes the orthogonal projector onto space spanned by $\{q_1, ...q_{j-1}\}$.

For each j, the GS algorithm computes a single orthogonal projection of rank $m - (j - 1)$, $v_j = P_j a_j$. Recall that: $P_{\perp q}$ denotes the rank $m - 1$ orthogonal projection onto the space orthogonal to q. By the definition of P_j, we can verify (without proof here):

$$P_j = P_{\perp q_{j-1}} ... P_{\perp q_2} P_{\perp q_1}, \tag{2}$$

and $P_{\perp q_1} = I$. As a result,

$$v_j = P_j a_j = P_{\perp q_{j-1}} ... P_{\perp q_2} P_{\perp q_1} a_j. \tag{3}$$

Specifically,

$$\begin{aligned}
v_j^1 &= a_j, \\
v_j^2 &= P_{\perp q_1} v_j^1 = v_j^1 - q_1 q_1^t v_j^1, \\
v_j^3 &= P_{\perp q_2} v_j^2 = v_j^2 - q_2 q_2^t v_j^2, \\
&\cdots \quad \cdots \\
v_j &= P_{\perp q_{j-1}} v_j^{j-1} = v_j^{j-1} - q_{j-1} q_{j-1}^t v_j^{j-1}.
\end{aligned}$$

We summarize the algorithm in 2.

Algorithm 2: Modified Gram Schmidt

for $i = 1$ *to* n **do**
  $v_i = a_i$
for $i = 1$ *to* n **do**
  $r_{ii} = \|v_i\|$
  $q_i = v_i / r_{ii}$
  for $j = i + 1$ *to* n **do**
    $r_{ij} = q_i^t v_j$
    $v_j = v_j - r_{ij} q_i$

2.1 Operation counts

Each addition, subtraction, multiplication, division and square root counts as one flop. Operation count is the number of flops an algorithm requires.

Theorem 2.1. The Gram-Schmidt algorithm requires $\sim 2mn^2$ flops for a matrix A of size $m \times n$.

Remark 1. The $\sim$ sign here is the asymptotic convergence, i.e.,

$$\lim_{m,n\to\infty} \frac{\text{the total number of flops}}{2mn^{2}} = \cancel{1}\ \text{const} \tag{4}$$

In discussing the operation count, it is standard to discard lower-order terms, since they are usually of little significance unless m and n are small.

Proof. In each i iteration, we have:

1. Line 7: m multiplication and $m-1$ addition.
2. Line 8: m multiplication and m subtraction.

In total we have $\sum_{i=1}^{n}\sum_{j=1}^{n}(4m-1)i \sim 2m^2n$. □

3 Housedolder triangularization

The target of the algorithm is to create a full QR of A. The idea is to applies a sequence of unitary matrices Q_k on the left of A such that, $Q_n ... Q_2 Q_1 A = R$ is upper triangular. Denote $Q = Q_1^t Q_2^t ... Q_n^t$, Q is also unitary. This implies that $A = QR$ is a full QR of A. We will discuss how to find Q_i.

3.1 Householder reflector

Each Q_k is chosen to introduce zeros below the diagonal in the $k-$th column while preserving all the zeros previously introduced. In general Q_k operates on rows $k, ..., m$. Each Q_k has the following format:

$$Q_k = \begin{bmatrix} I & 0 \\ 0 & F \end{bmatrix}, \tag{5}$$

where I is the identity matrix of the size $(k-1) \times (k-1)$ and F is unitray of size $(m-k+1)$. Multiplication by F will introduce zeros into $k-$th column. F is called a Householder reflector.

Suppose at the beginning of step k, the entries $k, ..., m$ of $k-$th column are given by the vector $x \in \mathbb{R}^{m-k+1}$. The Householder reflector F should introduce some zeros to x such that $Fx = [\|x\|, 0, ..., 0]^\intercal = \|x\|e_1$. The target now is to construct F such that F will map x to $\|x\|e_1$.

Let us define $v = x - \|x\|e_1$ (please check the picture). By the orthogonal projection formula, we have,

$$Px = (I - \frac{vv^t}{\|v\|^2})x = x - \frac{vv^t}{\|v\|^2}x. \tag{6}$$

This is the orthogonal projection of x onto space which is orthogonal to v. Move twice as far in the same direction; we will have the target vector, i.e.,

$$Fx = x - 2\frac{vv^t}{\|v\|^2}x = (I - 2\frac{vv^t}{\|v\|^2})x. \tag{7}$$

We now derive the Household projector: $F = I - 2\frac{vv^t}{\|v\|^2}$.