初等变换与 QR 分解

1.1 QR 分解

QR 分解将一个矩阵分解一个正交矩阵 (酉矩阵) 和一个三角矩阵的乘积. QR 分解被广泛应用于线性最小二乘问题的求解和矩阵特征值的计算.

1.1.1 QR 分解的存在唯一性

定理 1.1 (QR 分解) 设 $A \in \mathbb{C}^{m\times n}$ $(m \geq n)$. 则存在一个单位列正交矩阵 $Q \in \mathbb{C}^{m\times n}$ (即 $Q^*Q = I_{n\times n}$) 和一个上三角矩阵 $R \in \mathbb{C}^{n\times n}$, 使得

$$A = QR. \tag{1.1}$$

若 A 列满秩, 则存在一个具有正对角线元素的上三角矩阵 R 使得 (1.1) 成立, 且此时 QR 分解唯一, 即 Q 和 R 都唯一.

证明. 设 $A = [a_1, a_2, \ldots, a_n] \in \mathbb{C}^{m\times n}$. 若 A 列满秩, 即 $\mathrm{rank}(A) = n$. 则 QR 分解 (1.1) 就是对 A 的列向量组进行 Gram-Schmidt 正交化过程的矩阵描述 (见算法 1.1).

算法 1.1 Gram-Schmidt Process

```
r_11 = ||a_1||_2
q_1 = a_1/r_11
for j = 2 to n do
    q_j = a_j
    for i = 1 to j − 1 do
        r_ij = q_i^* a_j      % q_i^* 表示共轭转置
        q_j = q_j − r_ij q_i
    end for
    r_jj = ||q_j||_2
    q_j = q_j/r_jj
end for
```

☞ 如果 A 不是列满秩, 我们可以做类似的正交化过程, 具体实现可以参见相关资料.

下面证明满秩矩阵 QR 分解的存在唯一性.

存在性: 由于 A 列满秩, 由 Gram-Schmidt 正交化过程 (算法 1.1) 可知, 存在上三角矩阵 $R = [r_{ij}]_{n\times n}$ 满足 $r_{jj} > 0$, 使得 $A = QR$, 其中 Q 单位列正交.

唯一性: 假设 A 存在 QR 分解

$$A = Q_1R_1 = Q_2R_2,$$

其中 $Q_1, Q_2 \in \mathbb{C}^{m\times n}$ 单位列正交, $R_1, R_2 \in \mathbb{C}^{n\times n}$ 为具有正对角元素的上三角矩阵. 则有

$$Q_1 = Q_2 R_2 R_1^{-1}. \tag{1.2}$$

由于 R_1, R_2 均为上三角矩阵, 所以 $R_2R_1^{-1}$ 也是上三角矩阵, 且其对角线元素为 $R_2(i,i)/R_1(i,i)$, $i = 1, 2, \ldots, n$. 由 (1.2) 可得

$$1 = \|Q_1\|_2 = \|Q_2R_2R_1^{-1}\|_2 = \|R_2R_1^{-1}\|_2.$$

所以

$$\frac{R_2(i,i)}{R_1(i,i)} \le 1, \quad i = 1, 2, \ldots, n.$$

同理可证 $R_1(i,i)/R_2(i,i) \le 1$. 所以

$$R_1(i,i) = R_2(i,i), \quad i = 1, 2, \ldots, n.$$

又 $\|Q_1\|_F^2 = \mathrm{tr}(Q_1^*Q_1) = n$, 所以由 (1.2) 可知

$$\|R_2R_1^{-1}\|_F^2 = \|Q_2R_2R_1^{-1}\|_F^2 = \|Q_1\|_F^2 = n.$$

由于 $R_2R_1^{-1}$ 的对角线元素都是 1, 所以 $R_2R_1^{-1}$ 只能是单位矩阵, 即 $R_2 = R_1$. 因此 $Q_2 = AR_2^{-1} = AR_1^{-1} = Q_1$, 即 A 的 QR 分解是唯一的. □

☞ 如果 A 是实矩阵, 则上面证明中的运算都可以在实数下进行, 因此 Q 和 R 都可以是实矩阵.

☞ 基于 GS 正交化的 QR 分解算法 1.1 的运算量大约为 $2mn^2$. 在后面, 我们会介绍基于 Household 变换的 QR 分解, 在不需要计算 Q 的情况下, 运算量大约为 $2mn^2 - 2n^3/3$; 如果需要计算 Q, 则需另外大约 $2mn^2 - 2n^3/3$ 运算量.

1.2 初等变换矩阵

矩阵计算的一个基本思想就是把较复杂的问题转化为等价的较简单的, 易于求解问题. 而完成这个转化的基本工具就是初等变换矩阵, 其中使用较多的有两种正交变换: Householder 变换和 Givens 变换.

1.2.1 初等矩阵

我们考虑**初等矩阵**

$$E(u, v, \tau) = I - \tau uv^*,$$

其中 $u, v \in \mathbb{C}^n$ 是非零向量, τ 是一个非零复数. 事实上, $E(u,v,\tau)$ 是单位矩阵的一个秩 1 扰动.

定理 1.2 设 $E(u,v,\tau)$ 是一个初等矩阵, 我们有

(1) $\det(E(u,v,\tau)) = 1 - \tau v^*u$;

(2) 若 $1 - \tau v^*u \ne 0$, 则 $E(u,v,\tau)$ 非奇异, 且

$$(E(u,v,\tau))^{-1} = E(u,v,\gamma), \quad \text{其中 } \gamma = \frac{\tau}{\tau v^*u - 1}.$$

证明. (1) 易知

$$\begin{bmatrix} I & 0 \\ v^* & 1 \end{bmatrix} \begin{bmatrix} I-\tau uv^* & -\tau u \\ 0 & 1 \end{bmatrix} \begin{bmatrix} I & 0 \\ -v^* & 1 \end{bmatrix} = \begin{bmatrix} I & -\tau u \\ 0 & 1-\tau v^* u \end{bmatrix}.$$

由行列式的乘法可知

$$\det\left(\begin{bmatrix} I & 0 \\ v^* & 1 \end{bmatrix}\right) \cdot \det\left(\begin{bmatrix} I-\tau uv^* & -\tau u \\ 0 & 1 \end{bmatrix}\right) \cdot \det\left(\begin{bmatrix} I & 0 \\ -v^* & 1 \end{bmatrix}\right) = \det\left(\begin{bmatrix} I & -\tau u \\ 0 & 1-\tau v^* u \end{bmatrix}\right).$$

所以

$$\det\big(E(u,v,\tau)\big) = \det(I-\tau uv^*) = 1-\tau v^* u.$$

(2) 若 $1-\tau v^* u \neq 0$, 则 $\det\big(E(u,v,\tau)\big) \neq 0$, 所以 $E(u,v,\tau)$ 非奇异. 通过直接计算可知

$$\begin{aligned} E(u,v,\tau)E(u,v,\gamma) &= I - \tau uv^* - \gamma(1-\tau v^* u)uv^* \\ &= I - \tau uv^* - \frac{\tau}{\tau v^* u - 1}(1-\tau v^* u)uv^* \\ &= I. \end{aligned}$$

□

定理 1.3 设 $A \in \mathbb{C}^{n\times n}$, 则 A 非奇异当且仅当 A 可以分解成若干个初等矩阵的乘积.

1.2.2 Householder 变换

定义 1.1 我们称矩阵

$$H = I - 2vv^*, \quad v \in \mathbb{C}^n \text{ 满足 } \|v\|_2 = \sqrt{v^* v} = 1, \tag{1.3}$$

为 Householder 矩阵 (或 Householder 变换, 或 Householder 反射), 向量 v 称为 Householder 向量. 我们通常将矩阵 (1.3) 记为 $H(v)$.

从几何上看, 一个 Householder 变换就是一个关于超平面 $\mathrm{span}\{v\}^\perp$ 的反射. 对任意一个向量 $x \in \mathbb{C}^n$, 可将其写为

$$x = (v^* x)v + y \triangleq \alpha v + y,$$

其中 $\alpha v \in \mathrm{span}\{v\}$, $y \in \mathrm{span}\{v\}^\perp$. 则

$$Hx = x - 2vv^* x = x - 2\alpha v = -\alpha v + y,$$

即 Hx 与 x 在 $\mathrm{span}\{v\}^\perp$ 方向有着相同的分量, 而在 v 方向的分量正好相差一个符号. 也就是说, Hx 是 x 关于超平面 $\mathrm{span}\{v\}^\perp$ 的镜面反射, 见图 1.1. 因此, Householder 变换也称为反射变换.

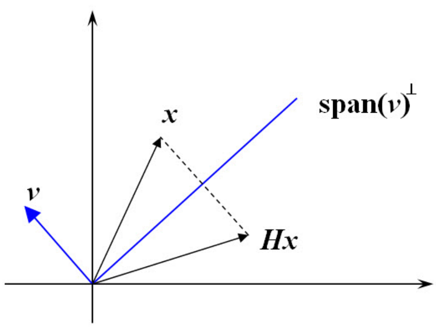

图 1.1 Householder 变换的几何意义

下面是关于 Householder 矩阵的几个基本性质.

定理 1.4 设 $H \in \mathbb{C}^{n\times n}$ 是一个 Householder 矩阵, 则

(1) $H^* = H$, 即 H Hermite 的;

(2) $H^*H = I$, 即 H 是酉矩阵;

(3) $H^2 = I$, 所以 $H^{-1} = H$;

(4) $\det(H) = -1$;

(5) H 有两个互异的特征值: $\lambda = 1$ 和 $\lambda = -1$, 其中 $\lambda = 1$ 的代数重数为 $n-1$.

Householder 矩阵的一个非常重要的应用就是可以将一个向量除第一个元素以外的所有元素都化为零. 我们首先给出一个引理.

引理 1.1 设 $x, y \in \mathbb{C}^n$ 为任意两个互异的向量, 则存在一个 Householder 矩阵 H 使得 $y = Hx$ 的充要条件是 $\|x\|_2 = \|y\|_2$ 且 $x^*y \in \mathbb{R}$.

证明. 若 $\|x\|_2 = \|y\|_2$ 且 $x^*y \in \mathbb{R}$, 则 $y^*y = x^*x$ 且 $x^*y = y^*x$. 于是

$$\|x-y\|_2^2 = (x-y)^*(x-y) = x^*x - y^*x - x^*y + y^*y = 2(x^*x - y^*x).$$

令 $v = x - y$, 则有

$$H(v)x = x - \frac{2(x-y)(x-y)^*x}{\|x-y\|_2^2} = x - \frac{2(x-y)(x^*x - y^*x)}{2(x^*x - y^*x)} = y,$$

即存在 Householder 矩阵 $H(v)$ 使得 $y = H(v)x$.

反之, 如果存在 Householder 矩阵 H 使得 $y = Hx$, 由于 H 是 Hermite 的, 所以 $x^*y = x^*Hx \in \mathbb{R}$. 又因为 H 是酉矩阵, 所以 $\|y\|_2 = \|Hx\|_2 = \|x\|_2$. □

由引理 1.1, 我们可以立即得到下面的定理.

定理 1.5 设 $x = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$ 是一个非零向量, 则存在 Householder 矩阵 H 使得 $Hx = \alpha e_1$, 其中 $\alpha = \|x\|_2$ (或 $\alpha = -\|x\|_2$), $e_1 = [1, 0, \dots, 0]^T \in \mathbb{R}^n$.

设 $x = [x_1, x_2, \dots, x_n]^T \in \mathbb{R}^n$ 是一个实的非零向量, 下面讨论如何计算其对应的 Householder 向量. 根据引理 1.1, H 所对应的 Householder 向量为

$$v = x - \alpha e_1 = [x_1 - \alpha, x_2, \dots, x_n]^T.$$

在实际计算中, 为了尽可能地减少舍入误差, 我们通常避免两个相近的数做减运算, 否则就会损失有效数字. 因此, 我通常取

$$\alpha = -\operatorname{sign}(x_1) \cdot \|x\|_2.$$

事实上, 我们也可以取 $\alpha = \operatorname{sign}(x_1)\|x\|_2$, 但此时为了减少舍入误差, 我们可以通过下面的公式来计算 v 的第一个分量 v_1

$$\alpha = \operatorname{sign}(x_1)\|x\|_2, \quad v_1 = x_1 - \alpha = \frac{x_1^2 - \|x\|_2^2}{x_1 + \alpha} = \frac{-(x_2^2 + x_3^2 + \cdots + x_n^2)}{x_1 + \alpha}.$$

无论怎样选取 α, 我们都有 $H = I - \beta vv^*$ 其中

$$\beta = \frac{2}{v^*v} = -\frac{1}{\alpha v_1}.$$

在实数域中计算 Householder 向量 v 的算法如下, 总运算量大约为 $3n$.

算法 1.2 计算 Householder 向量

% Given $x \in \mathbb{R}^n$, compute $v \in \mathbb{R}^n$ such that $Hx = \|x\|_2 e_1$, where $H = I - \beta vv^*$

function $[\beta, v]$ = **house**(x)
$n = \mathtt{length}(x)$ % Here $\mathtt{length}(x)$ denotes the dimension of x
$\eta = x_2^2 + x_3^2 + \cdots + x_n^2$
$v = x$
if $\eta = 0$ **then**
    $\beta = 0$
else
    $\alpha = \sqrt{x_1^2 + \eta}$ % The norm of x
    if $x_1 < 0$ **then**
        $v_1 = x_1 - \alpha$
    else
        $v_1 = x_1 + \alpha$
    end if
    $\beta = 2/(v_1^2 + \eta)$
end if

☞ 在实际计算时, 我们可以将向量 v 单位化, 使得 $v_1 = 1$. 这样, 我们就无需为 v 另外分配空间, 而是将 $v(2:n)$ 存放在 $x(2:n)$ 中, 因为经过 Householder 变换后, 向量 x 除第一个分量外, 其它都为零. 同时, 为了避免可能产生的溢出, 我们也可以事先将 x 单位化, 即令 $x = x/\|x\|_2$

1.2.3 Givens 变换

为简单起见, 我们这里这讨论实数域中的 Givens 变换. 设 $\theta \in [0, 2\pi]$, 我们称矩阵

$$G(i,j,\theta) = \begin{bmatrix} 1 & & & & & & \\ & \ddots & & & & & \\ & & c & & s & & \\ & & & \ddots & & & \\ & & -s & & c & & \\ & & & & & \ddots & \\ & & & & & & 1 \end{bmatrix} \in \mathbb{R}^{n \times n}$$

为 **Givens 变换** (或 **Givens 旋转**, 或 **Givens 矩阵**), 其中 $c = \cos(\theta)$, $s = \sin(\theta)$. 即将单位矩阵的 (i,i) 和 (j,j) 位置上的元素用 c 代替, 而 (i,j) 和 (j,i) 位置上的元素分别用 s 和 $-s$ 代替, 所得到的矩阵就是 $G(i,j,\theta)$.

定理 1.6 $G(i,j,\theta)$ 是正交矩阵, 且 $\det(G(i,j,\theta))=1$.

我们需要注意的是, 当一个矩阵左乘一个 Givens 矩阵时, 只会影响其第 i 行和第 j 行的元素. 而当一个矩阵右乘一个 Givens 矩阵时, 只会影响其第 i 和第 j 列的元素.

例 1.1 设 $x=\begin{bmatrix}x_1\\x_2\end{bmatrix}\in\mathbb{R}^2$, 则存在一个 Givens 变换 $G=\begin{bmatrix}c&s\\-s&c\end{bmatrix}\in\mathbb{R}^{2\times2}$ 使得 $Gx=\begin{bmatrix}r\\0\end{bmatrix}$, 其中 c,s 和 r 的值如下:

- 若 $x_1=x_2=0$, 则 $c=1, s=0, r=0$;
- 若 $x_1=0$ 但 $x_2\neq0$, 则 $c=0, s=x_2/|x_2|, r=|x_2|$;
- 若 $x_1\neq0$ 但 $x_2=0$, 则 $c=\mathrm{sign}(x_1), s=0, r=|x_1|$;
- 若 $x_1\neq0$ 且 $x_2\neq0$, 则 $c=x_1/r, s=x_2/r, r=\sqrt{x_1^2+x_2^2}$.

也就是说, 通过 Givens 变换, 我们可以将向量 $x\in\mathbb{R}^2$ 的第二个分量化为 0.

事实上, 对于任意一个向量 $x\in\mathbb{R}^n$, 我们都可以通过 Givens 变换将其任意一个位置上的分量化为 0. 更进一步, 我们也可以通过若干个 Givens 变换, 将 x 中除第一个分量外的所有元素都化为 0.

算法 1.3 Givens 变换

% Given $x=[a,b]^T\in\mathbb{R}^2$, compute c and s such that $Gx=[r,0]^T$ where $r=\|x\|_2$

function $[c,s]$ = **givens**(a,b)
if $b=0$ **then**
  if $a\geq0$ **then**
    $c=1,\quad s=0$
  else
    $c=-1,\quad s=0$
  end if
else
  if $|b|>|a|$ **then**
    $\tau=\dfrac{a}{b},\quad s=\dfrac{\mathrm{sign}(b)}{\sqrt{1+\tau^2}},\quad c=s\tau$
  else
    $\tau=\dfrac{b}{a},\quad c=\dfrac{\mathrm{sign}(a)}{\sqrt{1+\tau^2}},\quad s=c\tau$
  end if
end if

1.3 QR 分解的实现

这里我们给出 QR 分解的具体实现方法. 下面我们分别介绍基于 MGS 过程, Householder 变换和 Givens 变换的 QR 分解算法.

1.3.1 基于 MGS 的 QR 分解

在证明 QR 分解的存在性时,我们利用了 Gram-Schmidt 正交化过程. 但由于数值稳定性方面的原因,在实际计算中,我们一般不采用 Gram-Schmidt 过程,取而代之的是 **修正的 Gram-Schmidt 过程** (modified Gram-Schmidt process),即 MGS . 本算法的运算量大约为 $2mn^2$.

算法 1.4 基于 MGS 的 QR 分解

% Given $A \in \mathbb{R}^{m\times n}$, compute $Q = [q_1, \ldots, q_n] \in \mathbb{R}^{m\times n}$ and $R \in \mathbb{R}^{n\times n}$ such that $A = QR$

Set $R = [r_{ik}] = 0_{n\times n}$ (the $n \times n$ zero matrix)
if $a_1 = 0$ **then**
  $q_1 = 0$
else
  $r_{11} = \|a_1\|_2$
  $q_1 = a_1/\|a_1\|_2$
end if
for $k = 2$ to n **do**
  $q_k = a_k$
  for $i = 1$ to $k-1$ **do**
    $r_{ik} = q_i^T q_k$
    $q_k = q_k - r_{ik} q_i$
  end for
  if $q_k \neq 0$ **then**
    $r_{kk} = \|q_k\|_2$
    $q_k = q_k / r_{kk}$
  end if
end for

1.3.2 基于 Householder 变换的 QR 分解

由定理 1.5 可知,通过 Householder 变换,我们可以将任何一个非零变量 $x \in \mathbb{R}^n$ 转化成 $\|x\|_2 e_1$,即除第一个元素外,其它都为零. 下面我们就考虑通过 Householder 变换来实现矩阵的 QR 分解.

我们首先考虑 $m = n$ 时的情形. 设矩阵 $A \in \mathbb{R}^{n\times n}$,令 $H_1 \in \mathbb{R}^{n\times n}$ 为一个 Householder 变换,满足

$$H_1 \begin{bmatrix} a_{11} \\ a_{21} \\ \vdots \\ a_{n1} \end{bmatrix} = \begin{bmatrix} r_1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}.$$

于是

$$H_1 A = \left[\begin{array}{c|ccc} r_1 & \tilde{a}_{12} & \cdots & \tilde{a}_{1n} \\ \hline 0 & & & \\ \vdots & & \tilde{A}_2 & \\ 0 & & & \end{array}\right],$$

其中 $\tilde{A}_2 \in \mathbb{R}^{(n-1)\times(n-1)}$. 同样地, 我们可以构造一个 Householder 变换 $\tilde{H}_2 \in \mathbb{R}^{(n-1)\times(n-1)}$, 将 $\tilde{A}_2$ 的第一列中除第一个元素外的所有元素都化为 0, 即

$$\tilde{H}_2\tilde{A}_2 = \left[\begin{array}{c|ccc} r_2 & \tilde{a}_{23} & \cdots & \tilde{a}_{2n} \\ \hline 0 & & & \\ \vdots & & \tilde{A}_3 & \\ 0 & & & \end{array}\right].$$

令

$$H_2 = \begin{bmatrix} 1 & 0 \\ 0 & \tilde{H}_2 \end{bmatrix}.$$

则 $H_2 \in \mathbb{R}^{n\times n}$, 且

$$H_2H_1A = \left[\begin{array}{cc|ccc} r_1 & \tilde{a}_{12} & \tilde{a}_{13} & \cdots & \tilde{a}_{1n} \\ 0 & r_2 & \tilde{a}_{23} & \cdots & \tilde{a}_{2n} \\ \hline 0 & 0 & & & \\ \vdots & \vdots & & \tilde{A}_3 & \\ 0 & 0 & & & \end{array}\right].$$

不断重复上述过程. 这样, 我们就得到一系列的矩阵

$$H_k = \begin{bmatrix} I_{k-1} & 0 \\ 0 & \tilde{H}_k \end{bmatrix}, \quad k = 1, 2, 3, 4, \ldots, n-1$$

使得

$$H_{n-1}\cdots H_2H_1A = \begin{bmatrix} r_1 & \tilde{a}_{12} & \cdots & \tilde{a}_{1n} \\ 0 & r_2 & \cdots & \tilde{a}_{2n} \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & r_n \end{bmatrix} \triangleq R.$$

由于 Householder 变换都是正交矩阵, 因此 $H_1, H_2, \ldots, H_{n-1}$ 也都是正交矩阵. 令

$$Q = (H_{n-1}\cdots H_2H_1)^{-1} = H_1^{-1}H_2^{-1}\cdots H_{n-1}^{-1} = H_1H_2\cdots H_{n-1},$$

则 Q 也是正交矩阵, 且

$$A = (H_{n-1}\cdots H_2H_1)^{-1}R = QR.$$

以上就是基于 Householder 变换的 QR 分解的具体实现过程. 最后所得到的上三角矩阵 R 就存放在 A 的上三角部分. 矩阵 Q 可通过下面的算法实现

$$\begin{cases} Q = I_n, \\ Q = QH_k, \quad k = 1, 2, \ldots, n-1. \end{cases}$$

☞ 如果不需要生成 Q, 则基于 Householder 变换的 QR 分解的总运算量大约为 $2mn^2 - 2/3n^3$.

如果保留了每一步的 Householder 向量, 则 Q 也可以通过下面的向后累积方法实现:

$$
\begin{aligned}
&Q = I_n, \\
&Q = H_k Q, \quad k = n-1, n-2, \ldots, 1.
\end{aligned}
$$

这样做的好处是一开始 Q 会比较稀疏, 随着迭代的进行, Q 才会慢慢变满. 而前面的计算方法, 第一步就将 Q 变成了一个满矩阵. 采用这种方法计算 Q 的运算量大约为 $4(m^2n - mn^2 + \frac{1}{3}n^3)$.

算法 1.5 基于 Householder 变换的 QR 分解

% Given $A \in \mathbb{R}^{n\times n}$, compute Q and R such that $A = QR$ where $Q \in \mathbb{R}^{n\times n}$ and $R \in \mathbb{R}^{n\times n}$

% The upper triangular part of R is stored in the upper triangular part of A

Set $Q = I_{n\times n}$

for $k = 1$ to $n-1$ **do**

  $x = A(k:n, k)$

  $[\beta, v_k]$ = **house**(x) % $\tilde{H}_k = I - \beta v_k v_k^T$

  更新 $A(k:n, k:n)$, 即

$$
\begin{aligned}
A(k:n, k:n) &= (I_{n-k+1} - \beta v_k v_k^T) A(k:m, k:n) \\
&= A(k:n, k:n) - \beta v_k [v_k^T A(k:n, k:n)]
\end{aligned}
$$

  更新 $Q(:, k:n)$, 即

$$
\begin{aligned}
Q(:, k:n) &= Q(:, k:n)(I_{n-k+1} - \beta v_k v_k^T) \\
&= Q(:, k:n) - \beta [Q(:, k:n) v_k] v_k^T
\end{aligned}
$$

end for

☞ 上面的算法只是关于利用 Householder 变换来实现 QR 分解的一个简单描述, 并没有考虑运算量问题. 在实际计算时, 我们通常会保留所有的 Householder 向量. 由于第 k 步中 $\tilde{H}_k$ 所对应的 Householder 向量 v_k 的长度为 $m-k+1$, 因此我们先把 v_k 单位化, 使得 v_k 的第一元素为 1, 这样就只要存储 $v_k(2\!:\!end)$, 共 $m-k$ 个元素, 我们可以把这些元素存放在 A 的严格下三角部分. A 的上三角部分仍然存放 R. 在计算 Q 时采用向后累积的方法, 所以总的运算量大约为 $4m^2n - 2mn^2 + \frac{2}{3}n^3$. 若 $m = n$, 则运算量大约为 $\frac{8}{3}n^3$.