正交变换

张晓平

October 22, 2018

1 Householder变换

Gauss变换将一个矩阵约化为上三角阵，是基于如下事实：$\forall x \in \mathbb{R}^n$，可构造一个初等下三角阵L，使得$Lx = \alpha e_1$。

本节的目标：构造一个初等正交矩阵H，使其具有矩阵L的功能。

本节的目标：构造一个初等正交矩阵H，使其具有矩阵L的功能。

定义 1. 设$w \in \mathbb{R}^n$满足$\|w\|_2 = 1$，定义$H \in \mathbb{R}^{n\times n}$为

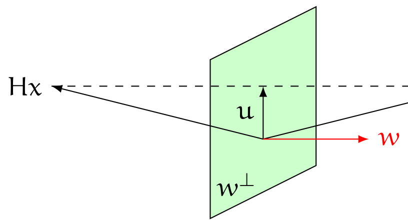
$$H = I - 2ww^T, \tag{1}$$

则称H为*Householder*变换。

Householder变换也叫做"初等反射矩阵"或"镜像变换"。

定理 1. 设H是由*(1)*定义的*Householder*变换，则H满足

1. （对称性）$H^T = H$
2. （正交性）$H^T H = I$
3. （对合性）$H^2 = I$
4. （反射性）$\forall x \in \mathbb{R}^n$，$Hx$是$x$关于$w$的垂直超平面的镜像反射。



Proof. (1)显然。(2)和(3)可由(1)导出。事实上，

$$H^T H = H^2 = (I - 2ww^T)(I - 2ww^T) = I - 4ww^T + 4ww^T = I.$$

(4)设$x \in \mathbb{R}^n$，则x可表示为 $x = u + \alpha w$, 其中$u \in \text{span}\{w\}^\perp$, $\alpha \in \mathbb{R}$。利用$u^T w = 0$和$w^T w = 1$可得

$$Hx = (I - 2ww^T)(u + \alpha w) = u + \alpha w - 2ww^T u - 2\alpha ww^T w = u - \alpha w.$$

□

定理 2. 设$0 \neq x \in \mathbb{R}^n$，则可构造单位向量$w \in \mathbb{R}^n$，使由*(1)*定义的*Householder*变换H满足

$$Hx = \alpha e_1,$$

其中$\alpha = \pm\|x\|_2$。

Proof. 由于H为正交矩阵，故

$$|\alpha| = \|Hx\|_2 = \|x\|_2$$

即$\alpha = \pm\|x\|_2$。又因为

$$Hx = (I - 2ww^T)x = x - 2(w, x)w$$

故欲使$Hx = \alpha e_1$，两边关于x做内积得

$$(x, x) - 2(w, x)(w, x) = \alpha(e_1, x)$$

即

$$2(w, x)^2 = \|x\|_2^2 - \alpha x_1$$

于是w应为

$$w = \frac{x - \alpha e_1}{\|x - \alpha e_1\|_2}.$$

□

该定理告诉我们，

- $\forall 0 \neq x \in \mathbb{R}^n$都可构造出Householder矩阵H，使Hx的后$n-1$个分量为0。
- 如何确定w?
  1. 计算$v = x \pm \|x\|_2 e_1$
  2. 计算$w = v/\|v\|_2$.

实际计算中，我们需要注意以下几点：

- 为使α为正，应取
$$v = x - \|x\|_2 e_1.$$
- 若$x \to e_1$，则计算 $v_1 = x_1 - \|x\|_2$ 时会出现两个相近的数相减。可做如下处理：当$x_1 > 0$时，使用
$$v_1 = x_1 - \|x\|_2 = \frac{x_1^2 - \|x\|^2}{x_1 + \|x\|_2} = \frac{-(x_2^2 + \cdots + x_n^2)}{x_1 + \|x\|_2}.$$
来计算v_1，就会避免出现两个相近的数相减。
- 注意到
$$H = I - 2ww^T = I - \frac{2}{v^T v}vv^T := I - \beta vv^T,$$
就没有必要求出w，而只需求出β和v即可。
- 实际计算中，可将v规格化为第一个分量为1的向量，这样正好可以把v的后$n-1$个分量保存在x的后$n-1$个化为0的分量位置上，而v的第一个分量1无需保存。
- 当$v^T v$很小时，可能会出现下溢，即被自动置为零。而若x的分量太大，平方可能会出现上溢。考虑到$\alpha v(\alpha \neq 0)$与v的单位化向量相同，为了避免溢出现象，可用$x/\|x\|_\infty$代替x来构造v。

```
function [v, beta] = householder(x)
n = length(x);
eta = max(abs(x));
x = x / eta;
sigma = x(2:n)' * x(2:n);
v = zeros(n, 1);
v(2:n) = x(2:n);
if sigma == 0
    beta = 0;
else
    alpha = sqrt(x(1)^2 + sigma);
    if x(1) <= 0
        v(1) = x(1) - alpha;
    else
        v(1) = -sigma / (x(1) + alpha);
```

```
    end
    beta = 2 * v(1)^2 / (sigma + v(1)^2);
    v = v / v(1);
end
end
```

利用Householder变换并不局限于$Hx = \alpha e_1$的形式，它可以将向量中任何若干相邻的元素化为零。例如，欲在$x \in \mathbb{R}^n$中从$k+1$到j位置引入零元素，只需定义

$$v = (0, \cdots, 0, x_k - \alpha, x_{k+1}, \cdots, x_j, 0, \cdots, 0)$$

即可，其中$\alpha^2 = \sum_{i=k}^{j} x_i^2$.

例 1. 构造*Householder*变换将

$$\begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \end{bmatrix}$$

中的第$3-4$个元素置为零。

若想计算矩阵$H = I - \beta vv^T \in \mathbb{R}^{m\times m}$和矩阵$A \in \mathbb{R}^{m\times n}$的乘积，H并不需要显示给出，可按如下方式计算：

$$HA = (I - \beta vv^T)A = A - \beta v(A^T v)^T = A - vu^T, \quad u = \beta A^T v,$$

即

1. 计算$u = \beta A^T v$;
2. 计算$B = A - vu^T$.

2 Givens变换

1. 欲把一个向量中的多个分量化为0，可用Householder变换；
2. 若只将其中一个分量化为0，则应采用Givens变换。

定义 2 (Givens变换)**.** *Givens*矩阵定义如下

$$\begin{aligned} G(i,k,\theta) &= I + s(e_i e_k^T - e_k e_i^T) + (c-1)(e_i e_i^T + e_k e_k^T) \\ &= \begin{array}{cc} \begin{pmatrix} 1 & & \vdots & & \vdots & & \\ & \ddots & \vdots & & \vdots & & \\ \cdots & \cdots & c & \cdots & s & \cdots & \cdots \\ & & \vdots & & \vdots & & \\ \cdots & \cdots & -s & \cdots & c & \cdots & \cdots \\ & & \vdots & & \vdots & \ddots & \\ & & \vdots & & \vdots & & 1 \end{pmatrix} & \begin{array}{c} \\ \\ i \\ \\ k \\ \\ \\ \end{array} \\ \begin{array}{ccccccc} & & i & & k & & \end{array} & \end{array} \end{aligned}$$

$$c = \cos\theta, s = \sin\theta$$

设$x \in \mathbb{R}^n$，令$y = G(i,k,\theta)x$，则有

$$\begin{aligned} y_i &= \quad cx_i + sx_k, \\ y_k &= -sx_i + cx_k, \\ y_j &= x_j, \quad j \neq i,k. \end{aligned}$$

若要$y_k = 0$，只要取

$$c = \frac{x_i}{\sqrt{x_i^2 + x_k^2}}, \quad s = \frac{x_k}{\sqrt{x_i^2 + x_k^2}},$$

便有

$$y_i = \sqrt{x_i^2 + x_k^2}, \quad y_k = 0.$$

从几何上看，$G(i,k,\theta)$是在(i,k)坐标平面内将x按顺时针方向做了θ度的旋转，所以Givens变换也称作平面旋转变换。

```
function [c, s] = givens_rotation(a, b)
if b == 0
    c = 1;
    s = 0;
else
    if abs(b) > abs(a)
        r = a / b;
        s = 1 / sqrt(1 + r^2);
        c = s * r;
    else
        r = b / a;
        c = 1 / sqrt(1 + r^2);
        s = c * r;
    end
end

end
```