

第八章 子空间迭代方法

魏华祎，易年余，陆讯

湘潭大学数学与计算科学学院

September 27, 2019

基本思想

在一个维数较低的子空间中寻找解析解的一个最佳近似. 子空间迭代算法的主要过程可以分解为下面三步:

- (1) 寻找合适的子空间;
- (2) 在该子空间中求“最佳近似”;
- (3) 若这个近似解满足精度要求, 则停止计算; 否则, 重新构造一个新的子空间, 并返回第 (2) 步.

这里主要涉及到的两个关键问题是:

- (1) 如果选择和更新子空间;
- (2) 如何在给定的子空间中寻找“最佳近似”.

关于第一个问题, 目前较成功的解决方案就是使用Krylov 子空间.

- 1 Krylov 子空间
- 2 GMRES 算法
- 3 共轭梯度法 (CG)
- 4 收敛性分析
- 5 其它 Krylov 子空间迭代算法

Krylov 子空间

设 $A \in \mathbb{R}^{n \times n}$, $r \in \mathbb{R}^n$, 则由 A 和 r 生成的 m 维 Krylov 子空间 定义为

$$\mathcal{K}_m = \mathcal{K}_m(A, r) \triangleq \text{span} \{r, Ar, A^2r, \dots, A^{m-1}r\}, \quad m \leq n$$

设 $\dim \mathcal{K}_m = m$, 令 v_1, v_2, \dots, v_m 是 \mathcal{K}_m 的一组基, 则 $\forall x \in \mathcal{K}_m$ 可表示为

$$x = y_1 v_1 + y_2 v_2 + \dots + y_m v_m \triangleq V_m y$$

寻找“最佳近似” $x^{(m)}$ 转化为

- (1) 寻找一组合适的基 v_1, v_2, \dots, v_m ;
- (2) 求出 $x^{(m)}$ 在这组基下面的表出系数 $y^{(m)}$.

基的选取: Arnoldi 过程

最简单的基: $\{r, Ar, A^2r, \dots, A^{m-1}r\} \mapsto$ 非正交, 稳定性得不到保证.

Arnoldi 过程: 将 $\{r, Ar, A^2r, \dots, A^{m-1}r\}$ 单位正交化

```
1:  $v_1 = r / \|r\|_2$ 
2: for  $j = 1$  to  $m$  do
3:    $z = Av_j$ 
4:   for  $i = 1$  to  $j$  do    % MGS 正交化过程
5:      $h_{i,j} = (v_i, z)$ ,    $z = z - h_{i,j}v_i$ 
6:   end for
7:    $h_{j+1,j} = \|z\|_2$    % if  $h_{j+1,j} = 0$  break, endif
8:    $v_{j+1} = z / h_{j+1,j}$ 
9: end for
```

Arnoldi 过程的矩阵表示

记 $V_m = [v_1, v_2, \dots, v_m]$

$$H_{m+1,m} = \begin{bmatrix} h_{1,1} & h_{1,2} & h_{1,3} & \cdots & h_{1,m} \\ h_{2,1} & h_{2,2} & h_{2,3} & \cdots & h_{2,m} \\ & h_{3,2} & h_{3,3} & \cdots & h_{3,m} \\ & & \ddots & \ddots & \vdots \\ & & & h_{m,m-1} & h_{m,m} \\ & & & & h_{m+1,m} \end{bmatrix} \in \mathbb{R}^{(m+1) \times m}$$

则由 Arnoldi 过程可知

$$Av_j = h_{1,j}v_1 + h_{2,j}v_2 + \cdots + h_{j,j}v_j + h_{j+1,j}v_{j+1}$$

所以有

$$AV_m = V_{m+1}H_{m+1,m} = V_m H_m + h_{m+1,m}v_{m+1}e_m^T \quad (7.1)$$

其中 $H_m = H_{m+1,m}(1:m, 1:m)$, $e_m = [0, \dots, 0, 1]^T \in \mathbb{R}^m$. 由于 V_m 是列正交矩阵, 上式两边同乘 V_m^T 可得

$$V_m^T A V_m = H_m \quad (7.2)$$

等式 (7.1) 和 (7.2) 是 Arnoldi 过程的两个重要性质.

Lanczos 过程

若 A 对称, 则 H_m 为对称三对角, 记为 T_m , 即

$$T_m = \begin{bmatrix} \alpha_1 & \beta_1 & & \\ \beta_1 & \ddots & \ddots & \\ & \ddots & \ddots & \beta_{m-1} \\ & & \beta_{m-1} & \alpha_m \end{bmatrix} \quad (7.3)$$

Lanczos 过程的性质与 [三项递推公式](#) (令 $v_0 = 0$ 和 $\beta_0 = 0$)

$$AV_m = V_m T_m + \beta_m v_{m+1} e_m^T \quad (7.4)$$

$$V_m^T A V_m = T_m \quad (7.5)$$

$$\beta_j v_{j+1} = A v_j - \alpha_j v_j - \beta_{j-1} v_{j-1}, \quad j = 1, 2, \dots$$

Lanczos 过程

```
1: Set  $v_0 = 0$  and  $\beta_0 = 0$ 
2:  $v_1 = r / \|r\|_2$ 
3: for  $j = 1$  to  $m$  do
4:    $z = Av_j$ 
5:    $\alpha_j = (v_j, z)$ 
6:    $z = z - \alpha_j v_j - \beta_{j-1} v_{j-1}$ 
7:    $\beta_j = \|z\|_2$ 
8:   if  $\beta_j = 0$  then break, end if
9:    $v_{j+1} = z / \beta_j$ 
10: end for
```

Krylov 子空间算法的一般过程

- (1) 令 $m = 1$;
- (2) 定义 Krylov 子空间 $\mathcal{K}_m(A, r_0)$;
- (3) 找出仿射空间 $x^{(0)} + \mathcal{K}_m$ 中的“最佳近似”解;
- (4) 如果这个近似解满足精度要求, 则迭代结束;
否则令 $m \leftarrow m + 1$, 返回第 (2) 步.

Krylov 子空间迭代算法基本框架

- 1: 选取初始向量 $x^{(0)}$
 - 2: 计算 $r_0 = b - Ax^{(0)}$, $v_1 = r_0 / \|r_0\|_2$
 - 3: 寻找“最佳近似”解: $x^{(1)} \in x^{(0)} + \mathcal{K}_1 = x^{(0)} + \text{span}\{v_1\}$
 - 4: if $x^{(1)}$ 满足精度要求 then
 - 5: 终止迭代
 - 6: end if
 - 7: for $m = 2$ to n do
 - 8: 调用 Arnoldi 或 Lanczos 过程计算向量 v_m
 - 9: 寻找“最佳近似”解: $x^{(m)} \in x^{(0)} + \mathcal{K}_m = x^{(0)} + \text{span}\{v_1, \dots, v_m\}$
 - 10: if $x^{(m)}$ 满足精度要求 then
 - 11: 终止迭代
 - 12: end if
 - 13: end for
-

如何计算 $x^{(0)} + \mathcal{K}_m$ 中的“最佳近似” $x^{(m)}$

首先, 我们必须给出“最佳”的定义, 不同的定义会导致不同的算法. 最直接的方式: $\|x^{(m)} - x_*\|_2$ 达到最小. 但由于 x_* 不知道, 因此不实用.

什么是“最佳”

(1) $\|r_m\|_2 = \|b - Ax^{(m)}\|_2$ 达到最小 A 对称 \rightarrow MINRES, A 非对称 \rightarrow GMRES

(2) A 对称正定, 极小化 $\|x_* - x^{(m)}\|_A \rightarrow$ CG(共轭梯度法)

本讲主要介绍GMRES算法和CG算法.

子空间迭代方法

- 1 Krylov 子空间
- 2 GMRES 算法
- 3 共轭梯度法 (CG)
- 4 收敛性分析
- 5 其它 Krylov 子空间迭代算法

GMRES 算法是目前求解非对称线性方程组的最常用算法之一. “最佳近似”解的判别方法为使得 $\|r_m\|_2 = \|b - Ax^{(m)}\|_2$ 最小

对任意向量 $x \in x^{(0)} + \mathcal{K}_m$, 可设 $x = x^{(0)} + V_m y$, 其中 $y \in \mathbb{R}^m$. 于是

$$r = b - Ax = r_0 - AV_m y = V_{m+1} (\beta e_1 - H_{m+1,m} y)$$

这里 $\beta = \|r_0\|_2$. 由于 V_{m+1} 列正交, 所以

$$\|r\|_2 = \|V_{m+1} (\beta e_1 - H_{m+1,m} y)\|_2 = \|\beta e_1 - H_{m+1,m} y\|_2$$

于是最优性条件就转化为

$$y^{(m)} = \arg \min_{y \in \mathbb{R}^m} \|\beta e_1 - H_{m+1,m} y\|_2 \quad (7.6)$$

用基于 Givens 变换的 QR 分解来求解即可.

GMRES 算法的基本框架

算法 2.1 GMRES 迭代算法基本框架

- 1: 选取初值 $x^{(0)}$, 停机标准 $\varepsilon > 0$, 以及最大迭代步数 IterMax
- 2: $r_0 = b - Ax^{(0)}$, $\beta = \|r_0\|_2$
- 3: $v_1 = r_0/\beta$
- 4: for $j = 1$ to IterMax do 5: $w = Av_j$
- 6: for $i = 1$ to j do % Arnoldi 过程
- 7: $h_{i,j} = (v_i, w)$
- 8: $w = w - h_{i,j}v_i$
- 9: end for
- 10: $h_{j+1,j} = \|w\|_2$

11: if $h_{j+1,j} = 0$ then
12: $m = j$, break
13: end if
14: $v_{j+1} = w/h_{j+1,j}$
15: 5 relres = $\|r_j\|_2 / \beta$
16: if relres $< \varepsilon$ then
17: $m = j$, break
18: end if
19: end for
20: 解最小二乘问题 (7.6), 得到 y
21: $x^{(m)} = x^{(0)} + V_m y^{(m)}$

实施细节

需要解决下面两个问题:

- (1) 如何计算残量 $r_m \triangleq b - Ax^{(m)}$ 的范数?
- (2) 如何求解最小二乘问题 (7.6)?

这两个问题可以同时处理.

最小二乘问题的求解

设 $H_{m+1,m}$ 的 QR 分解为

$$H_{m+1,m} = Q_{m+1}^T R_{m+1,m}$$

其中 Q_{m+1} 是正交矩阵, $R_{m+1,m} \in \mathbb{R}^{(m+1) \times m}$ 是上三角矩阵. 则

$$\|\beta e_1 - H_{m+1,m} y\|_2 = \|\beta Q_{m+1} e_1 - R_{m+1,m} y\|_2 = \left\| \beta q_1 - \begin{bmatrix} R_m \\ 0 \end{bmatrix} y \right\|_2$$

其中 $R_m \in \mathbb{R}^{m \times m}$ 非奇异 (假定 $H_{m+1,m}$ 不可约). 所以

$$\begin{aligned} y^{(m)} &= \beta R_m^{-1} q_1(1:m) \\ \|r_m\|_2 &= \|b - Ax^{(m)}\|_2 = \|\beta e_1 - H_{m+1,m} y^{(m)}\|_2 = \beta \cdot |q_1(m+1)| \end{aligned}$$

其中 $q_1(m+1)$ 表示 q_1 的第 $m+1$ 个分量

$H_{m+1,m}$ 的 QR 分解的递推计算方法

由于 $H_{m+1,m}$ 是上 Hessenberg 矩阵, 因此我们采用 Givens 变换.

(1) 当 $m = 1$ 时 $H_{21} = \begin{bmatrix} h_{11} \\ h_{21} \end{bmatrix}$, 构造 Givens 变换 G_1 使得

$$\bar{G}_1 H_{21} = \begin{bmatrix} * \\ 0 \end{bmatrix} = R_{21}, \text{ 即 } H_{21} = G_1^T R_{21}$$

(2) 假定存在 G_1, G_2, \dots, G_{m-1} 使得

$$(G_{m-1} \cdots G_2 G_1) H_{m,m-1} = R_{m,m-1}$$

即

$$H_{m,m-1} = (G_{m-1} \cdots G_2 G_1)^T R_{m,m-1} \triangleq Q_m^T R_{m,m-1}$$

为了书写方便, 这里假定 G_i 的维数自动扩张, 以满足矩阵乘积的需要.

(3) 考虑 $H_{m+1,m}$ 的 QR 分解. 易知

$$H_{m+1,m} = \begin{bmatrix} H_{m,m-1} & h_m \\ 0 & h_{m+1,m} \end{bmatrix} \text{ 其中 } h_m = [h_{1m}, h_{2m}, \dots, h_{mm}]^T$$

所以有

$$\begin{bmatrix} Q_m & 0 \\ 0 & 1 \end{bmatrix} H_{m+1,m} = \begin{bmatrix} R_{m,m-1} & Q_m h_m \\ 0 & h_{m+1,m} \end{bmatrix} = \begin{bmatrix} R_{m-1} & \tilde{h}_{m-1} \\ 0 & \hat{h}_{mm} \\ 0 & h_{m+1,m} \end{bmatrix}$$

其中 \tilde{h}_{m-1} 是 $Q_m h_m$ 的前 m 个元素组成的向量, \hat{h}_{mm} 是 $Q_m h_m$ 的最后一个元素.

构造 Givens 变换 G_m :

$$G_m = \begin{bmatrix} I_{m-1} & 0 & 0 \\ 0 & c_m & s_m \\ 0 & -s_m c_m & \end{bmatrix} \in \mathbb{R}^{(m+1) \times (m+1)}$$

其中 $c_m = \frac{\hat{h}_{m,m}}{\tilde{h}_{m,m}}, s_m = \frac{h_{m+1,m}}{\tilde{h}_{m,m}}, \tilde{h}_{m,m} = \sqrt{\hat{h}_{m,m}^2 + h_{m+1,m}^2}$ 令

$$Q_{m+1} = G_m \begin{bmatrix} Q_m & 0 \\ 0 & 1 \end{bmatrix}$$

则

$$Q_{m+1} H_{m+1,m} = G_m \begin{bmatrix} R_{m-1} & \tilde{h}_{m-1} \\ 0 & \hat{h}_{j,j} \\ 0 & h_{m+1,m} \end{bmatrix} = \begin{bmatrix} R_{m-1} & \tilde{h}_{m-1} \\ 0 & \tilde{h}_{j,j} \\ 0 & 0 \end{bmatrix} \triangleq R_{m+1,m}$$

所以可得 $H_{m+1,m}$ 的 QR 分解 $H_{m+1,m} = Q_{m+1}^T R_{m+1,m}$.

由 $H_{m,m-1}$ 的 QR 分解到 $H_{m+1,m}$ 的 QR 分解, 我们需要

- (1) 计算 $Q_m h_m$, 即将之前的 $m-1$ 个 Givens 变换作用到 $H_{m+1,m}$ 的最后一列的前 m 个元素上, 所以我们需要保留所有的 Givens 变换;
- (2) 残量计算: $\|r_m\|_2 = |\beta q_1(m+1)| = |\beta Q_{m+1}(m+1, 1)|$, 即

$$G_m G_{m-1} \cdots G_2 G_1 (\beta e_1)$$

的最后一个分量的绝对值. 由于在计算 r_{m-1} 时就已经计算出 $G_{m-1} \cdots G_2 G_1 (\beta e_1)$ 因此这里只需做一次 Givens 变换即可;

- (3) $y^{(m)}$ 的计算: 当相对残量满足精度要求时, 需要计算 $y^{(m)} = R_m^{-1} q_1(1:m)$ 而 q_1 即为 $G_m G_{m-1} \cdots G_2 G_1 (\beta e_1)$

实用 GMRES 算法

算法 2.2 实用 GMRES 算法

```
1: 给定初值  $x^{(0)}$ , 停机标准  $\varepsilon > 0$ , 最大迭代步数 IterMax
2:  $r_0 = b - Ax^{(0)}$ ,  $\beta = \|r_0\|_2$ 
3: if  $\beta < \varepsilon$  then
4:   停止计算, 输出近似解  $x^{(0)}$ 
5: end if
6:  $v_1 = r_0/\beta$ 
7:  $\xi = \beta e_1$       记录  $q_1$ 
8: for  $j = 1$  to IterMax do
9:    $w = Av_j$ 
10:  for  $i = 1$  to  $j$  do      % Arnoldi 过程
11:     $h_{i,j} = (v_i, w)$ 
12:     $w = w - h_{i,j}v_i$ 
13:  end for
```

```

14:    $h_{j+1,j} = \|w\|_2$ 
15:   if  $h_{j+1,j} = 0$  then           % 迭代中断
16:        $m = j$ , break
17:   end if
18:    $v_{j+1} = w/h_{j+1,j}$ 
19:   for  $i = 1$  to  $j - 1$  do           % 计算  $G_{j-1} \cdots G_2 G_1 H_{j+1,j}(1:j,j)$ 
20:        $\begin{bmatrix} h_{ij} \\ h_{i+1,j} \end{bmatrix} = \begin{bmatrix} c_i & s_i \\ -s_i & c_i \end{bmatrix} \begin{bmatrix} h_{ij} \\ h_{i+1,j} \end{bmatrix}$ 
21:   end for 22:   if  $|h_{jj}| > |h_{j+1,j}|$  then           % 构造 Givens 变换  $G_j$ 
23:        $\tau = h_{j+1,j}/h_{jj}$ ,  $c_j = 1/\sqrt{1 + \tau^2}$ ,  $s_j = c_j \tau$ 
24:   else
25:        $\tau = h_{jj}/h_{j+1,j}$ ,  $s_j = 1/\sqrt{1 + \tau^2}$ ,  $c_j = s_j \tau$ 
26:   end if
27:    $h_{jj} = c_j h_{jj} + s_j h_{j+1,j}$            % 计算  $G_j H_{j+1,j}(1:j,j)$ 
28:    $h_{j+1,j} = 0$ 

```



```

29:    $\begin{bmatrix} \xi_j \\ \xi_{j+1} \end{bmatrix} = \begin{bmatrix} c_j & s_j \\ -s_j & c_j \end{bmatrix} \begin{bmatrix} \xi_j \\ 0 \end{bmatrix}$  % 计算  $G_j (\beta G_{j-1} \cdots G_2 G_1 e_1)$ 
30:   relres =  $|\xi_{j+1}| / \beta$  % 相对残量
31:   if relres <  $\epsilon$  then
32:       m = j, break
33:   end if
34: end for
35: m = j
36:  $y^{(m)} = H(1 : m, 1 : m) \backslash \xi(1 : m)$  % 最小二乘问题, 回代求解
37:  $x^{(m)} = x^{(0)} + V_m y^{(m)}$ 
38: if relres <  $\epsilon$  then
39:     输出近似解 x 及相关信息
40: else
41:     输出算法失败信息
42: end if

```

GMRES 算法的中断

在上面的 GMRES 算法中, 当执行到某一步时有 $h_{j+1,j} = 0$, 则算法会中断 (breakdown). 如果出现这种中断, 则我们就可以找到精确解

定理

设 $A \in \mathbb{R}^{n \times n}$ 非奇异且 $r_0 \neq 0$. 若 $h_{i+1,i} \neq 0, i = 1, 2, \dots, k-1$ 则 $h_{k+1,k} = 0$ 当且仅当 $x^{(k)}$ 是方程组的精确解. (不考虑舍入误差)

带重启的 GMRES 算法

由于随着迭代步数的增加, GMRES 算法的每一步所需的运算量和存储量都会越来越大. 因此当迭代步数很大时, GMRES 算法就不太实用. **重启技术** 事先设定一个重启迭代步数 k , 如果 GMRES 达到这个迭代步数时仍不收敛, 则计算出 $x^{(0)} + \mathcal{K}_k$ 中的最佳近似解 $x^{(k)}$, 然后令 $x^{(0)} = x^{(k)}$, 重新开始新的 GMRES 迭代.

算法 2.3 带重启的 GMRES 算法

- 1: 设定重启步数 $k(\ll n)$
- 2: 给定初值 $x^{(0)}$, 停机标准 $\varepsilon > 0$, 最大迭代步数 IterMax
- 3: $r_0 = b - Ax^{(0)}, \beta = \|r_0\|_2$
- 4: if $\beta < \varepsilon$ then
- 5: 停止计算, 输出近似解 $x = x^{(0)}$
- 6: end if
- 7: for iter = 1 to ceil(IterMax/k) do
- 8: $v_1 = r_0/\beta$
- 9: $\xi = \beta e_1$
- 10: for $j = 1$ to k do
- 11: 调用 GMRES 循环
- 12: end for
- 13: $m = j$
- 14: $y^{(m)} = H(1 : m, 1 : m) \backslash \xi(1 : m)$

```

15:     $x^{(m)} = x^{(0)} + V_m y^{(m)}$ 
16:    if relres  $< \varepsilon$  then      % 收敛。退出循环
17:        break 18:    end if 19:     $x^{(0)} = x^{(m)}$  % 重启 GMRES
20:     $r_0 = b - Ax^{(0)}, \beta = \|r_0\|_2$ 
21: end for
22: if relres  $< \varepsilon$  then
23:     输出近似解  $x^{(m)}$  及相关信息
24: else
25:     输出算法失败信息
26: end if

```

带重启的 GMRES 算法需要注意的问题

- (1) 如何选取合适的重启步数 k ?
一般只能依靠经验来选取, 如 $k = 20, 50$.
- (2) 不带重启的 GMRES 算法能保证算法的收敛性, 但带重启的 GM-RES 算法却无法保证, 有时可能出现停滞现象 (stagnation).

子空间迭代方法

- 1 Krylov 子空间
- 2 GMRES 算法
- 3 共轭梯度法 (CG)
- 4 收敛性分析
- 5 其它 Krylov 子空间迭代算法

“最佳近似” : $\|x_* - x^{(m)}\|_A$ 最小

首先给出“最佳近似”解 $x^{(m)}$ 的一个性质. **定理** 设 A 对称正定, 则

$$x^{(m)} = \arg \min_{x \in x^{(0)} + \mathcal{K}_m} \|x - x_*\|_A \quad (7.7)$$

当且仅当

$$x^{(m)} \in x^{(0)} + \mathcal{K}_m \quad \text{H} \quad b - Ax^{(m)} \perp \mathcal{K}_m \quad (7.8)$$

Lanczos 过程

Lanczos 过程的三项递推公式:

$$\begin{aligned}AV_m &= V_{m+1}T_{m+1,m} = V_mT_m + \beta_m v_{m+1}e_m^T \\ V_m^T AV_m &= T_m\end{aligned}$$

其中 $T_m = \text{tridiag}(\beta_i, \alpha_{i+1}, \beta_{i+1})$

由前面的结论可知, 此时我们需要在 $x^{(0)} + \mathcal{K}_m$ 寻找最优解 $x^{(m)}$, 满足

$$b - Ax^{(m)} \perp \mathcal{K}_m \quad (7.9)$$

下面就根据这个性质推导 CG 算法的迭代公式.

CG 算法的推导

首先, 设 $\mathbf{x}^{(m)} = \mathbf{x}^{(0)} + \mathbf{V}_m \mathbf{z}^{(m)}$, 其中 $\mathbf{z}^{(m)} \in \mathbb{R}^m$. 由 (7.9) 可知

$$0 = \mathbf{V}_m^T (\mathbf{b} - \mathbf{A} \mathbf{x}^{(m)}) = \mathbf{V}_m^T (\mathbf{r}_0 - \mathbf{A} \mathbf{V}_m \mathbf{z}^{(m)}) = \beta \mathbf{e}_1 - \mathbf{T}_m \mathbf{z}^{(m)}$$

因此,

$$\mathbf{z}^{(m)} = \mathbf{T}_m^{-1} (\beta \mathbf{e}_1)$$

设 \mathbf{T}_m 的 LDL^T 分解为 $\mathbf{T}_m = \mathbf{L}_m \mathbf{D}_m \mathbf{L}_m^T$. 于是

$$\mathbf{x}^{(m)} = \mathbf{x}^{(0)} + \mathbf{V}_m \mathbf{z}^{(m)} = \mathbf{x}^{(0)} + \mathbf{V}_m \mathbf{T}_m^{-1} (\beta \mathbf{e}_1) = \mathbf{x}^{(0)} + (\mathbf{V}_m \mathbf{L}_m^{-T}) (\beta \mathbf{D}_m^{-1} \mathbf{L}_m^{-1} \mathbf{e}_1)$$

如果 $\mathbf{x}^{(m)}$ 满足精度要求, 则计算结束. 否则我们需要计算

$$\mathbf{x}^{(m+1)} = \mathbf{x}^{(0)} + \mathbf{V}_{m+1} \mathbf{T}_{m+1}^{-1} (\beta \mathbf{e}_1) = \mathbf{x}^{(0)} + (\mathbf{V}_{m+1} \mathbf{L}_{m+1}^{-T}) (\beta \mathbf{D}_{m+1}^{-1} \mathbf{L}_{m+1}^{-1} \mathbf{e}_1)$$

这里 $\mathbf{T}_{m+1} = \mathbf{L}_{m+1} \mathbf{D}_{m+1} \mathbf{L}_{m+1}^T$

记

$$\begin{aligned}\tilde{P}_m &\triangleq V_m L_m^{-T} = [\tilde{p}_1, \tilde{p}_2, \dots, \tilde{p}_m] \in \mathbb{R}^{n \times m} \\ y_m &\triangleq \beta D_m^{-1} L_m^{-1} e_1 = [\eta_1, \dots, \eta_m]^T \in \mathbb{R}^m\end{aligned}$$

\tilde{P}_m 和 y_m 的递推关系式(由 T_{m+1} 的 LDL^T 分解可得)

$$\begin{aligned}\tilde{P}_{m+1} &\triangleq V_{m+1} L_{m+1}^{-T} = [\tilde{P}_m, \tilde{p}_{m+1}] \\ y_{m+1} &\triangleq \beta D_{m+1}^{-1} L_{m+1}^{-1} e_1 = [y_m^T, \eta_{m+1}]^T, \quad m = 1, 2, \dots\end{aligned}$$

\tilde{P}_{m+1} 的递推关系式

$$\tilde{p}_{m+1} = -l_m \tilde{p}_m + v_{m+1}$$

x^{m+1} 的递推关系式

$$x^{(m+1)} = \tilde{P}_{m+1} y_{m+1} = \begin{bmatrix} \tilde{P}_m & \tilde{p}_{m+1} \end{bmatrix} \begin{bmatrix} y_m \\ \eta_{m+1} \end{bmatrix} = x^{(m)} + \eta_{m+1} \tilde{p}_{m+1}$$

r_{m+1} 的递推关系式 (收敛性判断)

$$r_{m+1} = b - Ax^{(m+1)} = b - A \left(x^{(m)} + \eta_{m+1} \tilde{p}_{m+1} \right) = r_m - \eta_{m+1} A \tilde{p}_{m+1}$$

另一方面, 我们有

$$r_m = b - Ax^{(m)} = r_0 - AV_m z^{(m)} = -\beta_m \left(e_m^T z^{(m)} \right) v_{m+1}$$

即 r_m 与 v_{m+1} 平行. 记 $r_m = T_m v_{m+1}$, 其中

$$\tau_0 = \beta = \|r_0\|_2, \quad \tau_m = -\beta_m \left(e_m^T z^{(m)} \right), \quad m = 1, 2, \dots$$

p^{m+1} 的递推关系式

(定义 $p_m = \tau_{m-1}\tilde{p}_m$)

$$p_{m+1} = \tau_m \tilde{p}_{m+1} = \tau_m (v_{m+1} - l_m \tilde{p}_m) = r_m + \mu_m p_m \quad (7.10)$$

其中 $\mu_m = -l_m \tau_m / \tau_{m-1}$, $m = 1, 2, \dots$ $x^{(m+1)}$ 和 r_{m+1} 的新递推关系式

$$x^{(m+1)} = x^{(m)} + \eta_{m+1} \tilde{p}_{m+1} = x^{(m)} + \xi_{m+1} p_{m+1} \quad (7.11)$$

$$r_{m+1} = r_m - \eta_{m+1} A \tilde{p}_{m+1} = r_m - \xi_{m+1} A p_{m+1} \quad (7.12)$$

其中 $\xi_{m+1} = \eta_{m+1} / \tau_m$, $m = 1, 2, \dots$

系数 ξ_{m+1} 和 μ_m 的计算方法

引理

下面的结论成立:

- (1) r_1, r_2, \dots, r_m 相互正交;
- (2) p_1, p_2, \dots, p_m 相互 A -共轭 (A -正交), 即当 $i \neq j$ 时有 $p_i^T A p_j = 0$.

在等式 (7.10) 两边同时左乘 $p_{m+1}^T A$ 可得

$$p_{m+1}^T A p_{m+1} = p_{m+1}^T A r_m + \mu_m p_{m+1}^T A p_m = r_m^T A p_{m+1}$$

再用 r_m^T 左乘方程 (7.12) 可得

$$0 = r_m^T r_{m+1} = r_m^T r_m - \xi_{m+1} r_m^T A p_{m+1}$$

于是

$$\xi_{m+1} = \frac{\mathbf{r}_m^T \mathbf{r}_m}{\mathbf{r}_m^T \mathbf{A} \mathbf{p}_{m+1}} = \frac{\mathbf{r}_m^T \mathbf{r}_m}{\mathbf{p}_{m+1}^T \mathbf{A} \mathbf{p}_{m+1}} \quad (7.13)$$

等式 (7.10) 两边同时左乘 $\mathbf{p}_m^T \mathbf{A}$ 可得

$$0 = \mathbf{p}_m^T \mathbf{A} \mathbf{p}_{m+1} = \mathbf{p}_m^T \mathbf{A} \mathbf{r}_m + \mu_m \mathbf{p}_m^T \mathbf{A} \mathbf{p}_m \implies \mu_m = -\frac{\mathbf{r}_m^T \mathbf{A} \mathbf{p}_m}{\mathbf{p}_m^T \mathbf{A} \mathbf{p}_m}$$

为了进一步减少运算量, 将上式简化. 用 \mathbf{r}_{m+1}^T 左乘方程 (7.12) 可得

$$\mathbf{r}_{m+1}^T \mathbf{r}_{m+1} = \mathbf{r}_{m+1}^T \mathbf{r}_m - \xi_{m+1} \mathbf{r}_{m+1}^T \mathbf{A} \mathbf{p}_{m+1} = -\xi_{m+1} \mathbf{r}_{m+1}^T \mathbf{A} \mathbf{p}_{m+1}$$

于是

$$\xi_{m+1} = -\frac{\mathbf{r}_{m+1}^T \mathbf{r}_{m+1}}{\mathbf{r}_{m+1}^T \mathbf{A} \mathbf{p}_{m+1}} \implies \xi_m = -\frac{\mathbf{r}_m^T \mathbf{r}_m}{\mathbf{r}_m^T \mathbf{A} \mathbf{p}_m}$$

即 $\mathbf{r}_m^T \mathbf{A} \mathbf{p}_m = -\mathbf{r}_m^T \mathbf{r}_m / \xi_m$ 于是

$$\mu_m = -\frac{\mathbf{r}_m^T \mathbf{A} \mathbf{p}_m}{\mathbf{p}_m^T \mathbf{A} \mathbf{p}_m} = \frac{\mathbf{r}_m^T \mathbf{r}_m}{\mathbf{p}_m^T \mathbf{A} \mathbf{p}_m} \cdot \frac{1}{\xi_m} = \frac{\mathbf{r}_m^T \mathbf{r}_m}{\mathbf{r}_{m-1}^T \mathbf{r}_{m-1}} \quad (7.14)$$

注意, 以上递推公式是从 $m = 1$ 开始的. 因此 $m = 0$ 时需要另外推导. 首先, 由 $\tilde{\mathbf{p}}_1$ 的定义可知

$$\tilde{\mathbf{p}}_1 = \tilde{\mathbf{P}}_1 = \mathbf{V}_1 \mathbf{L}_1^{-T} = \mathbf{v}_1 \implies \mathbf{p}_1 = \tau_0 \tilde{\mathbf{p}}_1 = \beta \mathbf{v}_1 = \mathbf{r}_0$$

其次, 由 Lanczos 过程可知 $\mathbf{T}_1 = \alpha_1 = \mathbf{v}_1^T \mathbf{A} \mathbf{v}_1$. 注意到 $\beta = \mathbf{r}_0^T \mathbf{r}_0$, 于是

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \mathbf{V}_1 \mathbf{T}_1^{-1} (\beta \mathbf{e}_1) = \mathbf{x}^{(0)} + \frac{\beta}{\mathbf{v}_1^T \mathbf{A} \mathbf{v}_1} \mathbf{v}_1 = \mathbf{x}^{(0)} + \frac{\mathbf{r}_0^T \mathbf{r}_0}{\mathbf{p}_1^T \mathbf{A} \mathbf{p}_1} \mathbf{p}_1$$

令 $\xi_1 = \frac{r_0^T r_0}{p_1^T A p_1}$ (注: 之前的 ξ_{m+1} 计算公式 (7.13) 只对 $m \geq 1$ 有定义), 则当 $m = 0$ 时关于 $x^{(m+1)}$ 的递推公式仍然成立.
最后考虑残量. 易知

$$r_1 = b - Ax^{(1)} = b - Ax^{(0)} - \frac{r_0^T r_0}{p_1^T A p_1} A p_1 = r_0 - \xi_1 A p_1$$

即当 $m = 0$ 时关于 r_{m+1} 的递推公式也成立.

共轭梯度法

算法 3.1 共轭梯度法

- 1: 给定初值 $\mathbf{x}^{(0)}$, 停机标准 $\varepsilon > 0$, 最大迭代步数 IterMax
- 2: $\mathbf{r}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}^{(0)}$,
- 3: $\beta = \|\mathbf{r}_0\|_2$
- 4: if $\beta < \epsilon$ then
- 5: 停止计算, 输出近似解 $\mathbf{x}^{(0)}$
- 6: end if
- 7: for $m = 1$ to IterMax do
- 8: $\rho = \mathbf{r}_{m-1}^T \mathbf{r}_{m-1}$
- 9: if $m > 1$ then
- 10: $\mu_{m-1} = \rho / \rho_0$
- 11: $\mathbf{p}_m = \mathbf{r}_{m-1} + \mu_{m-1} \mathbf{p}_{m-1}$
- 12: else

```

13:       $p_m = r_0$ 
14:  end if
15:   $q_m = Ap_m$ 
16:   $\xi_m = \rho / (p_m^T q_m)$ 
17:   $x^{(m)} = x^{(m-1)} + \xi_m p_m$ 
18:   $r_m = r_{m-1} - \xi_m q_m$ 
19:   $\text{relres} = \|r_m\|_2 / \beta$ 
20:  if  $\text{relres} < \varepsilon$  then
21:      停止迭代, 输出近似解  $x^{(m)}$ 
22:  end if
23:   $\rho_0 = \rho$ 
24: end for
25: if  $\text{relres} < \varepsilon$  then
26:     输出近似解  $x^{(m)}$  及相关信息
27: else
28:     输出算法失败信息

```

29:end if CG 算法的每个迭代步的主要运算为一个矩阵向量乘积和两个
向量内积;

子空间迭代方法

- 1 Krylov 子空间
- 2 GMRES 算法
- 3 共轭梯度法 (CG)
- 4 收敛性分析**
- 5 其它 Krylov 子空间迭代算法

CG 算法的收敛性

设 x_* 是解析解, $x^{(m)}$ 是 CG 算法在 $x^{(0)} + \mathcal{K}_m$ 中找到的近似解, 即

$$x^{(m)} = \arg \min_{x \in x^{(0)} + \mathcal{K}_m} \|x - x_*\|_A$$

记 \mathbb{P}_k 为所次数不超过 k 的多项式的集合. 对任意 $x \in x^{(0)} + \mathcal{K}_m$, 存在 $p(t) \in \mathbb{P}_{m-1}$, 使得

$$x = x^{(0)} + p(A)r_0$$

于是有

$$x - x_* = \varepsilon_0 + p(A) \left(b - Ax^{(0)} \right) = \varepsilon_0 + p(A) \left(Ax_* - Ax^{(0)} \right) \triangleq q(A)\varepsilon_0$$

其中 $\varepsilon_0 = x^{(0)} - x_*$ 多项式 $q(t) = 1 - tp(t) \in \mathbb{P}_m$ 且 $q(0) = 1$. 所以

$$\|x - x_*\|_A^2 = \varepsilon_0^T q(A)^T A q(A) \varepsilon_0$$

设 $A = Q\Lambda Q^T$, $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$, 记 $y = [y_1, y_2, \dots, y_n]^T \triangleq Q^T \varepsilon_0$.

$$\begin{aligned}
 \left\| x^{(m)} - x_* \right\|_A^2 &= \min_{x \in x^{(0)} + \mathcal{K}_m} \|x - x_*\|_A^2 \\
 &= \min_{q \in \mathbb{P}_m, q(0)=1} \varepsilon_0^T Q q(\Lambda)^T \Lambda q(\Lambda) Q^T \varepsilon_0 \\
 &= \min_{q \in \mathbb{P}_m, q(0)=1} \sum_{i=1}^n y_i^2 \lambda_i q(\lambda_i)^2 \\
 &\leq \min_{q \in \mathbb{P}_m, q(0)=1} \max_{1 \leq i \leq n} \left\{ q(\lambda_i)^2 \right\} \sum_{i=1}^n y_i^2 \lambda_i \\
 &= \min_{q \in \mathbb{P}_m, q(0)=1} \max_{1 \leq i \leq n} \left\{ q(\lambda_i)^2 \right\} y^T \Lambda y \\
 &= \min_{q \in \mathbb{P}_m, q(0)=1} \max_{1 \leq i \leq n} \left\{ q(\lambda_i)^2 \right\} \varepsilon_0^T A \varepsilon_0 \\
 &= \min_{q \in \mathbb{P}_m, q(0)=1} \max_{1 \leq i \leq n} \left\{ q(\lambda_i)^2 \right\} \|\varepsilon_0\|_A^2
 \end{aligned}$$

引理

设 $x^{(m)}$ 是 CG 算法迭代 m 步后得到的近似解. 则

$$\frac{\|x^{(m)} - x_*\|_A}{\|x^{(0)} - x_*\|_A} \leq \min_{q \in \mathbb{P}_m, q(0)=1} \max_{1 \leq i \leq n} |q(\lambda_i)|$$

当 A 的特征值不知道时, 可用区间代替, 即

$$\frac{\|x^{(m)} - x_*\|_A}{\|x^{(0)} - x_*\|_A} \leq \min_{q \in \mathbb{P}_m, q(0)=1} \max_{\lambda_n \leq \lambda \leq \lambda_1} |q(\lambda)|$$

由 Chebyshev 多项式的最佳逼近性质可知, 上式的解为

$$\tilde{q}(t) = \frac{T_m\left(\frac{2t - (\lambda_1 + \lambda_n)}{\lambda_1 - \lambda_n}\right)}{T_m\left(-\frac{\lambda_1 + \lambda_n}{\lambda_1 - \lambda_n}\right)} \implies |\tilde{q}(t)| \leq \frac{1}{2} \left(\frac{\sqrt{\kappa(A)} + 1}{\sqrt{\kappa(A)} - 1} \right)^m$$

定理

设 $A \in \mathbb{R}^{n \times n}$ 对称正定, $x^{(m)}$ 是 CG 算法迭代 m 步后得到的近似解. 则

$$\frac{\|x^{(m)} - x_*\|_A}{\|x^{(0)} - x_*\|_A} \leq 2 \left(\frac{\sqrt{\kappa(A)} - 1}{\sqrt{\kappa(A)} + 1} \right)^m$$

其中 $\kappa(A) = \lambda_1 / \lambda_n$

CG 算法的超收敛性

如果我们能够获得 A 的更多的特征值信息, 则能得到更好的误差限.

定理

设 $A \in \mathbb{R}^{n \times n}$ 对称正定, 特征值为

$$0 < \lambda_n \leq \cdots \leq \lambda_{n+1-i} \leq b_1 \leq \lambda_{n-i} \leq \cdots \leq \lambda_{j+1} \leq b_2 \leq \lambda_j \leq \cdots \leq \lambda_1$$

则当 $m \geq i + j$ 时有

$$\frac{\|x^{(m)} - x_*\|_A}{\|x^{(0)} - x_*\|_A} \leq 2 \left(\frac{b-1}{b+1} \right)^{m-i-j} \max_{\lambda \in [b_1, b_2]} \left\{ \prod_{k=n+1-i}^n \left(\frac{\lambda - \lambda_k}{\lambda_k} \right) \prod_{k=1}^j \left(\frac{\lambda_k - \lambda}{\lambda_k} \right) \right\}$$

其中 $b = (b_2/b_1)^{\frac{1}{2}} \geq 1$.

由此可知, 当 b_1 与 b_2 非常接近时, 迭代 $i + j$ 步后, CG 收敛会非常快!

推论

设 A 对称正定, 特征值为

$$\begin{aligned} 0 < \delta \leq \lambda_n \leq \cdots \leq \lambda_{n+1-i} \leq \\ 1 - \varepsilon \leq \lambda_{n-i} \leq \cdots \leq \lambda_{j+1} \leq 1 + \varepsilon \\ \leq \lambda_j \leq \cdots \leq \lambda_1 \end{aligned}$$

则当 $m \geq i + j$ 时有

$$\frac{\|X^{(m)} - X_*\|_A}{\|X^{(0)} - X_*\|_A} \leq 2 \left(\frac{1 + \varepsilon}{\delta} \right)^i \varepsilon^{m-i-j} \quad (7.16)$$

GMRES 算法的收敛性

正规矩阵情形: $A = UAU^*$

定理

设 $A \in \mathbb{R}^{n \times n}$ 是正规矩阵, $x^{(m)}$ 是 GMRES 得到的近似解, 则

$$\frac{\|b - Ax^{(m)}\|_2}{\|r_0\|_2} \leq \min_{q \in \mathbb{P}_m, q(0)=1} \max_{1 \leq i \leq n} |q(\lambda_i)| \quad (7.17)$$

需要指出的是, 上界 (7.17) 是紧凑的.

设 $\Omega \subset \mathbb{C}$ 是包含 A 的所有特征值的一个区域 (不能包含原点), 则

$$\frac{\|b - Ax^{(m)}\|_2}{\|r_0\|_2} \leq \min_{q \in \mathbb{P}_k, q(0)=1} \max_{\lambda \in \Omega} |q(\lambda)|$$

通常 Ω 必须是连通的, 否则求解非常困难, 即使两个区间的并都没法求解.

非正规情形

设 $A \in \mathbb{R}^{n \times n}$ 可对角化, 即 $A = X\Lambda X^{-1}$, 则

$$\left\| b - Ax^{(k)} \right\|_2 = \min_{x \in x^{(0)} + \mathcal{K}_k(A, r_0)} \|b - Ax\|_2 = \min_{q \in \mathbb{P}_k, q(0)=1} \|q(A)r_0\|_2 \quad (7.18)$$

相类似地, 我们可以得到下面的结论.

定理

设 $A = X\Lambda X^{-1}$ 其中 $X \in \mathbb{C}^{n \times n}$ 非奇异, Λ 是对角矩阵, $x^{(k)}$ 是 GMRES 算法得到的近似解, 则

$$\begin{aligned} \frac{\|b - Ax^{(k)}\|_2}{\|r_0\|_2} &\leq \|X\|_2 \left\| X^{-1} \right\|_2 \min_{q \in \mathbb{P}_k, q(0)=1} \max_{1 \leq i \leq n} |q(\lambda_i)| \\ &= \kappa(X) \min_{q \in \mathbb{P}_k, q(0)=1} \max_{1 \leq i \leq n} |q(\lambda_i)| \end{aligned} \quad (7.19)$$

其中 $\kappa(X)$ 是 X 的谱条件数.

如果 A 接近正规, 则 $\kappa(X) \approx 1$. 此时上界 (7.19) 在一定程度上能描述 GMRES 的收敛速度.

当如果 X 远非正交, 则 $\kappa(X)$ 会很大, 此时该上界就失去实际意义了.

需要指出的是, 上面的分析并不意味着非正规矩阵就一定比正规矩阵收敛慢. 事实上, 对任意一个非正规矩阵, 总存在一个相应的正规矩阵, 使得 GMRES 算法的收敛速度是一样的.

虽然 GMRES 算法的收敛性与系数矩阵的特征值有关, 但显然并不仅仅取决于特征值的分布. 事实上, 我们有下面的结论.

定理

对于任意给定的特征值分布和一条不增的收敛曲线, 则总存在一个矩阵 A 和一个右端项 b , 使得 A 具有指定的特征值分布, 且 GMRES 算法的收敛曲线与给定的收敛曲线相同.

考虑线性方程组 $Ax = b$ 其中

$$A = \begin{bmatrix} 0 & 1 & & & \\ & 0 & 1 & & \\ & & \ddots & \ddots & \\ & & & 0 & 1 \\ a_0 & a_1 & a_2 & \cdots & a_{n-1} \end{bmatrix}, \quad b = e_1$$

当 $a_0 \neq 0$ 时, A 非奇异. 易知, A 的特征值多项式为

$$p(x) = \lambda^n - a_{n-1}\lambda^{n-1} - a_{n-2}\lambda^{n-2} - \cdots - a_1\lambda - a_0$$

方程组的精确解为

$$x = [-a_1/a_0, 1, 0, \dots, 0]^T$$

以零向量为迭代初值, 则 GMRES 迭代到第 n 步时才收敛. (前 $n-1$ 步残量范数不变)

如果 A 不可以对角化

我们在分析 GMRES 算法的收敛性时, 通常会想办法用一个新的极小化问题来近似原来的极小化问题 (7.18). 当然, 这个新的极小化问题应该比较容易求解的.

事实上, 我们有

$$\begin{aligned}\frac{\|b - Ax^{(k)}\|_2}{\|r_0\|_2} &= \frac{\min_{q \in \mathbb{P}_k, q(0)=1} \|q(A)r_0\|_2}{\|r_0\|_2} \\ &\leq \max_{\|v\|_2=1} \min_{q \in \mathbb{P}_k, q(0)=1} \|q(A)v\|_2 \\ &\leq \min_{q \in \mathbb{P}_k, q(0)=1} \|q(A)\|_2\end{aligned}$$

不等式 (7.20) 右端代表的是在最坏情况下的 GMRES 收敛性, 而且是紧凑的, 即它是所能找到的不依赖于 r_0 的最好上界. 但我们仍然不清楚, 到底是 A 的那些性质决定着这个上界 [?]

可以证明, 当 A 是正规矩阵时, 上界 (7.20) 和 (7.21) 是相等的 [? ?]. 但是, 对于大多数非正规矩阵而言, 这两者是否相等或者非常接近, 迄今仍不太清楚. 最后需要指出的是, 算法的收敛性也依赖于迭代初值和右端项. 所以上定理中的上界描述的都是最坏情况下的收敛速度. 也就是说, 在实际计算中, 算法的收敛速度可能会比预想的要快得多.

子空间迭代方法

- 1 Krylov 子空间
- 2 GMRES 算法
- 3 共轭梯度法 (CG)
- 4 收敛性分析
- 5 其它 Krylov 子空间迭代算法

其它 Krylov 子空间迭代算法

对称	CG (1952)	对称正定, 正交投影法 (Galerkin)
	MINRES (1975)	对称不定, 斜投影法 (Petrov-Galerkin)
	SYMMLQ (1975)	对称不定
	SQMR (1994)	对称不定
非对称	FOM (1981)	正交投影法, Arnoldi
	GMRES (1984)	斜投影法 (Petrov-Galerkin), Arnoldi
	BiCG (1976)	双正交 (biorthogonalization)
	QMR (1991)	双正交 (biorthogonalization)
	CGS (1989)	Transpose free
	BiCGStab (1992) Tr	Transpose free, smoother convergence
	TFQMR (1993)	Transpose free, smoother convergence
正规方程	FGMRES (1993)	
正规方程	CGLS (1982)	最小二乘 (法方程)
	LSQR (1982)	最小二乘 (法方程)