

数值分析

数值分析

$$p_{k+1} = r_k + \beta_{k+1} p_k \quad 2n \text{ flops}$$

注: CG至多 n 步即可计算 x^* (在精确运算)可能“幸运中断”即 $p_0 = 0$ 或 $r_{k-1} = 0$.

$$r_{k-1} = 0 \Leftrightarrow x_{k-1} = x^*$$

$$p_k = 0 \Leftrightarrow p_k^T A p_k = 0$$

$$r_{k-1}^T r_{k-1} = r_{k-1}^T (r_{k-1} + \beta_k p_{k-1}) \|r_{k-1}\|^2 = r_{k-1}^T p_k = 0$$

$$\Rightarrow r_{k-1} = 0$$

计算量: 对每个 k , 需要一个 A 乘向量 $+12n$ flops. 设 A 为一个稠密阵, 执行到 n 步, 则需 $2n^3 + 12nflops \approx 2n^3$ 对大的 n .

Cholesky分解: $A = LL^T$, $\frac{1}{3}n^3$ flops, CG = 6倍的Cholesky.

如果 A 稀疏, 即只有少量的非零元。常见的情况 A 只有 $O(n)$ 个非零元。架设 A 每行的平均非零元为 $n_\alpha (<< n)$ 此时 Ax 只

需 $2n_\alpha nflops$. 所以执行 n 步的代

价 $2n_\alpha n^2 + 12n^2 flops << \frac{1}{3}n^3$ for large n .

如果只进行 k 步, (kjin, 通常kjin), 则

需 $2n_\alpha kn + 12kn flops \leq O(n^2) flops$

优点:

- ▶ 计算量小;
- ▶ 存储量: x_k, r_k, p_k, Ap_k , 4个向量非常小;
- ▶ 不对A做任何变换, 唯一的作用是形成和向量乘积, 可能只需存储A的非零元;
- ▶ SOR, Jacobi, G-S, CG中不需估计任何参数。

定理2.2.3(收敛性结果) 令 $A = A^T$ 正定, λ_1 和 λ_n 为A的最大最小特征值, 定义 $k = \frac{\lambda_1}{\lambda_n}$,

则 $\|x_k - x^*\|_A \leq 2\left(\frac{\sqrt{k}-1}{\sqrt{k}+1}\right)^k \|x_0 - x^*\|_A$. k 为A的条件数。 k 越接近于1, CG收敛越快, 比最速下降法快! 当 $k \gg 1$ 时, CG可能收敛很慢。

理论上算法很漂亮, 但在有限精度下计算结果和理论论结果差别往往很大。计算出 $r_j^T r_j \neq O(\epsilon)$ 甚至近似线性相关。同样的问题对 p_k 也成立。 \Rightarrow CG不稳定! \Rightarrow 1952 1970方法被抛弃!

1970: 首先发现CG应该作为一种迭代法使用, 即可对 $k_{j:n}$ 可能有 $\|x_k - x^*\|$ 很小!

结论: CG的不稳定性并不影响其收敛性! 只是延误了收敛性。

两者结合 \Rightarrow CG实用!

书31页，例 $n=5$ ，得到一些结论。

原理：要得到一个算法数值性质的一般结论，比如对CG，需对大量的实验，并且对大规模 $n \gg 1$ 。

如果要找一个算法的缺点，则可对特定的问题实验， n 大小不管。

CG的一般性数值性质：

A的中间特征值分布对CG的收敛速度影响巨大！

典型的情形： $\lambda_1 \gg \lambda_2 \gg \cdots \gg \lambda_n$

则经过几步后，CG的收敛因子将是 $\frac{\sqrt{\frac{\lambda_2}{\lambda_{n-1}}}-1}{\sqrt{\frac{\lambda_2}{\lambda_{n-1}}}+1}$ ，而非 $\frac{\sqrt{\frac{\lambda_1}{\lambda_n}}-1}{\sqrt{\frac{\lambda_1}{\lambda_n}}+1}$

比如 $\lambda_1 = 10^4, \lambda_n = 1, \lambda_2 = \cdots = \lambda_{n-1} = 2$

此时实际上只需 $k=3$ ，即得 $x - 2 = x^*2$)A的特征向量对CG的收敛性影响很大。如果 b 只由 m 个特征向量的组合生成，则CG的 m 步必然收敛 $x_m = x^*$

预处理的CG法(PCG)

$$\|x_k - x^*\|_A \leq 2\left(\frac{\sqrt{k}-1}{\sqrt{k}+1}\right)^k \|x_0 - x^*\|_A$$

其中 $k = \frac{\lambda_1}{\lambda_n} (\geq 1)$

当 k 越接近于 1, 则 CG 可能收敛很快!

当 $k \gg 1$, 则 CG 可能收敛很慢!

⇒ CG 可能不实用。

$$\Rightarrow Ax = b \Leftrightarrow \hat{A}\hat{x} = \hat{b}$$

用 $k(A)$ 和 $k(\hat{A})$ 表示 A 和 \hat{A} 的条件数, 若 $k(\hat{A}) \ll k(A)$, $B\hat{A}\hat{x} = \hat{b}$ 应用 CG, 则收敛一定很快。

此称为预处理(条件)技术.(precondition)

现状:

- ▶ 没有通用的有效的预处理技术
- ▶ 一般很难证明一种具体的预处理为何好
- ▶ 好的预处理技术+迭代法, 可能产生非常有效的算法。

有三种形式:

1. 左预处理技术 $M^{-1}Ax = M^{-1}b$
2. 右预处理技术 $AM^{-1}y = b, x = M^{-1}y$
3. 分裂预处理技术 $M_1^{-1}AM_2^{-1}y = M^{-1}b, x = M_2^{-1}y$

只讨论第三种技术

要求 $M = M_1 M_2 = M^T > 0$

$$M_1 = M_2 = m^{\frac{1}{2}} = C = C^T$$

$$\Rightarrow C^{-1} A C^{-1} y = C^{-1} b$$

据第一章中的定理1.2可知 $C^{-1} A C^{-1}$ 对称正定矩阵！

希望：

1. $k(C^{-1} A C^{-1}) < k(A)$
2. 不显示形式 $C^{-1} A C^{-1}$ ，因代价可能很大
3. $M = C^2$ 比 A “简单”，即解 $Mz = r$ 比 $Ax = b$ 要容易，即很快地高精度计算出 z 。对 $C^{-1} A C^{-1} \hat{x} = C^{-1} b$ 应用CG

CG算法：

1) 选取 $\hat{X}_0 \in R^n$ ，计算 $\hat{r}_0 = \hat{b} - \hat{A} \hat{x}_0$, $\hat{p}_1 = \hat{r}_0$

对 $k=1, 2, \dots$ 直到 $\|\hat{r}_k\| \leq \hat{\epsilon}$

$$\hat{\alpha}_k = \frac{\hat{r}_{k-1}^T \hat{r}_{k-1}}{\hat{p}_k^T \hat{A} \hat{p}_k}$$

$$\hat{x}_k = \hat{x}_{k-1} - \hat{\alpha}_k \hat{p}_k$$

$$\hat{r}_k = \hat{r}_{k-1} - \hat{\alpha}_k \hat{A} \hat{p}_k$$

$$\hat{\beta}_{k+1} = \frac{\hat{r}_k^T \hat{r}_k}{\hat{r}_{k-1}^T \hat{r}_{k-1}}$$

$$\hat{p}_{k+1} = \hat{r}_k + \hat{\beta}_{k+1} \hat{p}_k$$

end

算法中的 $\hat{A} = C^{-1}AC^{-1}$ 要显式形式，如何避免？

定义 $\hat{x}_0 = C_{x_0}$, $\hat{r}_0 = C_{r_0}^{-1}$

$\hat{p}_1 = C_{r_0}^{-1} = C_{p_1}^{-1}$

$\hat{x}_k = C_{x_k}$, $\hat{r}_k = C_{r_k}^{-1}$

$\hat{p}_0 = C_{p_0}$, $\hat{\alpha}_k = \alpha_k$, $\hat{\beta}_{k+1} = \beta_{k+1}$

则得到实用的PCG算法:

1)选择 $x_0 \in R^n$, $r_0 = b - Ax_0$, $p_1 = r_0$, 解 $Mz_0 = r_0$

2)对 $k = 1, 2, \dots$,直到 $\|r_k\| \leq \epsilon$

$$\alpha_k = \frac{r_{k-1}^T z_{k-1}}{p_k^T A p_k}$$

$$x_k = x_{k-1} + \alpha_k p_k$$

$$r_k = r_{k-1} - \alpha_k p_k$$

$$\text{解 } Mz_k = r_k$$

$$\beta_{k+1} = \frac{r_k^T z_k}{r_{k-1}^T z_{k-1}}$$

$$p_{k+1} = z_k \beta_{k+1} p_k$$

end

注: 若 $M=I$, PCG化为CG

$$K(C^{-1}AC^{-1}) < k(A)$$

$Mz=r$ 要比 $Ax=b$ 容易求解得多。希望1)和3)是矛盾的。

当 $M = I, z_k = r_k$ 当 $M=A$ 则 CG 一步即收敛, 但 $Mz = r \Leftrightarrow Ax = b$
对 $A = A^T > 0$ 讨论 M 的选择。

$$Ax = b \Leftrightarrow x = Bx + f$$

构造迭代法

$$x_{k+1} = Bx_k + f, k = 0, 1, \dots$$

如果 $\rho(B)$ (B 的谱半径) ($\rho(B) = \max_{1 \leq i \leq n} |\lambda_i(B)|$) 小于 1, 则 $\lim_{k \rightarrow \infty} x_k = x^*$

$$\rho(B) < 1 \Leftrightarrow \lim_{k \rightarrow \infty} x_k = x^*$$

且 $\rho(B)$ 越小, 则收敛一般越快。

$\Rightarrow B$ 越趋于 0, 则收敛一般越快

$$\text{分裂 } A \text{ 为 } A = M - N \Rightarrow I - M^{-1}A = M^{-1}N$$

由 $(M-N)x = b$ 即可得迭代法

$$x_{k+1} = M^{-1}Nx_k + M^{-1}b = Bx_k + f, k = 0, 1, \dots$$

$$M^{-1}N \approx 0 \Leftrightarrow I - M^{-1}n = M^{-1}A \approx I \Leftrightarrow M \approx A$$

具体的取 $M = D = \text{diag}(a_{11}, a_{22}, \dots, a_{nn}) > 0, N = L + L^T$

$L: A$ 的严格下三角部分, (变号) 得到 Jacobi 迭代

$$\text{法 } M = D = D^{\frac{1}{2}} D^{\frac{1}{2}}$$

取 $C = D^{\frac{1}{2}}$ 得到用方法预处理 CG 的算法。

取 $M = D - L, N = L^T$ 得到 Gauss-Seidel (G-S) 方法.

$$B = M^{-1}N = (D - L)^{-1}L^T$$

当 $A = A^T > 0$, M 和 B 都不对称, $x_{k+1} = Bx_k + f$, 依次计

算 $x_{k+1,1}, \dots, x_{k+1,n}$

使用一次以后, 现在反序计算 $x_{k+1,n}, \dots, x_{k+1,1}$ 。合并得到一个
新的迭代法, 称为对称SG-S。此

时 $M = (D - L)^{-T}L(D - L)^{-1}L^T = M^T > 0$ 。

记 $M = CC^T$. 对 $C^{-1}AC^{-T}$, 可类似推出PCG。另有SSOR(w)

第二类预处理技术:

$A = LL^T$ 。(cholesky分解)

注意: A 大型稀疏

对如此的 A 进行 LL^T 分解, 代价一般也是 $\frac{1}{3n^3}$, L : 满阵

设想 (不完全的分解): 要求 L 保持给定的稀疏结构。如和 A 的
结构相同, 得 $A = \hat{L}\hat{L}^T + E$

不完全的cholesky分解.

ILL^T 分解算法:

$$\tau_{11} = a_{11}^{\frac{1}{2}}$$

$$\text{对 } j = 1, 2, \dots, n \tau_{jj} = (a_{jj} - \sum_{k=1}^{j-1} \tau_{jk}^2)^{\frac{1}{2}}$$

注：分解可能不存在！有多种变形，效果有差别。对一般的 $Ax=b$ 常用左，右预处理

$$M^{-1}Ax = M^{-1}b$$

Jacob: $M=D$

G-S: 令 $A = D - L - U$, $M = (D - L)^{-1}U$

SOR: $M = (D - \omega L)^{-1}((p - \omega)I + \omega U)$

不完全: LU(ILU): $A=LU + E$, $M=LU$

解对称不定 $Ax=b$ 的Lanczos方法