

共轭梯度法(Conjugate Gradient Methods, CG)

本章讨论的共轭梯度法，既是一种求解大规模稀疏线性系统的方法，也是一种求解非线性优化问题的基本方法。本章我们会同时讨论它的线性和非线性算法。

线性的CG法最早由Hestenes和Stiefel在1950年代作为一种针对对称正定线性系统的求解算法引入。有别于Gauss消去法，CG法的效率由系数矩阵的特征分布决定，因此适当的预处理显得格外重要。最早的非线性CG算法则有Fletcher和Reeves于1960年代引入，它的一个早期应用就是针对大规模非线性优化问题。CG法的一个特点是它不需要存储矩阵(作为优化算法时)，同时它的算法效率比最速下降法高。

线性CG法

在本节，我们集中讨论线性算法，因此默认算法和模型都是线性的，不再刻意强调。最早的CG法是针对如下的线性系统的求解算法：

$$Ax = b, \quad (5.1)$$

其中 A 是 $n \times n$ 的对称正定矩阵。问题(5.1)等价于优化问题：

$$\min \phi(x) \stackrel{\text{def}}{=} \frac{1}{2} x^T A x - b^T x, \quad (5.2)$$

显然，问题(5.1)和(5.2)是同解的。这一点不难通过

$$\nabla \phi(x) = Ax - b \stackrel{\text{def}}{=} r(x) \quad (5.3)$$

看出。特别地，对 $x = x_k$ ，我们记

$$r_k = Ax_k - b, \quad (5.4)$$

共轭方向法(conjugate direction methods, CD)

首先给出向量组共轭的概念：向量组 $\{p_0, p_1, \dots, p_{n-1}\}$ 被称为是关于 n 阶对称正定矩阵 A 共轭的，若

$$p_i^T A p_j = 0, \forall i \neq j. \quad (5.5)$$

注：显然，共轭向量组是线性无关向量组，故至多包含 n 个向量。

假设我们已经有一组共轭向量组，那么我们依次沿个向量对 ϕ 的最小值做精确搜索，即

$$x_{k+1} = x_k + \alpha_k p_k, \quad (5.6)$$

其中 α_k 是精确步长

$$\alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k}, \quad (5.7)$$

则不论初值如何取，第 n 步的结果 x_n 就是全局最优解 x^* 。这件事情可以整理成如下定理：

注：这里

$$\begin{aligned}
& (A(x_k + \alpha_k p_k) - b)^T p_k = 0 \\
\Rightarrow & (Ax_k - b)^T p_k + \alpha_k p_k^T A p_k = 0 \\
\Rightarrow & \alpha_k = -\frac{r_k^T p_k}{p_k^T A p_k}.
\end{aligned}$$

定理5.1 对任意 $x_0 \in \mathbb{R}^n$, 由公式(5.6)和(5.7)产生的序列 $\{x_k\}$, n 步收敛至线性系统(5.1)的最优解 x^* 。(同时也是(5.2)的最优解。)

证明: 由 $\{p_i, i = 0, \dots, n-1\}$ 线性无关, 故向量 $x^* - x_0 \in \mathbb{R}^n$ 可由其线性表出, 即

$$x^* - x_0 = \sigma_0 p_0 + \sigma_1 p_1 + \dots + \sigma_{n-1} p_{n-1},$$

其中 $\sigma_k \in \mathbb{R}$, $k = 0, 1, \dots, n-1$ 。上式两边同乘以 $p_k^T A$ 再运用共轭性质(5.5), 有

$$\sigma_k = \frac{p_k^T A(x^* - x_0)}{p_k^T A p_k}. \quad (5.8)$$

注: 由(5.5), $p_k^T A p_i = 0$, 除了 $i = k$ 以外。

若 x_k 由(5.6)和(5.7)产生, 则

$$x_k = x_0 + \alpha_0 p_0 + \alpha_1 p_1 + \dots + \alpha_{k-1} p_{k-1}.$$

同样地, 上式两边同乘以 $p_k^T A$ 再运用共轭性质(5.5), 有

$$p_k^T A(x_k - x_0) = 0,$$

因此

$$p_k^T A(x^* - x_0) = p_k^T A(x^* - x_k + x_k - x_0) = p_k^T A(x^* - x_k) + p_k^T A(x_k - x_0),$$

注: 上式第二项是0。

即

$$p_k^T A(x^* - x_0) = p_k^T A(x^* - x_k) = p_k^T (b - Ax_k) = -p_k^T r_k.$$

注: $Ax^* = b$, $r_k = \nabla \phi(x_k) = Ax_k - b$ 。

由(5.8), $\sigma_k = \alpha_k$ 。□

在进一步讨论如何产生共轭组之前, 我们先考虑一下共轭方向法的几何意义。

如果(5.2)的矩阵 A 是一个对角阵, 那么几何上 $\phi(x)$ 的等高线就是一系列长短轴和坐标轴平行的(超)椭圆。于是我们只要沿各个坐标方向 e_1, e_2, \dots, e_n 做精确搜索, 那么至多 n 步就能收敛到 $\phi(\cdot)$ 的全局最优解。但如果 A 不是对角阵, 那么其等高线仍然是椭圆, 但长短轴不再和坐标轴平行。我们也不能沿坐标方向做 n 步精确搜索就得到全局最优解。书上提供了2D的例子图像, 大家可以自己在Matlab中绘制一下。在这种情况下, 一个明显的思路是通过(线性)坐标变换将系数为 A 的二次型 $\phi(\cdot)$ 变换为对角化二次型, 也即定义 \hat{x} 满足

$$\hat{x} = S^{-1}x, \quad (5.9)$$

其中 S 是 $n \times n$ 矩阵, 且

$$S = [p_0, p_1, \dots, p_{n-1}],$$

这里 S 的列向量 $\{p_0, p_1, \dots, p_{n-1}\}$ 是关于 A 两两共轭的向量组。于是(5.2)对应的变换为

$$\hat{\phi}(\hat{x}) \stackrel{\text{def}}{=} \phi(S\hat{x}) = \frac{1}{2}\hat{x}^T(S^TAS)\hat{x} - (S^Tb)^T\hat{x}.$$

注意这里 S^TAS 是对角阵，因此在此变换下，我们可以实现沿坐标轴 $\hat{e}_1, \hat{e}_2, \dots, \hat{e}_n$ 做 n 步精确搜索得到 $\phi(\hat{x})$ 的全局最优解。等价于在原 x 坐标系下，沿

$$p_{i-1} = S\hat{e}_i$$

做搜索。也即，在 x 坐标系下，沿 $\{p_0, p_1, \dots, p_{n-1}\}$ 做 n 步精确搜索，会得到 $\phi(x)$ 的全局最优解。

这里我们顺便提一下一个接下去会用来简化算法的事实：

$$r_{k+1} = r_k + \alpha_k Ap_k. \quad (5.10)$$

注：由(5.4)，

$$\begin{aligned} r_{k+1} &= Ax_{k+1} - b \\ &= A(x_k + \alpha_k p_k) - b \\ &= Ax_k - b + \alpha_k Ap_k \\ &= r_k + \alpha_k Ap_k. \end{aligned}$$

下面这个定理进一步证明了不但共轭方向法的最终解是最优的，而且它的每一步解 x_k 在其走过的方向张成的子空间中也是最优的。

定理 5.2 子空间扩充(Expanding Subspace Minimization) 令 $x_0 \in \mathbb{R}^n$ 是任意初值，假设序列 $\{x_k\}$ 是由共轭方向法(5.6)和(5.7)产生，则

$$r_k^T p_i = 0, i = 0, 1, \dots, k-1, \quad (5.11)$$

并且 x_k 是正定二次问题 $\phi(x) = \frac{1}{2}x^T Ax - b^T x$ 在子空间

$$\{x | x = x_0 + \text{span}\{p_0, p_1, \dots, p_{k-1}\}\} \quad (5.12)$$

上的最优解。

证明：先证点 \tilde{x} 是 ϕ 在子空间(5.12)的最优解当且仅当

$$r(\tilde{x})^T p_i = 0, i = 0, 1, \dots, k-1.$$

定义

$$h(\sigma) = \phi(x_0 + \sigma_0 p_0 + \dots + \sigma_{k-1} p_{k-1}),$$

其中 $\sigma = (\sigma_0, \sigma_1, \dots, \sigma_{k-1})^T$ 。因为 $h(\sigma)$ 是严格凸的，所以有唯一的极值点 σ^* 满足

$$\frac{\partial h(\sigma^*)}{\partial \sigma_i} = 0, i = 0, 1, \dots, k-1.$$

由链式法则及(5.3)，有

$$r(\tilde{x})^T p_i = \nabla \phi(x_0 + \sigma_0^* p_0 + \dots + \sigma_{k-1}^* p_{k-1})^T p_i = 0, i = 0, 1, \dots, k-1.$$

注：由链式法则，

$$\frac{\partial h(\sigma^*)}{\partial \sigma_i} = \nabla \phi(x_0 + \sigma_0^* p_0 + \dots + \sigma_{k-1}^* p_{k-1})^T p_i, i = 0, 1, \dots, k-1.$$

再证 x_k 满足(5.11)。对 $k = 1$ ，我们知道

$$x_1 = x_0 + \alpha_0 p_0$$

是 ϕ 沿 p_0 方向的最小值，且满足 $r_1^T p_0 = 0$ 。假设对 $i = 0, 1, \dots, k-2$ ，均有 $r_{k-1}^T p_i = 0$ ，则

$$r_k = r_{k-1} + \alpha_{k-1} A p_{k-1},$$

因此由 α_{k-1} 定义(5.7)，有

$$p_{k-1}^T r_k = p_{k-1}^T r_{k-1} + \alpha_{k-1} p_{k-1}^T A p_{k-1} = 0.$$

同理，对 $i = 0, 1, \dots, k-2$ ，有

$$p_i^T r_k = p_i^T r_{k-1} + \alpha_{k-1} p_i^T A p_{k-1} = 0,$$

这里 $p_i^T r_{k-1} = 0$ 是因为归纳假设，而 $\alpha_{k-1} p_i^T A p_{k-1} = 0$ 则是由于共轭性质。由此我们已经证明对 $i = 0, 1, \dots, k-1$ 有 $r_k^T p_i = 0$ 。由归纳法，命题得证。□

这个定理告诉我们， r_k 和之前的全部搜索方向都是正交的。（换言之，第 k 步的残量在之前所有已经搜索过的方向上投影都是零，也即之前的方向上的搜索都已经完成，是最优的。）现在我们只要有一组关于 A 的极大共轭方向组，我们就能用共轭方向法在 n 步完成 $\phi(x)$ 的最小值搜索。比如， A 的全部特征向量显然符合要求。但是，这件事情本身的代价就不低于寻找 $\phi(x)$ 的最小值或者求解线性方程组 $Ax = b$ 。一个更好的方法是去构造一组极大共轭方向组。我们在高等代数中，学习过如何构造极大正交组的Schmidt方法，而现在，Gram-Schmidt方法可以看作是在共轭组上的推广。由此，我们可以在共轭方向法的基础上，一边构建共轭组，一边沿共轭方向搜索 $\phi(x)$ 的全局最优解。这就是共轭梯度法(CG)。

CG法的基本性质

CG法实现的关键是在已知 p_0, p_1, \dots, p_{k-1} 的基础上构建 p_k 。而实际上，和用Gram方法构建正交组一样，我们只需要利用 p_{k-1} 和 r_k 就够了。更早的方向并不需要用到，因为 r_k 的性质， p_k 会自然和它们共轭。令

$$p_k = -r_k + \beta_k p_{k-1}, \quad (5.13)$$

这里我们通过参数 β_k 来调节使得 p_k 在这个组合下和 p_{k-1} 共轭，即两边同乘以 $p_{k-1}^T A$ ，有

$$p_{k-1}^T A p_k = -p_{k-1}^T A r_k + \beta_k p_{k-1}^T A p_{k-1} = 0,$$

于是

$$\beta = \frac{r_k^T A p_{k-1}}{p_{k-1}^T A p_{k-1}}.$$

而对 p_0 ，我们可以选取在 x_0 的最速下降方向。由此，我们得到CG法的具体形式：

算法5.1 (简单CG法, CG-Preliminary Version)

给定 x_0 ;

设置 $r_0 \leftarrow Ax_0 - b$, $p_0 \leftarrow -r_0$, $k \leftarrow 0$;

while $r_k \neq 0$

$$\alpha_k \leftarrow -\frac{r_k^T p_k}{p_k^T A p_k}; \quad (5.14a)$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k; \quad (5.14b)$$

$$r_{k+1} \leftarrow Ax_{k+1} - b; \quad (5.14c)$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T A p_k}{p_k^T A p_k}; \quad (5.14d)$$

$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1}p_k; \quad (5.14e)$$

$$k \leftarrow k + 1;$$

end while;

这个算法是之前叙述的直接表示，适合用于理解CG法的性质。但是用于实际计算和编程时，应该采用我们后面将会介绍的进阶版本。下面的定理给出了CG法的两个重要性质：首先，全部残量 r_i 是两两正交的， $i = 0, 1, \dots, n-1$ ；其次，直到 k 步的残量 r_k 和搜索方向 p_k 各自分别张成的线性子空间是同一个：

$$K(r_0, k) \stackrel{\text{def}}{=} \text{span}\{r_0, Ar_0, \dots, A^k r_0\}. \quad (5.15)$$

该子空间称为 k 阶克基于 r_0 的雷洛夫子空间(the Krylov subspace of degree k for r_0)。

定理 5.3 假设CG迭代的第 k 步没有收敛到 $\phi(x)$ 的全局极小值 x^* ，则有：

$$r_k^T r_i = 0, i = 0, 1, \dots, k-1, \quad (5.16)$$

$$\text{span}\{r_0, r_1, \dots, r_k\} = \text{span}\{r_0, Ar_0, \dots, A^k r_0\}, \quad (5.17)$$

$$\text{span}\{p_0, p_1, \dots, p_k\} = \text{span}\{r_0, Ar_0, \dots, A^k r_0\}, \quad (5.18)$$

$$p_k^T A p_i = 0, i = 0, 1, \dots, k-1. \quad (5.19)$$

因此，序列 $\{x_k\}$ 至多 n 步收敛到 x^* 。

证明：归纳法(induction)。结论(5.17)和(5.18)对 $k=0$ 是平凡的(trivial)。而由算法(5.19)对 $k=1$ 成立。现假设以上结论对 k 成立（归纳假设，the induction hypothesis），则需证 $k+1$ 的情形亦成立。

事实上，由归纳假设，

$$r_k \in \text{span}\{r_0, Ar_0, \dots, A^k r_0\}, p_k \in \text{span}\{r_0, Ar_0, \dots, A^k r_0\},$$

对第二式两边同乘以 A ，有

$$A p_k \in \text{span}\{Ar_0, A^2 r_0, \dots, A^{k+1} r_0\}. \quad (5.20)$$

再由(5.10)及第一式，有

$$r_{k+1} \in \text{span}\{r_0, Ar_0, \dots, A^{k+1} r_0\}.$$

因此

$$\text{span}\{r_0, r_1, \dots, r_{k+1}\} \subset \text{span}\{r_0, Ar_0, \dots, A^{k+1} r_0\}.$$

另一面，由(5.18)，我们有

$$A^{k+1} r_0 = A(A^k r_0) \in \text{span}\{A p_0, A p_1, \dots, A p_k\},$$

再由(5.10)，

$$A p_i = (r_{i+1} - r_i)/\alpha_i, i = 0, 1, \dots, k,$$

于是

$$A^{k+1} r_0 \in \text{span}\{r_0, r_1, \dots, r_{k+1}\},$$

即

$$\text{span}\{r_0, Ar_0, \dots, A^{k+1} r_0\} \subset \text{span}\{r_0, r_1, \dots, r_{k+1}\},$$

因此(5.17)得证。

而用类似的从 k 到 $k+1$ 的推导可证明(5.18)，即：

$$\begin{aligned}
& \text{span}\{p_0, p_1, \dots, p_k, p_{k+1}\} \\
= & \text{span}\{p_0, p_1, \dots, p_k, r_{k+1}\} \quad \text{由 (5.14e)} \\
= & \text{span}\{r_0, Ar_0, \dots, A^k r_0, r_{k+1}\} \quad \text{由归纳假设} \\
= & \text{span}\{r_0, r_1, \dots, r_k, r_{k+1}\} \quad \text{由 (5.17)} \\
= & \text{span}\{r_0, Ar_0, \dots, A^{k+1} r_0\}. \quad \text{由 (5.17)}
\end{aligned}$$

接下去继续归纳证明(5.19)。对(5.14e)两边同乘以 Ap_i , $i = 0, 1, \dots, k$, 有

$$p_{k+1}^T Ap_i = -r_{k+1}^T Ap_i + \beta_{k+1} p_k^T Ap_i. \quad (5.21)$$

由 β_k 定义(5.14d), 上式右端当 $i = k$ 时为零。而对 $i \leq k-1$, 由(5.19)的归纳假设, p_0, p_1, \dots, p_k 均两两共轭, 由定理5.2,

$$r_{k+1}^T p_i = 0, i = 0, 1, \dots, k. \quad (5.22)$$

再由(5.18), 我们有对 $i = 0, 1, \dots, k-1$, 成立

$$\begin{aligned}
Ap_i \in \text{Aspan}\{r_0, Ar_0, \dots, A^i r_0\} &= \text{span}\{Ar_0, A^2 r_0, \dots, A^{i+1} r_0\} \\
&\subset \text{span}\{p_0, p_1, \dots, p_{i+1}\}.
\end{aligned} \quad (5.23)$$

联合(5.22)和(5.23), 我们有

$$r_{k+1}^T Ap_i = 0, i = 0, 1, \dots, k-1,$$

因此对 $i = 0, 1, \dots, k-1$, (5.21)的第一项为零。再由(5.19)的归纳假设, 其第二项也为零。综合有

$$p_{k+1}^T Ap_i = 0, i = 0, 1, \dots, k.$$

也即(5.19)成立。

于是CG法产生的各搜索方向确实是共轭的, 由定理5.1, 此方法至多 n 步收敛至全局最优解。

而对(5.16)的证明不需要归纳法。因为 $\{p_i\}$ 是共轭组, 由(5.11)我们有

$$r_k^T p_i = 0, i = 0, 1, \dots, k-1; k = 1, 2, \dots, n-1.$$

由(5.14e),

$$p_i = -r_i + \beta_i p_{i-1},$$

于是有

$$r_i \in \text{span}\{p_i, p_{i-1}\}, i = 1, 2, \dots, k-1.$$

于是由

$$r_k^T r_0 = -r_k^T p_0 = 0$$

以及(5.11), (5.16)成立。□

注: 这里

$$r_k^T r_i = r_k^T (a_1 p_i + a_0 p_{i-1}),$$

由(5.11), 皆为零。这个定理的证明过程并不复杂, 也不困难, 主要阅读障碍可能来自语言。这是一篇很不错的专业英语范文。

这个证明的起点是 p_0 必须是 x_0 的最速下降方向 $-r_0$, 否则结论不成立。而现在我们知道这里两两共轭的是各搜索方向 p_k , 而各梯度 r_k 实际上是两两正交的。因此"共轭梯度法"其实有歧义。实际上是搜索方向关于 A 共轭而不是梯度。

实用(A Practical Form)共轭梯度法

利用定理5.2和5.3, 我们将CG法改造的更加适用于编程实现。首先由(5.14e)和(5.11), 可以将(5.14a)改为

$$\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}.$$

其次, 由(5.10),

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

于是再次应用(5.14e)和(5.11), 我们可以简化 β_{k+1} 的形式:

$$\beta_{k+1} = \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}.$$

最终, 我们将算法改造成:

算法5.2 (CG)

给定 x_0 ;

设置 $r_0 \leftarrow Ax_0 - b$, $p_0 \leftarrow -r_0$, $k \leftarrow 0$;

while $r_k \neq 0$

$$\alpha_k \leftarrow \frac{r_k^T r_k}{p_k^T A p_k}; \quad (5.24a)$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k; \quad (5.24b)$$

$$r_{k+1} \leftarrow r_k + \alpha_k A p_k; \quad (5.24c)$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}; \quad (5.24d)$$

$$p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k; \quad (5.24e)$$

$$k \leftarrow k + 1;$$

end while;

这里主要减少了矩阵向量乘法的工作量, 同时增加了可以缓存重复实用的中间变量。这里需要指出的是, CG方法主要针对大规模问题, 对于规模不大的问题($n < 10^5$), 直接分解效果可能更好。

收敛率

尽管 n 步精确收敛是一个看上去很不错的收敛速度, 但由于CG法处理的总是大规模问题 ($n \gg 10^5$), 所以实际上这个速度仍然是不够的。幸好, CG法当具备一些性质时, 其实际收敛速度可以远远高于 n 步。

由(5.24b)和(5.18), 我们有

$$\begin{aligned} x_{k+1} &= x_0 + \alpha_0 p_0 + \cdots + \alpha_k p_k \\ &= x_0 + \gamma_0 r_0 + \gamma_1 A r_0 + \cdots + \gamma_k A^k r_0, \end{aligned} \quad (5.25)$$

其中 γ_i 是在基 $\{r_0, A r_0, \cdots, A^k r_0\}$ 下的标出系数。现定义 $P_k^*(\cdot)$ 为系数为 $\gamma_0, \gamma_1, \cdots, \gamma_k$ 的 k 次多项式, 则

$$P_k^*(A) = \gamma_0 I + \gamma_1 A + \cdots + \gamma_k A^k,$$

由(5.25), 我们有

$$x_{k+1} = x_0 + P_k^*(A)r_0. \quad (5.26)$$

我们现在指出全部头 k 步限制在由(5.15)定义的Krylov子空间的算法中(即所有采取这 k 个方向的搜索算法), 算法5.2是最优的。这里, 我们的最优是基于算子范数

$$\|z\|_A^2 = z^T A z. \quad (5.27)$$

由 ϕ 的定义(5.2)以及 x^* 是 ϕ 的全局最优解, 我们有

$$\frac{1}{2}\|x - x^*\| = \frac{1}{2}(x - x^*)^T A(x - x^*) = \phi(x) - \phi(x^*). \quad (5.28)$$

而定理5.2指出 x_{k+1} 是 ϕ 在

$$\{x \mid x = x_0 + \text{span}\{p_0, p_1, \cdots, p_k\}\}$$

上的最优解, 由定理5.3之(5.18), 上述空间就是

$$\{x \mid x_0 + \text{span}\{r_0, Ar_0, \cdots, A^k r_0\}\}.$$

于是在所有以 $\{I, A, A^2, \cdots, A^k\}$ 为基的多项式 $P_k(A)$ 中, P_k^* 是问题

$$\min_{P_k} \|x_0 + P_k(A)r_0 - x^*\|_A \quad (5.29)$$

的最优解。

现在,

$$r_0 = Ax_0 - b = Ax_0 - Ax^* = A(x_0 - x^*),$$

所以

$$x_{k+1} - x^* = x_0 + P_k^*(A)r_0 - x^* = [I + P_k^*(A)A](x_0 - x^*). \quad (5.30)$$

令 $0 < \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$ 是 A 的特征值, 而 $\nu_1, \nu_2, \cdots, \nu_n$ 是对应特征向量(因此两两正交), 于是

$$A = \sum_{i=1}^n \lambda_i \nu_i \nu_i^T.$$

注: 等价于

$$A = Q\Lambda Q^T,$$

其中 Q 是对角阵, 其列向量为特征向量, $\Lambda = \text{diag}\{\lambda_1, \lambda_2, \cdots, \lambda_n\}$.

由于特征向量组是 \mathbb{R}^n 的一组基, 我们有

$$x_0 - x^* = \sum_{i=1}^n \xi_i \nu_i, \quad (5.31)$$

这里 ξ_i 是表出系数。由高等代数中的定理, $P_k(\lambda_i)$ 是 $P_k(A)$ 的特征值, 且 ν_i 亦是 $P_k(A)$ 的对用特征向量, 因此

$$P_k(A)\nu_i = P_k(\lambda_i)\nu_i, i = 1, 2, \cdots, n.$$

将上式和(5.31)一起代入(5.30), 有

$$x_{k+1} - x^* = \sum_{i=1}^n [1 + \lambda_i P_k^*(\lambda_i)] \xi_i \nu_i,$$

由

$$\begin{aligned} \|z\|_A^2 &= z^T A z \\ &= z^T \sum_{i=1}^n \lambda_i \nu_i \nu_i^T z \\ &= \sum_{i=1}^n \lambda_i (\nu_i^T z)^2, \end{aligned}$$

我们有

$$\begin{aligned} \|x_{k+1} - x^*\|_A^2 &= \sum_{i=1}^n \lambda_i \left[\nu_i^T \sum_{i=1}^n [1 + \lambda_i P_k^*(\lambda_i)] \xi_i \nu_i \right]^2 \\ &= \sum_{i=1}^n \lambda_i [1 + \lambda_i P_k^*(\lambda_i)]^2 \xi_i^2 \quad (\text{正交性}). \end{aligned} \quad (5.32)$$

由 P_k^* 的最优性, 我们有

$$\begin{aligned} \|x_{k+1} - x^*\|_A^2 &= \min_{P_k} \sum_{i=1}^n \lambda_i [1 + \lambda_i P_k(\lambda_i)]^2 \xi_i^2 \\ &\leq \min_{P_k} \max_{1 \leq i \leq n} [1 + \lambda_i P_k(\lambda_i)]^2 \left(\sum_{j=1}^n \lambda_j \xi_j^2 \right) \\ &= \min_{P_k} \max_{1 \leq i \leq n} [1 + \lambda_i P_k(\lambda_i)]^2 \|x_0 - x^*\|_A^2, \end{aligned} \quad (5.33)$$

这里, 注意到

$$\|x_0 - x^*\|_A^2 = \sum_{j=1}^n \lambda_j \xi_j^2.$$

公式(5.33)提示我们, CG法的收敛速度, 和因子

$$\min_{P_k} \max_{1 \leq i \leq n} [1 + \lambda_i P_k(\lambda_i)]^2 \quad (5.34)$$

相关。

定理 5.4 若 A 只有 r 个互异的特征值, 则CG法 r 步收敛到最优解。

证明: 假设 A 的特征值 $\lambda_1, \lambda_2, \dots, \lambda_n$ 只有 r 个互异的值 $\tau_1 < \tau_2 < \dots < \tau_r$ 。定义多项式

$$Q_r(\lambda) = \frac{(-1)^r}{\tau_1 \tau_2 \cdots \tau_r} (\lambda - \tau_1)(\lambda - \tau_2) \cdots (\lambda - \tau_r),$$

则有

$$Q_r(\lambda_i) = 0, i = 1, 2, \dots, n,$$

以及

$$Q_r(0) = 1.$$

因此 $Q(\lambda) - 1$ 是一个 r 次多项式, 且0是它的根。于是

$$\bar{P}_{r-1}(\lambda) = \frac{Q_r(\lambda) - 1}{\lambda}$$

是一个 $r - 1$ 次多项式。在(5.34)中, 令 $k = r - 1$, 则有

$$0 \leq \min_{P_{r-1}} \max_{1 \leq i \leq n} [1 + \lambda_i P_{r-1}(\lambda_i)]^2 \leq \max_{1 \leq i \leq n} [1 + \lambda_i \bar{P}_{r-1}(\lambda_i)]^2 = \max_{1 \leq i \leq n} Q_r^2(\lambda_i) = 0.$$

因此(5.34)对应的常数对 $k = r - 1$ 为0, 也即

$$\|x_r - x^*\|_A^2 = 0,$$

即 $x_r = x^*$ 成立。□

进一步的分析 (Luenberger[195]) 能给出下面更精确的结果:

定理 5.5 若 A 的特征值为 $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_n$, 我们有

$$\|x_{k+1} - x^*\|_A^2 \leq \left(\frac{\lambda_{n-k} - \lambda_1}{\lambda_{n-k} - \lambda_n} \right)^2 \|x_0 - x^*\|_A^2. \quad (5.35)$$

我们不会在这门课给出这个定理的证明。但是这个定理告诉我们的一个事实是假设 A 的特征值中的 $\lambda_1, \lambda_2, \lambda_{n-m}$ 都是相对接近1的小量, 而其余 m 个则远大于1。那么该定理告诉我们

$$\|x_{m+1} - x^*\| \approx \epsilon \|x_0 - x^*\|,$$

这里 $\epsilon = \lambda_{n-m} - \lambda_1$ 是一个相对小量。

预处理(Preconditioning)

上面的分析给我们的启示是如果能调整 A 的特征值分布, 那么我们能够提高CG法的求解速度。这一技术被称为预处理, 并且在当前

的大规模计算中是必须的手段。它的基本想法仍然是线性变换

$$\hat{x} = Cx. \quad (5.37)$$

于是 ϕ 变为

$$\hat{\phi}(\hat{x}) = \frac{1}{2} \hat{x}^T (C^{-T} A C^{-1}) \hat{x} - (X^{-T} b)^T \hat{x}. \quad (5.38)$$

如果 $C^{-T} A C^{-1}$ 能够有更好的特征值分布, 那么显然 $\hat{\phi}$ 在CG法中会更快收敛。本着这个想法, 我们实际上并不需要显示地做这个变换和逆变换, 而是只需要将等效算子作用在算法中实现即可。或者说, 令对称正定矩阵

$$M = C^T C,$$

我们只需将CG算法改成:

算法5.3 (Preconditioned CG)

给定 x_0 , 预处理子(Preconditioner) M ;

设置 $r_0 \leftarrow Ax_0 - b$;

解线性方程组 $My_0 = r_0$;

设置 $p_0 \leftarrow -y_0$, $k \leftarrow 0$;

while $r_k \neq 0$

$$\alpha_k \leftarrow \frac{r_k^T y_k}{p_k^T A p_k}; \quad (5.39a)$$

$$x_{k+1} \leftarrow x_k + \alpha_k p_k; \quad (5.39b)$$

$$r_{k+1} \leftarrow r_k + \alpha_k A p_k; \quad (5.39c)$$

$$\text{解线性方程组 } M y_{k+1} = r_{k+1}; \quad (5.39d)$$

$$\beta_{k+1} \leftarrow \frac{r_{k+1}^T y_{k+1}}{r_k^T y_k}; \quad (5.39e)$$

$$p_{k+1} \leftarrow -y_{k+1} + \beta_{k+1} p_k; \quad (5.39f)$$

$$k \leftarrow k + 1;$$

end while;

注意当 $M = I$ 是，算法5.3就是算法5.2。也即不做任何预处理。由于预处理几乎是必须的，因此算法5.3才是真正意义上实用的线性CG法。

实用预处理子

很遗憾，预处理子是基于具体问题的，并不存在“最佳”预处理子。但是当我们直接面对矩阵（失去矩阵如何产生的具体物理信息时），有一些常见的预处理办法值得参考。比如对称超松弛(symmetric successive over-relaxion, SSOR)算子，不完全Cholesky(incomplete Cholesky)分解等等。请大家自行参阅文献。

非线性共轭梯度法

当我们的目标函数 f 是一般非线性函数时，CG方法自然失去了 n 步精确收敛的特性，共轭也只发生在局部，因此两两共轭也无从谈起。然而，理论分析和实际测试都表明，按照CG法的思路构建的非线性搜索方法，在实际工作中有不错的效果。

Fletcher-Reeves方法，FR

Fletcher和Reeves在[107]中提出了这种方法，基本思路就是设法将线性CG法移植到非线性问题中。关键改动在于对 α_k 采用了非精确搜索，将 r_k 用局部 ∇f_k 代替。由于预处理失去意义，这里的移植实际上是基于算法5.2的。

算法5.4 (FR)

给定 x_0 ;

计算 $f_0 = f(x_0)$, $\nabla f_0 = \nabla f(x_0)$;

设置 $p_0 \leftarrow -\nabla f_0$, $k \leftarrow 0$;

while $\nabla f_k \neq 0$

 计算 α_k , 用某种非精确策略，如 Wolfe 条件;

$$x_{k+1} \leftarrow x_k + \alpha_k p_k;$$

 计算 ∇f_{k+1} ;

$$\beta_{k+1}^{\text{FR}} \leftarrow \frac{\nabla f_{k+1}^T \nabla f_{k+1}}{\nabla f_k^T \nabla f_k}; \quad (5.41a)$$

$$p_{k+1} \leftarrow -\nabla f_{k+1} + \beta_{k+1}^{\text{FR}} p_k; \quad (5.41b)$$

$$k \leftarrow k + 1;$$

end while;

这个算法看上去更加简单。它本质上会在一个不断共轭的方向上搜索问题的局部最优。但是注意它每一步都更新了局部信息 ∇f_k 的值，因此它的 n 步整体效率有可能优于一步Newton法。（事实上，由于CG法实际迭代步数远小于 n ，因此这里实际等效一步Newton法的步数也同样可能更小。）这里对(5.41a)还有一些其他选择，比如：

Polak-Ribiere方法，PR

$$\beta_{k+1}^{\text{PR}} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{\|\nabla f_k\|^2}. \quad (5.44)$$

一个更加稳定的做法是：

$$\beta_{k+1}^+ = \max\{\beta_{k+1}^{\text{PR}}, 0\}. \quad (5.45)$$

此外还有大量针对一般情形和特殊情形的变形，这里的一个原则是希望这一项能尽可能在计算上稳定，同时能更多引入新的局部信息。比如：

$$\beta_{k+1}^{\text{HS}} = \frac{\nabla f_{k+1}^T (\nabla f_{k+1} - \nabla f_k)}{(\nabla f_{k+1} - \nabla f_k)^T p_k},$$

等等。

重启

之前提到过，非线性CG法 n 步迭代的效果近似于一步Newton法。事实上，这里有一个更加实际的问题，就是局部信息的继承。从 x_0 出发的局部信息，通过迭代被之后的方向和步长所吸收，但是：其一，当经过一定的步数以后，实际上它已经失效了，因为如果 f 是一个正定二次型，它应该已经收敛了；其二，当 x_k 离 x_0 足够远时，就像信任域方法中讨论的那样，这种继承信息不但无助于算法提升效率，反而有害。因此，每隔一定步数 m ，我们应该重新设置整个算法，即将 p_m 重新设置为 $-\nabla f(x_m)$ ，将计数器 k 置为0。这个技术被称为重启(restart)。在实际计算中，对大规模问题，不论时线性（因为受到机器扰动的影响）还是非线性的Krylov子空间迭代法（CG法是其中之一），重启几乎都是必须的。