

章节 6.6 共轭梯度法

线性共轭梯度法

我们考虑二次问题：

$$\min f(x) = \frac{1}{2}x^T A x - b^T x, \quad (118)$$

这里, $A \in \mathbb{R}^{n \times n}$ 。最简单的情形, 如果 A 是对角矩阵, 并且考虑下图中的二维情况下, 我们可以每次沿着坐标轴对二次函数求最小值。这样, 只需要两步即可得到最优解。

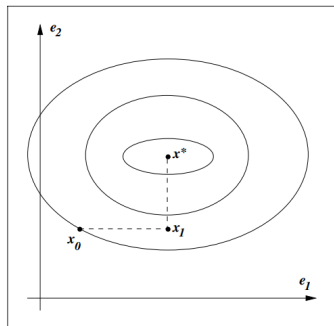


Figure: 图片来源: Numerical optimization, J. Nocedal, S. J. Wright.

但是，若 A 不是对角矩阵，若是仍然沿着坐标轴最小化函数，如图，迭代点不会很快收敛。

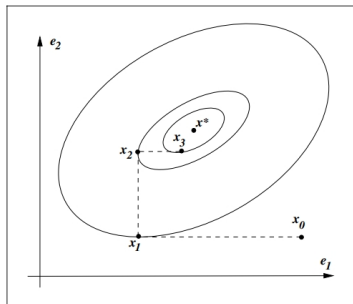


Figure: 图片来源: Numerical optimization, J. Nocedal, S. J. Wright.

共轭方向

定义： 设 A 是 $n \times n$ 正定阵。对于 \mathbb{R}^n 中的任一组非零向量 $\{p_0, p_1, \dots, p_k\}$, 如果 $p_i^T A p_j = 0 (i \neq j)$, 则称 p_0, p_1, \dots, p_k 是关于 A 共轭的。

共轭是正交概念的推广, 当取 $A = I$ 时, 共轭即为正交。

假设 $\{p_0, p_1, \dots, p_{n-1}\}$ 是给定的关于 A 的共轭方向。令 $S = [p_0, p_1, \dots, p_{n-1}]$. 若我们对 $f(x)$ 做变换:

$$\hat{f}(\hat{x}) = f(S\hat{x}) = \frac{1}{2} \hat{x}^T (S^T A S) \hat{x} - (Sb)^T \hat{x}.$$

由共轭方向的定义, 我们知道矩阵 $S^T A S$ 是对角正定矩阵。因此, 对于 $\hat{f}(\hat{x})$ 我们可以在 \hat{x} 的各个坐标轴方向 $\{e_1, e_2, \dots, e_n\}$ 求解最小化问题 (对应于 x 的坐标在 p_0, p_1, \dots, p_{n-1}), 最终得到最优解。

线性共轭梯度方法便是通过构造出 A 的共轭方向 $\{p_0, p_1, \dots, p_{n-1}\}$, 快速求解(118)。当然, 矩阵 A 的特征根方向是共轭方向, 但是若 A 的规模太大, 特征根分解很慢。也可以通过改进 Gram-Schmidt 正交化得到共轭方向, 但是其需要存储所有的 p_0, p_1, \dots, p_{n-1} 。线性共轭梯度法的优点在于, 在某个迭代 k 时, 只需要 p_{k-1} , 即可构造出 p_k , 并不需要所有的 p_0, \dots, p_{k-2} 。

线性共轭梯度方法

由于优化二次函数(118)等价于求解方程 $Ax = b$, 该方法叫做“线性”共轭梯度法。而名字中“梯度”时来源于第一个方向 p_0 取最速下降方向, 即 $p_0 = -\nabla f(x_0)$. 具体的来说, 线性共轭梯度算法如下

算法 线性共轭梯度方法 CG

Require: 初始点 x_0

- 1: $r_0 \leftarrow Ax_0 - b, p_0 \leftarrow -r_0, k \leftarrow 0$
 - 2: **while** $r_k \neq 0$ **do**
 - 3: $\alpha_k \leftarrow \frac{-r_k^T p_k}{p_k^T A p_k}$
 - 4: $x_{k+1} \leftarrow x_k + \alpha_k p_k$
 - 5: $r_{k+1} \leftarrow Ax_{k+1} - b$
 - 6: $\beta_{k+1} \leftarrow \frac{r_{k+1}^T A p_k}{p_k^T A p_k}$
 - 7: $p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k$
 - 8: $k \leftarrow k + 1$
 - 9: **end while**
-

我们下面讲解为何如此选取 $\alpha_k, x_{k+1}, r_{k+1}, p_{k+1}$ 。

- (0) 给定正定阵 A , 选取初始点 x_0 , $p_0 = -\nabla f(x_0)$ 保证第一步为下降方向。
记

$$r_k = Ax_k - b = \nabla f(x_k) \quad (119)$$

- (1) 由于我们想每步迭代在 p_k 方向最小化函数值, 即求精确的一维搜索步长 α_k , 即 $\alpha_k = \arg \min_{\alpha > 0} f(x_k + \alpha p_k)$. 由此可得

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

- (2) 更新迭代点 $x_{k+1} = x_k + \alpha_k p_k$, 并构造 p_{k+1} 是负梯度方向和前一个共轭方向 p_k 的线性组合, 即

$$p_{k+1} = -r_{k+1} + \beta_{k+1} p_k. \quad (120)$$

由于 $p_{k+1}^T A p_k = 0$, 可得

$$\beta_{k+1} = \frac{r_{k+1}^T A p_k}{p_k^T A p_k}$$

虽然构造 p_{k+1} 的方式, 仅保证了 p_k 与 p_{k+1} 互为共轭, 但是如下定理告诉我们, p_{k+1} 与所有的 p_0, \dots, p_k 均共轭。

定理 2.4 (线性共轭梯度法性质)

设线性共轭梯度法的第 k 步迭代的结果 x_k 不是问题 (118) 的解, 那么有以下结论成立

- ① $\text{span}(r_0, r_1, \dots, r_k) = \text{span}(r_0, Ar_0, \dots, A^k r_0)$
- ② $\text{span}(p_0, p_1, \dots, p_k) = \text{span}(r_0, Ar_0, \dots, A^k r_0)$
- ③ $r_k^T p_i = 0, \forall i < k$
- ④ $p_k^T A p_i = 0, \forall i < k$
- ⑤ $r_k^T r_i = 0, \forall i < k$

[思考题: 证明上述定理...]

下面的定理阐述了线性共轭算法的重要性质：

定理 2.5

严格凸二次函数 $f(x) = \frac{1}{2}x^T A x + c^T x$, 共轭方向法执行精确一维搜索, 则每步迭代点 x_{k+1} 是 $f(x)$ 在空间

$$\mathcal{V}_k = \{x \mid x = x_0 + \sum_{j=0}^k \beta_j p_j, \forall \beta_j \in \mathbb{R}\}$$

中的唯一极小点。因此, 最多需要 n 步, x_k 收敛到最优点 $x^* = A^{-1}b$.

证明: 由共轭方向的定义知, $\{p_0, p_1, \dots, p_{n-1}\}$ 线性无关。若 x_n 是 \mathcal{V}_{n-1} 上的最小值, 那么是 \mathbb{R}^n 上的最小值。

若要证明 x_{k+1} 是 \mathcal{V}_k 上最小值, 下面只要证: 对所有 $k < n$ 成立

$$r_{k+1}^T p_j = 0, j = 0, 1, \dots, k.$$

即在点 x_{k+1} 处的函数梯度 $r_{k+1} = \nabla f(x_{k+1})$ 与子空间 $\text{span}\{p_0, p_1, \dots, p_k\}$ 正交。

由线性共轭梯度法性质定理知, 对 $\forall j = 0, 1, \dots, k$ 有如下关系成立

$$r_{k+1}^T p_j = 0.$$

因此, 得证。

线性共轭梯度法

进一步，由以下关系，可以得到更加实用的线性共轭梯度算法。

首先 $p_k^T r_k = (-r_k + \beta_k p_{k-1})^T r_k = -r_k^T r_k$. 又因为

$$r_{k+1} - r_k = A(x_{k+1} - x_k) = \alpha_k A p_k. \quad (121)$$

故(121)两边同时乘以 r_{k+1}^T 和 p_k 分别得 $\alpha_k r_{k+1}^T A p_k = \alpha_k r_{k+1}^T r_{k+1}$, $\alpha_k = \frac{r_k^T r_k}{p_k^T A p_k}$

算法 线性共轭梯度方法 CG (Practical form)

Require: 初始点 x_0

- 1: $r_0 \leftarrow Ax_0 - b, p_0 \leftarrow -r_0, k \leftarrow 0$
 - 2: **while** $r_k \neq 0$ **do**
 - 3: $\alpha_k \leftarrow \frac{-r_k^T p_k}{p_k^T A p_k} \iff \alpha_k \leftarrow \frac{r_k^T r_k}{p_k^T A p_k};$
 - 4: $x_{k+1} \leftarrow x_k + \alpha_k p_k$
 - 5: $r_{k+1} \leftarrow Ax_{k+1} - b$
 - 6: $\beta_{k+1} \leftarrow \frac{r_{k+1}^T A p_k}{p_k^T A p_k} \iff \beta_{k+1} \leftarrow \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k}$
 - 7: $p_{k+1} \leftarrow -r_{k+1} + \beta_{k+1} p_k$
 - 8: $k \leftarrow k + 1$
 - 9: **end while**
-

结论 1: 若 A 有特征根 $\lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n$, 我们有

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

该结论说明共轭梯度法的收敛速度和特征根的分布有关。若 A 有 m 个较大的特征根, 剩余的 $n - m$ 的特征根都约等于 1。令 $\epsilon = \lambda_{n-m} - \lambda_1$. 那么, 只有在 $m + 1$ 步后, 我们有

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

$m + 1$ 步后共轭梯度法收敛较快, 而之前都比较慢。

此外, 我们还有如下结论:

结论 2: 记 $\kappa = \|A\|_2 \|A^{-1}\|$ 为矩阵 A 的条件数, 那么

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

相较于梯度法, 系数 $\frac{\sqrt{\kappa}-1}{\sqrt{\kappa}+1}$ 比梯度法的 $\frac{\kappa-1}{\kappa+1}$ 更好。

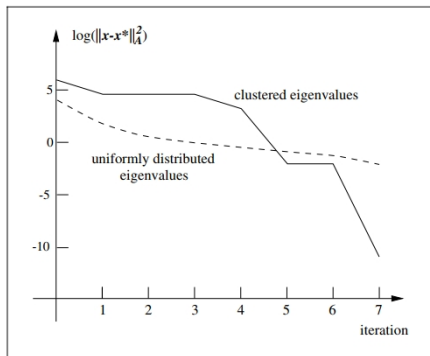


Figure: 共轭梯度法：矩阵 A 有较均匀特征根和集中的特征根（图片来源：Numerical optimization）

预条件方法的主要思路是对问题做一个线性变换, 使得新线性系统矩阵的条件数降低, 并且矩阵的特征根分布更加均匀。对 x 做变换 $\hat{x} = Cx$, 这里 C 是一个可逆矩阵。考虑最小化

$$g(\hat{x}) = f(C^{-1}\hat{x}) = \frac{1}{2}\hat{x}^T(C^{-T}AC^{-1})\hat{x} - (C^{-T}b)^T\hat{x}.$$

通过选取适当的 C , 可以使得新矩阵 $C^{-T}AC^{-1}$ 特征根分布更加均衡, 即理想情况下, 让 $C^{-T}AC^{-1} \approx I$ 。例如, PDE 数值求解种, 可以通过取 $C^TC \approx A$, 即 C^TC 为 A 的三对角部分。

将共轭梯度法推广到非二次函数的极小化问题，其迭代为

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

步长 α_k 由精确或者非精确一维搜索决定， p_{k+1} 的构造如下：

$$p_{k+1} = -r_{k+1} + \beta_{k+1} p_k.$$

同样，我们记 $r_k = \nabla f(x_k)$.

有如下 4 种最为经典的选取方式

$$\beta_k := \frac{r_{k+1}^T r_{k+1}}{r_k^T r_k} \quad (\text{Fletcher} - \text{Reeves})$$

$$\beta_k := \frac{r_{k+1}^T (r_{k+1} - r_k)}{\rho_k^T (r_{k+1} - r_k)} \quad (\text{Hestenes} - \text{Stiefel})$$

$$\beta_k := \frac{r_{k+1}^T (r_{k+1} - r_k)}{r_k^T r_k} \quad (\text{Polak} - \text{Ribiere} - \text{Polyak})$$

$$\beta_k := \frac{r_{k+1}^T r_{k+1}}{\rho_k^T (r_{k+1} - r_k)} \quad (\text{Dai} - \text{Yuan})$$

引理 2.6

设 $\{x_k\}$ 为使用 *Fletcher-Reeves* 格式 (也即 $\beta_{k+1} = \frac{\nabla f(x_{k+1})^T \nabla f(x_{k+1})}{\nabla f(x_k)^T \nabla f(x_k)}$) 非线性共轭梯度法得到的迭代点序列。 α_k 为非精确线搜索强 *Wolfe* 条件得到的步长, *Wolfe* 条件的系数满足 $0 < c_1 < c_2 < 0.5$, 那么搜索方向 p_k 满足

$$-\frac{1}{1-c_2} \leq \frac{\nabla f(x_k)^T p_k}{\|\nabla f(x_k)\|^2} \leq \frac{2c_2-1}{1-c_2}. \quad (122)$$

因此, p_k 为下降方向。

证明: 我们用归纳法证明, 首先, 对于 $k=0$, 因为 $p_0 = -\nabla f(x_0)$ 显然成立。如果对于 k 成立, 对于 $k+1$, 注意 $p_{k+1} = -\nabla f(x_{k+1}) + \beta_{k+1} p_k$, 有

$$\frac{\nabla f(x_{k+1})^T p_{k+1}}{\|\nabla f(x_{k+1})\|^2} = -1 + \beta_{k+1} \frac{\nabla f(x_{k+1})^T p_k}{\|\nabla f(x_{k+1})\|^2} = -1 + \frac{\nabla f(x_{k+1})^T p_k}{\|\nabla f(x_k)\|^2}$$

最后的一项是因为 $\beta_{k+1} = \frac{\|\nabla f(x_{k+1})\|^2}{\|\nabla f(x_k)\|^2}$ 。由强 *Wolfe* 条件中的第二个不等式, 我们有

$$\left| \nabla f(x_{k+1})^T p_k \right| \leq -c_2 \nabla f(x_k)^T p_k$$

因此

$$-1 + c_2 \frac{\nabla f(x_k)^T p_k}{\|\nabla f(x_k)\|^2} \leq -1 + \frac{\nabla f(x_{k+1})^T p_k}{\|\nabla f(x_k)\|^2} \leq -1 - c_2 \frac{\nabla f(x_k)^T p_k}{\|\nabla f(x_k)\|^2}$$

代入我们的归设即可得到结论。

非线性共轭梯度法

Fletcher-Reeves 格式的问题:

定义 $\cos \theta_k = \frac{-\nabla f(x_k)^T p_k}{\|\nabla f(x_k)\| \|p_k\|}$. 若我们有 $\cos \theta_k \approx 0$, 即 p_k 是一个比较差的下降方向。
在(122)两边同时乘 $\frac{\|\nabla f(x_k)\|}{\|p_k\|}$, 得到

$$\frac{1 - 2c_2}{1 - c_2} \frac{\|\nabla f_k\|}{\|p_k\|} \leq \cos \theta_k \leq \frac{1}{1 - c_2} \frac{\|\nabla f_k\|}{\|p_k\|}, \quad \text{for all } k = 0, 1, \dots$$

以上不等式告诉我们 $\cos \theta_k \approx 0$ 当且仅当

$$\|\nabla f(x_k)\| \ll \|p_k\|.$$

因为 p_k 几乎正交于梯度, 那么 x_k 到 x_{k+1} 的移动非常小, 即, $x_{k+1} \approx x_k$. 那么, 我们有 $\nabla f_{k+1} \approx \nabla f_k$, 因此

$$\beta_{k+1}^{\text{FR}} \approx 1$$

由上述结果以及 $\|\nabla f_{k+1}\| \approx \|\nabla f_k\| \ll \|p_k\|$, 我们得到

$$p_{k+1} \approx p_k,$$

所以新的搜索方向几乎与之前的相同。从而, 算法陷入停滞。

Polak-Ribiere (PR) 算法在上述 Fletcher-Reeves (FR) 遇到的问题中, 表现大不相同。假设 $\cos \theta_k \approx 0$, 那么 $\nabla f_{k+1} \approx \nabla f_k$ 得到

$$\beta_{k+1}^{\text{PR}} \approx 0.$$

由

$$p_{k+1} = -r_{k+1} + \beta_k p_k.$$

可知新的 $p_{k+1} \approx \nabla f(x_{k+1})$. 也就是说, 算法选择了负梯度方向重新开始。Hestenes-Stiefel (HS) 同样重新开始。但是, PR 和 HS 算法并没有全局收敛性。主要因为所产生的搜索方向 $p_{k+1} = -r_{k+1} + \beta_k p_k$ 可能不再是下降方向。

因此, 对于 FR 算法, 我们应该周期性采用梯度下降方向作为搜索方向, 例如, 每 n 步重新选择负梯度方向作为搜索方向即令 $p_{(\ell n)} = -r_{(\ell n)}$, $\ell = 1, 2, \dots$

这种策略称为重启策略, 这样的共轭梯度法也称作重启共轭梯度法。Fletcher-Reeves 方法在实现中必须使用重启。实际中, 也可以通过判定比值

$$\frac{\nabla f(x_{k+1})^T \nabla f(x_k)}{\|\nabla f(x_{k+1})\|^2}$$

确定是否需要重启。如果比值很小, 则说明梯度相差较大, 是理想的情况; 反之则需重启。

Dai(戴彧虹)-Yuan (袁亚湘) 方法, 是另一个可以保证全局收敛的算法, 仅需要弱 Wolfe 线搜索条件。

以上 4 种方法, 各有优势。实际中还需结合具体问题确定方法。

从实际计算效率及稳定性来看, 共轭梯度法未必比拟牛顿法好。但是, 共轭梯度法中搜索方向的计算仅仅用到目标函数的梯度, 而不必像拟牛顿法那样在每次迭代中更新 Hesse 矩阵 (或其逆) 的近似阵并记忆之。所以, 当问题的规模大而且有稀疏结构时, 共轭梯度法有高效执行计算的好处。

令 $m = 1$, $H_k^0 \equiv I_n$, BFGS 更新公式变为了

$$H_{k+1} = \left(I_n - \frac{s_k y_k^\top}{y_k^\top s_k} \right) \left(I_n - \frac{y_k s_k^\top}{y_k^\top s_k} \right) + \frac{s_k s_k^\top}{y_k^\top s_k}.$$

使用精确线搜索时我们总有 $d_k^\top \nabla f(x_{k+1}) = 0$, $s_k^\top \nabla f(x_{k+1}) = 0$, 于是

$$d_{k+1} = -H_{k+1} \nabla f(x_{k+1}) = -\nabla f(x_{k+1}) + \frac{s_k y_k^\top}{y_k^\top s_k} \nabla f(x_{k+1}).$$

$$\text{注意到 } \frac{s_k y_k^\top}{y_k^\top s_k} \nabla f(x_{k+1}) = \frac{y_k^\top \nabla f(x_{k+1})}{y_k^\top s_k} s_k = \frac{(g_{k+1} - g_k)^\top g_{k+1}}{y_k^\top d_k} d_k = -\frac{(g_{k+1} - g_k)^\top g_{k+1}}{g_k^\top d_k} d_k$$

, 而分母为 $g_k^\top d_k = g_k^\top (-g_k + c d_{k-1}) = -g_k^\top g_k$, 故

$$d_{k+1} = -g_{k+1} + \frac{(g_{k+1} - g_k)^\top g_{k+1}}{g_k^\top g_k} d_k.$$

Polak-Ribiere 法的共轭梯度的更新为:

$$d_{k+1} = -g_{k+1} + \beta_k d_k, \quad \beta_k = \frac{(g_{k+1} - g_k)^\top g_{k+1}}{g_k^\top g_k}.$$

故两种方法在此时等价。