数值最优化与 Matlab 程序设计

第四章 共轭梯度法

前面介绍的最速下降法和牛顿法都具有其自身的局限性. 本章将要介绍的共轭梯度法是介于最速下降法与牛顿法之间的一种无约束优化算法, 它具有超线性收敛速度, 而且算法结构简单, 容易编程实现. 此外, 跟最速下降法相类似, 共轭梯度法只用到了目标函数及其梯度值, 避免了二阶导数 (Hesse 阵) 的计算, 从而降低了计算量和存储量, 因此它是求解无约束优化问题的一种比较有效而实用的算法.

§4.1 共轭方向法

共轭方向法的基本思想是在求解 n 维正定二次目标函数极小点时产生一组共轭方向作为搜索方向, 在精确线搜索条件下算法至多迭代 n 步即能求得极小点. 经过适当的修正后共轭方向法可以推广到求解一般非二次目标函数情形. 下面先介绍共轭方向的概念.

定义 4.1 设 G 是 n 阶对称正定矩阵, 若 n 维向量组 $d_1, d_2, \cdots, d_m$

$(m \leq n)$ 满足

$$d_i^T G d_j = 0, \quad i \neq j,$$

则称 $d_1, d_2, \cdots, d_m$ 是 G 共轭的.

显然, 向量组的共轭是正交的推广, 即当 $G = I$(单位阵) 时, 上述定义变成向量组正交的定义. 此外, 不难证明, 对称正定矩阵 G 的共轭向量组必然是线性无关的.

下面我们考虑求解正定二次目标函数极小点的共轭方向法. 设

$$\min f(x) = \frac{1}{2} x^T G x + b^T x + c, \tag{4.1}$$

其中 G 是 n 阶对称正定阵, b 为 n 维常向量, c 为常数. 我们有下面的算法:

算法 4.1 (共轭方向法)

步 0 给定迭代精度 $0 \leq \varepsilon \ll 1$ 和初始点 x_0. 计算 $g_0 = \nabla f(x_0)$. 选取初始方向 d_0 使得 $d_0^T g_0 < 0$. 令 $k := 0$.

步 1 若 $\|g_k\| \leq \varepsilon$, 停算, 输出 $x^* \approx x_k$.

步 2 利用精确线搜索方法确定搜索步长 α_k.

步 3 令 $x_{k+1} := x_k + \alpha_k d_k$, 并计算 $g_{k+1} = \nabla f(x_{k+1})$.

步 4 选取 d_{k+1} 满足下降性和共轭性条件:

$$d_{k+1}^T g_{k+1} < 0, \quad d_{k+1}^T G d_i = 0, \quad i = 0, 1, \cdots, k.$$

步 5 令 $k := k + 1$, 转步 1.

下面给出算法 4.1 的收敛性定理.

定理 4.1 设目标函数 f 由 (4.1) 定义. $\{x_k\}$ 是算法 4.1 产生的迭代序列. 则每一步迭代点 x_{k+1} 都是 $f(x)$ 在 x_0 和方向 $d_0, d_1, \cdots, d_k$

所张成的线性流形

$$S_k = \left\{ x \,\middle|\, x = x_0 + \sum_{i=0}^{k} \alpha_i d_i,\ \forall \alpha_i \right\}$$

中的极小点. 特别, $x_n = x^* = -G^{-1}b$ 是问题 (4.1) 的唯一极小点.

证 由算法 4.1 可知, $d_0, d_1, \cdots, d_{n-1}$ 是 G 共轭的, 因而是线性无关的, 故有 $S_{n-1} = \mathbb{R}^n$. 于是我们只需证明 x_{k+1} 是 f 在线性流形 S_k 中的极小点即可.

显然有

$$x_{k+1} = x_k + \alpha_k d_k = \cdots = x_0 + \sum_{i=0}^{k} \alpha_i d_i \in S_k.$$

另一方面, 对任何 $x \in S_k$, 存在 $\beta_i \in \mathbb{R}$, $i = 0, 1, \cdots, k$, 使得

$$x = x_0 + \sum_{i=0}^{k} \beta_i d_i.$$

记

$$h_{k+1} = x - x_{k+1} = \sum_{i=0}^{k} (\beta_i - \alpha_i) d_i.$$

利用泰勒展开公式, 有

$$\begin{aligned} f(x) &= f_{k+1} + g_{k+1}^T h_{k+1} + \frac{1}{2} h_{k+1}^T G h_{k+1} \\ &\geq f_{k+1} + g_{k+1}^T h_{k+1} \\ &= f_{k+1} + \sum_{i=0}^{k} (\beta_i - \alpha_i) g_{k+1}^T d_i. \end{aligned}$$

下面只需证明

$$g_{k+1}^T d_i = 0, \quad \forall\, i = 0, 1, \cdots, k \tag{4.2}$$

即可. 事实上, 因

$$g_{j+1} - g_j = G(x_{j+1} - x_j) = \alpha_j G d_j,$$

故当 $i < k$ 时有

$$\begin{aligned} g_{k+1}^T d_i &= g_{i+1}^T d_i + \sum_{j=i+1}^{k} (g_{j+1} - g_j)^T d_i \\ &= g_{i+1}^T d_i + \sum_{j=i+1}^{k} \alpha_j d_j^T G d_i = 0, \end{aligned}$$

其中上式的第一项与求和项为 0 分别由精确线搜索和共轭性得到. 当 $i = k$ 直接由精确线搜索可得 $g_{k+1}^T d_k = 0$. 从而 (4.2) 式成立. 至此, 定理的结论已经得到证明. □

注 从定理 4.1 可知, 在精确线搜索下, 用算法 4.1 求解正定二次目标函数极小化问题 (4.1), 至多在 n 步内即可求得其唯一的极小点. 这种能在有限步内求得二次函数极小点的性质通常称为二次终止性.

§4.2 共轭梯度法

共轭梯度法是在每一迭代步利用当前点处的最速下降方向来生成关于凸二次函数 f 的 Hesee 阵 G 的共轭方向, 并建立求 f 在 $\mathbb{R}^n$ 上的极小点的方法. 这一方法最早是由 Hesteness 和 Stiefel 于 1952 年为求解对称正定线性方程组而提出来的, 后经 Fletcher 等人研究并应用于无约束优化问题取得了丰富的成果, 共轭梯度法也因此成为当前求解无约束优化问题的重要算法类.

设函数 f 由 (4.1) 所定义, 则 f 的梯度和 Hesse 阵分别为

$$g(x) = \nabla f(x) = Gx + b, \quad G(x) = \nabla^2 f(x) = G. \tag{4.3}$$

下面我们来讨论算法 4.1 中共轭方向的构造. 我们取初始方向 d_0 为初始点 x_0 处的负梯度方向, 即

$$d_0 = -\nabla f(x_0) = -g_0. \tag{4.4}$$

从 x_0 出发沿 d_0 方向进行精确线搜索得求步长 α_0, 令

$$x_1 = x_0 + \alpha_0 d_0,$$

其中 α_0 满足条件

$$\nabla f(x_1)^T d_0 = g_1^T d_0 = 0. \tag{4.5}$$

在 x_1 处, 用 f 在 x_1 的负梯度方向 $-g_1$ 与 d_0 的组合来生成 d_1, 即

$$d_1 = -g_1 + \beta_0 d_0, \tag{4.6}$$

然后选取系数 β_0 使 d_1 与 d_0 关于 G 共轭, 即令

$$d_1^T G d_0 = 0 \tag{4.7}$$

来确定 β_0. 将 (4.6) 代入 (4.7) 得

$$\beta_0 = \frac{g_1^T G d_0}{d_0^T G d_0}. \tag{4.8}$$

由 (4.3) 得

$$g_1 - g_0 = G(x_1 - x_0) = \alpha_0 G d_0. \tag{4.9}$$

另外, 由定理 4.1 可知 $g_2^T d_i = 0\,(i = 0, 1)$. 故由 (4.4)∼(4.6) 可得

$$g_2^T g_0 = 0, \quad g_2^T g_1 = 0, \quad d_0^T g_0 = -g_0^T g_0, \quad d_1^T g_1 = -g_1^T g_1.$$

现假设已得到相互共轭的搜索方向 $d_0, d_1, \cdots, d_{k-1}$, 精确线搜索得到的步长为 $\alpha_0, \alpha_1, \cdots, \alpha_{k-1}$, 且满足

$$\begin{cases} d_{k-1}^T G d_i = 0, & i = 0, 1, \cdots, k-2, \\ d_i^T g_i = -g_i^T g_i, & i = 0, 1, \cdots, k-1, \\ g_k^T g_i = 0, \quad g_k^T d_i = 0, & i = 0, 1, \cdots, k-1. \end{cases} \tag{4.10}$$

现令

$$d_k = -g_k + \beta_{k-1} d_{k-1} + \sum_{i=0}^{k-2} \beta_k^{(i)} d_i, \tag{4.11}$$

其中 β_{k-1}, $\beta_k^{(i)}$ $(i = 0, 1, \cdots, k-2)$ 的选择要满足

$$d_k^T G d_i = 0, \quad i = 0, 1, \cdots, k-1. \tag{4.12}$$

用 $d_i^T G$ $(i = 0, 1, \cdots, k-1)$ 左乘 (4.11) 得

$$\beta_{k-1} = \frac{g_k^T G d_{k-1}}{d_{k-1}^T G d_{k-1}}, \quad \beta_k^{(i)} = \frac{g_k^T G d_i}{d_i^T G d_i}, \quad i = 0, 1, \cdots, k-2. \tag{4.13}$$

类似于 (4.9), 我们有

$$g_{i+1} - g_i = G(x_{i+1} - x_i) = \alpha_i G d_i, \quad i = 0, 1, \cdots, k-1,$$

及

$$\alpha_i G d_i = g_{i+1} - g_i, \quad i = 0, 1, \cdots, k-1. \tag{4.14}$$

于是由归纳法假设 (4.10) 可得

$$\beta_k^{(i)} = \frac{g_k^T G d_i}{d_i^T G d_i} = \frac{g_k^T (g_{i+1} - g_i)}{d_i^T (g_{i+1} - g_i)} = 0, \quad i = 0, 1, \cdots, k-2.$$

于是, 第 k 步的搜索方向为

$$d_k = -g_k + \beta_{k-1} d_{k-1}, \tag{4.15}$$

其中 β_{k-1} 由 (4.13) 确定, 即

$$\beta_{k-1} = \frac{g_k^T G d_{k-1}}{d_{k-1}^T G d_{k-1}}. \tag{4.16}$$

同时有 $d_k^T g_k = -g_k^T g_k$. 这样 (4.4), (4.15) 和 (4.16) 确定了一组由负梯度方向形成的共轭方向, 而把沿着这组方向进行迭代的方法称之为共轭梯度法.

上面得推导实际上已经证明了下述结论:

定理 4.2 对于正定二次函数的极小化问题 (4.1), 由 (4.4), (4.15) 和 (4.16) 确定搜索方向 d_k 并采用精确线搜索确定步长因子 α_k 的共轭方向法, 至多 n 步迭代即可求得问题 (4.1) 的极小点, 并且对所有

的 $k\,(1 \le k \le n)$, 有

$$
\begin{aligned}
& d_k^T G d_i = 0, \quad i = 0, 1, \cdots, k-1, \\
& d_i^T g_i = -g_i^T g_i, \quad i = 0, 1, \cdots, k, \\
& g_{k+1}^T g_i = 0, \ g_{k+1}^T d_i = 0, \quad i = 0, 1, \cdots, k.
\end{aligned}
$$

为了使算法能够适应于求解非二次目标函数的极小点, 需要设法消去 (4.16) 中的矩阵 G. 由定理 4.2 及 (4.14) 得

$$
\begin{aligned}
\alpha_{k-1} g_k^T G d_{k-1} &= g_k^T (g_k - g_{k-1}) = g_k^T g_k, \\
\alpha_{k-1} d_{k-1}^T G d_{k-1} &= (-g_{k-1} + \beta_{k-2} d_{k-2})^T (g_k - g_{k-1}) \\
&= g_{k-1}^T g_{k-1}.
\end{aligned}
$$

由此, (4.16) 可化为

$$
\beta_{k-1} = \frac{g_k^T g_k}{g_{k-1}^T g_{k-1}}. \tag{4.17}
$$

下面我们给出共轭梯度法求解无约束优化问题 (4.1) 极小点的算法步骤.

算法 4.2 (共轭梯度法)

步 0 给定迭代精度 $0 \leq \varepsilon \ll 1$ 和初始点 x_0. 计算 $g_0 = \nabla f(x_0)$. 令 $k := 0$.

步 1 若 $\|g_k\| \leq \varepsilon$, 停算, 输出 $x^* \approx x_k$.

步 2 计算搜索方向 d_k :

$$d_k = \begin{cases} -g_k, & k = 0, \\ -g_k + \beta_{k-1} d_{k-1}, & k \geq 1, \end{cases}$$

其中当 $k \geq 1$ 时, β_{k-1} 由 (4.17) 确定.

步 3 利用精确线搜索方法确定搜索步长 α_k.

步 4 令 $x_{k+1} := x_k + \alpha_k d_k$, 并计算 $g_{k+1} = \nabla f(x_{k+1})$.

步 5 令 $k := k + 1$, 转步 1.

注 公式 (4.17) 是由 Fletcher 和 Reeves 给出的, 故称之为 Fletcher-Reeves 公式, 简称 FR 公式, 算法 4.2 也称之为 FR 共轭梯度法. 除 FR 公式外, 尚有下列著名公式:

$$\beta_k = \frac{g_{k+1}^T g_{k+1}}{-d_k^T g_k}, \quad \text{(Dixon 公式)},$$

$$\beta_k = \frac{g_{k+1}^T g_{k+1}}{d_k^T (g_{k+1} - g_k)}, \quad \text{(Dai-Yuan 公式)},$$

$$\beta_k = \frac{g_{k+1}^T (g_{k+1} - g_k)}{d_k^T (g_{k+1} - g_k)}, \quad \text{(Hesteness-Stiefel, HS 公式)},$$

$$\beta_k = \frac{g_{k+1}^T (g_{k+1} - g_k)}{g_k^T g_k}, \quad \text{(Polak, Ribiére, Polyak, PRP 公式)}.$$

下面我们来证明算法 4.2 的收敛性定理.

定理 4.3 设 $\{x_k\}$ 是由算法 4.2 产生的序列, 假定函数 $f(x)$ 一阶连续可微且水平集 $\mathcal{L}(x_0) = \{x \,|\, f(x) \le f(x_0)\}$ 是有界的. 那么算法

4.2 或者有限步终止, 或者 $\lim\limits_{k\to\infty} g(x_k) = 0$.

证 不失一般性, 不妨假设 $\{x_k\}$ 是无穷序列. 此时有 $g(x_k) \neq 0$. 因 $d_k = -g_k + \beta_{k-1}d_{k-1}$, 故有

$$g_k^T d_k = -\|g_k\|^2 + \beta_{k-1} g_k^T d_{k-1} = -\|g_k\|^2 < 0,$$

即 d_k 是下降方向. 从而由精确线搜索规则可知, $\{f(x_k)\}$ 是单调下降的, 故 $\{x_k\} \subset \mathcal{L}(x_0)$. 于是 $\{x_k\}$ 是有界的, 因而必有聚点 x^*, 即存在子列 $\{x_k : k \in K_1\}$ 收敛到 x^*. 由 f 的连续性, 有

$$f^* = \lim_{k(\in K_1)\to\infty} f(x_k) = f\left(\lim_{k(\in K_1)\to\infty} x_k\right) = f(x^*).$$

类似地, $\{x_{k+1} : k \in K_1\}$ 也是有界序列, 故存在子列 $\{x_{k+1} : k \in K_2\}$ 收敛到 $\bar{x}^*$, 这里 $K_2 \subset K_1$ 是无穷子序列. 于是可得

$$f^* = \lim_{k(\in K_2)\to\infty} f(x_k) = f\left(\lim_{k(\in K_2)\to\infty} x_k\right) = f(\bar{x}^*).$$

故有

$$f(\bar{x}^*) = f(x^*) = f^*. \tag{4.18}$$

下面用反证法证明 $g(x^*) = 0$. 如果不然, 即 $g(x^*) \neq 0$, 则对于充分小的 $\alpha > 0$, 有

$$f(x^* + \alpha d^*) < f(x^*).$$

注意到, 对任意的 $\alpha > 0$, 有

$$f(x_{k+1}) = f(x_k + \alpha_k d_k) \leq f(x_k + \alpha d_k).$$

对于 $k \in K_2 \subset K_1$, 令 $k \to \infty$ 对上式取极限得

$$f(\bar{x}^*) \leq f(x^* + \alpha d^*) < f(x^*),$$

这与 (4.18) 式矛盾, 从而证明了 $g(x^*) = 0$. 证毕. □

若在算法 4.2 中采用非精确搜索确定步长因子 α_k, 比如 Wolfe 准

则 (2.10) 和 (2.11), 则利用一般下降类算法的全局收敛性定理, 可以得到非精确搜索下的共轭梯度法的收敛性定理.

定理 4.4 设 $\{x_k\}$ 是由算法 4.2 利用 Wolfe 准则 (2.10)-(2.11) 产生的序列, 假定函数 $f(x)$ 一阶连续可微且有下界, 其梯度函数 $g(x)$ 在 $\mathbb{R}^n$ 上 Lipschitz 连续, 即存在常数 $L > 0$, 使得

$$\|g(u) - g(v)\| \leq L\|u - v\|, \quad \forall\, u, v \in \mathbb{R}^n.$$

若选取的搜索方向 d_k 与 $-g_k$ 的夹角 θ_k 满足条件

$$0 \leq \theta_k \leq \frac{\pi}{2} - \mu, \quad \mu \in \left(0, \frac{\pi}{2}\right).$$

那么算法 4.2 或者有限步终止, 或者 $\lim\limits_{k\to\infty} g(x_k) = 0$.

证 不失一般性, 不妨假设 $\{x_k\}$ 是无穷序列. 由 Lipschitz及连续

条件和 Wolfe 准则 (2.11) 式得

$$\begin{aligned}\alpha_k L\|d_k\|^2 &\geq d_k^T[g(x_k+\alpha_k d_k)-g_k] \geq -(1-\sigma)d_k^T g_k \\ &= (1-\sigma)\|d_k\|\|g_k\|\cos\theta_k,\end{aligned}$$

即

$$\alpha_k\|d_k\| \geq \frac{(1-\sigma)}{L}\|g_k\|\cos\theta_k.$$

于是利用上式及 Wolfe 准则 (2.10) 式可得

$$\begin{aligned}&f(x_k)-f(x_k+\alpha_k d_k) \\ &\geq -\rho\alpha_k d_k^T g_k = \rho\alpha_k\|d_k\|\|g_k\|\cos\theta_k \\ &\geq \rho\frac{(1-\sigma)}{L}\|g_k\|^2\cos^2\theta_k \\ &\geq \frac{\rho(1-\sigma)}{L}\|g_k\|^2\sin^2\mu.\end{aligned}$$

注意到 $f(x)$ 是有下界的, 由上式不难推得

$$\sum_{k=0}^{\infty} \|g\|^2 < +\infty,$$

这蕴含了当 $k \to \infty$ 时, 有 $\|g_k\| \to 0$. 证毕. □

§4.3 共轭梯度法的 Matlab 程序

在共轭梯度法的实际使用中, 通常在迭代 n 步或 $n+1$ 步之后, 重新取负梯度方向作为搜索方向, 我们称之为再开始共轭梯度法. 这是因为对于一般非二次函数而言, n 步迭代后共轭梯度法产生的搜索方向往往不再具有共轭性. 而对于大规模问题, 常常每 $m(m < n$ 或 $m \ll n)$ 步就进行再开始. 此外, 当搜索方向不是下降方向时, 也插入负梯度方向作为搜索方向.

本节给出基于 Armijo 非精确线搜索的再开始 FR 共轭梯度法的

程序 4.1 (FR共轭梯度法程序)

```
function [x,val,k]=frcg(fun,gfun,x0)
% 功能: 用FR共轭梯度法求解无约束问题: min f(x)
%输入: x0是初始点, fun, gfun分别是目标函数和梯度
%输出: x, val分别是近似最优点和最优值, k是迭代次数.
maxk=5000;   %最大迭代次数
rho=0.6;sigma=0.4;
k=0;  epsilon=1e-4;
n=length(x0);
while(k<maxk)
    g=feval(gfun,x0);  %计算梯度
    itern=k-(n+1)*floor(k/(n+1));
    itern=itern+1;
    %计算搜索方向
    if(itern==1)
        d=-g;
    else
        beta=(g'*g)/(g0'*g0);
```

```
        d=-g+beta*d0;  gd=g'*d;
        if(gd>=0.0)
            d=-g;
        end
    end
    if(norm(g)<epsilon), break; end    %检验终止条件
    m=0; mk=0;
    while(m<20)    %Armijo搜索
        if(feval(fun,x0+rho^m*d)<feval(fun,x0)+sigma*rho^m*g'*d)
            mk=m; break;
        end
        m=m+1;
    end
    x0=x0+rho^mk*d;
    val=feval(fun,x0);
    g0=g;  d0=d;
    k=k+1;
end
x=x0;
val=feval(fun,x);
```

例 4.1 利用程序 4.1 求解无约束优化问题

$$\min_{x\in\mathbb{R}^2} f(x) = 100(x_1^2 - x_2)^2 + (x_1 - 1)^2.$$

该问题有精确解 $x^* = (1,1)^T$, $f(x^*) = 0$.

解 利用程序 4.1, 终止准则取为 $\|\nabla f(x_k)\| \le 10^{-4}$. 取不同的初始点, 数值结果如下表.

表 4.1 FR 共轭梯度法的数值结果.

初始点 (x_0)	迭代次数 (k)	目标函数值 $(f(x_k))$
$(0,0)^T$	122	7.2372×10^{-9}
$(0.5,0.5)^T$	44	3.5498×10^{-10}
$(1.2,-1)^T$	56	1.1698×10^{-8}
$(-1.2,1)^T$	44	2.9396×10^{-9}
$(-1.2,-1)^T$	58	2.0235×10^{-9}

由上表可以看出, 共轭梯度法的收敛速度是比较令人满意的.

说明 上述程序的调用方式是:

```
x0=[-1.2 1]';
[x,val,k]=frcg('fun','gfun',x0)
```

其中 fun, gfun 分别是求目标函数值及其梯度的 M 函数文件.