MATH 3011 运筹学	**秋季 2023**

Lecture 13: 无约束优化 拟牛顿法

Lecturer: 陈士祥 *Scribes:* 陈士祥

致谢：本节感谢北京大学文再文老师提供的《最优化方法》参考讲义

1 问题形式

无约束最优化问题

$$\min_{x\in\mathbb{R}^n} \quad f(x) \tag{13.1}$$

其目标函数 f 是定义在 $\mathbb{R}^n$ 上的实值函数，决策变量 x 的可取值之集合是全空间 $\mathbb{R}^n$. f 是一次可微的。

2 拟牛顿法

牛顿法的突出优点是局部收敛很快（具有二阶收敛速率），
但运用牛顿法需要计算二阶导，而且目标函数的 Hesse 矩阵 $\nabla^2 f(x^k)$ 可能非正定，甚至奇异。为了克服这些缺点，
人们提出了拟牛顿法。其基本思想是：用不含二阶导数的矩阵 H_k 近似牛顿法中的 Hesse 矩阵的逆 $\nabla^2 f(x^k)^{-1}$.

由构造近似矩阵的方法不同，有不同的拟牛顿法。

2.1 割线方程

首先，对于 Hesse 矩阵的近似，我们有如下的要求。设第 k 次迭代后得到 x^{k+1}, 将目标函数 $f(x)$ 在 x^{k+1} 处二阶 Taylor 展开：

$$f(x)\approx \quad f(x^{k+1})+\nabla f(x^{k+1})^T(x-x^{k+1})+\frac{1}{2}(x-x^{k+1})^T\nabla^2 f(x^{k+1})(x-x^{k+1}),$$

进一步有

$$\nabla f(x)\approx \nabla f(x^{k+1})+\nabla^2 f(x^{k+1})(x-x^{k+1}),$$

于是令 $x=x^k$ 得

$$\nabla f(x^k)\approx \nabla f(x^{k+1})+\nabla^2 f(x^{k+1})(x^k-x^{k+1}).$$

记 $s^k = x^{k+1} - x^k$, $y^k = \nabla f(x^{k+1}) - \nabla f(x^k)$, 则有

$$\nabla^2 f(x^{k+1}) s^k \approx y^k \quad \text{or} \quad \nabla^2 f(x^{k+1})^{-1} y^k \approx s^k.$$

这样，计算出 s^k 和 y^k 后，可依上式估计在 x^{k+1} 处的 Hessian 矩阵或者 Hessian 的逆矩阵。要求在迭代中构造出 Hesse 矩阵的近似 B_{k+1}, 使其满足

$$B_{k+1} s^k = y^k. \tag{13.2}$$

我们有理由要求在迭代中构造出 Hesse 矩阵逆的近似 H_{k+1}, 使其满足

$$H_{k+1} y^k = s^k. \tag{13.3}$$

通常把式(13.2)和(13.3)称作割线方程，也称为拟牛顿条件。

曲率条件 由于近似矩阵必须保证迭代收敛, 正如牛顿法要求 Hesse 矩阵正定, B^k 正定也是必须的, 即有必要条件

$$\left(s^k\right)^{\mathrm{T}} B^{k+1} s^k > 0 \Longrightarrow \left(s^k\right)^{\mathrm{T}} y^k > 0,$$

Definition 13.1 曲率条件在迭代过程中满足 $\left(s^k\right)^{\mathrm{T}} y^k > 0, \forall k \in \mathbb{N}^+$.

如果线搜索使用 Powell-Wolfe 准则:

$$\nabla f\left(x^k + \alpha d^k\right)^{\mathrm{T}} d^k \geqslant c_2 \nabla f\left(x^k\right)^{\mathrm{T}} d^k,$$

其中 $c_2 \in (0,1)$. 上式即 $\nabla f\left(x^{k+1}\right)^{\mathrm{T}} s^k \geqslant c_2 \nabla f\left(x^k\right)^{\mathrm{T}} s^k$. 在不等式两边同时减去 $\nabla f\left(x^k\right)^{\mathrm{T}} s^k$, 由于 $c_2 - 1 < 0$ 且 s^k 是下降方向, 因此最终有

$$\left(y^k\right)^{\mathrm{T}} s^k \geqslant (c_2 - 1) \nabla f\left(x^k\right)^{\mathrm{T}} s^k > 0.$$

拟牛顿法的迭代格式如下：

Algorithm 1 拟牛顿算法 (Quasi-Newton method)

Require: 选取初始点 x^0, 令 $H_0 = I$ 或 $B_0 = I$, $k := 0$.

while 未满足终止条件: **do**
  计算搜索方向 $d^k = -H_k \nabla f(x^k)$ 或者 $d^k = -(B_k)^{-1} \nabla f(x^k)$.
  采用一维搜索确定步长因子 α_k, 令 $x^{k+1} = x^k + \alpha_k d^k$.
  基于 x^k 到 x^{k+1} 的梯度变化，更新 H_{k+1} 或者 B_{k+1}.
  $k := k + 1$
end while

下面我们就来讨论怎样构造及确定满足拟牛顿条件的 Hesse 矩阵逆的近似 H_{k+1}.

2.2 SR1 公式

设 H_k 是第 k 次迭代的 Hesse 矩阵逆的近似，我们希望以 H_k 来产生 H_{k+1}, 即

$$H_{k+1} = H_k + E_k,$$

其中 E_k 是一个低秩的矩阵。

为此，可采用对称秩一 (SR1) 校正

$$H_{k+1} = H_k + auu^T,\ (a \in \mathbb{R}, u \in \mathbb{R}^n).$$

由拟牛顿条件(13.3)知

$$H_{k+1}y^k = H_k y^k + (au^T y^k)u = s^k$$

故 u 必与方向 $s^k - H_k y^k$ 一致，且假定 $s^k - H_k y^k \neq 0$.

不妨取 $u = s^k - H_k y^k$, 此时 $a = \dfrac{1}{u^T y^k}$, 从而得到

$$H_{k+1} = H_k + \frac{(s^k - H_k y^k)(s^k - H_k y^k)^T}{(s^k - H_k y^k)^T y^k}. \tag{13.4}$$

上式称为对称秩一校正。

同理，由 $B_{k+1}s_k = y_k$ 得

$$B_{k+1} = B_k + \frac{uu^T}{u^T s^k}, \quad u = y^k - B_k s^k.$$

SR1 的缺陷：

对称秩一校正的缺点是，不能保持迭代矩阵 H_{k+1} 的正定性。

仅当 H_k 正定以及 $(s^k - H_k y^k)^T y^k > 0$ 时，对称秩一校正才能保持正定性。
证明： 设 $0 \neq w \in \mathbb{R}^n$, 则

$$w^T H_{k+1} w = w^T H_k w + \frac{(u^T w)^2}{u^T y_k} > 0.$$

而这个条件往往很难保证，即使 $(s^k - H_k y^k)^T y^k > 0$ 满足，它也可能很小从而导致数值上的困难。这些都使得对称秩一校正的拟牛顿法应用有较大局限性。

2.3 DFP 公式

采用对称秩二 (SR2) 校正

$$H_{k+1} = H_k + auu^T + bvv^T,$$

并使得拟牛顿条件(13.3)成立，则有

$$H_{k+1}y^k = H_k y^k + (au^T y^k)u + (bv^T y^k)v = s^k.$$

这里 u, v 显然不是唯一确定的，但有一种明显的选择是:

$$\begin{cases} u = s^k, & au^T y^k = 1; \\ v = H_k y^k, & bv^T y^k = -1. \end{cases}$$

因此有

$$H_{k+1} = H_k + \frac{s^k {s^k}^T}{{s^k}^T y^k} - \frac{H_k y^k {y^k}^T H_k}{{y^k}^T H_k y^k}. \tag{13.5}$$

上式称为 DFP(Davidon-Fletcher-Powell) 校正公式，由 Davidon(1959) 提出，后经 Fletcher & Powell(1963) 修改而来。

2.4 BFGS 公式

BFGS 是由 Broyden、Fletcher、Goldfarb、Shanno 4 人从不同角度提供了一种新的拟牛顿公式。

根据割线方程, 将秩二更新的待定参量式代入, 得

$$B^{k+1}s^k = \left(B^k + auu^{\mathrm{T}} + bvv^{\mathrm{T}}\right)s^k = y^k,$$

整理可得

$$\left(a \cdot u^{\mathrm{T}} s^k\right)u + \left(b \cdot v^{\mathrm{T}} s^k\right)v = y^k - B^k s^k.$$

简单的取法是令 $\left(a \cdot u^{\mathrm{T}} s^k\right)u$ 对应 y^k 相等, $\left(b \cdot v^{\mathrm{T}} s^k\right)v$ 对应 $-B^k s^k$ 相等, 即有

$$a \cdot u^{\mathrm{T}} s^k = 1, \quad u = y^k, \quad b \cdot v^{\mathrm{T}} s^k = -1, \quad v = B^k s^k.$$

将上述参量代入割线方程, 即得 BFGS 更新公式

$$B^{k+1} = B^k + \frac{uu^{\mathrm{T}}}{\left(s^k\right)^{\mathrm{T}} u} - \frac{vv^{\mathrm{T}}}{\left(s^k\right)^{\mathrm{T}} v}.$$

利用 SMW 公式以及 $H^k = \left(B^k\right)^{-1}$, 可以推出关于 H^k 的 BFGS 公式.

Definition 13.2 *BFGS 公式 在拟牛顿类算法中, 基于 B^k 的 BFGS 公式为*

$$B^{k+1} = B^k + \frac{y^k\left(y^k\right)^{\mathrm{T}}}{\left(s^k\right)^{\mathrm{T}} y^k} - \frac{B^k s^k\left(B^k s^k\right)^{\mathrm{T}}}{\left(s^k\right)^{\mathrm{T}} B^k s^k},$$

基于 H^k 的 BFGS 公式为

$$H^{k+1}=\left(I-\frac{s^k\left(y^k\right)^{\mathrm{T}}}{\left(s^k\right)^{\mathrm{T}}y^k}\right)^{\mathrm{T}}H^k\left(I-\frac{s^k\left(y^k\right)^{\mathrm{T}}}{\left(s^k\right)^{\mathrm{T}}y^k}\right)+\frac{s^k\left(s^k\right)^{\mathrm{T}}}{\left(s^k\right)^{\mathrm{T}}y^k}.$$

推导 H^k 的 BFGS 公式, 我们需要下面的 Sherman-Morrison-Woodbury（SMW）公式。

Sherman-Morrison-Woodbury 公式： 设 $A\in\mathbb{R}^{n\times n}$ 是非奇异阵，$u,v\in\mathbb{R}^n$ 是任意向量。若 $1+v^TA^{-1}u\neq 0$, 则 A 的秩一校正 $A+uv^T$ 非奇异，且其逆可以表示为

$$(A+uv^T)^{-1}=A^{-1}-\frac{A^{-1}uv^TA^{-1}}{1+v^TA^{-1}u}. \tag{13.6}$$

Sherman-Morrison-Woodbury 推广公式： 设 $A\in\mathbb{R}^{n\times n}$ 是非奇异阵，$U\in\mathbb{R}^{n\times k},V\in\mathbb{R}^{n\times k}$ 是任意矩阵。若 $I_k+V^TA^{-1}U$ 可逆, 则 $A+UV^T$ 非奇异，且其逆可以表示为

$$(A+UV^T)^{-1}=A^{-1}-A^{-1}U(I_k+V^TA^{-1}U)^{-1}V^TA^{-1}. \tag{13.7}$$

推导 H^k 的 BFGS 公式之提示:
对于可逆矩阵 $B\in\mathbb{R}^{n\times n}$ 与矩阵 $U\in\mathbb{R}^{n\times m},V\in\mathbb{R}^{n\times m}$, 根据 SMW 公式(13.7)为:

$$\left(B+UV^{\mathrm{T}}\right)^{-1}=B^{-1}-B^{-1}U\left(I+V^{\mathrm{T}}B^{-1}U\right)^{-1}V^{\mathrm{T}}B^{-1}.$$

在 BFGS 的推导中, 关于 B^k 的更新公式为:

$$B_{k+1}=B_k+\frac{y_ky_k^{\mathrm{T}}}{s_k^{\mathrm{T}}y_k}-\frac{B_ks_k\left(B_ks_k\right)^{\mathrm{T}}}{s_k^{\mathrm{T}}B_ks_k}=B_k+\begin{pmatrix}-\frac{B_ks_k}{s_k^{\mathrm{T}}B_ks_k} & \frac{y_k}{s_k^{\mathrm{T}}y_k}\end{pmatrix}\begin{pmatrix}s_k^{\mathrm{T}}B_k\\ y_k^{\mathrm{T}}\end{pmatrix}.$$

对照 SMW 公式(13.7), 令式中 $B=B_k$, 且

$$U_k=\begin{pmatrix}-\frac{B_ks_k}{s_k^{\mathrm{T}}B_ks_k} & \frac{y_k}{s_k^{\mathrm{T}}y_k}\end{pmatrix},\quad V_k=\begin{pmatrix}B_ks_k & y_k\end{pmatrix},$$

此时公式的左端就等于 B_{k+1}^{-1}, 且右端只需计算一个 2 阶矩阵的逆. 假设 $B_k^{-1}=H_k$, 由 SMW 公式就得到

$$H_{k+1}=\left(B_k+U_kV_k^{\mathrm{T}}\right)^{-1}=\left(I-\frac{s_ky_k^{\mathrm{T}}}{s_k^{\mathrm{T}}y_k}\right)H_k\left(I-\frac{y_ks_k^{\mathrm{T}}}{s_k^{\mathrm{T}}y_k}\right)+\frac{s_ks_k^{\mathrm{T}}}{s_k^{\mathrm{T}}y_k}.$$

作业 13.1 利用 *SMW* 公式，由 $H_{k+1}^{(DFP)}$ 推导 $B_{k+1}^{(DFP)}$.

BFGS 公式的有效性

BFGS 公式产生的 B^{k+1} 或 H^{k+1} 是否正定呢?

Theorem 13.1 *BFGS 公式使拟牛顿矩阵正定的充分条件使用秩二更新公式从 B^k 或 H^k 更新 B^{k+1} 或 H^{k+1}, 拟牛顿矩阵正定的充分条件可以是:*
(1) B^k 或 H^k 正定;
(2) 满足曲率条件 $\left(s^k\right)^{\mathrm{T}} y^k > 0, \forall k \in \mathbb{N}^+$.

证明上述定理, 只需要从基于 H^k 的 BFGS 公式分析即可, 从而得到 H^{k+1} 和其逆 B^{k+1} 均正定.

因为在确定步长时使用某一 Wolfe 准则线搜索即可满足曲率条件, 因此 BFGS 公式产生的拟牛顿矩阵有望保持正定.

2.4.1 从优化意义理解 BFGS 公式

基于 H^k 的 BFGS 格式恰好是优化问题

$$\begin{aligned} \min_H \quad & \left\|H - H^k\right\|_W, \\ \text{s.t.} \quad & H = H^{\mathrm{T}}, \\ & H y^k = s^k. \end{aligned}$$

的解. 上式中 $\|\cdot\|_W$ 是加权范数, 定义为

$$\|H\|_W = \left\|W^{1/2} H W^{1/2}\right\|_F,$$

且 W 满足割线方程, 即 $W s^k = y^k$. 使用 $\|\cdot\|_W$ 可以让得到的拟牛顿公式同样满足 仿射不变性（请回顾: “牛顿法为什么好” -牛顿法的仿射不变性质）。注意 $H y^k = s^k$ 是割线方程, 因此优化问题的意义是在满足割线方程的对称矩阵中找到距离 H^k 最近的矩阵 H 作为 H^{k+1}. 因此我们可以进一步认知, BFGS 格式更新的拟牛顿矩阵是正定对称的, 且在满足割线方程的条件下采取的是最佳逼近策略.

2.4.2 从优化意义理解 DFP 公式

有了 BFGS 公式的优化意义做铺垫, 讨论 DFP 公式的优化意义显得十分简单. 利用对偶性质, 基于 B^k 的 DFP 格式将是优化问题

$$\begin{aligned} \min_B \quad & \left\|B - B^k\right\|_W, \\ \text{s.t.} \quad & B = B^{\mathrm{T}}, \\ & B s^k = y^k. \end{aligned}$$

的解. 上式中 $\|\cdot\|_W$ 是加权范数, 定义为

$$\|B\|_W = \left\|W^{1/2} B W^{1/2}\right\|_F,$$

λ	0.1	0.01	10^{-4}	10^{-8}
10	5	6	8	10
100	7	8	10	12
10^4	12	13	15	17
10^6	17	18	20	22
10^9	24	25	27	29

表 13.1: BFGS 方法的迭代次数

λ	0.1	0.01	10^{-4}	10^{-8}
10	10	13	16	19
30	25	32	37	40
100	80	99	107	111
300	237	290	307	313
10^3	787	958	1006	1014

表 13.2: DFP 方法的迭代次数

且 W 满足另一割线方程, 即 $Wy^k = s^k$. 注意 $Bs^k = y^k$ 是另一割线方程, 因此优化问题的意义是在满足割线方程的对称矩阵中找到距离 B^k 最近的矩阵 B 作为 B^{k+1}.

DFP 公式的缺陷

尽管 DFP 格式与 BFGS 对偶, 但从实际效果而言, DFP 格式的求解效率整体上不如 BFGS 格式. M.J.D. Powell 曾求解问题

$$\min_{x\in\mathbb{R}^2} f(x) = \frac{1}{2}\|x\|_2^2.$$

设置初始值

$$B^0 = \begin{pmatrix} 1 & 0 \\ 0 & \lambda \end{pmatrix}, \quad x_1 = \begin{pmatrix} \cos\psi \\ \sin\psi \end{pmatrix},$$

其中 $\tan^2\psi = \lambda$. 当误差阈 $\epsilon = 10^{-4}$ 时, 分别取 λ 为不同的值, 使用 BFGS 算法与 DFP 算法所产生的迭代步数分别如表 13.1和 13.2所示. 由此看出, 在本问题中, BFGS 算法的求解效率要远高于 DFP 算法. (参考文献: Powell M J D. How bad are the BFGS and DFP methods when the objective function is quadratic?[J]. Mathematical Programming, 1986, 34(1): 34-47.)

2.4.3 BFGS 另一种解释

$X = H_{k+1}$ 是下面优化问题的最优解：

$$\begin{aligned} &\min_X Tr(H_k^{-1}X) - \log\det(H_k^{-1}X) - n \\ &\text{s.t.} \quad Xs_k = y_k, X^T = X. \end{aligned} \tag{13.8}$$

上述问题中的目标函数，是概率分布 $N(0,X)$ 和 $N(0,H_k)$ 的相对熵。

作业 13.2 证明以下结论：

1. 问题(13.8) 目标函数值是非负的。目标值为 0 仅当 $X = H_k$.
2. 证明 BFGS 的迭代公式 H_{k+1} 是该问题的最优解。

(提示：$-\log\det(X)$ 是关于 X 的凸函数，并且 $\frac{\partial \log\det X}{\partial X} = X^{-T}$, $\frac{\partial Tr(C^TX)}{\partial X} = C$.)

2.5 BFGS 算法收敛性

Theorem 13.2 *BFGS 全局收敛性: 设初始矩阵 B^0 是对称正定矩阵, 目标函数 $f(x)$ 是二阶连续可微函数, 下水平集*

$$\mathcal{L} = \left\{x \in \mathbb{R}^n \mid f(x) \leqslant f\left(x^0\right)\right\}$$

凸, 且存在 $m, M \in \mathbb{R}^+$ 使得对 $\forall z \in \mathbb{R}^n, x \in \mathcal{L}$ 满足

$$m\|z\|^2 \leqslant z^{\mathrm{T}}\nabla^2 f(x)z \leqslant M\|z\|^2$$

(即 $z^{\mathrm{T}}\nabla^2 f(x)z$ 被 $\|z\|$ 控制), 那么 BFGS 格式结合 Wolfe 线搜索的拟牛顿算法全局收敛到 $f(x)$ 的极小值点 x^*.

局部收敛速度：进一步假设 f 的 Hessian 在最优点邻域内是 Lipschitz 连续，那么 BFGS 的迭代点最终超线性收敛到最优点 x^*.

我们略过证明。

3 有限内存 BFGS 方法

基本思路: 标准的拟牛顿近似矩阵的更新公式可以记为

$$B^{k+1} = g\left(B^k, s^k, y^k\right), s^k = x^{k+1} - x^k, y^k = \nabla f\left(x^{k+1}\right) - \nabla f\left(x^k\right).$$

若变量维度太大，那么存储 H_k 需要大量内存，并且更新的计算量为 $O(n^2)$. 如果只保存最近的 m 组数据, 那么迭代公式可以写成

$$B^{k+1} = g\left(g\left(\cdots g\left(B^{k-m+1}, s^{k-m+1}, y^{k-m+1}\right)\right)\right).$$

考虑 BFGS 方法:

$$d^k = -\left(B^k\right)^{-1} \nabla f\left(x^k\right) = -H^k \nabla f\left(x^k\right).$$

重写 BFGS 更新公式为

$$H^{k+1} = \left(V^k\right)^{\mathrm{T}} H^k V^k + \rho_k s^k \left(s^k\right)^{\mathrm{T}},$$

其中

$$\rho_k = \frac{1}{\left(y^k\right)^{\mathrm{T}} s^k}, \quad V^k = I_{n\times n} - \rho_k y^k \left(s^k\right)^{\mathrm{T}}.$$

将上式递归地展开 m 次, 即

$$\begin{aligned}
H^k = & \left(\prod_{j=k-m}^{k-1} V^j\right)^{\mathrm{T}} H^{k-m} \left(\prod_{j=k-m}^{k-1} V^j\right) + \\
& \rho_{k-m} \left(\prod_{j=k-m+1}^{k-1} V^j\right)^{\mathrm{T}} s^{k-m} \left(s^{k-m}\right)^{\mathrm{T}} \left(\prod_{j=k-m+1}^{k-1} V^j\right) + \\
& \rho_{k-m+1} \left(\prod_{j=k-m+2}^{k-1} V^j\right)^{\mathrm{T}} s^{k-m+1} \left(s^{k-m+1}\right)^{\mathrm{T}} \left(\prod_{j=k-m+2}^{k-1} V^j\right) + \cdots + \\
& \rho_{k-1} s^{k-1} \left(s^{k-1}\right)^{\mathrm{T}}.
\end{aligned}$$

为了节省内存, 我们只展开 m 次, 利用 H^{k-m} 进行计算, 即可求出 H^{k+1}. 下面介绍一种不计算 H^k, 只利用展开式计算 $d^k = -H^k \nabla f\left(x^k\right)$ 的巧妙算法: 双循环递归算法. 它利用迭代式的结构尽量节省计算 d^k 的开销.

由于我们只需要得到 $-d^k = H^k \nabla f(x_k)$，将等式两边同右乘 $\nabla f\left(x^k\right)$. 观察等式右侧需要计算

$$V^{k-1} \nabla f\left(x^k\right), \cdots, V^{k-m} \cdots V^{k-1} \nabla f\left(x^k\right).$$

这些计算可以递归地进行. 同时在计算 $V^{k-l} \cdots V^{k-1} \nabla f\left(x^k\right)$ 的过程中, 可以计算上一步的 $\rho_{k-l} \left(s^{k-l}\right)^{\mathrm{T}} \left[V^{k-l+1} \cdots V^{k-1} \nabla f\left(x^k\right)\right]$, 这是一个标量. 记

$$\begin{aligned}
q &= V^{k-m} \cdots V^{k-1} \nabla f\left(x^k\right), \\
\alpha_{k-l} &= \rho_{k-l} \left(s^{k-l}\right)^{\mathrm{T}} \left[V^{k-l+1} \cdots V^{k-1} \nabla f\left(x^k\right)\right],
\end{aligned}$$

因此递归公式可化为如下的形式:

$$H^k \nabla f\left(x^k\right)=\left(\prod_{j=k-m}^{k-1} V^j\right)^{\mathrm{T}} H^{k-m} q+\left(\prod_{j=k-m+1}^{k-1} V^j\right)^{\mathrm{T}} s^{k-m} \alpha_{k-m}+\cdots+s^{k-1} \alpha_{k-1}$$

在双循环递归算法中, 除了上述第一个循环递归过程 (自下而上) 外, 还有以下第二个循环递归过程。我们需要在公式中自上而下合并每一项. 以前两项为例, 它们有公共的因子 $\left(V^{k-m+1} \cdots V^{k-1}\right)^{\mathrm{T}}$, 提取后可以将前两项写为 (注意将 V^{k-m} 的定义回代)

$$\begin{aligned}
&\left(V^{k-m+1} \cdots V^{k-1}\right)^{\mathrm{T}}\left[\left(V^{k-m}\right)^{\mathrm{T}} r+\alpha_{k-m} s^{k-m}\right] \\
&=\left(V^{k-m+1} \cdots V^{k-1}\right)^{\mathrm{T}}\left(r+\left(\alpha_{k-m}-\beta\right) s^{k-m}\right),
\end{aligned}$$

这正是第二个循环的迭代格式. 注意合并后原递归式的结构仍不变, 因此可以递归地计算下去. 最后, 变量 r 就是我们期望的结果 $H^k \nabla f\left(x^k\right)$.

Algorithm 2 算法 L-BFGS 双循环递归

Require: 初始化 $q \leftarrow \nabla f\left(x^k\right)$.

Ensure: r, 即 $H^k \nabla f\left(x^k\right)$.

```
1: for i = k-1, ..., k-m do
2:     计算并保存 α_i ← ρ_i (s^i)^T q.
3:     更新 q ← q - α_i y^i
4: end for
5: 初始化 r ← Ĥ^{k-m} q, 其中 Ĥ^{k-m} 是 H^{k-m} 的近似矩阵.
6: for i = k-m, ..., k-1 do
7:     β ← ρ_i (y^i)^T r
8:     更新 r ← r + (α_i - β) s^i
9: end for
```

L-BFGS 双循环递归算法约需要 $4mn$ 次乘法运算, $2mn$ 次加法运算; 若近似矩阵 $\hat{H}^{k-m}$ 是对角矩阵, 则额外需要 n 次乘法运算. 由于 m 不会很大, 因此算法的复杂度是 $\mathcal{O}(mn)$. 算法需要的额外存储为临时变量 α_i, 其大小是 $\mathcal{O}(m)$.

进一步的参考资料

- R. Fletcher, Practical Methods of Optimization (2nd Edition). John Wiley & Sons, 1987.
- D. C. Liu and J. Nocedal, On the Limited Memory Method for Large Scale Optimization. Mathematical Programming B, 45(3), pp. 503-528, 1999.

- Nocedal, Jorge, and Stephen J. Wright, eds. Numerical optimization. New York, NY: Springer New York, 1999.