

## 目录

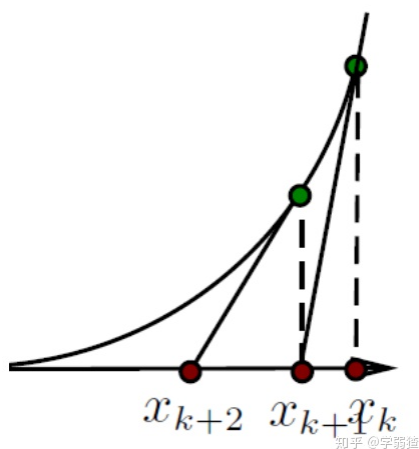
- 割线法：拟牛顿法的前身
- SR1 方法
- BFGS 方法
- BFGS 方法的实操细节
- DFP 方法
- Broyden 族
- 统一拟牛顿方法的 DM 条件

## Source

- 厦门大学课堂笔记，教授主页：<https://www.math.fsu.edu/~whuang2/index.html>
- R. H. Byrd, H. F. Khalfan, and R. B. Schnabel. *Analysis of a symmetric rank-one trust region method.*
- R. H. Byrd and J. Nocedal. *A tool for the analysis of quasi-Newton methods with application to unconstrained minimization.*
- Y.-H. Dai. *A perfect example for the BFGS method.*
- J. E. Dennis and J. J. More. *Quasi-Newton methods, motivation and theory.*
- J. E. Dennis and R. B. schnabel. *Numerical methods for unconstrained optimization and nonlinear equations.*
- D.-H. Li and M. Fukushima. *A modified BFGS method and its global convergence in nonconvex minimization.*
- D.-H. Li and M. Fukushima. *On the global convergence of the BFGS method for nonconvex unconstrained optimization problems.*

## 割线法：拟牛顿法的前身

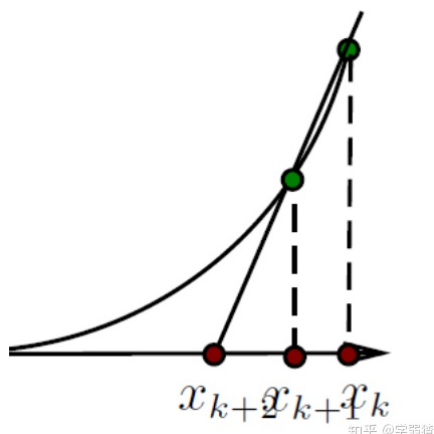
要说拟牛顿法（Quasi-Newton Method）必然要先提到上一节说的牛顿法。如果我们不用一般的情况来看它，而直接考虑一元的情况，其实对应的就是下面这张图



注意这张图是某一个函数的**导函数的图像**。在这种情况下，我们可以看出，如果在导函数上某一点做切线，这条切线的斜率就是二次导函数，并且对应的就是下面这个式子

$$f''(x_k) = -\frac{f'(x_k)}{x_{k+1} - x_k}$$

化简一下，就是  $x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)}$ ，这个就是牛顿法的一元形式。如果我们不使用导函数的切线，而是割线，那么这个时候，会有下面这张图



我们可以看出，这个时候考虑的是导函数之前两个点所形成的割线，那么这个时候会得到一个式为

$$B_k(x_k - x_{k-1}) = f'(x_k) - f'(x_{k-1})$$

这里的  $B_k$  就是**这条割线的斜率**。所以割线法其实就是拟牛顿法的前身，因为如果我们设  $s_k = x_{k+1} - x_k$ ,  $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ ，式子就会变成

$$B_k s_{k-1} = y_{k-1}$$

这就是拟牛顿法的本质。拟牛顿法可以好用，一个很重要的地方在于它**不需要精确计算二阶信息**。因为二阶信息本质上是一个矩阵，在大规模问题上计算一个矩阵的复杂度是难以接受的，所以拟牛顿法自然就有了它独特的优势。

但是我相信你也发现了，对于一般的  $n$  元的情况， $B_k$  为一个**对称正定矩阵**的情况下，这个方程组只有  $n-1$  个，这是因为我们需要两个初始条件，就必然会使得方程组的条件减少一个。因此我们需要其他额外的条件，而这些额外的条件就促使了各种各样的拟牛顿法的诞生。

## SR1 方法

SR1 方法也就是**对称秩一更新** (Symmetric Rank-one Update) 方法，这个方法继承的思路是：我希望每一步对于  $B_k$  的更新能够尽可能的小。这个“尽可能小”得含义，其实就是

$$B_{k+1} = \arg \min_{B^T=B} \|B - B_k\|$$

所以我们提出的一个思路就是考虑这样的更新

$$B_{k+1} = B_k + \frac{1}{a} v v^T$$

这个更新的思路在于我们的下一步的矩阵只在上一个矩阵的基础上加了一个秩为 1 的对称半正定矩阵。那么如何求解  $a, v$  呢？

首先我们代入我们希望求解的条件

$$\begin{aligned} B_{k+1} s_k &= y_k \\ B_k s_k + \frac{1}{a} v v^T s_k &= y_k \\ \frac{v^T s_k}{a} v &= y_k - B_k s_k \end{aligned}$$

这一点说明了 $v$ 是与 $y_k - B_k s_k$ 同方向的，所以我们可以设 $v = \tilde{a}(y_k - B_k s_k)$ 。那么再代回之前 $B_{k+1}$ 的式子，就可以得到

$$B_{k+1} = B_k + \frac{\tilde{a}^2}{a}(y_k - B_k s_k)(y_k - B_k s_k)^T$$

设 $b = \frac{\tilde{a}^2}{a}$ ，那么左右两边乘上 $s_k$ ，可以得到

$$y_k = B_k s_k + b(y_k - B_k s_k)(y_k - B_k s_k)^T s_k$$

这就可以解出来 $b = \frac{1}{(y_k - B_k s_k)^T s_k}$ 。

这里有一个小细节是 $a, v$ 并不是相互独立的，当然这里的含义不是概率统计里面的那个，而是说 $a, v$ 的解是相互制约的，因此在这种情况下，我们可以令 $\tilde{a} = 1$ ，这个时候可以得到对应的解为

$$a = (y_k - B_k s_k)^T s_k, v = y_k - B_k s_k$$

但是这里有一个问题就是，如果说 $a = 0$ ，这个时候问题自然就失败了（分母不能为0）。所以实际的更新一般会采用的方法为

$$B_{k+1} = \begin{cases} B_k & \frac{|(y_k - B_k s_k)^T s_k|}{\|s_k\| \|y_k - B_k s_k\|} \geq \gamma, \gamma \in (0, 1) \\ B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k} & \text{otherwise} \end{cases}$$

当然了，这个更新是不保正定的。事实上，如果 $a < 0$ ，那么 $B_{k+1}$ 不一定是正定矩阵。这个性质一定程度上限制了这个方法的效力，但是有一个有意思的事情是下面这个性质。

Proposition 1:

设 $\{x_0, \dots, x_n\}$ 为 $n + 1$ 步连续的SR1方法的更新，设优化问题为

$$f(x) = \frac{1}{2}x^T A x - b^T x, \text{ 设 } (y_k - B_k s_k)^T s_k \neq 0, s_i \text{ 线性无关, 那么}$$

$$B_n = A$$

我们证明一下这个结论。这里我们采用的依然是线性空间的那一套思路，也就是说考虑 $B_n$ 与 $A$ 在线性算子的表示下相同。为此自然可以考虑数学归纳法。

当 $k = 1$ 的时候，对应的就是 $B_1 s_0 = y_0$ ，这个要求自然是可以满足的。

现在我们假设对于 $B_i s_j = y_j$ 对于任意的 $j < i$ 都成立，那么我们考虑一下如何证明使得 $B_{i+1} s_j = y_j, j < i + 1$ 。事实上，当我们写开我们的迭代公式，可以发现

$$B_{i+1} s_j = [B_i + \frac{(y_i - B_i s_i)(y_i - B_i s_i)^T}{(y_i - B_i s_i)^T s_i}] s_j$$

如果要说明这个式子就等于 $y_j$ ，自然要说明后面的那一串是不起作用的。这就取决于 $(y_i - B_i s_i)^T s_j$ 这一项。又因为

$$(y_i - B_i s_i)^T s_j = y_i^T s_j - s_i^T y_j$$

结合  $y_i = As_i$  (因为我们考虑的是二次凸问题) 即可得到上式为 0, 代回就可以证明出  $k+1$  的情况。证明好这个, 就可以得到

$$B_n[s_0, \dots, s_{n-1}] = [y_0, \dots, y_{n-1}] = A[s_0, \dots, s_{n-1}]$$

根据线性无关即可得到  $B_n = A$ 。

基于这个性质, 我们可以观察到 SR1 算法可以使得最终选择的  $B_k$  会一步步的接近于函数的海塞矩阵。这样子有什么好处呢? 注意到牛顿法的每一步的下降方向就是要使用海塞矩阵的, 因此我们可以发现, 如果可以逼近它, 那么有理由相信它之后会越来越好的去逼近海塞矩阵, 也就可以恢复牛顿法的二次局部收敛速度。

下面是 SR1 算法的流程

### A trust region method with symmetric rank-one update

**Input:**  $x_0, B_0, \Delta_{\max} > 0, \Delta_0 \in (0, \Delta_{\max}], \eta \in [0, 0.25), \gamma \in (0, 1)$ ;

Set  $k \leftarrow 0$ ;

**while** not accurate enough **do**

Approximately solve

$$s_k \leftarrow \arg \min_{p \in \mathbb{B}(\Delta_k)} m_k(p) = f(x_k) + \nabla f(x_k)^T p + \frac{1}{2} p^T B_k p;$$

Compute  $\rho_k \leftarrow \frac{f(x_k) - f(x_k + s_k)}{m_k(0) - m_k(s_k)}$  and  $y_k \leftarrow \nabla f(x_{k+1}) - \nabla f(x_k)$ ;

Set

$$x_{k+1} \leftarrow \begin{cases} x_k + s_k & \text{if } \rho_k > \eta \\ x_k & \text{otherwise} \end{cases}$$

Set

$$\Delta_{k+1} \leftarrow \begin{cases} 0.25\Delta_k & \text{if } \rho_k < 0.25 \\ \min(2\Delta_k, \Delta_{\max}) & \text{if } \rho_k > 0.75 \text{ and } \|p_k\| = \Delta_k \\ \Delta_k & \text{otherwise} \end{cases}$$

Set

$$B_{k+1} \leftarrow \begin{cases} B_k & \text{if } \frac{|(y_k - B_k s_k)^T s_k|}{\|s_k\| \|y_k - B_k s_k\|} \geq \gamma \\ B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k} & \text{otherwise;} \end{cases}$$

$k \leftarrow k + 1$ ;

**end while**

知乎 @学弱渣

为什么这个算法采取的是信赖域算法的框架呢? 这是因为一方面, 矩阵  $B_k$  不保正定, 如果使用线搜索会导致无法找到下降方向的问题。另一方面, 矩阵可以很好的逼近海塞矩阵, 但是线搜索每一次只会利用到一个方向上的海塞矩阵的信息 (比方说  $d_j^T B d_j \leq 0$  表示矩阵在  $d_j$  这个方向上是负定的, 在  $B$  的其它方向上是什么样并不关心)。信赖域可以利用到全部信息, 所以信赖域算法也就相对更高效点。

下面我们给出它的全局收敛性和局部收敛性。它的证明是非常精细的, 所以我们同样的不会给出具体证明, 大家知道结论就好。

先给出全局收敛性

Theorem 1:

设在信赖域算法中  $\eta = 0$ , 设存在  $\beta$ , 使得  $\|B_k\| \leq \beta$ , 并且存在  $\delta$ , 使得  $f$  在  $\mathcal{N}_{x_0}$  上存在下界, 且在  $\mathcal{N}_{x_0}^\delta = \{x : \|x - y\| \leq \delta, \exists y \in \mathcal{N}_{x_0}\}$  上 Lipschitz 连续, 并且子问题的解均满足  $m_k(0) - m_k(p_k^c) \geq c_1 \|\nabla f(x_k)\| \min(\Delta_k, \frac{\|\nabla f(x_k)\|}{\|B_k\|})$ , 那

么有  $\liminf_{k \rightarrow \infty} \|\nabla f(x_k)\| = 0$

这里有一个假设是  $\|B_k\| \leq \beta$ ，相当于要求得到的  $B_k$  的模长是有一致的上界的。但是这个条件理论上为什么成立其实一直都没有解决.....

再给出局部收敛性

Theorem 2:

设  $\{x_k\}$  表示利用 SR1 信赖域框架更新得到的迭代点，信赖域算法的子问题用带截断的 CG 算法求解，参数为  $\kappa \in (0, 1), \theta > 0$ 。设  $\{x_k\}$  会收敛到极小值点  $x^*$ ， $f \in C^2$ ，且存在  $\beta$ ，使得  $\|B_k\| \leq \beta$ ， $\nabla^2 f(x^*)$  正定，且在  $x^*$  附近海塞矩阵 Lipschitz 连续，并且对于所有的  $k$  有  $\frac{|(y_k - B_k s_k)^T s_k|}{\|s_k\| \|y_k - B_k s_k\|} \geq \gamma$ ，那么有  $\lim_{k \rightarrow \infty} \frac{\|x_{k+n+1} - x^*\|}{\|x_k - x^*\|} = 0$ 。

虽然不是一般的超线性，但是依然算是超线性的收敛速度（准确来说是  $n + 1$  步 Q-超线性收敛速度）

## BFGS 方法

BFGS 方法应该算是拟牛顿算法中最有名的方法之一了。它的想法和 SR1 类似，只不过它对于“最小变化”的这个条件的诠释有所不同。我们希望做到的是，在给定了  $B_k$  的情况下，如何构造出一个  $B_{k+1}$ ，使得对称正定阵的条件依然满足，并且依然有“最小变化”？也就是说，如果给定一个对称正定阵  $A$ ，但  $Ax \neq b$ ，可否通过一些方法构造出新的对称正定阵  $\bar{A}$ ，使得  $\bar{A}x = b$ ？这就归结到了下面这个定理。

Theorem 3:

若  $b^T x > 0$ ，那么存在一个实的非奇异的矩阵  $J_+$ ，满足  $J_+ J_+^T x = b$ ，也即  $\bar{A} = J_+ J_+^T$ 。并且存在  $v$ ，使得  $J_+^T x = v$ ， $J_+ v = b$ ，并且可以通过如下方法找到  $J_+$ ：

(1)  $J_+ = \arg \min_{J \in Q(v, b)} \|J - L\|_F$ ，其中  $Q(v, b) = \{M : Mv = b\}$

(2) 找到满足  $J_+^T x = v$  的  $v$

事实上，最终的求解的公式为  $J_+ = L + \frac{(b - Lv)v^T}{v^T v}$ ，

$$\bar{A} = A + \frac{bb^T}{b^T x} - \frac{Axx^T A}{x^T Ax}, \quad v = \alpha L^T x, \quad \alpha^2 = \frac{b^T x}{x^T Ax}$$

这里的  $L$  满足  $A = LL^T$ 。

这个定理乍一看不知道在说啥，但是我们一步一步理其实就会清楚很多。其实这并不是希望我们自己去开天辟地，而是验证一下给定的表达式是否满足要求即可。

首先， $J_+ \in Q(v, b)$  吗？我们看看给定的公式是否满足这个要求，这个是显然的，交给读者吧。其次，我们要说明  $J_+$  是给定的条件中最小的那个，那么自然的，我们希望做的是说明对于任意的  $J \in Q(v, b)$ ，都有

$$\|J_+ - L\|_F \leq \|J - L\|_F$$

事实上，我们有

$$\|J_+ - L\|_F = \left\| \frac{(b - Lv)v^T}{v^T v} \right\|_F$$

要想利用上  $J$  的性质，就是要用上  $Jv = b$  的这个条件，所以代入可以得到

$$\|J_+ - L\|_F = \left\| \frac{(J - L)vv^T}{v^T v} \right\|_F \leq \left\| \frac{vv^T}{v^T v} \right\|_F \|J - L\|_F$$

注意到  $\frac{vv^T}{v^T v}$  为一个秩一投影阵，也就是说它只有一个特征值 1，其余的都为 0，数值分析告诉我们，

这样的话就会得到它的  $F$  范数为 1。所以自然的就可以证明完这一个小结论。

最后，就只需要看一下  $J_+^T x = v$  是否满足即可。注意到我们是验证，所以可以反过来推，看能不能得到什么有趣的东西。

我们首先代入  $J_+$ ，可以得到

$$L^T x + \frac{v(b - Lv)^T x}{v^T v} = v$$

这就可以说明  $v = \alpha L^T x$ ，其中  $\alpha$  是一个标量（想想为什么？）。接下来只要求  $\alpha$  即可。把这个式子代入，就可以得到

$$L^T x + \frac{\alpha L^T x (b - \alpha L L^T x)^T x}{\alpha^2 x^T L L^T x} = \alpha L^T x$$

注意到  $L$  是一个对称正定阵拆解出来的矩阵，所以  $L^T x$  两边都可以约掉，所以有

$$1 + \frac{(b - \alpha L L^T x)^T x}{\alpha x^T L L^T x} = \alpha$$

左边的拼在一起，结合  $L L^T = A$  就得到了结论  $\alpha^2 = \frac{b^T x}{x^T A x}$ 。一般来说我们都会取方向为正。

我们看一下怎么利用这个定理。注意到一般来说，我们是不能有  $B_k s_k = y_k$  的，这是因为，对于拟牛顿法来说，我们会有  $B_k p_k = -\nabla f(x_k)$ ，这里的  $p_k = x_{k+1} - x_k$  等于搜索方向的某一个倍数。那么这个时候，注意到  $p_k = s_k$ ，所以我们相当于得到了一个结论是  $y_k = -\nabla f(x_k)$ 。也就是  $\nabla f(x_{k+1}) = 0$ ，也就是说下一步就收敛了，那自然没有什么讨论的必要了。

既然我们一般有  $B_k s_k \neq y_k$ ，又希望  $B_{k+1} s_k = y_k$ ，那么其实对应到上面的这个定理，就是

$$A = B_k, x = s_k, b = y_k, \bar{A} = B_{k+1}$$

然后找到那个  $A$  的更新方式，就会得到下面的这个更新公式

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$$



这就是 **BFGS 更新公式**，因为 Broyden, Fletcher, Goldfarb, Shanno 各自独立发现了这个公式，所以起了这么一个名字。

事实上，根据那个定理，如果  $y_k^T s_k > 0$ ，那么就能够使得我们的矩阵保正定。事实上，这可以通过 Wolfe 条件的第二个不等式（Curvature 条件）得到，这个性质我们在第 3 节也用到过，我们就不再证明了。

好的，现在我们就可以把 BFGS 的更新公式放在下面了。

### A line search BFGS algorithm

**Input:** Initial iterate  $x_0$ , initial SPD  $B_0$ ;

1, Set  $k \leftarrow 0$ ;

**while** not accurate enough **do**

2, Compute  $p_k$  such that  $B_k p_k = -\nabla f(x_k)$ ;

3,  $x_{k+1} \leftarrow x_k + \alpha_k p_k$  with appropriate  $\alpha_k$  (the Wolfe conditions);

4, Compute  $s_k = \alpha_k p_k$  and  $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$ ;

4, Compute

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k};$$

5,  $k \leftarrow k + 1$ ;

**end while**

知乎 @学弱渣

当然，如果我们利用上耳熟能详的 **Sherman-Morrison 公式**

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

我们就可以得到我们的逆形式的 BFGS 更新公式，也即

$$H_{k+1} = (I_n - \frac{s_k y_k^T}{y_k^T s_k}) H_k (I_n - \frac{y_k s_k^T}{y_k^T s_k}) + \frac{s_k s_k^T}{y_k^T s_k}$$

这里  $H_k = B_k^{-1}$ 。事实上，它也是下面这个优化目标的解

$$H_{k+1} = \arg \min_H \|H - H_k\|_{W_H} \quad \text{s.t. } s_k = H y_k, H^T = H$$

这里定义  $\|A\|_{W_H} = \|W_H^{1/2} A W_H^{1/2}\|_F$ 。同样的，我们也不解释为什么它是这个优化目标的解，因为太复杂了.....

当然了，BFGS 方法的全局收敛性也是有理论保障的，我们写在下面。

Theorem 3:

设  $x_0$  为初值， $f \in C^2$ ,  $\mathcal{N}_{x_0} = \{x : f(x) \leq f(x_0)\}$  凸，并且存在  $m, M > 0$

，使得  $m\|z\|^2 \leq z^T \nabla^2 f(x) z \leq M\|z\|^2$  对任意的  $z \in \mathbb{R}^n, x \in \mathcal{N}_{x_0}$  成立。

设  $B_0$  为任意的初始的对称正定阵，那么 BFGS 更新会使得  $\{x_k\}$  收敛到  $f$  的极小值点  $x^*$ 。

这个证明还是有点复杂的，我们一步步来看。这里既然是线搜索方法，那么自然会希望通过 Zoutendijk 条件来说明收敛性。下降方向自然也不必说明，因为我们的矩阵是对称正定阵。如果要说明收敛性，无非就是要说明搜索方向与负梯度的夹角不要取得太差。这也是我们下面证明的一个思路。

如果我们有更新公式为  $B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}$ ，那么通过一个公式，我们可以说明

$$\begin{aligned}\text{tr}(B_{k+1}) &= \text{tr}(B_k) - \frac{\|B_k s_k\|^2}{s_k^T B_k s_k} + \frac{\|y_k\|^2}{y_k^T s_k} \\ |B_{k+1}| &= |B_k| \frac{y_k^T s_k}{s_k^T B_k s_k}\end{aligned}$$

当然了主要是行列式的那个式子需要依赖我们下面给出的公式。我们给出这个公式及其证明，但是中间的转换留给读者思考。

Lemma 1:

$$|I + xy^T + uv^T| = (1 + y^T x)(1 + v^T u) - (x^T v)(y^T u)$$

它的证明属于典型的“想到就很简单，没想到就很复杂”的类型，因为需要考虑到一个变换

$$|I + xy^T + uv^T| = |I + \begin{bmatrix} x & u \end{bmatrix} \begin{bmatrix} y^T \\ v^T \end{bmatrix}| = |I + \begin{bmatrix} x^T \\ u^T \end{bmatrix} \begin{bmatrix} y & v \end{bmatrix}|$$

计算一下行列式即可。

$$\begin{aligned}\text{设 } \cos \theta_k &= \frac{s_k^T B_k s_k}{\|s_k\| \|B_k s_k\|} \quad (\text{这个其实就是夹角，但是不算是很显然}), \quad q_k = \frac{s_k^T B_k s_k}{s_k^T s_k} \\ , \text{ 并且 } m_k &= \frac{y_k^T s_k}{s_k^T s_k}, \quad M_k = \frac{y_k^T y_k}{y_k^T s_k}, \text{ 那么就会有}\end{aligned}$$

$$\begin{aligned}\frac{\|B_k s_k\|^2}{s_k^T B_k s_k} &= \frac{q_k}{\cos^2 \theta_k} \\ |B_{k+1}| &= |B_k| \frac{m_k}{q_k}\end{aligned}$$

这里注意到  $\lambda - \ln \lambda > 0, \forall \lambda > 0$ ，我们会有

$$\Psi(B) = \text{tr}(B) - \ln(\det(B)) > 0$$

我们的目标是通过构造一个不等式，使得我们可以有办法证明出  $\cos \theta_k$  是有下界的。而这个式子的构造就是利用了我们的特征值的表示方法。具体的内容也会看到。

如果有了这个式子，那么我们代入  $B_{k+1}$ ，就会有

$$\begin{aligned}\Psi(B_{k+1}) &= \text{tr}(B_k) + M_k - \frac{q_k}{\cos^2 \theta_k} - \ln(\det(B_k)) - \ln m_k + \ln q_k \\ &\leq \Psi(B_k) + (M_k - \ln m_k - 1) + \ln \cos^2 \theta_k\end{aligned}$$

(这个证明也是有跳步的，读者可以补上这些细节) 所以我们就得到了一个这样的不等式。这个不等式的意义在于我们找到了一个递推，同时注意到  $m_k \geq m, M_k \leq M$  (这是通过题目中的条件得到的，想想为什么?)，所以我们有



$$\Psi(B_{k+1}) \leq \Psi(B_0) + (M - \ln m - 1)(k + 1) + \sum_{j=0}^k \ln \cos^2 \theta_j$$

可以看出，如果我们假设迭代会无休止的进行，那么右边的求和的式子如果没有界，必然会导出矛盾。不妨设  $\cos \theta_k \rightarrow 0$ ，那么这个时候，我们可以看出，肯定会存在一个  $K > 0$ ，使得对任意的  $j > K$ ，有

$$\ln(\cos^2 \theta_k) < -2(M - \ln m - 1)$$

事实上，右边的这个界可以随便取，关键看证明怎么样最方便。

这个时候，设  $C = M - \ln m - 1$ ，我们不难得到

$$0 < \Psi(B_0) + (k + 1)C + \sum_{j=0}^K \ln \cos^2 \theta_j + \sum_{j=K+1}^k (-2C) \rightarrow -\infty (k \rightarrow \infty)$$

这就矛盾了。因此利用 Zoutendijk 条件，我们也算是比较好的证明了最终的全局收敛性。

最后，我们再简单提一下 BFGS 方法的**局部收敛性**。它的局部收敛性对应着下面这两个定理

Theorem 4:

设  $f \in C^2$ ， $\{x_k\}$  通过 BFGS 更新收敛于  $x^*$ ，且  $\nabla^2 f(x^*)$  为对称正定矩阵，设  $B_0$  为任意的对称正定阵，那么存在  $0 \leq \nu < 1$ ， $K > 0$  为整数，使得对任意的  $k > K$ ，有

$$f(x_{k+1}) - f(x^*) \leq \nu^{k-K+1} (f(x_K) - f(x^*)), \text{ 且}$$

$$\sum_{k=0}^{\infty} \|x_k - x^*\| < \infty$$

这个定理对应的是它的 R - 超线性收敛速度，还有一个定理是这样的

Theorem 5:

除了 Theorem 4 的假设以外，还要求  $f$  的海塞矩阵在  $x^*$  处 Lipschitz 连续，那么有

$$\lim_{k \rightarrow \infty} \frac{\|x_{k+1} - x^*\|}{\|x_k - x^*\|} = 0.$$

这个对应的是 Q - 超线性收敛速度。因为证明复杂，所以我们都不会给出证明。

## BFGS 方法的实操细节

不知道是否有人注意到了 BFGS 方法的证明依赖了一个**海塞矩阵凸**的条件，也就是说要求问题是一个**凸问题**。事实上，13 年有人给出了一个极好的例子来说明**存在非凸函数使得 BFGS 方法无法收敛**。在这样的情况下，对于算法做修正就成了重中之重。这一部分我们会介绍两个迭代中可以做的修正方法。

第一个方法可以写成

$$B_{k+1} = \begin{cases} B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} & \frac{y_k^T s_k}{s_k^T s_k} \geq \epsilon \|\nabla f(x_k)\|^\kappa \\ B_k & \text{otherwise} \end{cases}$$

也就是说我们不会要求每一步都更新矩阵  $B_k$ 。这样修改有一个好处，就是一方面保证了问题在当前步的迭代是足够小心的。另一方面也不会影响到大局。这里有一个观察是当靠近迭代点的时候，不等式左边恒正，因为局部的凸性，而右边因为趋近于极小值点而趋近于 0，所以局部上这个不等式其实是失效的，也就不会影响到我们的迭代。

第二个方法可以写成

$$\begin{aligned}\tilde{y}_k &= y_k + r_k s_k \\ B_{k+1} &= B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{\tilde{y}_k \tilde{y}_k^T}{\tilde{y}_k^T s_k}\end{aligned}$$

可以看出我们这里对于  $y_k$  做了一个修正，并且使用了修正后的  $\tilde{y}_k$  做正常的迭代。这样的思想是：既然问题不是凸问题，那么就考虑加一个很凸的东西，使得问题变为凸问题。对于这里，其实就是将问题改成了

$$\tilde{f}(x) = f(x) + \frac{1}{2} r_k \|x_k\|^2$$

如果你对它求梯度，计算  $\tilde{y}_k$ ，发现它正好就是  $y_k + r_k s_k$ 。那么这个  $r_k$  就决定了你“拉凸”的程度。原始论文上提供的是

$$r_k = (1 + \max(-\frac{y_k^T s_k}{s_k^T s_k}, 0)) \|\nabla f(x_k)\|$$

这两个方法都可以使得最终的算法在非凸情况下也可以具有全局收敛性。

到此，我们算是介绍好了 BFGS 方法的内容，后面我们会介绍一些其它的算法和理论。

## DFP 方法

事实上，如果我们把所有的  $H_k$  都改成  $B_k$ ，就会有一个全新的更新公式叫做 **DFP 更新公式**，也就是

$$B_{k+1} = (I_n - \frac{s_k y_k^T}{y_k^T s_k}) B_k (I_n - \frac{y_k s_k^T}{y_k^T s_k}) + \frac{s_k s_k^T}{y_k^T s_k}$$

不过 DFP 公式用的是比较少的，原因有两处，一是收敛性证明目前还空缺，另外一个数值上存在一个巨大的缺陷。我们给一个例子说明。

Example 1:

考虑  $f(x) = \frac{1}{2} x^T A x$ ，其中  $A = O \text{diag}(1, 1, 1, 2, 4) O^T$ ， $O$  为一个正交矩阵，定义初值为  $x_0 = (1, 1, 1, 1, 1)^T$ ，初始的海塞矩阵的逆的估计为  $H_0 = \text{diag}(1, 1, 1, 1, L)$ 。

我们可以看到，如果  $L$  的取值比较小，相当于估计  $B_k$  的逆存在某一个很小的特征值，那么相当于说  $B_k$  存在一个特征值会与其它的特征值差距很大。下面给出了数值实验的结果。

	$L$					
	1	$10^{-1}$	$10^{-2}$	$10^{-3}$	$10^{-4}$	$10^{-5}$
BFGS iter	5	6	14	20	23	27
DFP iter	5	10	61	495	4836	48301

可以看出，对于 DFP 方法，它很难修正这一个情况。所以事实上它已经很少被人使用了。

## Broyden 族

这个方法也是一个很有趣的方法。简要说来，可以把它写成

$$(1 - \phi_k)BFGS + \phi_k DFP$$

相当于对 BFGS 方法和 DFP 方法做了一个线性组合。如果我们设

$$\phi_k = \frac{s_k^T y_k}{s_k^T y_k - s_k^T B_k s_k}$$

那么这个时候得到的方法就是 SR1 方法。

这个方法在取  $\phi_k$  比较好的时候是会比 BFGS 方法的表现更好的，但是问题也恰好是在这里。一般来说  $\phi_k$  是**没有一个很好的准则来进行选取的**。所以我们这里也只是提一下，感兴趣的同学可以自己参考其他资料。

## 统一拟牛顿方法的 DM 条件

讲完了 BFGS，DFP 和 Brodyen-Family 等具体的拟牛顿算法，下面提供一个很有意思的**保证拟牛顿法收敛速度**的定理。

Theorem 6:

设更新公式为  $x_{k+1} = x_k + p_k$ ， $f \in C^2$ ，且  $\{x_k\}$  收敛到点  $x^*$ ，且  $\nabla f(x^*) = 0$ ， $\nabla^2 f(x^*)$  对称正定，那么迭代具有 Q - 超线性收敛速度当且仅当

$$\lim_{k \rightarrow \infty} \frac{\|\nabla f(x_k) + \nabla^2 f(x_k)p_k\|}{\|p_k\|} = 0.$$

我们证明一下这个结论，首先观察到，如果说取

$$p_k^N = -\nabla^2 f(x_k)^{-1} \nabla f(x_k)$$

那么代回去就会得到分子是 0，所以我们的这个极限的分子部分，就是**拟牛顿法逼近牛顿法搜索方向的程度**。所以我们先说明式子等价于  $p_k - p_k^N = o(\|p_k\|)$ 。

一方面，如果极限式子成立，那么我们有

$$\begin{aligned} p_k - p_k^N &= \nabla^2 f(x_k)^{-1} (\nabla^2 f(x_k)p_k + \nabla f(x_k)) \\ &= \|\nabla^2 f(x_k)^{-1}\| o(\|p_k\|) = o(\|p_k\|) \end{aligned}$$

(如果取  $k \rightarrow \infty$ ，那么在靠近极小值点的时候，海塞矩阵的逆的范数是有界的) 另一方面，如果有  $p_k - p_k^N = o(\|p_k\|)$ ，那么其实也就是把式子倒过来写，变成

$$\|\nabla f(x_k) + \nabla^2 f(x_k)p_k\| \leq \|\nabla^2 f(x_k)\| \|p_k - p_k^N\| = o(\|p_k\|)$$

所以也是一样的。接下来我们来看看为什么这个性质可以推出我们要的超线性收敛速度的结论。注意到首先，我们通过拆解容易得到

$$\|x_{k+1} - x^*\| \leq \|x_k + p_k^N - x^*\| + \|p_k - p_k^N\|$$

根据  $p_k^N$  就是牛顿法的方向，加上我们认为  $x_k$  应该已经靠近了迭代点  $x^*$ ，所以有

$$\|x_{k+1} - x^*\| = O(\|x_k - x^*\|^2) + o(\|p_k\|)$$

如果要说明是超线性收敛速度，那么自然的希望说明的就是  $\|p_k\|$  的量级应该是和  $\|x_k - x^*\|$  差不多的，这是因为如果是这样的话，那么加上小量的约束，就可以得到超线性收敛速度（这里要区分  $o(x_k)$  和  $O(x_k^2)$  的含义，二者并不等价，后者明显要更优一些）。

基于此，首先我们发现

$$\|p_k\| \leq \|x_{k+1} - x^*\| + \|x_k - x^*\|$$

之前我们已经给  $\|x_{k+1} - x^*\|$  估计过一个界了，代入我们就可以得到

$$\|p_k\| \leq O(\|x_k - x^*\|)$$

要说明相等，还差另外一边。但是另外一边的说明其实也并不困难。因为我们的三角不等式还有一个形式

$$\|p_k\| \leq \|x_k - x^*\| - \|x_{k+1} - x^*\|$$

所以同样的推断可以得到  $\|p_k\| \geq O(\|x_k - x^*\|)$ ，这就说明了  $\|p_k\|$  的量级。这个量级给定之后，就不难得到  $\|x_{k+1} - x^*\| = o(\|x_k - x^*\|)$ ，这就是 Q - 超线性收敛速度的含义，所以我们就证明了这个结论。

这个定理非常棒，但是有的人可能会问一个问题，就是这里的更新公式中的步长默认为 **1**，实际情况下**不需要做步长选取吗**？答案是肯定的，事实上，有定理保证在某一步之后所有的步长选取条件都不再需要，也就是说  $\alpha_k = 1$  是一定成立的。但是我们这里不提更多的细节。感兴趣的可以参考一下 Source 中的内容或者 Google 学术一下。

好的，到此我们就算是介绍好了所有的拟牛顿法的重要内容。

## 小结

这一节我们主要关注的是拟牛顿法的算法，理论和应用。因为它可以巧妙地避开牛顿法中**对海塞矩阵的逆**的求解，同时可以保证算法具有超线性的收敛速度。事实上从拟牛顿法开始，我们的理论部分就开始减少了，**对于实操的和理解的部分要求增多**。当然了，这样的思想在优化中，自然也是很重要的。