



# 拟牛顿法

王尧

西安交通大学智能决策与机器学习中心  
(Email: [yao.s.wang@gmail.com](mailto:yao.s.wang@gmail.com))

2022. 6

# 回顾：梯度法与牛顿法

Back to unconstrained, smooth convex optimization

$$\min_x f(x)$$

where  $f$  is convex, twice differentiable, and  $\text{dom}(f) = \mathbb{R}^n$ . Recall **gradient descent** update:

$$x^+ = x - t \nabla f(x)$$

and **Newton's method** update:

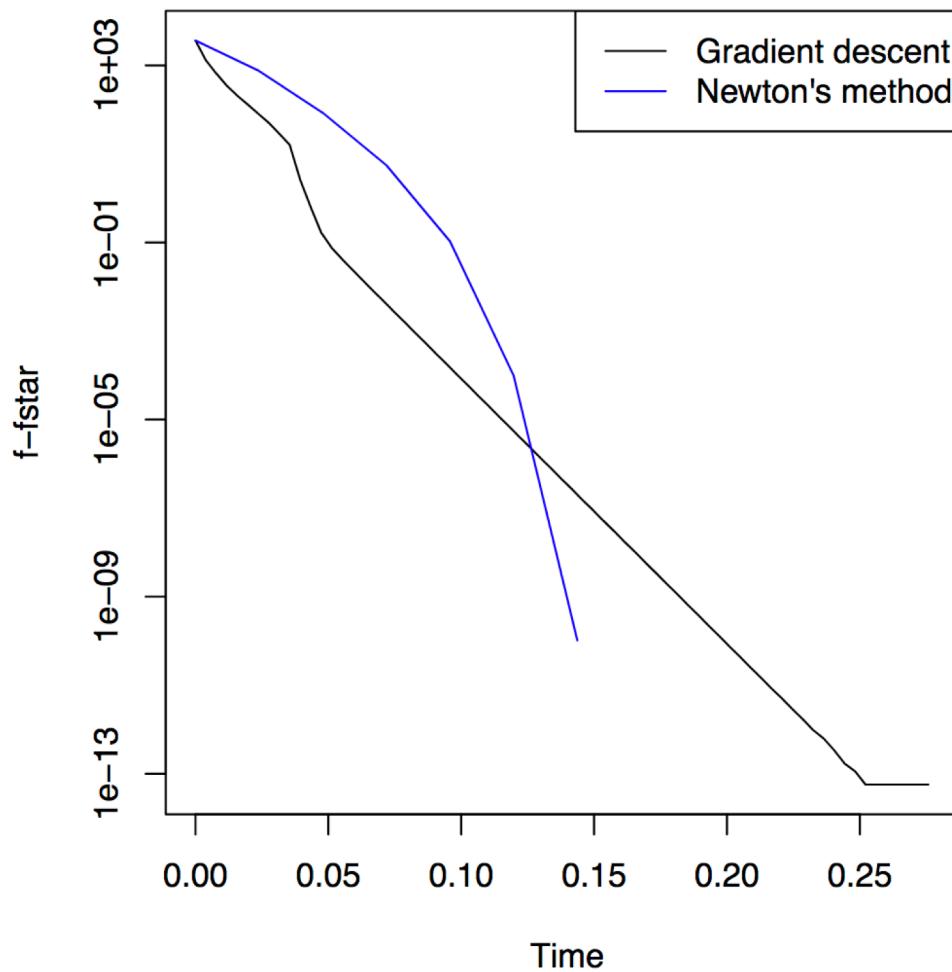
$$x^+ = x - t(\nabla^2 f(x))^{-1} \nabla f(x)$$

- Newton's method has (local) quadratic convergence, versus linear convergence of gradient descent
- But Newton iterations are much more expensive ...

# 回顾：梯度法与牛顿法

- **Memory**: each iteration of Newton's method requires  $O(n^2)$  storage ( $n \times n$  Hessian); each gradient iteration requires  $O(n)$  storage ( $n$ -dimensional gradient)
- **Computation**: each Newton iteration requires  $O(n^3)$  flops (solving a dense  $n \times n$  linear system); each gradient iteration requires  $O(n)$  flops (scaling/adding  $n$ -dimensional vectors)
- **Backtracking**: backtracking line search has roughly the same cost, both use  $O(n)$  flops per inner backtracking step

# Back to logistic regression example



# 回顾：拟牛顿法

Two main steps in Newton iteration:

- Compute Hessian  $\nabla^2 f(x)$
- Solve the system  $\nabla^2 f(x)s = -\nabla f(x)$

Each of these two steps could be expensive

Quasi-Newton methods repeat updates of the form

$$x^+ = x + ts$$

where direction  $s$  is defined by linear system

$$Bs = -\nabla f(x)$$

for some approximation  $B$  of  $\nabla^2 f(x)$ . We want  $B$  to be easy to compute, and  $Bs = g$  to be easy to solve

# 一点历史

- In the mid 1950s, W. Davidon was a mathematician/physicist at Argonne National Lab
- He was using coordinate descent on an optimization problem and his computer kept crashing before finishing
- He figured out a way to accelerate the computation, leading to the first quasi-Newton method (soon Fletcher and Powell followed up on his work)
- Although Davidon's contribution was a major breakthrough in optimization, his original paper was rejected
- In 1991, after more than 30 years, his paper was published in the first issue of the SIAM Journal on Optimization
- In addition to his remarkable work in optimization, Davidon was a peace activist (see the book "The Burglary")

# 割线方程

对 $\nabla f(x)$ 在点 $x^{k+1}$ 处一阶Taylor近似, 得

$$\nabla f(x) = \nabla f(x^{k+1}) + \nabla^2 f(x^{k+1})(x - x^{k+1}) + \mathcal{O}(\|x - x^{k+1}\|^2),$$

令 $x = x^k$ , 且 $s^k = x^{k+1} - x^k$ 为点差,  $y^k = \nabla f(x^{k+1}) - \nabla f(x^k)$ 为梯度差, 得

$$\nabla^2 f(x^{k+1})s^k + \mathcal{O}(\|s^k\|^2) = y^k.$$

这是Hesse矩阵满足的方程.

现忽略高阶项 $\|s^k\|^2$ , 只希望近似Hesse矩阵的矩阵 $B^{k+1}$ 满足方程

$$B^{k+1}s^k = y^k,$$

或其逆矩阵 $H^{k+1}$ 满足

$$H^{k+1}y^k = s^k.$$

# 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

$$(s^k)^T B^k s^k > 0 \implies (s^k)^T y^k > 0,$$

这一条件针对的是一切的 $B^k$ ，因此要在迭代过程中始终满足。

## 定义

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

# 拟牛顿算法

---

## 算法 1 拟牛顿算法框架

---

**Require:** 初始坐标  $x^0 \in \mathbb{R}^n$ , 初始矩阵  $B^0 \in \mathbb{R}^{n \times n}$ (或  $H^0$ ),  $k = 0$ .  
**Ensure:**  $x^K, B^K$ (或  $H^K$ ).

- 1: 检查初始元素.
  - 2: **while** 未达到停机准则 **do**
  - 3: 计算方向  $d^k = -(B^k)^{-1} \nabla f(x^k)$  或  $d^k = -H^k \nabla f(x^k)$ .
  - 4: 通过线搜索(Wolfe)产生步长  $\alpha_k > 0$ , 令  $x^{k+1} = x^k + \alpha_k d^k$ .
  - 5: 更新 Hesse 矩阵的近似矩阵  $B^{k+1}$  或其逆矩阵  $H^{k+1}$ .
  - 6:  $k \leftarrow k + 1$ .
  - 7: **end while**
-

# 秩1矩阵更新

秩一更新是一种迭代式更新矩阵的手段，它的结构非常简单。我们可以用秩一更新的方式更新拟牛顿矩阵。

## 定义

秩一更新 对于拟牛顿矩阵  $B^k \in \mathbb{R}^{n \times n}$ , 设  $0 \neq u \in \mathbb{R}^n$  且  $a \in \mathbb{R}$  待定, 则  $uu^T$  是秩一矩阵, 且有秩一更新

$$B^{k+1} = B^k + auu^T.$$

进一步我们确定参量  $u$  和  $a$ . 根据割线方程  $B^{k+1}s^k = y^k$ , 代入秩一更新的结果, 得到

$$(B^k + auu^T)s^k = y^k,$$

整理得

$$auu^T s^k = (a \cdot u^T s^k)u = y^k - B^k s^k.$$

由于  $a \cdot u^T s^k$  是标量, 因此上式表明  $u$  和  $y^k - B^k s^k$  同向.

# 秩1矩阵更新

根据

$$auu^T s^k = (a \cdot u^T s^k)u = y^k - B^k s^k,$$

我们推出 $u$ 和 $y^k - B^k s^k$ 同向. 此时将会有诸多选择以确定具体的参量. 处于简单考虑, 不妨就令 $u$ 和 $y^k - B^k s^k$ 相等, 即 $\textcolor{red}{u} = y^k - B^k s^k$ , 代入上式得

$$(a \cdot (y^k - B^k s^k)^T s^k)(y^k - B^k s^k) = y^k - B^k s^k,$$

再令 $(a \cdot (y^k - B^k s^k)^T s^k) \neq 0$ , 则可以确定 $a$ 为

$$\textcolor{red}{a} = \frac{1}{(y^k - B^k s^k)^T s^k}.$$

至此,  $u$ 和 $a$ 均被确定.

根据以上算法, 一种 $B^k$ 的秩一更新(SR1)公式为

$$\textcolor{red}{B^{k+1} = B^k + \frac{uu^T}{u^T s^k}, \quad u = y^k - B^k s^k.}$$

# 秩1矩阵更新

我们推出了基于 $B^k$ 的秩一更新公式. 由同样的过程(请推导), 可以推出基于 $H^k$ 的秩一更新公式.

## 定理

拟牛顿算法的秩一更新公式 拟牛顿矩阵 $B^k$ 的秩一更新公式为

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

拟牛顿矩阵 $H^k$ 的秩一更新公式为

$$H^{k+1} = H^k + \frac{vv^T}{v^T y^k}, \quad v = s^k - H^k y^k.$$

# 秩一更新公式的缺陷

秩一公式的推导很简单, 形式也很简洁, 但是由秩一公式更新的 $B^{k+1}$ 无法保证正定, 即使 $B^k$ 正定.

## 定理

秩一更新公式使 $B^{k+1}$ 正定的充分条件 使用秩一更新公式从 $B^k$ 更新 $B^{k+1}$ ,  $B^{k+1}$ 正定的充分条件可以是:

- (1)  $B^k$  正定;
- (2)  $u^T s^k > 0$ .

证明是简单的. 设 $0 \neq w \in \mathbb{R}^n$ , 则

$$\begin{aligned} w^T B^{k+1} w &= w^T B^k w + \frac{w^T u u^T w}{u^T s^k} = w^T B^k w + \frac{(u^T w)^2}{u^T s^k} \\ &> 0. \end{aligned}$$

注: 实际中上述条件很难保证

# Broyden-Fletcher-Goldfarb-Shanno (BFGS)

Broyden, Fletcher, Goldfarb, Shanno



# BFGS更新

BFGS公式的核心思想是对 $B^k$ 进行秩二更新，而不是秩一更新。

## 定义

**秩二更新** 对于拟牛顿矩阵 $B^k \in \mathbb{R}^{n \times n}$ , 设 $0 \neq u, v \in \mathbb{R}^n$ 且 $a, b \in \mathbb{R}$ 待定, 则有秩二更新形式

$$B^{k+1} = B^k + auu^T + bv v^T.$$

根据割线方程, 将秩二更新的待定参量式代入, 得

$$B^{k+1}s^k = (B^k + auu^T + bv v^T)s^k = y^k,$$

整理可得

$$(a \cdot u^T s^k)u + (b \cdot v^T s^k)v = y^k - B^k s^k.$$

上式的形式和秩一更新的情形类似, 不过是多加了一项.

# BFGS更新

类似处理, 我们通过选取合适的 $u, v, a, b$ 使得上式成立, 一个简单的取法是令 $(a \cdot u^T s^k)u$ 对应 $y^k$ 相等,  $(b \cdot v^T s^k)v$ 对应 $-B^k s^k$ 相等, 即有

$$a \cdot u^T s^k = 1, \quad u = y^k,$$

$$b \cdot v^T s^k = -1, \quad v = B^k s^k.$$

将上述参量代入割线方程, 即得**BFGS更新公式**

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

其中 $u, v$ 如上定义.

# BFGS公式

综上所述, BFGS公式的定义如下.

## 定义

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

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

基于 $H^k$ 的*BFGS*公式为

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

# BFGS公式

BFGS公式产生的 $B^{k+1}$ 或 $H^{k+1}$ 是否正定呢？我们通过一个充分性定理说明。

## 定理

**BFGS公式使拟牛顿矩阵正定的充分条件** 使用秩一更新公式从 $B^k$ 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  $(s^k)^T y^k > 0, \forall k \in \mathbb{N}^+$ .

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。

注：使用某一线搜索法(如后退法)即可满足曲率条件

# Davidon-Fletcher-Powell更新

DFP公式利用与BFGS公式类似的推导方法, 不同的是其以割线方程 $H^{k+1}y^k = s^k$ 为基础进行对 $H^k$ 的秩二更新. 读者可以再练习推导.

## 定义

*DFP公式* 基于 $H^k$ 的*DFP更新公式*为

$$H^{k+1} = H^k - \frac{H^k y^k (H^k y^k)^T}{(y^k)^T H^k y^k} + \frac{s^k (s^k)^T}{(y^k)^T s^k},$$

基于 $B^k$ 的*DFP更新公式*为

$$B^{k+1} = (I - \frac{y^k (s^k)^T}{(s^k)^T y^k})^T B^k (I - \frac{y^k (s^k)^T}{(s^k)^T y^k}) + \frac{y^k (y^k)^T}{(s^k)^T y^k}.$$

注: DFP更新与BFGS更新互为对偶关系

# 收敛性

Assume that  $f$  convex, twice differentiable, having  $\text{dom}(f) = \mathbb{R}^n$ , and additionally

- $\nabla f$  is Lipschitz with parameter  $L$
- $f$  is strongly convex with parameter  $m$
- $\nabla^2 f$  is Lipschitz with parameter  $M$

(same conditions as in the analysis of Newton's method)

**Theorem:** Both DFP and BFGS, with backtracking line search, converge globally. Furthermore, for all  $k \geq k_0$ ,

$$\|x^{(k)} - x^*\|_2 \leq c_k \|x^{(k-1)} - x^*\|_2$$

where  $c_k \rightarrow 0$  as  $k \rightarrow \infty$ . Here  $k_0, c_k$  depend on  $L, m, M$

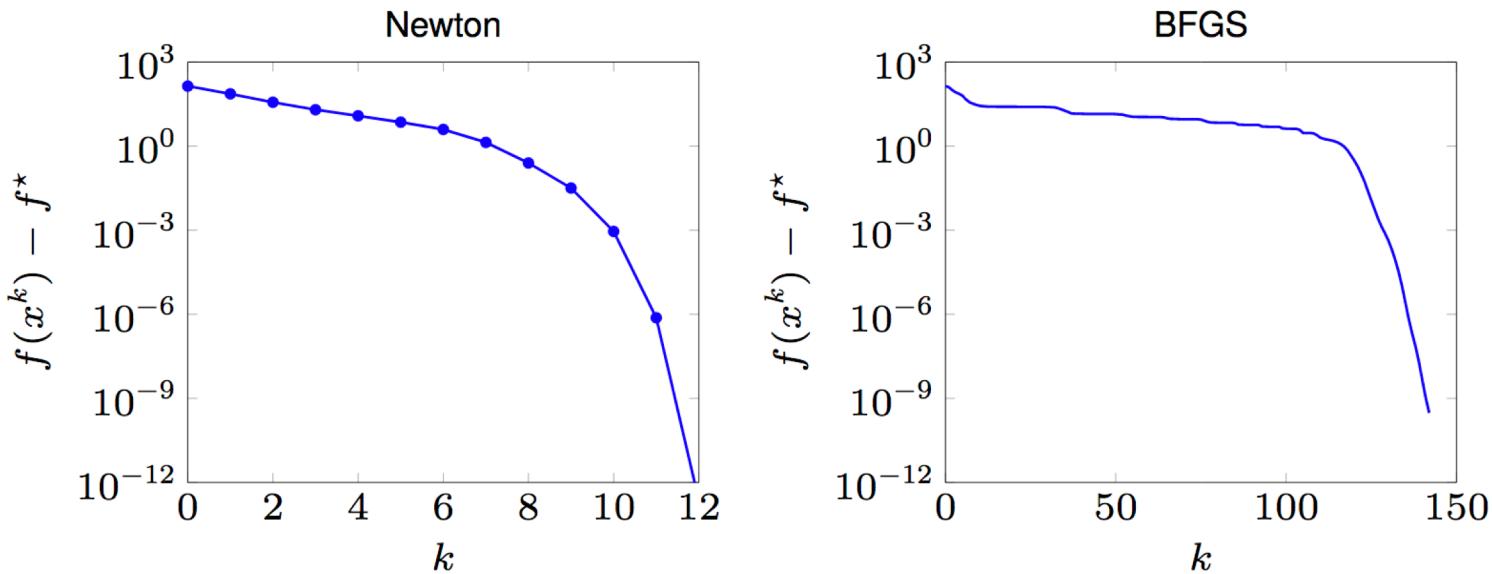
This is called **local superlinear convergence**

注：请仔细阅读袁亚湘版1.5节

# Example: Newton versus BFGS

Example from Vandenberghe's lecture notes: Newton versus BFGS on LP barrier problem, for  $n = 100$ ,  $m = 500$

$$\min_x \quad c^T x - \sum_{i=1}^m \log(b_i - a_i^T x)$$



Note that Newton update is  $O(n^3)$ , quasi-Newton update is  $O(n^2)$ .

*Thank you for your  
attentions !*