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)

2022. 6

牛顿法

Given unconstrained, smooth convex optimization

$$\min_x \ f(x)$$

where f is convex, twice differentable, and $\mathrm{dom}(f) = \mathbb{R}^n$. Recall that gradient descent chooses initial $x^{(0)} \in \mathbb{R}^n$, and repeats

$$x^{(k)} = x^{(k-1)} - t_k \cdot \nabla f(x^{(k-1)}), \quad k = 1, 2, 3, \ldots$$

In comparison, Newton's method repeats

$$x^{(k)} = x^{(k-1)} - \left(\nabla^2 f(x^{(k-1)})\right)^{-1} \nabla f(x^{(k-1)}), \quad k = 1, 2, 3, \ldots$$

Here $\nabla^2 f(x^{(k-1)})$ is the Hessian matrix of f at $x^{(k-1)}$

牛顿法

Recall the motivation for gradient descent step at x: we minimize the quadratic approximation

$$f(y) \approx f(x) + \nabla f(x)^T (y - x) + \frac{1}{2t}\|y - x\|_2^2$$

over y, and this yields the update $x^+ = x - t\nabla f(x)$

Newton's method uses in a sense a better quadratic approximation

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

and minimizes over y to yield $x^+ = x - (\nabla^2 f(x))^{-1} \nabla f(x)$

注：若Hessian为正定矩阵，则牛顿方向为下降方向

二次逼近的几何展示

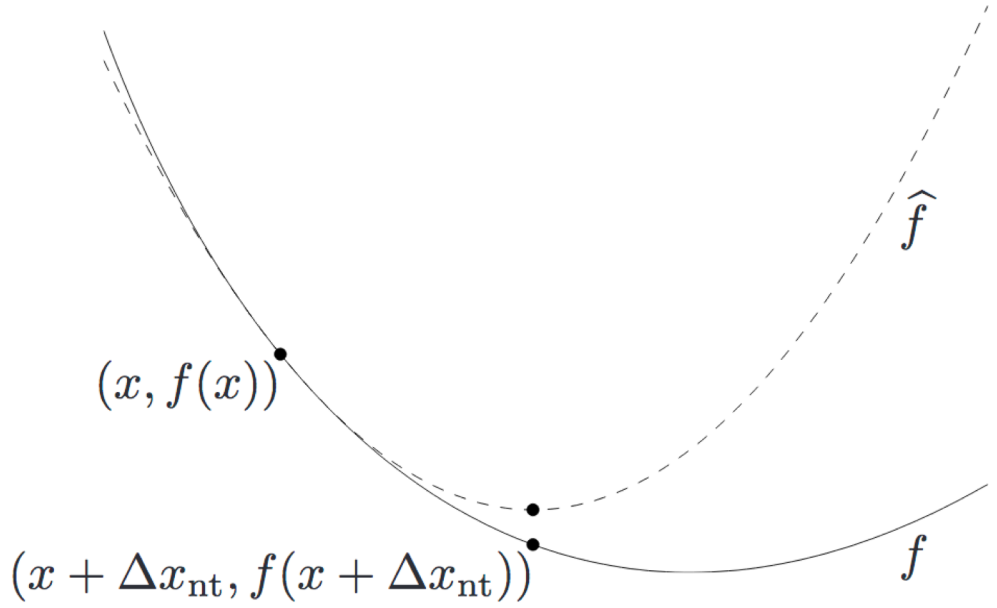


Figure 9.16 The function f (shown solid) and its second-order approximation $\widehat{f}$ at x (dashed). The Newton step Δx_{nt} is what must be added to x to give the minimizer of $\widehat{f}$.

Boyd版9.5.1节

线性化最优性条件

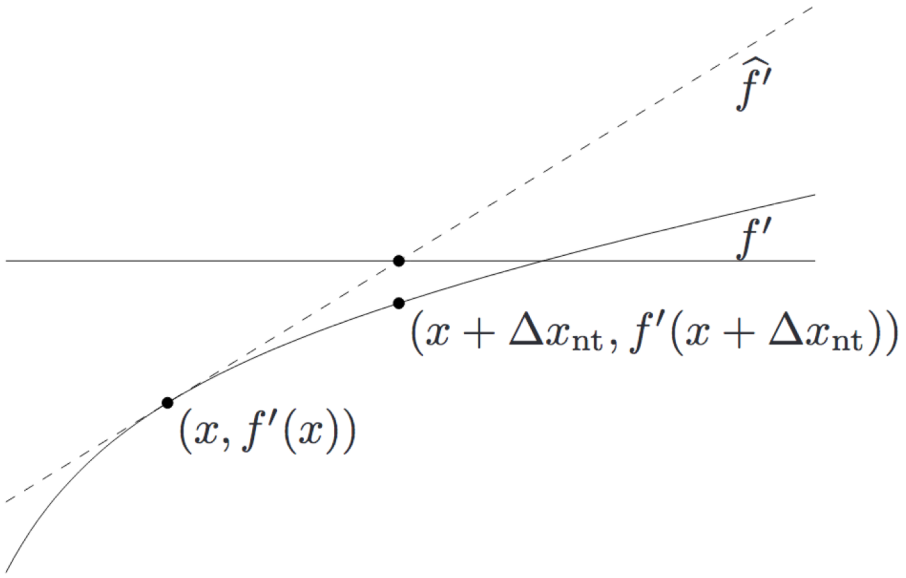


Figure 9.18 The solid curve is the derivative f' of the function f shown in figure 9.16. $\widehat{f'}$ is the linear approximation of f' at x. The Newton step Δx_{nt} is the difference between the root of $\widehat{f'}$ and the point x.

$$\nabla f(x+v) \approx \nabla f(x) + \nabla^2 f(x)v = 0$$

Boyd版9.5.2节

牛顿减量

At a point x, we define the Newton decrement as

$$\lambda(x) = \left(\nabla f(x)^T \left(\nabla^2 f(x)\right)^{-1} \nabla f(x)\right)^{1/2}$$

This relates to the difference between $f(x)$ and the minimum of its quadratic approximation:

$$\begin{aligned} & f(x) - \min_y \left(f(x) + \nabla f(x)^T (y-x) + \frac{1}{2}(y-x)^T \nabla^2 f(x)(y-x)\right) \\ &= f(x) - \left(f(x) - \frac{1}{2}\nabla f(x)^T \left(\nabla^2 f(x)\right)^{-1} \nabla f(x)\right) \\ &= \frac{1}{2}\lambda(x)^2 \end{aligned}$$

注：牛顿减量可用于迭代算法的终止条件

后退线性搜索

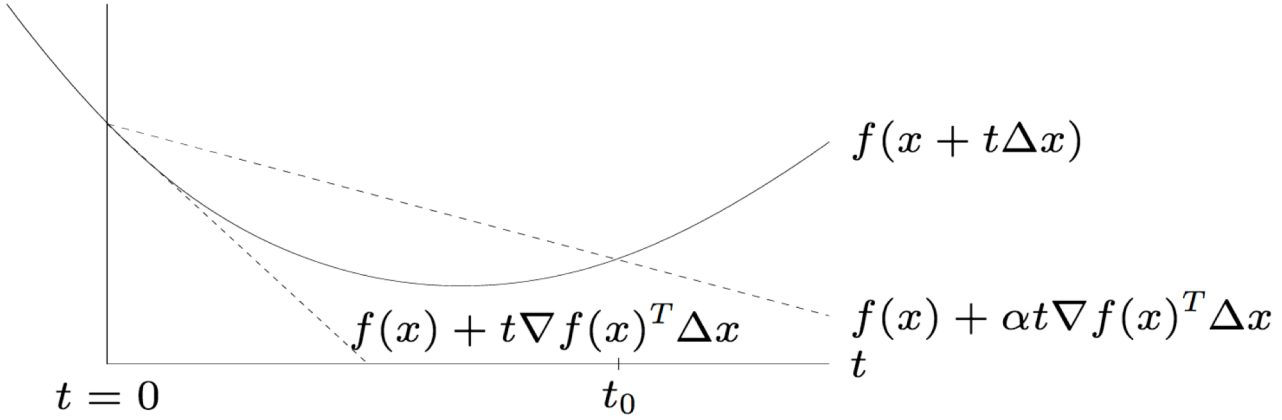
exact line search: $t = \operatorname{argmin}_{t>0} f(x + t\Delta x)$

backtracking line search (with parameters $\alpha \in (0, 1/2)$, $\beta \in (0, 1)$)

- starting at $t = 1$, repeat $t := \beta t$ until

$$f(x + t\Delta x) < f(x) + \alpha t \nabla f(x)^T \Delta x$$

- graphical interpretation: backtrack until $t \leq t_0$



对充分小的t, $f(x + t\Delta x) \approx f(x) + t\nabla f(x)^T \Delta x < f(x) + \alpha t \nabla f(x)^T \Delta x$

后退线性搜索

In practice, we use damped Newton's method (typically just called Newton's method), which repeats

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

Note that the pure method uses $t = 1$

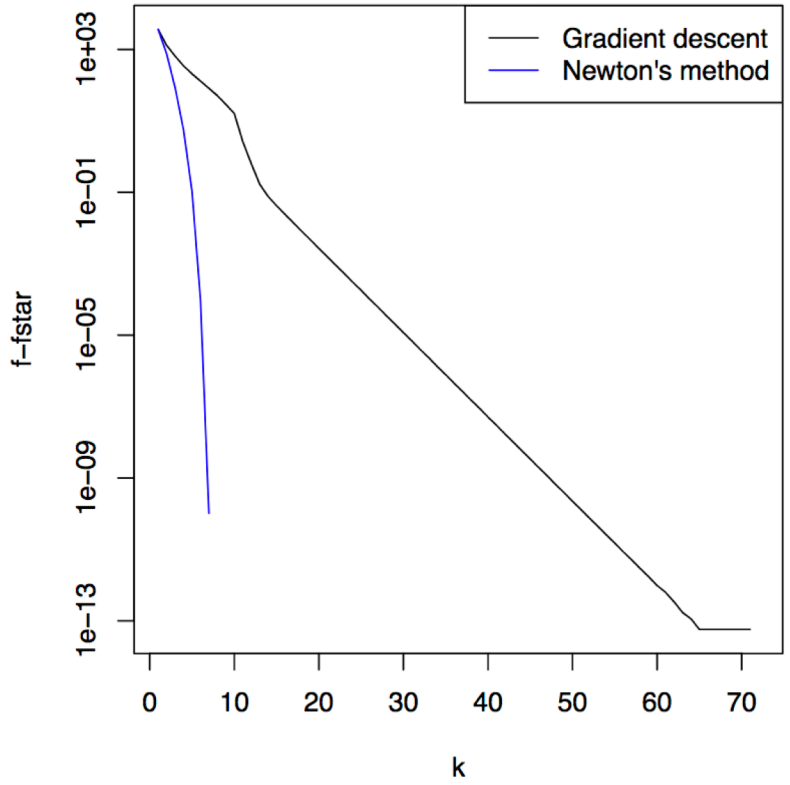
Step sizes here are chosen by backtracking search, with parameters $0 < \alpha \leq 1/2$, $0 < \beta < 1$. At each iteration, start with $t = 1$, while

$$f(x + tv) > f(x) + \alpha t \nabla f(x)^T v$$

we shrink $t = \beta t$, else we perform the Newton update. Note that here $v = -(\nabla^2 f(x))^{-1}\nabla f(x)$, so $\nabla f(x)^T v = -\lambda^2(x)$

Example: Logistic Regression

Logistic regression example, with $n = 500$, $p = 100$: we compare gradient descent and Newton's method, both with backtracking



Newton's method: in a totally different regime of convergence...!

收敛性分析

assumptions

- f strongly convex on S with constant m
- $\nabla^2 f$ is Lipschitz continuous on S, with constant $L > 0$:

$$\|\nabla^2 f(x) - \nabla^2 f(y)\|_2 \leq L\|x - y\|_2$$

outline: there exist constants $\eta \in (0, m^2/L)$, $\gamma > 0$ such that

- if $\|\nabla f(x)\|_2 \geq \eta$, then $f(x^{(k+1)}) - f(x^{(k)}) \leq -\gamma$
- if $\|\nabla f(x)\|_2 < \eta$, then

$$\frac{L}{2m^2}\|\nabla f(x^{(k+1)})\|_2 \leq \left(\frac{L}{2m^2}\|\nabla f(x^{(k)})\|_2\right)^2$$

收敛性分析

damped Newton phase ($\|\nabla f(x)\|_2 \geq \eta$)

- most iterations require backtracking steps
- function value decreases by at least γ
- if $p^\star > -\infty$, this phase ends after at most $(f(x^{(0)}) - p^\star)/\gamma$ iterations

quadratically convergent phase ($\|\nabla f(x)\|_2 < \eta$)

- all iterations use step size $t = 1$
- $\|\nabla f(x)\|_2$ converges to zero quadratically: if $\|\nabla f(x^{(k)})\|_2 < \eta$, then

$$\frac{L}{2m^2}\|\nabla f(x^l)\|_2 \leq \left(\frac{L}{2m^2}\|\nabla f(x^k)\|_2\right)^{2^{l-k}} \leq \left(\frac{1}{2}\right)^{2^{l-k}}, \qquad l \geq k$$

收敛性结论

conclusion: number of iterations until $f(x) - p^\star \leq \epsilon$ is bounded above by

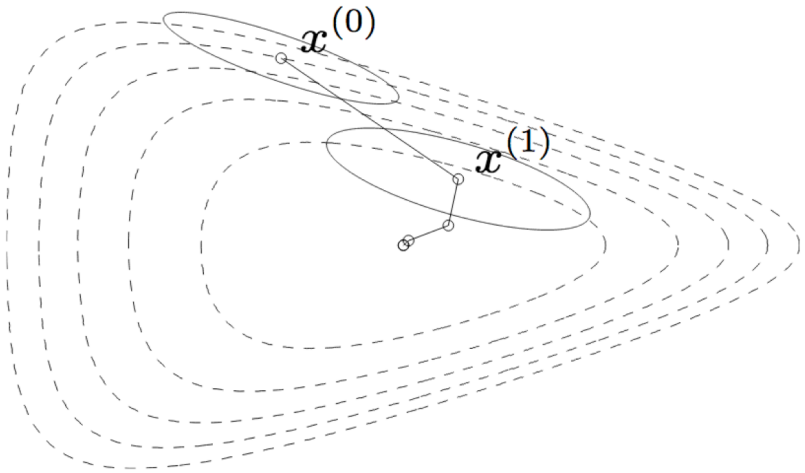
$$\frac{f(x^{(0)}) - p^\star}{\gamma} + \log_2 \log_2(\epsilon_0/\epsilon)$$

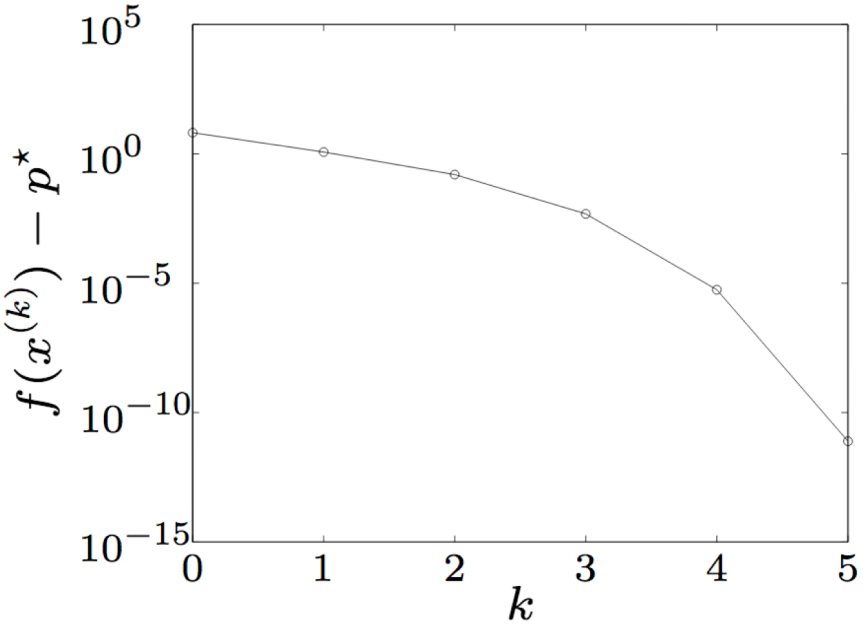
- γ, ϵ_0 are constants that depend on m, L, $x^{(0)}$
- second term is small (of the order of 6) and almost constant for practical purposes
- in practice, constants m, L (hence γ, ϵ_0) are usually unknown
- provides qualitative insight in convergence properties ($i.e.$, explains two algorithm phases)

注：上述收敛结果通常称为local convergence rate

一个简单例子

$$f(x_1, x_2) = e^{x_1+3x_2-0.1} + e^{x_1-3x_2-0.1} + e^{-x_1-0.1}$$



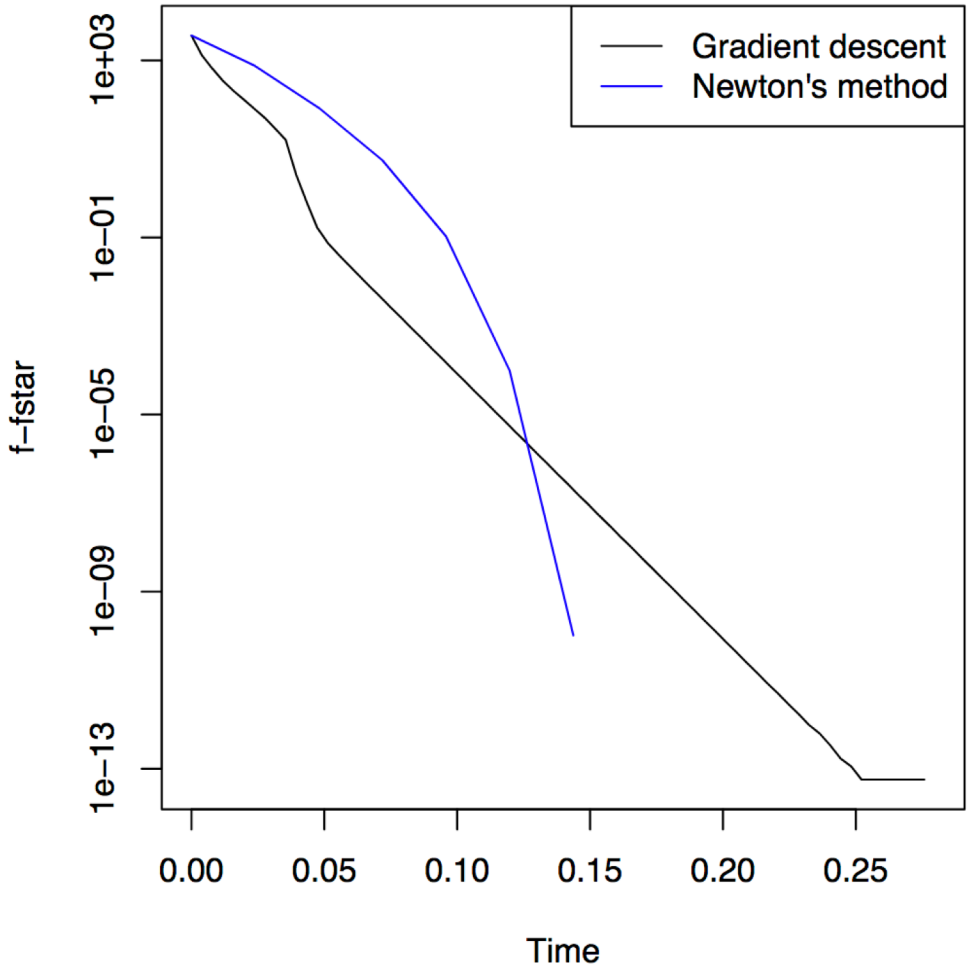


- backtracking parameters $\alpha = 0.1$, $\beta = 0.7$
- converges in only 5 steps

与梯度法的比较

- 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



拟牛顿法

If the Hessian is too expensive (or singular), then a quasi-Newton method can be used to approximate $\nabla^2 f(x)$ with $H \succ 0$, and we update according to

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

- Approximate Hessian H is recomputed at each step. Goal is to make H^{-1} cheap to apply (possibly, cheap storage too)
- Convergence is fast: superlinear, but not the same as Newton. Roughly n steps of quasi-Newton make same progress as one Newton step

Thank you for your attentions !