Convex Optimization

Part 5: Newton's method

Namhoon Lee

POSTECH

21 Nov 2022

Admin

Assignment 4

- available on PLMS
- due by Friday 9 December

Motivation

Consider unconstrained smooth convex optimization

$$\min_x f(x)$$

Gradient descent: starts from an initial point and repeats

$$x_{t+1} = x_t - \eta_t \nabla f(x_t)$$

- achieves linear convergence rate under strong convexity $\mathcal{O}(\kappa \log 1/\epsilon)$

Newton's method: starts from an initial point and repeats

$$x_{t+1} = x_t - \left(\nabla^2 f(x_t)\right)^{-1} \nabla f(x_t)$$

- achieves quadratic convergence rate under strong convexity $\mathcal{O}(\log\log(1/\epsilon))$

Newton step

For $x \in \operatorname{dom} f$, the vector

$$\Delta x_{\mathsf{nt}} = -\big(\nabla^2 f(x)\big)^{-1} \nabla f(x)$$

is called the Newton step (for f, at x).

Positive definiteness of $\nabla^2 f(x)$ implies that

$$\nabla f(x)^\top \Delta x_{\mathsf{nt}} = -\nabla f(x)^\top \big(\nabla^2 f(x)\big)^{-1} \nabla f(x) < 0$$

unless $\nabla f(x) = 0$, so the Newton step is a descent direction (unless x is optimal).

Minimizer of second-order approximation

The second-order Taylor approximation (or model) $\hat{f}$ of f at x is

$$\hat{f}(x+v) \approx f(x) + \nabla f(x)^\top v + \frac{1}{2} v^\top \nabla^2 f(x) v$$

- a convex quadratic function of v and minimized when $v = \Delta x_{\mathsf{nt}}$; the Newton step Δx_{nt} is what should be added to the point x to minimize the second-order approximation of f at x.
- If f is quadratic, then $x + \Delta x_{\mathsf{nt}}$ is the exact minimizer of f.
- If f is nearly quadratic, $x + \Delta x_{\mathsf{nt}}$ should be a very good estimate of the minimizer of f, *i.e.*, $x^\star$.

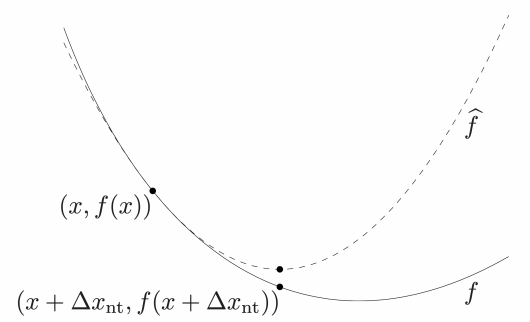

Figure: the function f (solid) and its second-order approximation $\hat{f}$ at x (dashed); $x + \Delta x_{\mathrm{nt}}$ is the minimizer of $\hat{x}$; figure from BV

Solution of linearized optimality condition

If we linearize the optimality condition $\nabla f(x^\star) = 0$ near x we obtain

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

which is a linear equation in v, with solution $v = \Delta x_{\mathsf{nt}}$. So the Newton step Δx_{nt} is what must be added to x so that the linearized optimality condition holds. Again, this suggests that when x is near $x^\star$ (so the optimality conditions almost hold), the update $x + \Delta x_{\mathsf{nt}}$ should be a very good approximation of $x^\star$.

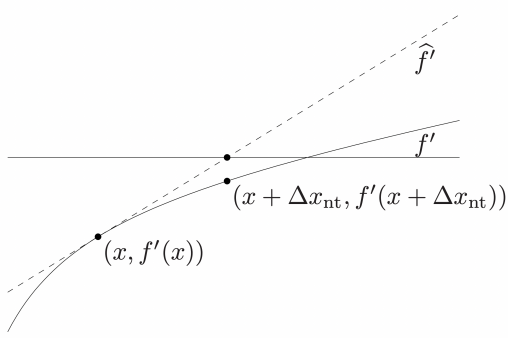

Figure: Illustration of the Newton step in the case of $n = 1$, *i.e.*, $f : \mathbb{R} \to \mathbb{R}$; figure from BV

Given our current approximation x of the solution, we form a first-order Taylor approximation of f' at x. The solution $x^\star$ of the minimization problem is characterized by $f'(x^\star) = 0$, *i.e.*, it is the zero-crossing of the derivative f', which is monotonically increasing since f is convex. The zero-crossing of this affine approximation is then $x + \Delta x_{\mathrm{nt}}$.

Affine invariance of the Newton step

Suppose $T \in \mathbb{R}^{n \times n}$ is nonsingular, define $\bar{f}(y) = f(Ty)$, and let $x = Ty$. Then we have

$$\nabla \bar{f}(y) = T^\top \nabla f(x), \qquad \nabla^2 \bar{f}(y) = T^\top \nabla^2 f(x) T$$

The Newton step for $\bar{f}$ at y is therefore

$$\begin{aligned} \Delta y_{\mathsf{nt}} &= -\big(T^\top \nabla^2 f(x) T\big)^{-1} \big(T^\top \nabla f(x)\big) \\ &= -T^{-1} \big(\nabla^2 f(x)\big)^{-1} \nabla f(x) \\ &= T^{-1} \Delta x_{\mathsf{nt}} \end{aligned}$$

where Δx_{nt} is the Newton step for f at x. Hence the Newton steps of f and $\bar{f}$ are related by the same linear transformation, and

$$x + \Delta x_{\mathsf{nt}} = T(y + \Delta y_{\mathsf{nt}}).$$

- The Newton step is is independent of linear (or affine) changes of coordinates.

Newton decrement

The Newton decrement at x is defined as

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

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

$$f(x) - \inf_y \hat{f}(y) = f(x) - \hat{f}(x + \delta x_{\mathsf{nt}}) = \frac{1}{2}\lambda(x)^2$$

i.e., can think of $\lambda^2(x)/2$ as an estimate of the suboptimality gap $f(x) - f^\star$

- We can also express the Newton decrement as

$$\lambda(x) = \left(\Delta x_{\mathsf{nt}}^{\top} \nabla^2 f(x) \Delta x_{\mathsf{nt}}\right)^{1/2}$$

  which shows that λ is the length of the Newton step, in the quadratic norm defined by the Hessian, *i.e.*, the norm

$$\|u\|_{\nabla^2 f(x)} = \left(u^{\top} \nabla^2 f(x) u\right)^{1/2}$$

- like the Newton step, the Newton decrement is affine invariant; *i.e.*, the Newton decrement of $\bar{f}(y) = f(Ty)$ at y, where T is nonsingular, is the same as the Newton decrement of f at $x = Ty$.

Newton's method

Newton's method

given a starting point $x \in \operatorname{dom} f$, tolerance $\epsilon > 0$

---

repeat

1. Compute the Newton step and decrement.

$$\Delta x_{\mathsf{nt}} := -\big(\nabla^2 f(x)\big)^{-1} \nabla f(x); \qquad \lambda^2 := \nabla f(x)^\top \big(\nabla^2 f(x)\big)^{-1} \nabla f(x).$$

2. Stopping criterion. **quit** if $\lambda^2/2 \leq \epsilon$.
3. Line search. Choose step size t by backtracking line search.
4. Update. $x := x + t\Delta x_{\mathsf{nt}}$.

Phase 1: damped (or guarded) Newton method

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

- when x is not close to $x^\star$ apply step size t using backtracking line search

Phase 2: undamped (or pure) Newton method

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

- when x is close enough to $x^\star$ apply unit step size $t = 1$
- this is where we achieve quadratic convergence

Backtracking line search

choose the step length to (approximately) minimize f along the ray $\{x + t\Delta x \mid t \geq 0\}$

Backtracking line search

given a descent direction Δx for f at $x \in \operatorname{dom} f$, $\alpha \in (0, 0.5)$, $\beta \in (0, 1)$.

$t := 1$.
while $f(x + t\Delta x) > f(x) + \alpha t \nabla f(x)^\top \Delta x, \qquad t := \beta t.$

Convergence analysis

Assume

- f is twice continuously differentiable
- f is strongly convex with constant m, *i.e.*, $\nabla^2 f(x) \succeq mI$
- ∇f is Lipschitz continuous with parameter M *i.e.*, $\nabla^2 f(x) \preceq MI$

Also assume the Hessian of f is Lipschitz continuous with constant L, *i.e.*,

$$\|\nabla^2 f(x) - \nabla^2 f(y)\|_2 \leq L\|x - y\|_2 \qquad \text{for all } x, y$$

- L can be interpreted as a bound on the third derivative of f; zero for a quadratic function
- L measures how well f can be approximated by a quadratic model; we can expect L to play a critical role in the performance of Newton's method
- Newton's method will work well for a function whose quadratic model varies slowly (*i.e.*, small L)

Theorem

Newton's method with backtracking line search satisfies the following two-stage convergence bounds

$$f(x^{(k)}) - f^\star \leq \begin{cases} (f(x^{(0)}) - f^\star) - \gamma k & k \leq k_0 \\ \dfrac{2m^3}{L^2}\left(\dfrac{1}{2}\right)^{2^{k-k_0+1}} & k > k_0 \end{cases}$$

where $\gamma = \alpha\beta^2\eta^2 m/M^2$, $\eta = \min\{1, 3(1-2\alpha)\}\frac{m^2}{L}$, *and* k_0 *is the number of steps until* $\|\nabla f(x^{(k_0+1)})\| < \eta$

Idea and outline of convergence proof

Convergence analysis reveals that there are η and γ with $0 < \eta \leq m^2/L$ and $\gamma > 0$ such that the following hold

1. (damped Newton phase) if $\|\nabla f(x^{(k)})\|_2 \geq \eta$, then
$$f(x^{(k+1)}) - f(x^{(k)}) \leq -\gamma$$
2. (pure Newton phase) if $\|\nabla f(x^{(k)})\|_2 < \eta$, then the backtracking line search selects $t^{(k)} = 1$ and
$$\frac{L}{2m^2}\|\nabla f(x^{(k+1)})\|_2 \leq \left(\frac{L}{2m}\|\nabla f(x^{(k)})\|_2\right)^2$$

- by recursively applying the second inquality for $l \geq k$ we can obtain
$$\frac{L}{2m^2}\|\nabla f(x^{(l)})\|_2 \leq \left(\frac{L}{2m}\|\nabla f(x^{(k)})\|_2\right)^{2^{l-k}} \leq \left(\frac{1}{2}\right)^{2^{l-k}}$$
and hence
$$f(x^{(l)}) - f^\star \leq \frac{1}{2m}\|\nabla f(x^{(l)})\|_2^2 \leq \frac{2m^3}{L^2}\left(\frac{1}{2}\right)^{2^{l-k+1}}$$

proof sketchy (see BV for full proof)

1. derive an upper bound on the number of iterations in the dampted Newton phase: f decreases by at least γ $(= \alpha\beta\eta^2 m/M^2)$ at each iteration, the number of dampted Newton steps cannot exceed

$$\frac{f(x^{(0)}) - f^\star}{\gamma}$$

2. bound the number of iterations in the quadratically convergent phase: the subobtimality gap implies that we must have $f(x) - f^\star \leq \epsilon$ after no more than

$$\log_2 \log_2(\epsilon_0/\epsilon)$$

iterations in the quadratically convergent phase, where $\epsilon_0 = 2m^3/L^2$

- compare this to linear convergence of gradient descent (under strong convexity)

overall, then, the number of iterations until $f(x) - f^\star \leq \epsilon$ is bounded above by

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

- $\log_2 \log_2(\epsilon_0/\epsilon)$ grows extremely slowly with required accuracy ϵ, and can be considered a constant for practical purposes, say five or six (Six iterations of the quadratically convergent stage gives an accuracy of about $\epsilon \approx 5 \cdot 10^{-20}\epsilon_0$)

Comparison to first-order methods

At a high-level:

- Memory: each iteration of Newton's method requires $\mathcal{O}(n^2)$ storage ($n \times n$ Hessian); each gradient iteration requires $\mathcal{O}(n)$ storage (n-dimensional gradient)
- Computation: each Newton iteartion requires $\mathcal{O}(n^3)$ flops (solving a dense $n \times n$ linear system); each gradient iteration requires $\mathcal{O}(n)$ flops (scaling/adding n-dimensional vectors)
- Backtracking: backtracking line search has roughly the same cost, both use $\mathcal{O}(n)$ flops per inner backtracking step
- Conditioning: Newton's method is not affected by a problem's conditioning, but gradient descent can seriously degrade

Summary

Newton's method has several advantages over gradient methods:

- Convergence of Newton's method is rapid in general, and quadratic near $x^\star$.
- Newton's method is affine invariant.
- Newton's method scales well with problem size.
- The good performance of Newton's method is not dependent on the choice of algorithm parameters.

The main disadvantage of Newton's method is the cost of forming and storing the Hessian, and the cost of computing the Newton step, which requires solving a set of linear equations.

Any questions?