Newton's method

Lecture 10

October 19, 2021

Beck, sections 5.1-5.2

Recall: gradient descent

$$\min_{\underline{x} \in \mathbb{R}^n} f(\underline{x})$$

Algorithm: $\underline{x}_0, \underline{x}_1, \dots \underline{x}_n, \dots$

hope: $\underline{x}_n \xrightarrow{n \uparrow \infty}$ stationary point of f.

$$\underline{x}_n = \underline{x}_{n-1} + t_n \underline{d}_n$$

descent direction

gradient descent: choose $\underline{d}_n = -\nabla f(\underline{x}_{n-1})$

t_n: stepsize

Gradient descent is a "first-order" method

- requires first derivatives
- convergence is "first-order"

if $\underline{x}_*$ is a stationary point of f, then

$$\|\underline{x}_{k+1} - \underline{x}^*\| \leq C \cdot \|\underline{x}_k - \underline{x}^*\|$$

convergence if $C < 1$

Newton's method: a second-order method

$$\min_{\underline{x} \in \mathbb{R}^n} f(\underline{x})$$

$$\underline{x}_{k+1} = \underline{x}_k - \underbrace{\left(\nabla^2 f(\underline{x}_k)\right)^{-1}}_{\text{Hessian}} \underbrace{\nabla f(\underline{x}_k)}_{\text{gradient}} \qquad (\text{"Pure Newton"})$$

Interpretations of Newton's method?

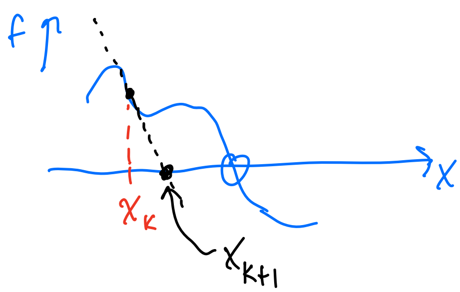
Note: Newton's method (in optimization)

$$\not\Updownarrow$$

Newton-Raphson method (for root-finding)

Recall: Newton-Raphson method

goal: compute x s.t. $f(x)=0$ $\quad (f: \mathbb{R} \to \mathbb{R})$



Newton-Raphson: is an iterative approach: $x_0, x_1, \ldots$

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

computes the exact root of a linear approx. to f at x_k

Multivariate version: $g: \mathbb{R}^n \to \mathbb{R}^n$

goal: compute $\underline{x}$ s.t. $g(\underline{x}) = \underline{0}$

Idea is the same: compute a local linear approx, compute exact root of this.

at $\underline{x} = \underline{x}_k$: $\underline{g}(\underline{x}) \simeq \underline{g}(\underline{x}_k) + \underbrace{\underline{\underline{J}}_g(\underline{x}_k)}_{\substack{\text{Jacobian,} \\ n \times n \\ \text{matrix}}} (\underline{x} - \underline{x}_k) = \underline{0}$

Taylor

Newton-Raphson

$$(\underline{\underline{J}}_g)_{ij} = \frac{\partial g_i}{\partial x_j}$$

$$\Rightarrow \underset{\underset{\underline{x}_{k+1}}{\shortparallel}}{\underline{x}} = \underline{x}_k - [\underline{\underline{J}}_g(\underline{x}_k)]^{-1} \underline{g}(\underline{x}_k)$$

Newton's method for optimization (Interpretation 1)

$$\min_{\underline{x} \in \mathbb{R}^n} f(\underline{x}) \quad , \quad f : \mathbb{R}^n \to \mathbb{R}$$

given $\underline{x}_k$, assign to $\underline{x}_{k+1}$ the vector corresponding to a Newton-Raphson update for ∇f.

(i.e., try to force $\nabla f = 0$)

"Solve" $\nabla f(\underline{x}) = \underline{0}$. $\quad \nabla f \in \mathbb{R}^n$

Let's call $g = \nabla f$, $g: \mathbb{R}^n \to \mathbb{R}^n$

Newton-Raphson on $g = \nabla f$: $\underline{x}_{k+1} = \underline{x}_k - [\underline{\underline{J}}_g(\underline{x}_k)]^{-1} \underline{g}(\underline{x}_k)$

Note: $g(\underline{x}_k) = \nabla f(\underline{x}_k)$

$\underline{\underline{J}}_g(\underline{x}_k) = \nabla^2 f(\underline{x}_k)$

$$\Longrightarrow \underline{x}_{k+1} = \underline{x}_k - [\nabla^2 f(\underline{x}_k)]^{-1} \nabla f(\underline{x}_k)$$

("Pure" Newton)

Newton's method for optimization on f

$\Updownarrow$

Newton-Raphson for root finding on ∇f.

Newton's method for optimization (Interpretation 2)

$$\min_{\underline{x} \in \mathbb{R}^n} f(\underline{x})$$

Main idea: compute $\underline{x}_{k+1}$ via exact minimization of a quadratic function constructed at $\underline{x}_k$.

Around $\underline{x} = \underline{x}_k$: $f(\underline{x}) \approx f(\underline{x}_k) + \nabla f(\underline{x}_k)^T (\underline{x} - \underline{x}_k)$

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

Taylor

Define $Q(\underline{x}) := f(\underline{x}_k) + \nabla f(\underline{x}_k)^T (\underline{x} - \underline{x}_k)$

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

Newton's method: $\underline{x}_{k+1} = \underset{\underline{x} \in \mathbb{R}^n}{\operatorname{argmin}} \; Q(\underline{x})$

This problem has a unique solution iff $\nabla^2 Q \succ \underline{0}$

$$\nabla^2 Q = \nabla^2 f(\underline{x}_k)$$

Assume: $\nabla^2 Q = \nabla^2 f(\underline{x}_k) \succ \underline{0}$.

Then: $\nabla Q(\underline{x}) = \underline{0}$ has a unique solution, which is the global minimum of Q.

$$\|$$

$$\nabla f(\underline{x}_k) + \nabla^2 f(\underline{x}_k)(\underline{x} - \underline{x}_k)$$

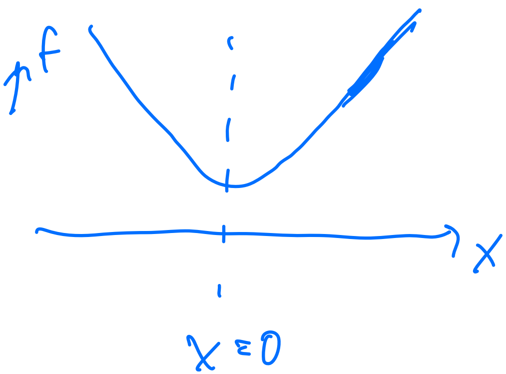
$$\Rightarrow \underline{x} = \underline{x}_k - \left(\nabla^2 f(\underline{x}_k)\right)^{-1} \nabla f(\underline{x}_k)$$

("Pure" Newton)

Newton's method doesn't always converge

Example (Example 5.1)

Consider Newton's method on $f(x) = \sqrt{1+x^2}$ for $x \in \mathbb{R}$.
For which values of initial guess x_0 does Newton's method converge?



$$\min_{x \in \mathbb{R}} f(x)$$

(x = 0 is the minimum location)

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

$$= x_k - \frac{f'(x_k)}{f''(x_k)}$$

$$f'(x) = \frac{x}{\sqrt{1+x^2}} \qquad f''(x) = \frac{1}{\sqrt{1+x^2}} + \frac{x(-\frac{1}{2})(2x)}{(1+x^2)^{3/2}}$$

$$= \frac{1}{\sqrt{1+x^2}} - \frac{x^2}{(1+x^2)^{3/2}} = \frac{1}{(1+x^2)^{3/2}}$$

$$\Rightarrow x_{k+1} = x_k - \frac{f'(x_k)}{f''(x_k)} = x_k - \frac{x_k}{\sqrt{1+x_k^2}}(1+x_k^2)^{3/2}$$

$$= -x_k^3$$

When does this work? When does $\lim_{k \to \infty} x_k = 0$?

This happens iff $\exists\, k$ s.t. $|x_k| < 1$.

iff $|x_0| < 1$.

This works only if x_0 is chosen "well-enough".

Local quadratic (!) convergence

The power of Newton's method: quadratic convergence. $\min_{\underline{x} \in S} f(\underline{x})$

Technical assumptions: (i) $\nabla^2 f(\underline{x}) \succeq m \underline{\underline{I}}$, for some $m > 0$.

$\left(\nabla^2 f(\underline{x}) - m \underline{\underline{I}} \succeq \underline{\underline{0}} \right)$

(ii) $\| \nabla^2 f(\underline{x}) - \nabla^2 f(\underline{y}) \| \leq M \| \underline{x} - \underline{y} \|$

("Lipschitz condition")

(iii) Assume $S = \mathbb{R}^n$

Then: let $\underline{x}_*$ be the global minimizer of f. We have:

$$\| \underline{x}_{k+1} - \underline{x}_* \| \leq \frac{M}{2m} \| \underline{x}_k - \underline{x}_* \|^2$$

Why is $\underbrace{\text{quadratic}}_{\text{vs. e.g., linear}}$ convergence nice?

Suppose $\underline{x}_0$ satisfies $\|\underline{x}_* - \underline{x}_0\| = R$

Under the assumptions above:

$$\|\underline{x}_1 - \underline{x}_*\| \leq \frac{M}{2m}\|\underline{x}_0 - \underline{x}_*\|^2 = \left(\frac{M}{2m}\right) R^2$$

$$\|\underline{x}_2 - \underline{x}_*\| \leq \frac{M}{2m}\left[\left(\frac{M}{2m}\right) R^2\right]^2 = \left(\frac{M}{2m}\right)^3 R^4$$

$$\|\underline{x}_3 - \underline{x}_*\| \leq \left(\frac{M}{2m}\right)^7 R^8$$

$$\|\underline{x}_4 - \underline{x}_*\| \leq \left(\frac{M}{2m}\right)^{15} R^{16}$$

$$\vdots$$

$$\|\underline{x}_k - \underline{x}_*\| \leq \left(\frac{M}{2m}\right)^{2^k - 1} R^{2^k} = \frac{2m}{M}\left[\frac{RM}{2m}\right]^{2^k}$$

Suppose, e.g, that $\frac{RM}{2m} \leq \frac{1}{2} \rightsquigarrow \underbrace{R \leq \frac{m}{M}}$

$\underline{x}_0$ is "close enough" to $\underline{x}_*$.

$$\Rightarrow \frac{M}{2m}\|\underline{x}_1 - \underline{x}_*\| \leq 2^{-2} \qquad (k=1) \qquad (\text{0 digits of acc.})$$

$$\frac{M}{2m}\|\underline{x}_2 - \underline{x}_*\| \leq 2^{-4} \qquad (k=2) \qquad (\sim 1 \text{ digit})$$

$$\frac{M}{2m}\|\underline{x}_3 - \underline{x}_*\| \leq 2^{-8} \qquad (\sim 2.5 \text{ digits})$$

$$\frac{M}{2m}\|\underline{x}_4 - \underline{x}_*\| \leq 2^{-16}$$

$$\frac{M}{2m}\|\underline{x}_5 - \underline{x}_*\| \leq 2^{-32} \qquad (\sim 9 \text{ digits})$$

I.e., Newton's method converges very quickly if $\|\underline{x}_* - \underline{x}_0\|$ is "small enough".

Alternative: gradient descent (assuming f is very "nice")

achieves: $\underbrace{\|\underline{x}_k - \underline{x}_0\| \lesssim 2^{-k}}_{\text{"linear convergence"}}$

(ie. $\|\underline{x}_{k+1} - \underline{x}_*\| \leq C\|\underline{x}_k - \underline{x}_*\|^1$)

Newton's method works awfully poorly when $\|\underline{x}_0 - \underline{x}_*\|$ is "large".

Newton's method in general does not guarantee convergence, nor does it guarantee that $f(\underline{x}_{k+1}) \leq f(\underline{x}_k)$.

Variant: Damped Newton's method

Pure Newton: $\underline{x}_{k+1} = \underline{x}_k - (\nabla^2 f(\underline{x}_k))^{-1} \nabla f(\underline{x}_k)$

Idea: employ backtracking linesearch in direction

$$\underline{d}_k = -(\nabla^2 f(\underline{x}_k))^{-1} \nabla f(\underline{x}_k)$$

Backtracking: specify $\alpha \in (0,1)$, $\beta \in (0,1)$

At iteration k: compute $\underline{d}_k = -(\nabla^2 f(\underline{x}_k))^{-1} \nabla f(\underline{x}_k)$

Set $t_k = 1$ (Stepsize)

Compare $\underbrace{f(\underline{x}_k) - f(\underline{x}_k + t_k \underline{d}_k)}_{\text{actual improvement}}$ vs. $\underbrace{-t_k \underline{d}_k^T \nabla f(\underline{x}_k)}_{\text{"expected" improvement in } f.}$

I.e., if $f(\underline{x}_k) - f(\underline{x}_k + t_k \underline{d}_k) < \left[-t_k \underline{d}_k^T \nabla f(\underline{x}_k)\right]\alpha$

then: $t_k \leftarrow t_k \beta$, go back to "compare" step.

else $\underline{x}_{k+1} = \underline{x}_k + t_k \underline{d}_k$

This is "damped" Newton's method

Problem: require $\nabla^2 f(\underline{x}_k) \succ \underline{0}$.

Variant: Hybrid Newton's method

Idea: run gradient descent if $\nabla^2 f(\underline{x}_k) \not\succ \underline{0}$.

Else, run Newton's method.

All this w/ backtracking:

at iteration k: If $\nabla^2 f(\underline{x}_k) \succ \underline{\underline{0}}$

$$\underline{d}_k = -\left(\nabla^2 f(\underline{x}_k)\right)^{-1} \nabla f(\underline{x}_k) \qquad \text{(Newton)}$$

Else:

$$\underline{d}_k = -\nabla f(\underline{x}_k) \qquad \text{(GD)}$$

Choose t_k according to backtracking linesearch. (α, β)

Set $\underline{x}_{k+1} = \underline{x}_k + t_k \underline{d}_k$.

This is fine: $\nabla^2 f(\underline{x}_k)$ & $\underline{Q}$ can be determined by computing eigenvalues of $\nabla^2 f$.

But, computing eigenvalues is slow, and there are better alternatives.