Newton's Method

Javier Peña
Convex Optimization 10-725/36-725

Last time: dual correspondences

Given a function $f : \mathbb{R}^n \to \mathbb{R}$, we define its conjugate $f^* : \mathbb{R}^n \to \mathbb{R}$,

$$f^*(y) = \max_x \; \big(y^T x - f(x)\big)$$

Properties and examples:

- Conjugate f^* is always convex (regardless of convexity of f)
- When f is a quadratic in $Q \succ 0$, f^* is a quadratic in Q^{-1}
- When f is a norm, f^* is the indicator of the dual norm unit ball
- When f is closed and convex, $x \in \partial f^*(y) \iff y \in \partial f(x)$

Fenchel duality

$$\begin{aligned} \text{Primal}: \quad & \min_x \; f(x) + g(x) \\ \text{Dual}: \quad & \max_u \; -f^*(u) - g^*(-u) \end{aligned}$$

Newton's method

Consider the unconstrained, smooth convex optimization problem

$$\min_x \ f(x)$$

where f is convex, twice differentable, and $\mathrm{dom}(f) = \mathbb{R}^n$.

Newton's method: choose initial $x^{(0)} \in \mathbb{R}^n$, and

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

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

Compare to gradient descent: choose initial $x^{(0)} \in \mathbb{R}^n$, and

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

Newton's method interpretation

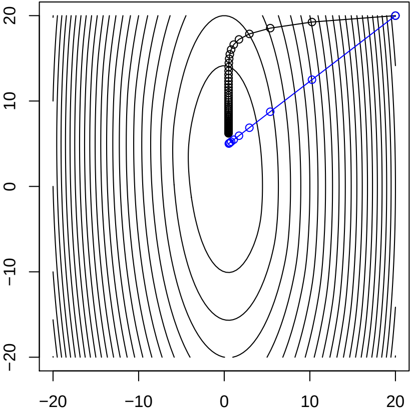
The Newton step $x^+ = x - (\nabla^2 f(x))^{-1} \nabla f(x)$ can be obtained by minimizing over y the *quadratic approximation*

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

By contrast the gradient descent step $x^+ = x - t\nabla f(x)$ can be obtained by minimizing over y the quadratic approximation

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

For $f(x) = (10x_1^2 + x_2^2)/2 + 5\log(1 + e^{-x_1 - x_2})$, compare gradient descent (black) to Newton's method (blue), where both take steps of roughly same length



(For our example we needed to consider a nonquadratic ... why?)

Outline

Today:

- Interpretations and properties
- Convergence analysis
- Backtracking line search
- Equality-constrained Newton
- Quasi-Newton methods

Newton's method for root finding

Assume $F : \mathbb{R}^n \to \mathbb{R}^n$ is a differentiable vector-valued function and consider the system of equations

$$F(x) = 0.$$

Newton's method for root-finding: choose initial $x^{(0)} \in \mathbb{R}^n$, and

$$x^{(k)} = x^{(k-1)} - F'(x^{(k-1)})^{-1} F(x^{(k-1)}), \quad k = 1, 2, 3, \ldots$$

Here $F'(x^{(k-1)})$ is the Jacobian matrix of F at $x^{(k-1)}$.

The Newton step $x^+ = x - F'(x)^{-1} F(x)$ can be obtained by solving over y the linear approximation

$$F(y) \approx F(x) + F'(x)(y - x) = 0.$$

Example: let $F : \mathbb{R} \to \mathbb{R}$ be defined as

$$F(x) = x^2 - 2.$$

Newton's method starting from $x^{(0)} = 1$:

k	0	1	2	3	4
$x^{(k)}$	1	1.5	1.4166	1.414216	1.4142135
$x^{(k)} - \sqrt{2}$	-0.4142	0.0858	0.0024	2.1×10^{-6}	1.6×10^{-12}

Newton's method for the optimization problem

$$\min_x \ f(x)$$

is the same as Newton's method for finding a root of

$$\nabla f(x) = 0.$$

History: The work of Newton (1685) and Raphson (1690) originally focused on finding roots of polynomials. Simpson (1740) applied this idea to general nonlinear equations and minimization.

Affine invariance of Newton's method

Important property Newton's method: affine invariance.

Assume $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable and $A \in \mathbb{R}^{n \times n}$ is nonsingular. Let $g(y) := f(Ay)$.

Newton step for g starting from y is

$$y^+ = y - \left(\nabla^2 g(y)\right)^{-1} \nabla g(y).$$

It turns out that the Newton step for f starting from $x = Ay$ is $x^+ = Ay^+$.

Therefore progress is independent of problem scaling. By contrast, this is not true of gradient descent.

Local convergence

Theorem: Assume $F : \mathbb{R}^n \to \mathbb{R}^n$ is continuously differentiable and $x^\star \in \mathbb{R}^n$ is a root of F, that is, $F(x^\star) = 0$ such that $F'(x^\star)$ is non-singular. Then

(a) There exists $\delta > 0$ such that if $\|x^{(0)} - x^\star\| < \delta$ then Newton's method is well defined and

$$\lim_{k\to\infty} \frac{\|x^{(k+1)} - x^\star\|}{\|x^{(k)} - x^\star\|} = 0.$$

(b) If F' is Lipschitz continuous in a neighborhood of $x^\star$ then there exists $K > 0$ such that

$$\|x^{(k+1)} - x^\star\| \le K\|x^{(k)} - x^\star\|^2.$$

Newton decrement

Consider again Newton's method for the optimization problem

$$\min_x \quad f(x).$$

At a point x, define the Newton decrement as

$$\lambda(x) = \Big(\nabla f(x)^T \big(\nabla^2 f(x)\big)^{-1} \nabla f(x)\Big)^{1/2}.$$

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

$$\begin{aligned} f(x) - \min_y \Big(f(x) + \nabla f(x)^T (y-x) + \frac{1}{2}(y-x)^T \nabla^2 f(x)(y-x)\Big) \\ = \frac{1}{2} \nabla f(x)^T \big(\nabla^2 f(x)\big)^{-1} \nabla f(x) = \frac{1}{2}\lambda(x)^2. \end{aligned}$$

Therefore can think of $\lambda^2(x)/2$ as an approximate bound on the suboptimality gap $f(x) - f^\star$

Another interpretation of Newton decrement: if Newton direction is $v = -(\nabla^2 f(x))^{-1} \nabla f(x)$, then

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

i.e., $\lambda(x)$ is the length of the Newton step in the norm defined by the Hessian $\nabla^2 f(x)$

Note that the Newton decrement, like the Newton steps, are affine invariant; i.e., if we defined $g(y) = f(Ay)$ for nonsingular A, then $\lambda_g(y)$ would match $\lambda_f(x)$ at $x = Ay$

Backtracking line search

We have seen pure Newton's method, which need not converge. In practice, we instead use damped Newton's method (i.e., Newton's method), which repeats

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

Note that the pure method uses $t = 1$

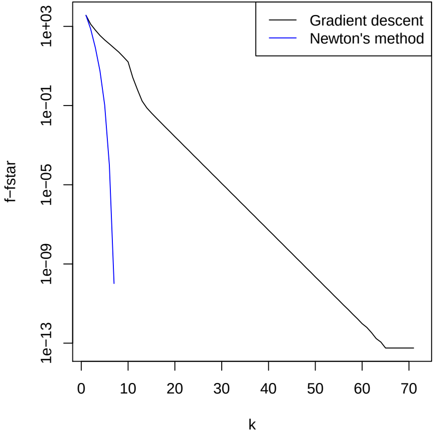
Step sizes here typically are chosen by backtracking search, with parameters $0 < \alpha \leq 1/2$, $0 < \beta < 1$. At each iteration, we start with $t = 1$ and while

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

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 has a different regime of convergence.

Convergence analysis

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

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

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

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

Here $\gamma = \alpha\beta^2\eta^2 m/L^2$, $\eta = \min\{1, 3(1-2\alpha)\}m^2/M$, and k_0 is the number of steps until $\|\nabla f(x^{(k_0+1)})\|_2 < \eta$

In more detail, convergence analysis reveals $\gamma > 0$, $0 < \eta \leq m^2/M$ such that convergence follows two stages

- Damped phase: $\|\nabla f(x^{(k)})\|_2 \geq \eta$, and

$$f(x^{(k+1)}) - f(x^{(k)}) \leq -\gamma$$

- Pure phase: $\|\nabla f(x^{(k)})\|_2 < \eta$, backtracking selects $t = 1$, and

$$\frac{M}{2m^2}\|\nabla f(x^{(k+1)})\|_2 \leq \Big(\frac{M}{2m^2}\|\nabla f(x^{(k)})\|_2\Big)^2$$

  Note that once we enter pure phase, we won't leave, because

$$\frac{2m^2}{M}\Big(\frac{M}{2m^2}\eta\Big)^2 < \eta$$

  when $\eta \leq m^2/M$

To reach $f(x^{(k)}) - f^\star \le \epsilon$, we need at most

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

iterations, where $\epsilon_0 = 2m^3/M^2$

- This is called quadratic convergence. Compare this to linear convergence (which, recall, is what gradient descent achieves under strong convexity)
- The above result is a local convergence rate, i.e., we are only guaranteed quadratic convergence after some number of steps k_0, where $k_0 \le \frac{f(x^{(0)})-f^\star}{\gamma}$
- Somewhat bothersome may be the fact that the above bound depends on L, m, M, and yet the algorithm itself does not

Self-concordance

A scale-free analysis is possible for self-concordant functions: on an open segment of $\mathbb{R}$, a convex function f is called self-concordant if

$$|f'''(t)| \leq 2f''(t)^{3/2} \quad \text{for all } t.$$

On an open convex domain of $\mathbb{R}^n$ a function is self-concordant if its restriction to every line in its domain is so.

Examples of self-concordant functions

- $f : \mathbb{R}^n_{++} \to \mathbb{R}$ defined by

$$f(x) = -\sum_{i=1}^{n} \log(x_j)$$

- $f : \mathbb{S}^n_{++} \to \mathbb{R}$ defined by

$$f(X) = -\log(\det(X))$$

Convergence analysis for self-concordant functions

Property of self-concordance: if g is self-concordant and A, b are of appropriate dimensions, then

$$f(x) := g(Ax - b)$$

is self-concordant.

Theorem (Nesterov and Nemirovskii): Newton's method with backtracking line search requires at most

$$C(\alpha, \beta)\big(f(x^{(0)}) - f^\star\big) + \log\log(1/\epsilon)$$

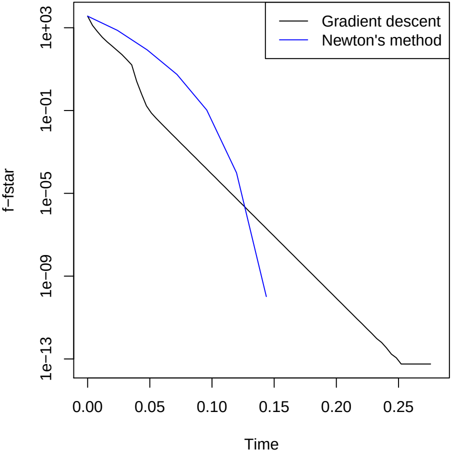
iterations to reach $f(x^{(k)}) - f^\star \leq \epsilon$, where $C(\alpha, \beta)$ is a constant that only depends on α, β

Comparison to first-order methods

At a high-level:

- 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
- Conditioning: Newton's method is not affected by a problem's conditioning, but gradient descent can seriously degrade
- Fragility: Newton's method may be empirically more sensitive to bugs/numerical errors, gradient descent is more robust

Back to logistic regression example: now x-axis is parametrized in terms of time taken per iteration



Each gradient descent step is $O(p)$, but each Newton step is $O(p^3)$

Sparse, structured problems

When the inner linear systems (in Hessian) can be solved efficiently and reliably, Newton's method can strive

E.g., if $\nabla^2 f(x)$ is sparse and structured for all x, say banded, then both memory and computation are $O(n)$ with Newton iterations

What functions admit a structured Hessian? Two examples:

- If $g(\beta) = f(X\beta)$, then $\nabla^2 g(\beta) = X^T \nabla^2 f(X\beta) X$. Hence if X is a structured predictor matrix and $\nabla^2 f$ is diagonal, then $\nabla^2 g$ is structured
- If we seek to minimize $f(\beta) + g(D\beta)$, where $\nabla^2 f$ is diagonal, g is not smooth, and D is a structured penalty matrix, then the Lagrange dual function is $-f^*(-D^T u) - g^*(-u)$. Often $-D\nabla^2 f^*(-D^T u)D^T$ can be structured

Equality-constrained Newton's method

Consider now a problem with equality constraints, as in

$$\min_x \; f(x) \;\; \text{subject to} \;\; Ax = b$$

Several options:

- Eliminating equality constraints (reduced-space approach): write $x = Fy + x_0$, where F spans null space of A, and $Ax_0 = b$. Solve in terms of y.
- Equality-constrained Newton: in many cases this is the most straightforward option.
- Deriving the dual: the Fenchel dual is $-f^*(-A^T v) - b^T v$ and strong duality holds. More on this later.

Equality-constrained Newton's method

Start with $x^{(0)} \in \mathbb{R}^n$. Then we repeat the update

$$x^+ = x + tv, \quad \text{where}$$
$$v = \underset{A(x+z)=b}{\operatorname{argmin}} \left(f(x) + \nabla f(x)^T z + \frac{1}{2} z^T \nabla^2 f(x) z \right)$$

From the KKT conditions it follows that for some w, v satisfies

$$\begin{bmatrix} \nabla^2 f(x) & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} v \\ w \end{bmatrix} = - \begin{bmatrix} \nabla f(x) \\ Ax - b \end{bmatrix}.$$

The latter is precisely the root-finding Newton step for the KKT conditions of the original equality-constrained problem, namely

$$\begin{bmatrix} \nabla f(x) + A^T y \\ Ax - b \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Quasi-Newton methods

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
- Very wide variety of quasi-Newton methods; common theme is to "propogate" computation of H across iterations.
- More on this class of methods later.

References and further reading

- S. Boyd and L. Vandenberghe (2004), "Convex optimization", Chapters 9 and 10
- Güler (2010), "Foundations of Optimization", Chapter 14.
- Y. Nesterov (1998), "Introductory lectures on convex optimization: a basic course", Chapter 2
- Y. Nesterov and A. Nemirovskii (1994), "Interior-point polynomial methods in convex programming", Chapter 2
- J. Nocedal and S. Wright (2006), "Numerical optimization", Chapters 6 and 7
- L. Vandenberghe, Lecture notes for EE 236C, UCLA, Spring 2011-2012