

Last time: dual correspondences

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

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

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, $x \in \partial f^*(y) \iff y \in \partial f(x)$

Relationship to duality (also called Fenchel duality):

Primal :	$\min_x f(x) + g(x)$
Dual :	$\max_u -f^*(u) - g^*(-u)$

2

Key properties

1. Conjugate f^* is always convex in y
2. When f is closed and convex, $x \in \partial f^*(y) \iff y \in \partial f(x)$
 - if x is in the subdifferential of the conjugate for a given value, then the value is in the subdiff of the primal
3. $\min_x f(x) + g(x) \iff \max_u -f^*(u) - g^*(-u)$

Newton's method

- this seems unrelated to duality, but the relationship will reveal itself!

Given smooth, unconstrained convex optimization

$$\min_x f(x)$$

Assuming f is convex, twice differentiable, and $\text{dom}(f) = \mathbb{R}^n$.

Gradient descent: $x^{(k)} = x^{(k-1)} - t_k \cdot \nabla f(x^{(k-1)})$

- move in the direction of negative gradient

Newton's method: $x^{(k)} = x^{(k-1)} - \nabla^2 f(x^{(k-1)})^{-1} \cdot \nabla f(x^{(k-1)})$

- this is a different direction entirely! What is the motivation for this?
- the inverse hessian is itself a matrix, so this is a linear operation on the gradient

A. Motivation

Gradient descent can be interpreted as minimizing the quadratic approximation

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

where $\frac{1}{2t}$ does not say anything about the geometry of the real problem.

In contrast, Newton's method minimizes a better quadratic approximation

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

Minimizing over y yields the update step.

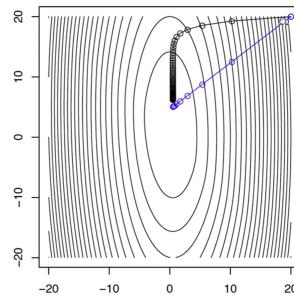
$$\begin{aligned} Q(y) &= f(x) + \nabla f(x)^T(y - x) + \frac{1}{2}(y - x)^T \nabla^2 f(x)(y - x) \\ \nabla Q(y) &= \nabla f(x) + \nabla^2 f(x)(y - x) . \\ y &= x - (\nabla^2 f(x))^{-1}(\nabla f(x)) \end{aligned}$$

For a quadratic criterion function, Newton's method will yield the solution in one step - since the gradient would be linear in the optimization variable - so you would get the exact solution (?)

- because the quadratic approximation IS the real criterion function too!
-

Consider minimizing $f(x) = (10x_1^2 + x_2^2)/2 + 5 \log(1 + e^{-x_1 - x_2})$
(this must be a nonquadratic ... why?)

We compare gradient descent (black) to Newton's method (blue), where both take steps of roughly same length

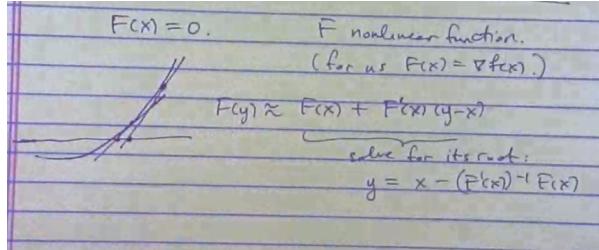


- Newton's method is intuitively better because it uses the curvature inherent to your problem, and not the assumption we make with GD i.e. that the curvature is some constant $\frac{1}{2t}$

Today

- Interpretation
- Backtracking line searing
- Convergence
- Equality constrained Newton
- Quasi Newton

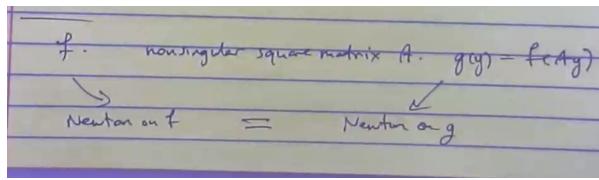
1. Interpretation



- Finding the roots of polynomial equations

Distinguishing properties of Newton's method

vs Gradient descent: Affine invariance



- doesn't just need to a linear transformation: can also be an affine transformation
- i.e. Newton update steps on $f(y) =$ Newton update steps on $g(y) = f(Ay + b)$

Note: when doing matrix derivatives e.g. $\partial(Ax)$: take everything before the variable of interest and transpose it!

Affine invariance of Newton's method

Important property Newton's method: **affine invariance**. Given f , nonsingular $A \in \mathbb{R}^{n \times n}$. Let $x = Ay$, and $g(y) = f(Ay)$. Newton steps on g are

$$\begin{aligned} y^+ &= y - (\nabla^2 g(y))^{-1} \nabla g(y) \\ &= y - (A^T \nabla^2 f(Ay) A)^{-1} A^T \nabla f(Ay) \\ &= y - A^{-1} (\nabla^2 f(Ay))^{-1} \nabla f(Ay) \end{aligned}$$

Hence

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

i.e.,

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

So progress is independent of problem scaling; recall that this is **not true** of gradient descent

8

- if you scaled your problem, Newton's method would be just as good
- This is not true for gradient descent! Where problem conditioning matters a lot of it to be useful

- else you would just improve the performance by scaling your function by some amount

Newton decrement

At a point x , the Newton decrement is $\lambda(x) = \left(\nabla f(x)^T (\nabla^2 f(x))^{-1} \nabla f(x) \right)^{1/2}$

Interpretation 1: This is analogous to the duality gap, to the extent that you think that the quadratic approximation is helpful: the difference between $f(x)$ and the min of the quadratic approximation to $f(x)$

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

This serves as an approximate upper bound on the suboptimality gap $f(x) - f_{opt}$

Newton decrement

At a point x , we define the **Newton decrement** as

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

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

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

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

9

Interpretation 2:

The norm induced by a matrix $\|x\|_A = x^T A x$

Seeing the Newton step as an update step,

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

where $v = \nabla^2 f(x)^{-1} \cdot \nabla f(x)$

Then, the length of the Newton update step in the norm defined by the hessian = square of the Newton decrement: $\|v\|_{\nabla^2 f(x)} = \lambda(x)^2$

- so in a sense, the Newton decrements tells you something about the size of each

update

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

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

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 $x = Ay$

2. Backtracking line search

Backtracking line search

So far what we've seen is called **pure Newton's method**. This need not converge. In practice, we use **damped Newton's method** (i.e., Newton's method), which repeats

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

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

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

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

11

- same interpretation/intuition as backtracking for gradient descent

3. Convergence analysis

- extremely fast! 'Quadratic convergence' - $\frac{1}{2}^{2^k}$!!
 - for quadratic problems, the optimum is found in one step, as discussed earlier

For a Hessian to be Lipschitz M means

$$\|\nabla^2 f(x) - \nabla^2 f(y)\|_{op} \leq M \|x - y\|_2$$

- operator norm = largest eigenvalue of the matrix
 - remember that the Hessian is a matrix!

Convergence analysis

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

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

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

13

- local quadratic convergence, since it only kicks in after a number of steps k_0
 - k_0 is the number of steps before the L2 norm of the Jacobian is less than η , i.e. it is smaller than some value
- The first phase is considered a 'damped phase', and the second is the 'pure' phase
- In the damped phase, we decrease the criterion by γk after every step k
 - backtracking picks some $t \neq 1$
- In the pure phase, backtracking picks $t = 1$
 - and you never get out of this phase! so there are two distinct phases

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

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

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

$$\frac{2m^2}{M} \left(\frac{M}{2m^2} \eta\right)^2 \leq \eta$$

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

Note the decrease in the pure phase - this is where 'quadratic convergence' comes from.

Note: if a function is m -strongly convex, that means

$$f(y) \geq f(x) + \nabla f(x)(y-x) + \frac{m}{2} \|y-x\|_2^2$$

Fact 1: this also means that $f(x) - f_{opt} \leq \frac{1}{2m} \|\nabla f(x)\|_2^2$

- the criterion difference depends on the gradient at point x !

$$\begin{aligned}
 f(y) &\geq f(x) + \nabla f(x)^T (y - x) + \frac{m}{2} \|y - x\|_2^2 \\
 \text{minimize both sides over } y. \\
 f^* &\geq \min_y (f(x) + \nabla f(x)^T (y - x) + \frac{m}{2} \|y - x\|_2^2) \\
 0 &= \nabla f(x) + m(y - x) \\
 y &= -\frac{1}{m} \nabla f(x) + x. \\
 \Rightarrow \text{plug in:} \\
 &= f(x) - \frac{1}{m} \|\nabla f(x)\|_2^2 + \frac{1}{2m} \|\nabla f(x)\|_2^2 \\
 &= f(x) - \frac{1}{2m} \|\nabla f(x)\|_2^2. \\
 \text{rearrange:} \\
 f(x) - f^* &\leq \frac{1}{2m} \|\nabla f(x)\|_2^2
 \end{aligned}$$

$$\underline{\text{Fact 2:}} \quad \frac{M}{2m^2} \|\nabla f(x^+)\|_2 \leq \left(\frac{M}{2m^2} \|\nabla f(x)\|_2 \right)^2$$

- The L2 norm of the gradient of the updated step is less than the squared L2 norm of the gradient at the current step, modulo a scaling factor

$$\begin{aligned}
 \text{Part of Fact 2:} \\
 \|\nabla f(x^+)\|_2 &= \|\nabla f(x+v)\|_2 \\
 &= \|\nabla f(x+v) - \nabla f(x) + \nabla f(x)v\|_2 \\
 &= \left\| \int_0^1 \nabla^2 f(x+tv)v dt - \nabla^2 f(x)v \right\|_2 \\
 &= \left\| \int_0^1 (\nabla^2 f(x+tv) - \nabla^2 f(x))v dt \right\|_2 \\
 &\leq \int_0^1 \|\nabla^2 f(x+tv) - \nabla^2 f(x)\|_2 dt \\
 &\leq \|\nabla^2 f(x+tv) - \nabla^2 f(x)\|_{op} \cdot \|v\|_2 \\
 &\leq M \cdot t \|v\|_2 \\
 &\leq M \|v\|_2 \int_0^1 t dt \\
 &= \frac{1}{2} M \|\nabla^2 f(x)\|_2 \|v\|_2 \\
 &\leq \frac{1}{2} M \|\nabla^2 f(x)\|_2 \|\nabla f(x)\|_2 \\
 &\leq \frac{M}{2m^2} \|\nabla f(x)\|_2
 \end{aligned}$$

$$\underline{\text{Fact 3:}} \quad f(x^{(k)}) - f_{opt} \leq \frac{2m^3}{M^2} \left(\frac{1}{2} \right)^{2^{k-k_0}}$$

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

ref Fact 3.

We have established $\frac{M}{2m^2} \|\nabla f(x^{(k)})\|_2 \leq \left(\frac{M}{2m} \|\nabla f(x^{(k)})\|_2\right)^2$

$a_k = \frac{M}{2m^2} \|\nabla f(x^{(k)})\|_2 \leq a_{k-1}$

$a_k \leq a_{k-1}$
 $\leq a_{k-2}$
 $\leq a_{k_0}$

$a_k \leq a_{k-1}$
 $\leq a_{k-2}$
 $\leq a_{k_0}$

$\frac{M}{2m^2} \|\nabla f(x^{(k)})\|_2 \leq \left(\frac{M}{2m} \|\nabla f(x^{(k)})\|_2\right)^{\frac{k-k_0}{2}}$

But at k_0 we know $\|\nabla f(x^{(k_0)})\|_2 < \eta$

so $\frac{M}{2m^2} \|\nabla f(x^{(k)})\|_2 \leq \left(\frac{1}{2}\right)^{\frac{k-k_0}{2}} \leq \frac{1}{2^k}$

finally $f(x^{(k)}) - f^* \leq \frac{1}{2^k} \|\nabla f(x^{(k)})\|_2^2$
 $\leq \frac{1}{2^k} \frac{m^2}{2m^2}$

Convergence analysis

Unraveling this result, what does it say? To get $f(x^{(k)}) - f^* \leq \epsilon$, we need at most

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

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 $k_0 \leq \frac{f(x^{(0)}) - f^*}{\gamma}$
- Somewhat bothersome may be the fact that the above bound depends on L, m, M , and yet the **algorithm itself does not ...**

15

Quadratic convergence because $k_0 = \log \log(\epsilon_0/\epsilon)$

- just need to make this many steps in the quadratic convergence phase
- this is a local rate - this is how fast it converges once you are close
- the global rate is just linear!
- but in practice, you typically enter the quadratic convergence very quickly, especially with a warm start
 - this is what interior point methods do - you never leave this phase

There is a contradiction here - if Newton's method is truly scale invariant, then why does these bounds depend on the Lipschitz/strong convexity properties of the objective function?

Self concordance

the canonical self concordant function is $-\log(x)$: gives equality for the bottom definition

$$|f'''(x)| \leq 2f''(x)^{3/2}$$

Self-concordance

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

$$|f'''(x)| \leq 2f''(x)^{3/2} \quad \text{for all } x$$

and on \mathbb{R}^n is called self-concordant if its projection onto every line segment is so

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

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

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

16

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

18