Gradient Descent

Thursday, February 4, 2021 10:43 PM

Solve $\min_x f(x)$ (no constraints, assume $f: \mathbb{R}^n \to \mathbb{R}$ not $f: \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$)

Assume $f \in \Gamma_0(\mathbb{R}^n)$ (i.e., proper, lsc, convex)
and ∇f is L-Lipschitz continuous aka strongly smooth

① (Failed) idea #1

$$X_{k+1} = \operatorname{argmin}_x \underbrace{f(x_k) + \langle \nabla f(x_k), x - x_k \rangle}_{g_k(x)} \quad 1^{st} \text{ order "surrogate"}$$

⚠ I often use x_k to mean the k^{th} iterate of the vector x, so $x_k \in \mathbb{R}^n$

... but sometimes write $x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$ so $x_i \in \mathbb{R}$

"abuse of notation"
... but it's ok because I almost have tenure. Sorry

This idea makes sense! applied mathematicians/engineers/physicists love linearizing. Linearization is only valid locally, so just update.

... but this rarely works. Why? Usually $\min_x g_k(x) = \boxed{-\infty}$!

* you can fix this if you add a compact constraint or "coercive" regularizer ... then it's called Frank-Wolfe aka conditional gradient
cf. Martin Jaggi '13

② Attempt #2

2nd order Taylor Series

$$X_{k+1} = \operatorname{argmin}_x \underbrace{f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \tfrac{1}{2} \langle x - x_k, \nabla^2 f(x_k)(x - x_k) \rangle}_{g_k(x)}$$

$g_k(x)$ surrogate is now a quadratic
(and if f is convex, so is g_k)

to minimize $g_k(x)$,
Fermat's rule:

$$0 = \nabla g_k(x)$$
$$= \nabla f(x_k) + \nabla^2 f(x_k)(x - x_k)$$

$$\Rightarrow \quad x = x_k - \nabla^2 f(x_k)^{-1} \nabla f(x_k) \quad \text{and} \quad x_{k+1} = x$$

This is Newton's Method

This doesn't look like Newton's method I learned in Calc + Numerical Analysis!

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

Ahh... but it is.

Newton (aka Newton-Raphson) for root-finding $F(x)=0$

For us, Newton for optimization solves $\min_x f(x)$, i.e., $\nabla f(x) = 0$

$F(x) = \nabla f(x)$ ← The connection

In optimization lingo, Newton's method is a 2nd order method meaning that it involves second derivatives

Rule of thumb

2nd order methods converge quickly (few iterations), but each iteration may be costly (i.e., inverting a matrix / solving system of lin. equations)

(only a general heuristic rule: sometimes 2nd order methods converge slowly. Sometimes you can invert $\nabla^2 f$ cheaply)

A 1st order method only uses $\nabla f(x)$ (not $\nabla^2 f$)

and usually converges more slowly than 2nd order methods

but each step is cheap.

Which to use? It depends!

- Structure matters (is $\nabla^2 f$ easy to invert? Is there ill-conditioning? Types of constraints. Repeated solves)
- Small/med. problem size, high accuracy ⇒ 2nd order (default for cvx)
- large problem, low accuracy ok ⇒ 1st order
- in between problems ⇒ unclear (try both?)

Other types:

3rd order: exist but not common. See recent Nesterov work.

0^{th} order: Only query $f(x)$ not $\nabla f(x)$. Slow.

Usually, if we can get $f(x)$, we can get $\nabla f(x)$ for little additional cost... more on this later. So 0^{th} order applicable only in special cases

coordinate descent: Doesn't fit into our classification scheme. Benefits depend heavily on structure

(3) Final attempt to derive gradient descent

Recall f is convex and L-strongly smooth, so $0 \preceq \nabla^2 f(x) \preceq L \cdot I$

$$\Rightarrow \forall y, \quad \tfrac{1}{2} \langle y, \nabla^2 f(x) \cdot y \rangle \leq \tfrac{1}{2} L \|y\|^2$$

so instead of...

$$x_{k+1} = \underset{x}{\operatorname{argmin}} \; f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \tfrac{1}{2} \langle x - x_k, \nabla^2 f(x_k)(x - x_k) \rangle$$

... try ...

$$x_{k+1} = \underset{x}{\operatorname{argmin}} \; \underbrace{f(x_k) + \langle \nabla f(x_k), x - x_k \rangle + \tfrac{1}{2} L \|x - x_k\|^2}_{g_k(x)}$$

$$= \underbrace{x_k - \tfrac{1}{L} \nabla f(x_k)}_{\text{cheap update!}}$$

(i.e., do Newton update w/ $\nabla^2 f = L \cdot I$ so $(\nabla^2 f)^{-1} = \tfrac{1}{L} I$)

cheap update! $O(n)$ computation (vs. $O(n^3)$ for Newton)

in 1D, $n=1$, Newton is just as cheap, which is 1 reason you focus on it in 1D root-finding

The fact $\nabla^2 f(x) \preceq L \cdot I$

means $g_k(x) \geq f(x) \quad \forall x$

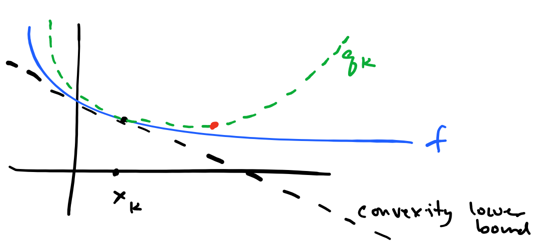
So... g_k is more than a linearization

or... a linearization w/ a penalty to show we don't trust linearization far away from x_k

... less than the full 2^{nd} order Taylor series

... it's a majorizer of f.

We repeatedly minimize the majorizer



MM framework "Majorization-minimization"

We'll give a specific proof for convergence of gradient descent later (using convexity)

... but MM applies even if f isn't convex.

Assume we can always construct a majorizer g_k s.t.

1) $\forall x, \quad f(x) \le g_k(x)$ (majorizes f)

2) $f(x_k) = g_k(x_k)$

Iterate: $x_{k+1} \in \underset{x}{\operatorname{argmin}}\ g_k(x)$

Then this algorithm is a descent algo, ie., $\underbrace{\text{it never makes things worse}}_{\text{rather weak but better than not having it!}}$:

$$\begin{aligned} f(x_{k+1}) &\le g_k(x_{k+1}) && \text{by (1)} \\ &\le g_k(x_k) && \text{since } x_{k+1} \text{ minimizes } g_k \\ &= f(x_k) && \text{by (2).} \end{aligned}$$

w/ a bit more work, a typical result might show:

- If $f(x)$ is bounded below, then $f(x_k)$ converges
- If (x_k) converges and f is lsc, then the limit $x_k \to x$ is a stationary point, $\nabla f(x) = 0$.

No convexity needed

Ex: (usually non-convex)

① Expectation Maximization (EM)
for max. likelihood estimation

② DC: Difference of Convex functions
AKA convex + concave

$$f(x) = g(x) - h(x)$$
g, h both convex

then $g_k(x) = g(x) - \big(\underbrace{h(x_k) + \langle \nabla h(x_k), x - x_k \rangle}_{\text{affine in } x}\big)$

is a majorizer, and since g_k is convex,

$\min_x g_k$ is often reasonable

Not all non-convex problems are equally hard

(though note EM, for example, still only gives a stationary pt.)