Lecture 7

Last time
- Preconditioning
- Newton's method

Today
- Quasi-Newton methods
- BFGS

We saw last class that Newton's has a couple of issues:

(i) It might converge to saddle points if started close to one. ($\lambda_{\min}(\nabla^2 f(w_*)) < 0$)

(ii) It is expensive.

Let us emphasize the second issue.

Given a dense PSD symmetric matrix M^* we can invert it exactly using a Cholesky factorization, which costs $O(d^2)$ memory & $O(d^3)$ flops.

We can invert it inexactly using conjugate gradient, which costs $O(d^2)$ memory and $(\sqrt{\kappa(M^*)} \log(1/\epsilon)\, d^2)$ flops.

It's hard to abstract these numbers. Let's see an example (see website).

Thus, there are different regimes

Scale	Range	What is reasonable?
Small	$d \leq 10^3$	d, d^2, d^3
Medium	$10^3 \leq d \leq 10^5$	d, d^2
Large	$d \geq 10^5$	d.

When dealing with $f_n(w) = \sum_{j=1}^{n} \ell_j(w)$ we increase the cost of computing gradients & Hessians

Today, we will try to get preconditioning to work up to medium scale.

Quasi-Newton methods

A natural idea is to substitute $\nabla^2 f(w)$ with something a bit simpler to compute.

The idea is

(QN) $$w_{k+1} \leftarrow w_k - \eta_k B_k^{-1} \nabla f(w).$$

We would like three things:

(1) Descent in $f(w_k)$. ← Intuitively this should help us avoid saddles.

(2) Emulates Newton near a minimizer. ↑ Fast convergence

(3) Updating & Inverting B_k is "cheap!"

Let us start with a general condition

that guarantees descent.

Lemma (D2.0) Suppose $f:\mathbb{R}^d \to \mathbb{R}$ is L-smooth and $B_k \succ 0$, then (QL) gives

$$f(w_{k+1}) \le f(w_k) - \underbrace{\left(\eta_k - \frac{L\eta_k^2}{2\lambda_{\min}(B_k)}\right)}_{\text{Positive when } \eta_k \in (0, \frac{2\lambda_{\min}(B_k)}{L})} \|\nabla f(w_k)\|^2_{B_k^{-1}}$$

The proof is very similar to that of the original descent lemma, and it right away tells us that we get substitute (1) with (1*) $B_k \succ 0$.

Now, let us try to materialize (2). Intuitively we want to capture the "curvature" of f. We can capture some of it with gradients, since

$$\nabla f(w_{k+1}) - \nabla f(w_k) = \underbrace{\left(\int_0^1 \nabla^2 f(w_k + t\, s_k)\, dt\right)}_{H_k} s_k. \quad (\star)$$

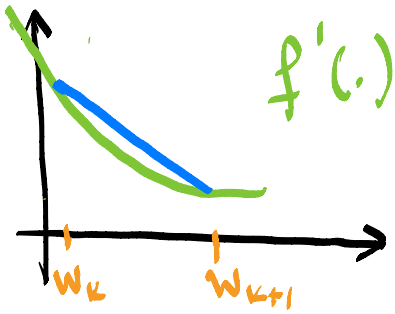
($s_k = w_{k+1} - w_k$)

When $\|s_k\|$ is small (near criticality) $H_k \approx \nabla^2 f(w_{k+1})$.

So we could use $(\star)$ as a way to constrain B_{k+1}. So we ask for

$$(2^*)\quad B_{k+1}\underbrace{(w_{k+1}-w_k)}_{s_k} = \underbrace{\nabla f(w_{k+1})-\nabla f(w_k)}_{y_k},$$

In 1D this corresponds to



$$B_{k+1} = \frac{f'(w_{k+1})-f'(w_k)}{w_{k+1}-w_k}.$$

In higher dimensions however B_k has $d(d+1)/2$ and (2^*) only has d constraints, thus, there are many B_k's that satisfy this.

For this we go back to (3), one simple way to do this is to request that

(3_r^*) $B_{k+1}-B_k$ has rank r.

A first natural idea is $\operatorname{rank}(B_{k+1}-B_k)=1$. However one can show that if (2^*) and (3_1^*), then (1^*) cannot hold. Thus, we focus on (3_2^*). We know that

$$B_{k+1}-B_k = \alpha uu^T + \beta vv^T \quad \text{with } \alpha,\beta\in\mathbb{R},\ u,v\in\mathbb{R}^d$$

Given that we expect (2^*) it is natural to take $u = B_k s_k$ and $v = y_k$.

Allows us to easily correct for (2*)

Then

$$y_k \overset{\text{want}}{=} B_{k+1} s_k = B_k s_k + \alpha \langle B_k s_k, s_k \rangle B s_k + \beta \langle y_k, s_k \rangle y_k$$

$$\Rightarrow \quad \alpha = -\frac{1}{\langle B_k s_k, s_k \rangle} \quad \text{and} \quad \beta = \frac{1}{\langle y_k, s_k \rangle}.$$

This leads to the update

$$B_{k+1} \leftarrow B_k - \frac{B_k s_k s_k^T B_k}{\langle B_k s_k, s_k \rangle} + \frac{y_k y_k^T}{\langle s_k, y_k \rangle}.$$

Proposition: Suppose that $B_k = B_k^T > 0$, $s_k \neq 0$, and $\langle s_k, y_k \rangle > 0$. Then, $B_{k+1} > 0$. ⊣

Proof: Take $v \in \mathbb{R}^d \setminus \{0\}$. Then,

$$v^T B_{k+1} v = v^T B_k v - \underbrace{\frac{(v^T B_k s_k)^2}{\langle B_k s_k, s_k \rangle}}_{\text{Nonnegative by Cauchy-Schwarz}} + \frac{\langle v, y_k \rangle^2}{\langle s_k, y_k \rangle}$$

$\langle B^{1/2} a, B^{1/2} b \rangle^2 \leq \langle Ba, a \rangle \langle Bb, b \rangle$

There are two possibilities.

▷ ($v = c s_k$) Then, first two terms equal zero and $v^T B_{k+1} v = c^2 \langle s_k, y_k \rangle > 0$.

▷ (otherwise) Cauchy-Schwarz is strict and

the first two terms are positive. (why?) □

Thus, we need to guarantee that $\langle y_k, s_k \rangle > 0 \quad \forall k$. This motivates the Wolfe conditions for finding η_k, namely given $0 < c_1 < c_2 < 1$, we want (Ensures the two are compatible)

(Descent)

$$f(w_k - \eta_k B_k^{-1} \nabla f(w_k)) \leq f(w_k) - c_1 \eta_k \langle B_k^{-1} \nabla f(w_k), \nabla f(w_k) \rangle,$$

(☺)

$$\langle \nabla f(\underbrace{w_k - \eta_k B_k^{-1} \nabla f(w_k)}_{w_{k+1}}), B_k^{-1} \nabla f(w_k) \rangle \leq c_2 \langle \nabla f(w_k), B^{-1} \nabla f(w_k) \rangle$$

(Ensures $\langle y_k, s_k \rangle > 0$.)

To see why we want the last condition, note

$$\begin{aligned} \langle y_k, s_k \rangle &= -\eta_k \langle \nabla f(w_{k+1}) - \nabla f(w_k), B_k^{-1} \nabla f(w_k) \rangle \\ &\geq \eta_k (1 - c_2) \langle \nabla f(w_k), B_k^{-1} \nabla f(w_k) \rangle \\ &> 0. \end{aligned}$$

BFGS (Broyden-Fletcher-Goldfarb-Shanno)

We have all the elements to state BFGS, with one small caveat, we want to be able to update the inverse B_k^{-1} efficiently, to do so we note that

$$(+) \quad B_{k+1}^{-1} = \left(I - \frac{s_k y_k^T}{y_k^T s_k}\right) B_k^{-1} \left(I - \frac{y_k s_k^T}{y_k^T s_k}\right) + \frac{s_k s_k^T}{y_k^T s_k}$$

Can be written with one matrix-vector product and two outer product.

(Check!)

Thus the total cost is $O(d^2)$ flops.

BFGS Algorithm

Start with w_0 and $B_0^{-1} = \gamma I$. ($\gamma > 0$.)

Step $k \geq 0$:

Choose $\eta_k > 0$ to satisfy (☺)

Set $w_{k+1} \leftarrow w_k - \eta_k B_k^{-1} \nabla f(w_k)$.

Update B_{k+1}^{-1} using (+).

This algorithm is great in practice, but when d gets large, $O(d^2)$ becomes too much of a burden. Instead, Nocedal proposed in 1980 a variant that only keeps the last m pairs of (s_k, y_k). The complexity of this variant is $O(dm)$ and usually people set $m = 10$ or $m = 20$. This variant is known as L-BFGS.

low memory

What is known about (L-)BFGS?

One can show that BFGS satisfies

$$w_k - w_0 \in \text{span}\{\nabla f(w_0), \ldots, \nabla f(w_k)\}.$$

So we should not expect faster convergence than for μ-SC + L-smooth.

	L-smooth	+ μ-SC	+ Q-Lipschitz Hess
BFGS	Fails to reach critrcality Mascarenhas (2004)	Global linear convergence Powell ('76)	Global linear Eventually superlinear Nocedal-Wright (1999)
L-BFGS	Fails to reach critrcality Dai (2002)	Global linear convergence Liu-Nocedal ('89)	Global linear convergence Liu-Nocedal ('89)

In practice L-BFGS seems to be much better!

See also Jin-Jiang-Mokhari (2024) for a more modern nonasymtotic treatment of BFGS.