Lecture 24

Quasi-Newton methods

- Faster than gradient based methods
- Don’t need Hessian
- Secant condition
- Curvature condition
- DFP
- BFGS

Lecture 25- Nonlinear equations

- Homotopy methods

Lecture 26 - Trust region methods

- Dogleg method

Idea w/ Quasi-Newton

- Hessians can (maybe?) annoying to compute.
- Can we go faster than first order w/o access to a Hessian?

Idea: "~~approximate the Hessian~~".

"approximate the function" with a quadratic surrogate.

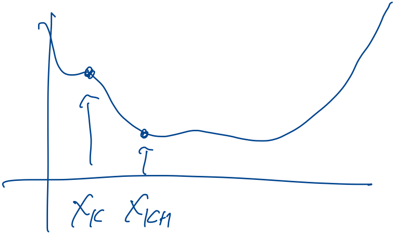
Is this hard?

Idea 0: finite differences?

Can't we do these?

Lots of little issues, see Lecture 23.

Let's see Quasi-Newton methods in 1d.



What does Newton do?

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

$$f''(x_k) \approx \frac{f'(x_k) - f'(x_{k-1})}{x_k - x_{k-1}}$$

Secant

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

⊛ This idea works! ☺

In 1d.

The Secant method Converges Superlinearly fast!

In general

Newton: $H(x_k) p_k = -g_k$

$\downarrow$ ~~"approx."~~ / "behaves-like"

B_k

Quasi-Newton, we use

$$B_k p_k = -g_k.$$

and then update $B_k \to B_{k+1}$.

Iteration

Initialize B_0 → How?

Very Important!

Until done

- Solve $B_k p_k = -g_k$ for a search dir
- Compute α_k for line search
- update $x_k \to x_{k+1} = x_k + \alpha_k p_k$.
- Update $B_k \to B_{k+1}$

⊛ We know x_{k+1} Now. ⊛

Q: What is different from Newton?

🟩 = differences

Are all updates $B_k \to B_{k+1}$ good?

No! $B_0 = I$, $B_{k+1} = B_k$ gives grad. descent, and we want better!

Idea: $B_k p_k = -g_k$ is based on the minimizer of $\frac{1}{2} p_k^T B_k p_k - g_k^T p_k + f_k$

Only true if B_k is sym. pos. def.

So Ensure B_{k+1} is spd!

Idea:

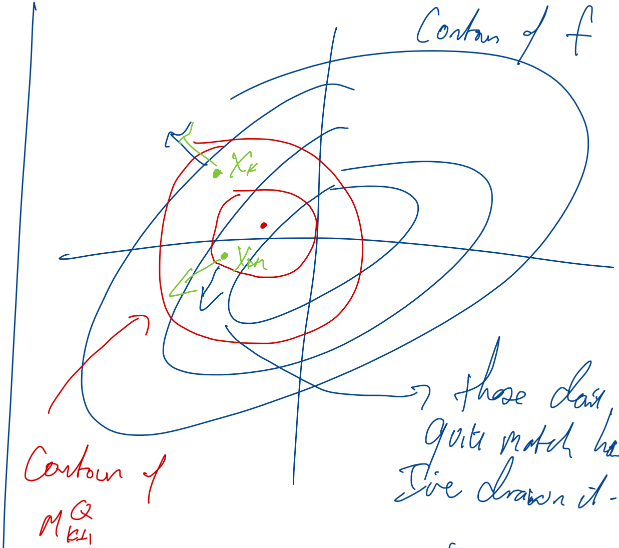
The surrogate function to minimize is

$$M_k^Q(p) = f_k + g_k^T p + \frac{1}{2} p^T B_k p$$

We'd like the gradient of M_{k+1}^Q to match at x_k, x_{k+1}.

$$\nabla M_{k+1}^Q(0) = g_{k+1} \quad ✓$$

$$\nabla M_{k+1}^Q(-\alpha p_k) = g_k$$



at x_{k+1}, x_k, we want to create matching gradients

As algebra: this is Easy!

$$M_k^Q(p) = f_k + g_k^T p + \frac{1}{2} p^T B_k p$$

$$M_{k+1}^Q(p) = f_k + g_{k+1}^T p + \frac{1}{2} p^T B_{k+1} p$$

$$\nabla M_{k+1}^Q(p) = g_{k+1} + B_{k+1} p$$

$$\nabla M_{k+1}^Q(0) = g_{k+1} \quad \checkmark \quad (\text{yay!})$$

I want

$$\boxed{\nabla M_{k+1}^Q(-\alpha p_k) = g_k}$$

$$\Longleftrightarrow$$

$$g_{k+1} + B_{k+1} \underbrace{(-\alpha p_k)}_{x_k - x_{k+1}} = g_k$$

let $s_k = x_k - x_{k+1}$

$$B_k s_k = \underbrace{g_k - g_{k+1}}_{y_k}$$

$$\Longleftrightarrow$$

We want

$$B_{k+1} s_k = y_k$$

called the secant condition.

Note: Why not $s_k = x_{k+1} - x_k$?

Convention ☹

All Quasi-Newton methods satisfy the secant condition!

Now, if $B_{k+1} \succ 0$.

What do we know about

$y_k^T s_k$?

Your quiz!

Yes, B_{k+1} satisfies secant condition.

$B_{k+1} s_k = y_k$

$B_{k+1} \succ 0 \implies$

$x^T B_{k+1} x > 0$ for all $x \neq 0$.

So

$$s_k^T \underbrace{B_{k+1} s_k}_{= y_k} > 0$$

$s_k^T y_k > 0$.

If you do a secant update, and you want $B_{k+1} > 0$.

But s_k, y_k are indep. of $B_k \to B$ update! So if we want $B_{k+1} \succ$ we will need

$s_k^T y_k > 0$

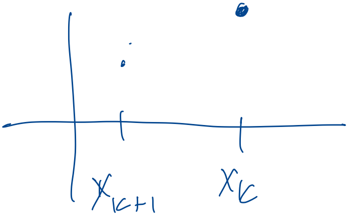
How do we Ensure t

Idea 0: Add to line search condi

Idea 1: Realize that Wolfe condition $\implies$

$s_k^T y_k > 0$.

Let's look at 1d to see what goes wrong.



$x_k - x_{k+1} = s_k > 0$ (I'm moving left!)

$y_k = g_k - g_{k+1}$

$s_k^T y_k < 0 \Rightarrow g_k - g_{k+1} < 0.$

$\Rightarrow g_{k+1}$ is steeper than g_k.

We get $s_k^T y_k < 0$ if g_{k+1} is steeper than g_k. Which doesn't happen w/ Wolfe line search!

So just use Wolfe line search!

Note $s_k = -\alpha p_k$.

Steps: show

$$g_{k+1}^T s_k \ge c_2 \, g_k^T s_k$$

$-g_k^T s_k$ from S1[illegible].

Then, you can get

$$y_k^T s_k \ge \underbrace{(c_2 - 1)}_{< 0} \alpha_k \underbrace{g_k^T p_k}_{< 0}$$

since $c_2 < 1$.

$\Rightarrow \quad \cancel{\geq} 0.$

What are these updates?

$$B_k \to B_{k+1}$$

DFP:

$$\rho_k = \frac{1}{y_k^T s_k}$$

$$B_{k+1} = (I - \rho_k y_k s_k^T) B_k (I$$
$$+ \rho_k y_k y_k^T.$$

If $B_k \succ 0$, then $B_{k+1} \succ 0$

$y_k^T s_k > 0$.

$-\rho_k s_k y_k^T)$

i/

$\min \|B - B_k\|_W$

s.t. $B = B^T$, $B s_k = y_k$

Where W is Based on the average Hessian for $x_k \to x_{k+1}$.

Why update $B_k \to B_{k+1}$?

Why not update

$B_k^{-1} \to B_{k+1}^{-1}$?

BFGS update

works w/

T_k behaves like

Secant condition for T_k

$T_{k+1} y_k = s_k$.

BFGS update:

$T_{k+1} = (I - \rho_k s_k y_k^T) T_k$

$+ \rho s_k s_k^T$.

⊕ Same derivation w/ the norm. ☺

H_k^{-1}

is

$(I - \rho_k y_k s_k^T)$

weighted

min $\|T - T_k\|_W$

st $T = T^T$, $T y_k = s_k$.

Note BFGS ≠ DFP on B_k^{-1}.

Both BFGS, DFP are in the Broyden class. These all give superlinear convergence!

I have a video showing this!

SR1 - update...

give up ☹

But these of trust methods.

Sym. rank 1

pos-def.

are usal

region