Building efficient Approximations to the Hessian

we want ⓐ $B\delta x = -g$

where B is p.d. (so we get a descent dir)

ⓑ B is easy to work with — particularly when there are lots of variables

ⓒ B is "like" the Hessian

Recall the model:

$$f(x+\delta x) \cong f(x) + \delta x^T \nabla f\Big|_x + \frac{1}{2}\delta x^T H \delta x$$

which implies:

$$\nabla f(x+\delta x) \cong \nabla f\Big|_x + H\delta x$$

So that

$$H\delta x = \nabla f\Big|_{x+\delta x} - \nabla f\Big|_x$$

This is known as a Secant Condition

often written

$$H\Delta x = \Delta g$$

Δg: change in gradient

Idea observing the change in grad. tells you something about H

$\Rightarrow$ require our approx B to meet secant cond, so that

$$B\Delta x = \Delta g.$$

This leads to a rich seam of algorithms

I: (Barzilai - Borwein)

- assume B is $\left(\frac{1}{\alpha}\right)I$.

then $\left(\frac{1}{\alpha}\right)\Delta x$ should be Δg

so compute

$$\min_{\alpha} \left\| \left(\frac{1}{\alpha}\right)\Delta x - \Delta g \right\|^2$$

yielding.

$$\alpha^l = \frac{\Delta x^T \Delta x}{\Delta x^T \Delta g} \quad \leftarrow \text{long step}$$

Notice that we could $\min_\alpha \| \Delta x - \alpha \Delta g \|^2$.

to get

$$\alpha^s = \frac{\Delta x^T \Delta g}{\Delta g^T \Delta g} \quad \leftarrow \text{short step}$$

Method:
- take each step
- choose one

↑ all sorts of ways of doing this – we don't really care.

Point:

If x is huge, we still have a little Hessian info, which is known to help.

Idea: maintain estimate of Hessian over steps.

at kth step, we have a model function

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

and we use this to choose a step.

Notice: - we'd like g_k to be $\nabla f|_{x_k}$

- B_k to be "good" (as above)

assume we take a step from k

$$p_k = -B_k^{-1} g_k$$

$$x_{k+1} = x_k + \alpha_k p_k$$

↑ this satisfies Wolfe conds.

our neco model is:

$$m_{k+1}(p) = f_{k+1} + g_{k+1}^T p + \frac{1}{2} p^T B_{k+1} p.$$

f_{k+1}, g_{k+1}: known; B_{k+1}: wanted

require B_{k+1} satisfy secant conditions.

step at k is $\alpha_k p_k = s_k$ (known)

require $B_{k+1} s_k = g_{k+1} - g_k$

write y_k

~~Notice $s_k^T s_k > 0 = s_k^T$~~

Notice

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

if we build B_{k+1} mgh

Want: B_{k+1} "like" B_k

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

B_{k+1} psd if B_k p.s.d.

Construction:

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

Check:

$$B_{k+1} s_k = B_k s_k - \frac{y_k s_k^T B_k s_k}{s_k^T y_k} - B_k s_k + \frac{y_k s_k^T B_k s_k}{s_k^T y_k} + y_k \quad \checkmark$$

$$B_{k+1} = A^T B_k A + C \quad \checkmark$$

$A^T B_k A$: psd if B_k psd

C: psd

- We are interested in B_{k+1}^{-1} and have an approx of B_{k+1}
- Write $B_{k+1}^{-1} = C_{k+1}$, we have

turns out:

$$C_{k+1} = C_k - \frac{C_k y_k y_k^T C_k}{y_k^T C_k y_k} + \frac{s_k s_k^T}{y^T s_k}$$

notice an interesting feature:

- if multiplication by C_k is "easy" then so is mult by C_{k+1}

This is DFP (Davidon-Fletcher-Powell)

of interest for
- historical reasons
- intro to major idea

BFGS

(Broyden - Fletcher - Goldfarb - Shanno)

- one of the most influential rejected papers ever written!

Idea: . maintain C_k $(= B_k^{-1})$ directly.

- from Secant condition, we have

$$B_k s_k = y_k$$

so $$s_k = C_{k+1} y_k$$

- and want:
  - secant
  - C_{k+1} "close to" C_k
  - C_{k+1} psd if C_k psd

We get

$$C_{k+1} = \left(I - \frac{s_k y_k^T}{s_k^T y_k}\right) C_k \left(I - \frac{y_k s_k^T}{s_k^T y_k}\right) + \frac{s_k s_k^T}{y_k^T s_k}.$$

(check as for DFP)

Notice we can write a recursion:

$$C_{k+1} = (V_k)^T C_k V_k + \frac{s_k s_k^T}{y_k^T s_k}$$

$$= V_k^T \left[V_{k-1}^T C_{k-1} V_{k-1} + \frac{s_{k-1} s_{k-1}^T}{s_k^T y_k} \right] V_k + \frac{s_k s_k^T}{s_k^T y_k}$$

etc

What is H_0?

options

- the Hessian inverse
- Identity
- scaled identity
- $\frac{1}{\text{diag}(\text{Hessian})}$

Big attraction: (A)

- If k is sufficiently large, H_0 wont matter much
- so look back only r steps

⇒ L-BFGS

↑ limited memory, cause you need to store only r s's and r y's and H_0

Big attraction: (B)

- with sensible choice of H_0, mult by H_{k+1} is easy

$$s_k s_k^T x = s_k (s_k^T x)$$

$$V_k x = \left(I - \frac{y_k s_k^T}{s_k^T y_k}\right) x = x - \frac{y_k}{s_k^T y_k}\left[s_k^T x\right]$$

(and notice efficiencies w/ $s_k^T x$)

LBFGS is the method of choice for big, unconstrained problems