

Building efficient Approximations to the Hessian ①

we want (a) $B \delta x = -g$

where

B is p.d.

(so we get a descent dir)

- (b) B is easy to work with
— particularly when there are lots of variables
- (c) B is "like" the Hessian

recall the model:

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

which implies:

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

so that

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

This is known as a secant condition ⁽²⁾

is often written

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

change in gradient

Idea you observing the change in grad. tells 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 $(\frac{1}{\alpha})I$.

then $(\frac{1}{\alpha})\Delta x$ 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
to get

$$\min_{\alpha} \|\Delta x - \alpha \Delta g\|^2$$

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

Method:

- take each step
- choose one
 - ↑ all sorts of ways of doing this - we don't really care.

Point:

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

Idea: maintain estimate of Hessian over steps.

at k th step, we have a model function

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

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 new model is :

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

\swarrow known \searrow \uparrow wanted

require

B_{k+1} satisfy secant conditions.

step at k is $\underbrace{\alpha_k p_k = s_k}_{\text{known}}$

require

$$B_{k+1} s_k = \underbrace{g_{k+1} - g_k}_{\text{write } y_k}$$

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

Notice

⑥

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

if we build B_{k+1} right

Want:

B_{k+1} "like" B_k

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

B_{k+1} psd if B_k psd.

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:

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

$$B_{k+1} = \underbrace{A^T B_k A}_{\substack{\uparrow \\ \text{psd if } B_k \text{ psd}}} + \underbrace{C}_{\substack{\uparrow \\ \text{psd}}} \quad \checkmark$$

• 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:

⑧

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

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

$$B_k s_k = y_k$$

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

- 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 DFB)

Notice we can write a recursion:

$$\begin{aligned} C_{k+1} &= (V_k)^T C_k V_k + \frac{S_k S_k^T}{y_k^T S_k} \\ &= V_k^T \left[V_{k-1}^T C_{k-1} V_{k-1} + \frac{S_{k-1} S_{k-1}^T}{S_{k-1}^T y_{k-1}} \right] V_k \\ &\quad + \frac{S_k S_k^T}{S_k^T y_k} \end{aligned}$$

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 won't matter much
- so look back only r steps

\Rightarrow h -BFGS

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

Big attraction:

(B)

(12)

- with sensible choice of H_0 , mult by

H_{k+1} is easy

$$S_k S_k^T x = S_k (S_k^T x)$$

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

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

LBFG-S is the method of choice for big, unconstrained problems