Taken from http://jkcray.maths.ul.ie/ms4327/Slides.pdf Annotated by Jean Honorio jhonorio@purdue.edu

6.1 The BFGS Method

BFGS (Broyden, Fletcher, Goldfarb & Shanno) is perhaps the most popular quasi-Newton method

DFP (Davidon, Fletcher & Powell) is the BFGS precursor

Start by forming the familiar quadratic model/approximation:

$$m_k(p) = f_k + g_k^T p + \frac{1}{2} p^T H_k p \tag{6.1}$$

- Here H_k is an $n \times n$ positive definite symmetric matrix (that is an approximation to the exact Hessian) .
- H_k will be updated at each iteration.
- For clarity, I will use H for approximations to the **Hessian** and J for approximations to the **Inverse Hessian**.

- The function and gradient values of the model at $p = 0$ match f_k and g_k.
- In other words $m_k(0) = f_k$ and $\nabla_p m_k(p)_{|p=0} = g_k$.
- The minimiser of this model wrt **p** is as usual:

$$p_k = -H_k^{-1} g_k \tag{6.2}$$

  and is used as the search direction.
- The new iterate is

$$x_{k+1} = x_k + \alpha_k p_k \tag{6.3}$$

  again as usual, where the step length α_k by line search

- Clearly if H_k is the exact Hessian, I have Newton's method — in this Chapter, H_k will be an approximation to the Hessian based on gradient values.

- Instead of computing H_k from scratch at each iteration, Davidon used the following clever argument:
- Suppose that I have generated a new iterate x_{k+1} and wish to construct a new quadratic model of the form

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

- It is reasonable to ask that the gradient of m_{k+1} should match the gradient of f at x_k & x_{k+1} .
- Since $\nabla m_{k+1}(0) \equiv g_{k+1}$, (they match at x_{k+1}) I need only check that they match at x_k — which means I require that:

$$\nabla m_{k+1}(-\alpha_k p_k) \equiv g_{k+1} - \alpha_k H_{k+1} p_k = \nabla m_k(0) \equiv g_k .$$

- Rearranging, I have

$$H_{k+1} \alpha_k p_k = g_{k+1} - g_k . \tag{6.4}$$

- First define:

$$s_k = x_{k+1} - x_k \equiv \alpha_k p_k \tag{6.5a}$$

$$y_k = g_{k+1} - g_k \tag{6.5b}$$

Then (6.4) gives us the **secant equation**

$$H_{k+1} s_k = y_k. \tag{6.6}$$

- I am taking H_{k+1} to be positive definite so ${s_k}^T H_{k+1} s_k > 0$ and so this equation is possible only if the step s_k and change in gradients y_k satisfy the **curvature condition**

$$s_k^T y_k > 0. \tag{6.7}$$

- In general, though, I need to enforce 6.7 by imposing restrictions on the line search procedure for choosing α_k.

- The problem is that there are infinitely many solutions for H_k as there are $n(n+1)/2$ degrees of freedom in a symmetric matrix and the secant equation represents only n conditions.
- Requiring that H_{k+1} be positive definite represents n inequality conditions but there are still degrees of freedom left.
- To determine H_{k+1} uniquely, I impose the additional condition that; **among all symmetric matrices satisfying the secant equation, H_{k+1} is "closest to" the current matrix H_k .**
- So I need to solve the problem:

$$\min_{H} \|H - H_k\| \tag{6.9a}$$

$$\text{subject to} \quad H = H^T, Hs_k = y_k \tag{6.9b}$$

- I can use any convenient matrix norm — a choice that simplifies the algebra (reduces the pain) is the "weighted Frobenius norm":

$$\|A\|_W \equiv \|W^{\frac{1}{2}} A W^{\frac{1}{2}}\|_F, \tag{6.10}$$

where $\|C\|_F^2 \equiv \sum_{i=1}^n \sum_{j=1}^n C_{ij}^2$ for any square matrix C.

- **Any** choice of the weight matrix W will do provided it is positive definite, symmetric and satisfies $Wy_k = s_k$.

- For example, I could take $W = \overline{H}_k^{\,-1}$, where $\overline{H}_k$ is the **average Hessian** defined by

$$\overline{H}_k = \int_0^1 \nabla^2 f(x_k + \tau \alpha_k p_k) d\tau. \tag{6.11}$$

- It follows that

$$y_k = \overline{H}_k \alpha_k p_k = \overline{H}_k s_k \tag{6.12}$$

by using Taylor's theorem.

I can now state my update formula for the Hessian estimate H_k as a Theorem:

Theorem 6.1 **a** *solution of (6.9a, 6.9b) is*

$$\textbf{DFP} \quad H_{k+1} = (I - \gamma_k y_k s_k^T) H_k (I - \gamma_k s_k y_k^T) + \gamma_k y_k y_k^T, \tag{6.13}$$

where

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

- H_k is my current estimate of the Hessian, usually initially the Identity matrix.
- H_{k+1} is my (I hope) improved estimate of the Hessian, using newly available information, namely the two vectors s_k & y_k .

6.2 Inverting the Hessian approximation

It would be very useful if I could calculate an estimate of the the **inverse** Hessian $\nabla^2 f$ — say $J_k \equiv H_k^{-1}$. This would allow us to calculate $p_k = -J_k g_k$ instead of solving $H_k p_k = -g_k$ for the search direction p_k — giving a speedup in the algorithm.

But how to transform Eq. 6.13 into an update formula for J_{k+1} in terms of J_k ?

I need a formula that gives the inverse of H_{k+1} in terms of the inverse of H_{k+1}.

The Sherman-Morrison-Woodbury formula is what I need.

It states that if a square non-singular matrix A is updated by

$$\hat{A} = A + RST^T$$

where R, T are $n \times p$ matrices for $1 \le p < n$ and S is $p \times p$ then

$$\hat{A}^{-1} = A^{-1} - A^{-1} R U^{-1} T^T A^{-1}, \tag{6.20}$$

where $U = S^{-1} + T^T A^{-1} R$.

Using the SMW formula, I can derive the following equation for the update of the inverse Hessian approximation, H_k that corresponds to the DFP update of B_k in Eq. 6.13;

$$\textbf{DFP} - \textbf{Inverse} \quad J_{k+1} = J_k - \frac{J_k y_k y_k^T J_k}{y_k^T J_k y_k} + \frac{s_k s_k^T}{y_k^T s_k}. \tag{6.21}$$

This is a rank-2 update as the two terms added to J_k are both rank-1.

The DFP method has been superseded by the BFGS (Broyden, Fletcher, Goldfarb & Shanno) method. It can be derived by making a small change in the derivation that led to Eq. 6.13. Instead of imposing conditions on the Hessian approximations H_k , I impose corresponding conditions on their inverses J_k . The updated approximation J_{k+1} must be symmetric and positive definite. It must satisfy the secant equation Eq. 6.6, now written as

$$J_{k+1} y_k = s_k. \tag{6.22}$$

and also the "closeness" condition

$$\min_J \|J - J_k\| \tag{6.23a}$$

$$\text{subject to} \quad J = J^T, J y_k = s_k. \tag{6.23b}$$

The matrix norm is again the weighted Frobenius norm, where the weight matrix is now any matrix satisfying $W s_k = y_k$.

(You can take W to be the "average" Hessian $\overline{H}_k$ defined in Eq. 6.11 above — though any matrix satisfying $Ws_k = y_k$ will do.)

Using the same reasoning as above, a solution to 6.23a is given by

$$\textbf{BFGS} \quad J_{k+1} = (I - \gamma_k s_k y_k^T) J_k (I - \gamma_k y_k s_k^T) + \gamma_k s_k s_k^T. \tag{6.24}$$

Note the symmetry between this equation and Eq. 6.13 — one transforms into the other by simply interchanging s_k and y_k — of course $\gamma_k = \frac{1}{{s_k}^T y_k}$ is invariant under this transformation.

J_0 is often taken to be just the identity matrix — possibly scaled.

Algorithm 6.1 (BFGS)

begin

Given x_0, tolerance $\varepsilon > 0$ and starting J_0

$k \leftarrow 0$;

while $\|g_k\| > \varepsilon$ do

$\boxed{p_k \leftarrow -J_k g_k}$ This takes O(n^2) time, while solving a linear system for the Newton method takes O(n^3)

$x_{k+1} \leftarrow x_k + \alpha_k p_k$ after performing line search

Define $s_k = x_{k+1} - x_k$ and $y_k = g_{k+1} - g_k$

Compute J_{k+1} using Eq. 6.24

$k \leftarrow k+1$;

end

end