Quasi-Newton Minimisation with Limited Storage

Introduction.

Our objective is to find the global minimum of a twice continuously differentiable function of many variables $F(x)=F(x^1,\ldots,x^N)$ where N can be arbitrarily large.

The classical Newton Minimisation Method delivers a fast-converging (superlinear) algorithm for solving this problem when a good initial approximation is known.

Following is a formal description of the Newton Minimisation algorithm for the above problem:

Let $x_0=(x_0^1,\ldots,x_0^N)$ be an initial approximation of the point where F takes its global minimum.
Let $k=1$ be the iteration counter, $\varepsilon>0$ - termination flag.

Do loop:

Evaluate the gradient $G_{k-1}=\left(\frac{\partial F}{\partial x^1},\ldots,\frac{\partial F}{\partial x^N}\right)$ at $x=x_{k-1}$.

Evaluate the Hessian matrix $H=\|\frac{\partial^2 F}{\partial x^i \partial x^j}\|\|_{x=x_{k-1}}$ at $x=x_{k-1}$.

Define a direction in which the next iteration will be searched:

$$p_k=-H^{-1}G_{k-1}. \qquad (1)$$

Minimise the univariate function

$$\begin{aligned}&\Phi(\alpha):=F(x_{k-1}+\alpha p_k),\\ &a=\alpha_{\min}:\Phi(\alpha_{\min})=\min_{\alpha}\Phi(\alpha).\end{aligned} \qquad (2)$$

Evaluate the next approximation to solution:

$$x_k=x_{k-1}+s_k,\ s_k=\alpha_{\min}p_k$$

Terminate loop if $|F(x_k)|<\varepsilon\max(1,\|x_k\|)$, otherwise resume loop with $k = k + 1$.
End of loop

Application of this algorithm runs into difficulties when the Hessian matrix cannot be easily obtained numerically.
Two problems are posed in connection with practical application of the Newton Method:

- ***Can the inverse of the Hessian matrix in (1) be numerically approximated using the output of previous iterations rather than calculated directly?***

- ***What univariate minimisation (i.e. line search) can best be used to minimise function (2) so as to achieve the fastest global convergence?***

1. Limited Memory quasi-Newton Minimisation Method.

A number of algorithms exist (collectively known as quasi-Newton methods) that address this problem. The main idea of the quasi-Newton methods is to use the search directions s_k and gradients G_k returned from previous iterations to estimate the value of the Hessian matrix.

All of the quasi-Newton methods are based on the following observation: If **H** is the Hessian matrix of F at $x=x_{k-1}$, $s_{k-i}=x_{k-i}-x_{k-i-1}$ are the increments of x for previous iterations, $y_{k-i}=G_{k-i}-G_{k-i-1}$ are increments of the gradient, and the index $i=1,\ldots,M$ identifies the M previous steps of the algorithm, then:

$$y_{k-i}=G_{k-i}-G_{k-i-1}=\nabla F|_{x=x_{k-i}}-\nabla F|_{x=x_{k-i-1}} ¿ H\left(x_{k-i}-x_{k-i-1}\right)=Hs_{k-i}$$

$$s_{k-i}\approx H^{-1}y_{k-i}\ , \qquad (3)$$

assuming that $s_{k-i}=x_{k-i}-x_{k-i-1}$ or H are “sufficiently small”.

In any quasi-Newton method, the Hessian matrix is approximated with a symmetric positive-definite matrix that satisfies (3) for $i=1,\ldots,M$, where M is much smaller than N (if $M = N$ then **H** can be restored uniquely from (3)). When N is very large and it is either impossible or impractical to store N^2 elements of the approximate Hessian matrix, any of the “limited storage” quasi-Newton methods can be used that approximate the inverse Hessian matrix “on the fly” along with evaluating the next search direction through formula (1).

Assuming that H_k is an approximation to the Hessian obtained at the kth step, let us chose a symmetric matrix H_{k+1} that satisfies (3) for i=0 exactly:

$$H_{k+1}s_k = y_k \tag{4}$$

First, we pick a symmetric rank-one update

$$\left(H_{k+1}\right)_i^i = \left(H_k\right)_i^i + u^i v_j \ . \tag{5}$$

Substituting (5) into (4) we can transform (5) into

$$\left(H_{k+1}\right)_i^i = \left(H_k\right)_i^i + \frac{1}{\langle v, s_k\rangle}\left(y_k - H_k s_k\right)^i v_j \tag{6}$$

assuming that v is not orthogonal to s_k . The requirement of symmetry for (6) implies that v be chosen as $v = y_k - H_k s_k$. The latter is assumed to be non-zero, as otherwise H_k would have already satisfied (4) in place of H_{k+1} so no correction would have been required. Finally, we arrive at the following **symmetric rank-one Hessian update**:

$$\left(H_{k+1}\right)_j^i = \left(H_k\right)_j^i + \frac{1}{\langle y_k - H_k s_k, s_k\rangle}\left(y_k - H_k s_k\right)^i \left(y_k - H_k s_k\right)_j \tag{7}$$

Formula (7) exhausts all possible symmetric rank-one updates. In order to obtain alternative Hessian correction formulae, we will need to look into updates of rank two and above.

Let us revert to the rank-one update (6), this time allowing v to be arbitrary (but not orthogonal to s_k), and symmetrise the resultant operator H_{k+1} :

$$\left(H^2_{k+1}\right)_j^i = \left(H_{k+1} + H^T_{k+1}\right)_j^i / 2 \ . \tag{8}$$

The operator H^2_{k+1} , although symmetric, does not satisfy (4) so long as v is arbitrary. However, we can apply formulae (6) and (8) iteratively:

$$\begin{aligned} &\left(\tilde{H}^n_{k+1}\right)_i^i = \left(H^n_{k+1}\right)_i^i + \frac{1}{\langle v, s_k\rangle}\left(y_k - H^n_{k+1}s_k\right)^i v_j \\ &\left(H^{n+1}_{k+1}\right)_j^i = \left(\tilde{H}^n_{k+1} + \tilde{H}^{nT}_{k+1}\right)_j^i / 2, \\ &n = 3,4,\ldots \end{aligned} \tag{9}$$

Passing n to limit in (9) we can see that the limit $H_{k+1}=\lim_{n\to\infty} H_{k+1}^{n}$ exists, satisfies (4), is symmetric and given by the following formula:

$$\left(H_{k+1}\right)_j^i=\left(H_k\right)_j^i+\frac{1}{\langle v,s_k\rangle}\left(\left(y_k-H_k s_k\right)^i v_j+v^i\left(y_k-H_k s_k\right)_j\right)- \\ -\frac{\langle y_k-H_k s_k,s_k\rangle}{\langle v,s_k\rangle^2}v^i v_j \qquad (10)$$

If v is taken as y_k in 10 (which is possible to do as long as the Hessian is positive defined; indeed $\langle s_k,y_k\rangle=\langle s_k,H_{k+1}s_k\rangle>0$) we get the so-called Davidon-Fletcher-Powell (DFP) update:

$$\left(H_{k+1}\right)_j^i=\left(H_k\right)_j^i-\frac{1}{\langle s_k,H_k s_k\rangle}\left(H_k s_k\right)^i\left(H_k^T s_k\right)_j+\frac{1}{\langle y_k,s_k\rangle}\left(y_k\right)^i\left(y_k\right)_j+ \\ +\langle s_k,H_k s_k\rangle\left(w_k\right)^i\left(w_k\right)_j \qquad (11)$$

where

$$\left(w_k\right)^i=\frac{1}{\langle y_k,s_k\rangle}\left(y_k\right)^i-\frac{1}{\langle s_k,H_k s_k\rangle}\left(H_k s_k\right)^i . \qquad (12)$$

Vector (12) is orthogonal to s_k hence any multiple of the rank-one matrix $\left(w_k\right)^i\left(w_k\right)_j$ can be added to H_{k+1} without affecting the symmetry and condition (4).

While adding such a matrix to (11) leads to a family of parameterised updates, the method based on using (11) with the multiple of $\left(w_k\right)^i\left(w_k\right)_j$ term altogether dropped is believed to be the most efficient one. Thus, we arrive at the following Broyden-Fletcher-Goldfarb-Shanno (BFGS) update:

$$\left(H_{k+1}\right)_j^i=\left(H_k\right)_j^i-\frac{1}{\langle s_k,H_k s_k\rangle}\left(H_k s_k\right)^i\left(H_k^T s_k\right)_j+\frac{1}{\langle y_k,s_k\rangle}\left(y_k\right)^i\left(y_k\right)_j . \qquad (13)$$

Before we can apply (13) in a quasi-Newton minimisation for calculating the search direction using formula (1), we need to invert the matrix H_{k+1} . In the event we use only one-step BFGS update and $H_k=I$ is the identity matrix, the formula (13) will be transformed into

$$\left(H_{k+1}\right)_j^i=\delta_j^i-\frac{1}{\langle s_k,s_k\rangle}\left(s_k\right)^i\left(s_k\right)_j+\frac{1}{\langle y_k,s_k\rangle}\left(y_k\right)^i\left(y_k\right)_j\ . \qquad (14)$$

We can show by direct substitution that the inverse of matrix (14) is given by

$$\delta_m^j-\frac{\left(y_k\right)^j\left(s_k\right)_m}{\langle y_k,s_k\rangle}+\frac{\left(s_k\right)^j\left(s_k\right)_m}{\langle y_k,s_k\rangle}-\frac{\left(s_k\right)^j\left(y_k\right)_m}{\langle y_k,s_k\rangle}+\frac{\left(s_k\right)^j\left(s_k\right)_m\langle y_k,y_k\rangle}{\langle y_k,s_k\rangle^2}\ . \qquad (15)$$

Indeed, the product of matrices (14) and (15) is

$$\left\{\delta_j^i-\frac{\left(s_k\right)^i\left(s_k\right)_j}{\langle s_k,s_k\rangle}+\frac{\left(y_k\right)^i\left(y_k\right)_j}{\langle y_k,s_k\rangle}\right\}\times ¿$$

$$¿\times\left\{\delta_m^j-\frac{\left(y_k\right)^j\left(s_k\right)_m}{\langle y_k,s_k\rangle}+\frac{\left(s_k\right)^j\left(s_k\right)_m}{\langle y_k,s_k\rangle}-\frac{\left(s_k\right)^j\left(y_k\right)_m}{\langle y_k,s_k\rangle}+\frac{\left(s_k\right)^j\left(s_k\right)_m\langle y_k,y_k\rangle}{\langle y_k,s_k\rangle^2}\right\}=$$

$$¿\delta_m^i-\frac{\left(y_k\right)^i\left(s_k\right)_m}{\langle y_k,s_k\rangle}+\frac{\left(s_k\right)^i\left(s_k\right)_m}{\langle y_k,s_k\rangle}-\frac{\left(s_k\right)^i\left(y_k\right)_m}{\langle y_k,s_k\rangle}+\frac{\left(s_k\right)^i\left(s_k\right)_m\langle y_k,y_k\rangle}{\langle y_k,s_k\rangle^2}-$$

$$-\frac{\left(s_k\right)^i\left(s_k\right)_m}{\langle s_k,s_k\rangle}+\frac{\left(y_k\right)^i\left(y_k\right)_m}{\langle y_k,s_k\rangle}+\frac{\left(s_k\right)^i\left(s_k\right)_m\langle y_k,s_k\rangle}{\langle s_k,s_k\rangle\langle y_k,s_k\rangle}-\frac{\left(s_k\right)^i\left(s_k\right)_m\langle s_k,s_k\rangle}{\langle s_k,s_k\rangle\langle y_k,s_k\rangle}+$$

$$+\frac{\left(s_k\right)^i\left(y_k\right)_m\langle s_k,s_k\rangle}{\langle s_k,s_k\rangle\langle y_k,s_k\rangle}-\frac{\left(s_k\right)^i\left(s_k\right)_m\langle s_k,s_k\rangle\langle y_k,y_k\rangle}{\langle s_k,s_k\rangle\langle y_k,s_k\rangle^2}-\frac{\left(y_k\right)^i\left(s_k\right)_m\langle y_k,y_k\rangle}{\langle y_k,s_k\rangle^2}+$$

$$+\frac{\left(y_k\right)^i\left(s_k\right)_m\langle y_k,s_k\rangle}{\langle y_k,s_k\rangle^2}-\frac{\left(y_k\right)^i\left(y_k\right)_m\langle y_k,s_k\rangle}{\langle y_k,s_k\rangle^2}+\frac{\left(y_k\right)^i\left(s_k\right)_m\langle y_k,s_k\rangle\langle y_k,y_k\rangle}{\langle y_k,s_k\rangle^3}=$$

$$¿\delta_m^i$$

- the identity matrix.

Using operator (15) in formula (1) we get the so-called limited-memory BFGS method (L-BFGS) that uses the output of only one previous iteration to approximate the inverse Hessian matrix. In the context of formula (1), the application of operator (15) will result in the following formula (we introduce an intermediate approximation $p_k^{(0)}$ to the new search direction p_k):

$$
\begin{aligned}
&p_k^{(0)} := -G_{k-1} \\
&p_k := p_k^{(0)} - \frac{\langle s_{k-1,} p_k^{(0)} \rangle}{\langle y_{k-1,} s_{k-1} \rangle} y_{k-1} + \frac{\langle s_{k-1,} p_k^{(0)} \rangle}{\langle y_{k-1,} s_{k-1} \rangle} s_{k-1} - \frac{\langle y_{k-1,} p_k^{(0)} \rangle}{\langle y_{k-1,} s_{k-1} \rangle} s_{k-1} + \\
&+ \frac{\langle s_{k-1,} p_k^{(0)} \rangle \langle y_{k-1,} y_{k-1,} \rangle}{\langle y_{k-1,} s_{k-1} \rangle^2} s_{k-1}.
\end{aligned} \tag{16}
$$

We can eliminate the last term in (16) by introducing another intermediate approximation $p_k^{(1)}$:

$$p_k^{(0)} := -G_{k-1} \,, \tag{17}$$

$$p_k^1 := p_k^{(0)} - \frac{\langle s_{k-1,} p_k^{(0)} \rangle}{\langle y_{k-1,} s_{k-1} \rangle} y_{k-1} \,, \tag{18}$$

$$p_k := p_k^{(1)} + \left[\frac{\langle s_{k-1,} p_k^{(0)} \rangle}{\langle y_{k-1,} s_{k-1} \rangle} - \frac{\langle y_{k-1,} p_k^{(1)} \rangle}{\langle y_{k-1,} s_{k-1} \rangle} \right] s_{k-1} \,. \tag{19}$$

Note that p_k is calculated in (19) from both $p_k^{(0)}$ and $p_k^{(1)}$. Equivalence of (17-19) to (16) can be demonstrated by direct substitution.

If we re-write (17-19) in the operator form, we obtain

$$p_k^{(0)} := -IG_{k-1} \tag{20}$$

$$p_k^1 := \left(I + A_{k-1}\right) p_k^{(0)}, \; A_{k-1} = \|a_j^i\| = \| - \frac{y_{k-1}^i s_{k-1}^q \delta_{qj}}{\langle y_{k-1,} s_{k-1} \rangle} \| \tag{21}$$

$$
\begin{aligned}
&p_k := \left(I + A_{k-1}^T, \right) p_k^{(1)} + B_{k-1} p_k^{(0)}, \; B_{k-1} = \|b_j^i\| = \| \frac{s_{k-1}^i s_{k-1}^q \delta_{qj}}{\langle y_{k-1,} s_{k-1} \rangle} \|, \\
&B_{k-1}^T = B_{k-1}.
\end{aligned} \tag{22}
$$

The identity matrix **I** in (20) can be replaced with an approximate inverse Hessian. Formula (3) suggests that

$$D = \frac{\langle y_{k-1,} s_{k-1} \rangle}{\langle y_{k-1,} y_{k-1} \rangle} I \tag{23}$$

can be used as the initial diagonal approximation for the inverse Hessian matrix.

Now, once the one-step L-BFGS update has been represented in an operator form, we can adapt it to perform multi-step Hessian updates. If the values

s_{k-i} and $y_{k-i}=G_{k-i}-G_{k-i-1}$ for M previous steps are known, then (21-22) can be used to make M iterative "corrections" of the inverse Hessian matrix (and hence the new search direction) as follows:

$$\begin{aligned}
&p_k^{(0)} := -IG_{k-1} \\
&p_k^{0,0} := p_k^{(0)} \\
&p_k^{0,i} := \prod_{j=1}^{i} \left(I + A_{k-j}\right) p_k^{(0)},\ i = 1,\ldots M, \\
&p_k^{0,M+1} = D p_k^{0,M}, \\
&p_k^{1,i} := \left(I + A_{k-i}^T\right) p_k^{1,i+1} + B_{k-i} p_k^{0,i-1},\ i = M,\ldots,1, \\
&p_k := p_k^{1,1}.
\end{aligned} \tag{24}$$

Note that operators (21) and (22) – i.e., iterations (18) and (19) – are applied in a "stacking" FILO order: (18) are applied in the direct order while (19), in the reverse order.

Alternatively, the above process can be defined as a recursive application of "corrections" contributed by each of the M previous iterations to the approximate inverse Hessian matrix as follows.

Let H_k^i denote the approximation to the inverse Hessian matrix **H** obtained using information from i steps k-M+i-1,...,k-M. There holds then the following recurrent formula:

$$\tilde{H}_k^0 := D\ , \tag{25}$$

$$\tilde{H}_k^{i+1} := \left(I + A_{k-M+i}^T\right) \circ \tilde{H}_k^i \circ \left(I + A_{k-M+i}\right) + B_{k-M+i},\ i = 0,\ldots M-1\ , \tag{26}$$

$$\tilde{H} = \tilde{H}_k \approx \tilde{H}_k^M \tag{27}$$

where D is defined in (23) and $\tilde{H}$ is the approximate inverse Hessian matrix to be used instead of H^{-1} in formula (1) for solving the new search direction. Note that the one-step limited-memory quasi-Newton minimisation method (16) (see [1]) is a particular case of (25-27) with M=1 and $D \equiv I$.

This algorithm is used in L-BFGS subroutine implemented by the author of [2].

2. Implementtaion

The original implementation of L-BFGS algorithm can be obtained from Jorge Nocedal's web page: http://www.ece.northwestern.edu/~nocedal/L-BFGS.html. Our derivation of the L-BFGS algorithm closely follows the original implementation. At each iteration, if the next iterate doesn't reduce

the objective function value, a backtracking line search is undertaken. The line search algorithm implemented in the original code (see [3]) is based on “bracketing” procedure (MCSRCH subroutine) that requires both gradient and objective function re-evaluation at each backtracking step. In cases when the gradient evaluation is more expensive than that of the objective function value, it is recommended to replace the existing line search with e.g., bisection, secants or polynomial line search. For relatively “flat” objective functions either bisection or secants algorithm demonstrates good results, a polynomial line search method (see [9]) should be used if the objective function is asymptotically polynomial.

3. Further Reading.

[1] provides an undergraduate-level introduction to L-BFGS and comparison with other unconstrained optimization techniques. The limited-memory BFGS was first introduced in [2]. [3] established a relationship between BFGS (as a member of a broader family of methods that use Broyden class of updates) and variable-metric Conjugate Gradients solvers. [5] demonstrated that the updates used in BFGS and DFP algorithms are particular cases of the Wedderburn rank-reduction formula. [6,7] propose an alternative reduced-Hessian quasi-Newton method based on the same idea of approximating the Hessian from curvature information accumulated in a sequence of expanding subspaces as the one implemented by L-BFGS. [8] demonstrated a successful application of a modified L-BFGS solver with a polynomial line search in sesimic imaging.

References.

1. Gill, Philip E.; Murray, Walter; Wright. Practical Optimization.
2. Nocedal, J. "Updating quasi-Newton matrices with limited storage", Mathematics of Computation, 1980, Vol.24, No.151, pp. 773-782.
3. JORGE J. MORE, DAVID J. THUENTE. MCSRCH Line Search. ARGONNE NATIONAL LABORATORY. MINPACK PROJECT. JUNE 1983.
4. Nazareth, Larry. A Relationship Between the BFGS and Conjugate Gradient Algorithms and Its Implications for New Algorithms. SIAM J. Numer. Anal. Vol. 16, No. 5, October 1979.
5. Chu, Moody T.; Funderlic, Golub. A rank-one Reduction Formula and Its Applications to Matrix Factorizations. SIAM Review, Vol. 37, No. 4, pp. 512-530, December 1995.
6. Gill, Philip E.; Leonard. Reduced-Hessian Quasi-Newton Methods for Unconstrained Optimization. SIAM J. Optim., Vol.12, No.1, pp. 209-237, 2001.

7. Gill, Philip E.; Leonard. Limited-Memory Reduced-Hessian Methods for Large-Scale Unconstrained Optimization. SIAM J. Optim., Vol. 14, No. 2, pp. 380-401, 2003.
8. Maharramov, M. A; Albertin. Locally-Optimal Image-Difference Wave Equation Tomography. 77th Annual International Meeting, SEG, Expanded Abstracts, 3009-3013.
9. Press, William H.; Teukolsky, Vetterling, Flannery. Numerical Recipes 3rd Edition: the Art of Scientific Computing. Cambridge University Press; 3 edition, September 10, 2007.