A NOTE ON SPARSE QUASI-NEWTON METHODS

by

Mukund Thapa

TECHNICAL REPORT SOL 79-13

Sep 79

Research and reproduction of this report were partially supported by the Department of Energy Contract DE-AS03-76SF00034; the National Science Foundation Grants MCS76-20019-A01 and ENG77-06761; and U.S. Army Research Office Contract DAAG29-79-C-0110.

Reproduction in whole or in part is permitted for any purposes of the United States Government. This document has been approved for public release and sale; its distribution is unlimited.
A NOTE ON SPARSE QUASI-NEWTON METHODS

by

Mukund Thapa

1. Introduction

Consider the unconstrained minimization problem

$$\min_{x \in \mathbb{R}^n} f(x)$$  \hspace{1cm} (1.0)

An important class of algorithms used to solve the above problem is that of Quasi-Newton algorithms [1]. The idea of these methods is to maintain a positive definite symmetric matrix that approximates the Hessian at each iteration. Given the point $x_k$ in $\mathbb{R}^n$, the algorithm obtains a direction of descent, $p_k$, by solving the system of equations

$$B_k p_k = -g_k,$$  \hspace{1cm} (1.1)

where $B_k$ is the approximation to the Hessian at iteration $k$ and $g_k$ is the gradient at $x_k$. The next point, $x_{k+1}$, is then set to $x_k + \alpha_k p_k$ where $\alpha_k$ is chosen to cause a "sufficient" decrease in the function value at $x_k$. If the new point, $x_{k+1}$, satisfies some convergence criteria, the algorithm is terminated; else, the above procedure is repeated after obtaining $B_{k+1}$, a new approximation to the Hessian, as follows:
\[ B_{k+1} = B_k + U_k, \tag{1.2} \]

where \( U_k \) is a matrix chosen so that \( B_{k+1} \) is symmetric, positive definite and satisfies the Quasi-Newton condition (henceforth referred to as the QN condition),

\[ B_{k+1} s_k = y_k, \tag{1.3} \]

with

\[ s_k = x_{k+1} - x_k, \quad \text{and} \quad y_k = g_{k+1} - g_k. \]

There are a number of different ways of choosing \( U_k \) in equation (1.2). Three possible choices are shown below.

**BFGS Update:**

\[
U_k^{\text{BFGS}} = \frac{y_k y_k^T T}{s_k y_k} - \frac{B_k s_k s_k^T B_k}{s_k B_k s_k} \tag{1.4}
\]

**DFP Update:**

\[
U_k^{\text{DFP}} = \frac{(y_k - B_k s_k)y_k^T + y_k(y_k - B_k s_k)^T}{y_k s_k} \tag{1.5}
\]

\[
- \frac{(y_k - B_k s_k)^T s_k y_k y_k^T}{(y_k s_k)^2}
\]

**Self-Scaling BFGS:**

\[
B_{k+1} = \left( B_k - \frac{B_k s_k s_k^T B_k}{s_k B_k s_k} \right) \frac{s_k y_k}{s_k y_k^T} + \frac{y_k y_k^T}{y_k s_k} \tag{1.6}
\]
Quasi-Newton methods have been very successful in solving unconstrained and constrained problems of moderate size. The difficulty in applying these methods to large problems is that a symmetric $n \times n$ matrix (or a factorization) must be stored. However, many large problems have a sparse Hessian whose sparsity pattern is known (or can be determined) a priori. In this case, it seems possible to maintain a suitably sparse approximation to the Hessian; and, much current research is being directed to this objective (see [2],[3],[4],[5]).

Updates of the type given by equations (1.4), (1.5) and (1.6) cause total fill-in (that is, they do not preserve any zeros of the Hessian approximation). Obtaining updates that preserve sparsity and satisfy the Quasi-Newton condition (1.3) requires the solution of a linear system of equations whose coefficient matrix has the same sparsity pattern as the Hessian. This does not guarantee positive definiteness; and, in fact, it is not possible to always satisfy the Quasi-Newton condition (1.3) and preserve positive definiteness while maintaining sparsity (see [3], for example). Furthermore, sparse updates are usually of rank $n$; and, hence it is not possible to easily update the factorization of the Hessian approximation. This results in the additional work of refactorizing the Hessian at each iteration.

Shanno [3] showed how the sparse analog of any symmetric update $U_k$ can be derived by variational means. This paper shows how these sparse analogs can be derived as a simple extension of Toint's derivation of a sparse update.
2. Definitions and Notation

In the rest of the paper the subscript \( k \) will be dropped and the subscript \( k + 1 \) will be replaced by the superscript \( * \).

Let \( B \) be the sparse symmetric matrix representing the approximation to the Hessian at the start of iteration \( k \).

Let \( N = \{(i,j): B_{ij} = 0\} \) that is, \( N \) represents the sparsity pattern assumed at the start of the algorithm. Note that the sparsity pattern is assumed to be fixed and any additional zeros created are treated as non-zeros.

Let

\[
\bar{N} = \{(i,j): i, j = 1, \ldots, n\} \setminus N
\]

\[
= \{(i,j): B_{ij} \neq 0\}.
\]

For any symmetric matrix \( A \), define matrices \( A_N \) and \( A_{\bar{N}} \) as follows:

\[
(A_N)_{ij} = \begin{cases} A_{ij} & (i,j) \in N \\ 0 & (i,j) \in \bar{N} \end{cases}
\]

\[
(A_{\bar{N}})_{ij} = \begin{cases} 0 & (i,j) \in N \\ A_{ij} & (i,j) \in \bar{N} \end{cases}
\]
In words, $A_N$ is the matrix $A$ with zeros in the positions corresponding to the non-zeros of $B$; and $A_N^c$ is the matrix $A$ with zeros in the positions corresponding to the zeros of $B$. Then $A$ can be written as

$$A = A_N + A_N^c.$$ 

Define $D_i$ to be a diagonal matrix whose diagonal elements are 0 or 1 depending on the sparsity pattern of the $i$th row of $B$. That is,

$$(D_i)_{jj} = \begin{cases} 1 & \text{if } (i,j) \in \mathbf{N} \\ 0 & \text{if } (i,j) \notin \mathbf{N}. \end{cases}$$

Finally, define $s^i = D_i s$ for any vector $s$.

An example that illustrates the above definitions and notations now follows.

**Example:**

$$B = \begin{pmatrix} 10 & 1 & 0 & 0 \\ 1 & 20 & 2 & 0 \\ 0 & 2 & 30 & 3 \\ 0 & 0 & 3 & 40 \end{pmatrix}, \quad A = \begin{pmatrix} 25 & 3 & 4 & 5 \\ 3 & 35 & 2 & 3 \\ 4 & 2 & 45 & 6 \\ 5 & 3 & 6 & 55 \end{pmatrix}$$
Then,

\[
A_N = \begin{pmatrix}
0 & 0 & 4 & 5 \\
0 & 0 & 0 & 3 \\
4 & 0 & 0 & 0 \\
5 & 3 & 0 & 0
\end{pmatrix}
\quad \quad \quad
A_\infty = \begin{pmatrix}
25 & 3 & 0 & 0 \\
3 & 35 & 2 & 0 \\
0 & 2 & 45 & 6 \\
0 & 0 & 6 & 55
\end{pmatrix}
\]

\[
D_1 = \begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0
\end{pmatrix}
\quad \quad \quad
s = (1 \ 2 \ 3 \ 4)^T
\]

\[
s^1 = D_1 s = (1 \ 2 \ 0 \ 0)^T
\]
3. Toint's Method

Toint [2] proposed finding a matrix $E$ such that: $E$ is closest to $B$ in some sense; $B^* = B + E$ has the same sparsity pattern as $B$ (thus, $E$ has the same sparsity pattern as $B$); and $B^*$ satisfies the Quasi-Newton condition (1.3). Formally, the problem can be stated as:

$$\text{(P1)} \quad \text{Min} \quad \|E\|_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} E_{ij}^2,$$

where $\|\cdot\|_F$ is the Frobenius norm (3.0)

such that $E_s = y - B_s$ \hspace{1cm} (3.1)

$E_{ij} = 0 \quad (i,j) \in N$ \hspace{1cm} (3.2)

$E = E^T$. \hspace{1cm} (3.3)

By variational means, Toint obtained the following result

$$E_{ij} = \begin{cases} 0 & (i,j) \in N \\ \lambda_i s_j + \lambda_j s_i & (i,j) \in \bar{N} \end{cases} \hspace{1cm} (3.4)$$

where $\lambda = (\lambda_1, \ldots, \lambda_n)^T$ is the solution of the linear system

$$\varphi \lambda = y - B_s \quad (= E\lambda) \hspace{1cm} (3.5)$$
with \( \varphi \) defined by

\[
\varphi_{ij} = (s^i)_j (s^j)_i + 1 s^i_2 \delta_{ij} \quad \forall i,j
\] (3.6)

and \( \delta_{ij} \) is the Kronecker delta.

Note that \( \varphi \) is symmetric and has the same sparsity pattern as \( B \). Furthermore, \( \varphi \) is positive definite if and only if \( 1 s^i_2 > 0 \) for all \( i \) (see Toint [2]).

In matrix notation,

\[
E = \sum_{i=1}^{n} \lambda_i [e_i (s^i)^T + s^i e_i^T],
\] (3.7)

where \( e_i \) is the unit vector with 1 in the \( i^{th} \) position, and

\[
\varphi = \sum_{j=1}^{n} [(s^j)_i s^j_i + 1 s^j_2 e_j e_j^T].
\] (3.8)

Toint also obtained a generalization by minimizing \( \text{IWEW}_F \) where \( W \) is a diagonal matrix given by

\[
W = \begin{pmatrix}
t_1 & 0 \\
0 & t_2 \\
& \ddots \\
& & 0 & t_n
\end{pmatrix}
\] with \( t_i > 0 \) for \( i = 1, \ldots, n \). (3.9)
In this case the $\varphi$ and $E$ matrices are defined by

$$
\varphi_{ij} = \frac{(s^i)_j(s^j)_i}{t_1 t_j} + \frac{n}{t_1 t_k} \delta_{ij} \sum_{k=1}^{n} \frac{(s^k)_i^2}{t_k t_k}
$$

(3.10)

$$
E_{ij} = \frac{1}{t_1 t_j} \left[ \lambda_1 (s^i)_j + \lambda_j (s^j)_i \right]
$$

(3.11)
4. Sparse Analogs of Symmetric Updates

Shanno [3] showed how sparse analogs of symmetric updates (using BFGS as an example) could be derived by variational means. This section shows how these sparse analogs and those using self-scaling can be derived as a simple extension of Toint's results.

Let $B^* = \eta B + U$, where $U$ is symmetric but in general will not have the same sparsity pattern as $B$; $\eta$ is some scale factor; and $B^* s = y$. Then, by definition we have

$$B^*_N = U_N$$  \hspace{1cm} (4.0)

$$B^*_N = \eta B^*_N + U_N \quad \text{(Note that } B^*_N = B)$$  \hspace{1cm} (4.1)

Now $B^*_N$ has the same sparsity pattern as $B$ but does not satisfy the Quasi-Newton condition (1.3). Hence, we want to find a $\hat{B}^*$ given by

$$\hat{B} = B^*_N + E$$  \hspace{1cm} (4.2)

such that $\hat{B}^*$ is symmetric, has the same sparsity pattern as $B$ and satisfies the Quasi-Newton condition (1.3).

Next, note that

$$\hat{B}^* s = (B^*_N + E)s$$

$$= (B^* - B^*_N + E)s$$

$$= y - (B^*_N - E)s$$  

10
Clearly, $\hat{B}^* s = y$ if and only if $(B_N^* - E)s = 0$ or

$$Es = B_N^* s.$$  \hspace{1cm} (4.4)

Thus $\hat{B}^*$ is obtained by solving the following problem

(P2) \hspace{1cm} \text{Min} \quad \|E\|_F^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} E_{ij}^2 \hspace{1cm} (4.4)

such that \hspace{1cm} Es = B_N^* s \hspace{1cm} (4.5)

$$E_{ij} = 0 \hspace{1cm} (i,j) \in N \hspace{1cm} (4.6)$$

$$E = E^T.$$ \hspace{1cm} (4.7)

Problem P2 is almost the same as problem P1. The only difference is in equation (4.5) of P2 and equation (3.1) of P1. Thus the solution to problem P2 is:

$$E_{ij} = \begin{cases} 
0 & (i,j) \in N \\
\lambda_i s_j + \lambda_j s_i & (i,j) \in \overline{N} 
\end{cases}$$ \hspace{1cm} (4.8)

where $\lambda = (\lambda_1, \ldots, \lambda_n)$ is the solution of the linear system
\[ \varphi \lambda = B_N^* s \quad (= E_s) \quad (4.9) \]

with \( \varphi \) defined by (3.6) or (3.8).

If the norm to be minimized is chosen to be \( \| W E W \|_F^2 \) with \( W \) given by (3.9), then \( E \) and \( \varphi \) are given by (3.10) and (3.11) respectively.
5. A Note on Computations

Shanno [3] indicated that the computation of $B_N^s$ does not require the storage of the elements of $U_N$ but does require the computation of the elements of $U_N$ (that is, those elements of $U$ corresponding to the zero elements of $B$). However, the following result shows that the elements of $U_N$ need not be computed.

$$B_N^s = U_N s \quad \text{(from (4.0))}$$

$$= (U - U_N) s \quad \text{(by definition of $U_N$)}$$

$$= Us - U_N s$$

$$= (B^* - \eta B) s - U_N s \quad \text{(since $B^* = \eta B + U$)}$$

$$= y - \eta Bs - U_N s.$$ 

6. Conclusion

This paper has shown how the sparse analogs of Quasi-Newton updates can be derived as a simple extension of Toint's results; and, how the computation of $B_N^s$ can be done efficiently. At present, research on the computational and theoretical aspects of sparse Quasi-Newton algorithms is continuing, and further results will be described in a later technical report.
7. Acknowledgements

I would like to thank Dr. Margaret H. Wright and Dr. Philip E. Gill, without whose motivation, guidance and enthusiasm this research would not have been possible.

8. References


A Note on Sparse Quasi Newton Methods

Mukund Thapa

Operations Research Department - SOL
Stanford University
Stanford, CA 94305

Mathematics Division
U.S. Army Research Office, Box CM, Duke Station
Durham, NC 27706

September 1979

This document has been approved for public release and sale; its distribution is unlimited.

Sparse
Quasi Newton
Unconstrained
Computation

SEE ATTACHED
Shanno's derivation of the sparse analog of any symmetric Quasi-Newton update is obtained as a simple extension of Toint's derivation of a sparse update. Furthermore, it is shown how to compute an intermediate quantity efficiently.