A NEW PRECONDITIONED QUASI-NEWTON METHOD FOR OPTIMIZATION

Issam A.R. Moghrabi
Lebanese American University, Natural Science Division,
P.O.Box 13-5053, Beirut, Lebanon, email: imoghrbi@lau.edu.lb

(Received August 31st 1998; accepted December 29th 1999)

ABSTRACT

We address in this paper quasi-Newton methods for nonlinear unconstrained optimization, concentrating on the derivation of a new updating formulae to be used in the preconditioning of the search direction in Conjugate Gradient methods. Such formula is used in obtaining at each iteration a 'better' set up of the inverse Hessian [or the Hessian itself], than the current estimate, by performing an alternative simple update operation that uses available iteration data, such as gradient points and iterates.

Keywords: Quasi-Newton Methods, Unconstrained optimization, Multi-step methods.

INTRODUCTION

The problem addressed here is that of solving min $f(x)$, $x \in \Re^n$, where $f(x)$ is the objective function we seek to minimize. We denote the gradient and the Hessian of f by g and G, respectively. Let X denote a differentiable path $\{x(\tau)\}$ in $\Re^n$, where $\tau \in \Re$. Then a straightforward application of the chain rule to the vector function $g(x(\tau))$ shows that, at any point on X (corresponding to $\tau = \tau^*$, say), must satisfy

$$G(x(\tau))\ \frac{dx}{d\tau}(\tau^*) = \frac{dg}{d\tau}(\tau^*),$$

the Newton equation. The standard technique for exploiting this relation in the construction of quasi-Newton algorithms is to focus attention on the situation in which a new point x_{i+1} has been generated, by some means, from a previous estimate x_i of the desired minimum. The curve X is defined, for this case, to be the straight line which interpolates these 2 iterates:

$$x(\tau) \equiv x_i + \tau s_i ,$$

where $s_i \equiv x_{i+1} - x_i$.

Thus,

$$\frac{dx}{d\tau} = s_i , \forall \tau ,$$

Following a similar strategy for approximating $\frac{dg}{d\tau}$ at $\tau = 1$, we obtain

$$\frac{dg}{d\tau} \approx g(x_{i+1}) - g(x_1) \equiv y_i .$$

Upon substituting the quantities for $\frac{dg}{d\tau}$ and $\frac{dx}{d\tau}$ in the Newton equation and taking $\tau = 1$ (because we wish to derive a relation satisfied by $G(x_{i+1}) = G(x(1))$, in order to be able to approximate $G(x_{i+1})$), we obtain

$$G(x_{i+1})\, s_i \cong y_i .$$

Requiring that the Hessian be replaced by its approximant B_{i+1} and that the relation holds as an equality, we therefore have

$$B_{i+1}\, s_i = y_i ,$$

or, equivalently,

$$H_{i+1} y_i = s_i ,$$

for $H_{i+1} \equiv B_{i+1}^{-1}$, the so-called Secant Equation. Therefore, quasi-Newton methods for unconstrained optimization start with a current estimate of the minimum of the objective function, x_i, a corresponding gradient point, $g(x_i)$, and a current Hessian (or inverse Hessian) approximation B_i (or H_i). The next iterate is generated by computing a search direction p_i , by means of

$$p_i = -H_i g(x_i) ,$$

where a line search is then performed along the ray $\{x(t): x(t) = x_i + t\, p_i , t \geq 0 \}$ to determine a step $s_i = t_i\, p_i$. Once the next iterate has been determined the next step is to compute the next Hessian (or inverse Hessian) approximation, where, in general, it is computed using a recurrence of the form:

$$B_{i+1} = B_i + U(B_i, s_i, y_i) ,$$

with the precondition that the secant equation is satisfied. The best-known example of such a formula is the BFGS formula (Biggs, 1973; McDowell, 1983; Buckley, 1983; More, Garbow and Hillstrom, 1981)

$$B_{i+1} = B_i - \frac{B_i s_i s_i^T B_i}{s_i^T B_i s_i} + \frac{y_i y_i^T}{s_i^T y_i}.$$

A NEW UPDATING FORMULA BASED ON NON-QUADRATIC PROPERTIES

Biggs suggested that the matrix H_i used to approximate the inverse of the quadratic function Hessian, G^{-1}, should be updated at each iteration so as to satisfy

$$H_{i+1} y_i = \mu_i s_i,$$

where $y_i = g_{i+1} - g_i$ and $s_i = x_{i+1} - x_i$. This last equation is a modified Secant condition tailored in this case for general functions. The scalar $\mu_i > 0$ is intended to reflect the non-quadratic properties of $f(x)$. If $f(x)$ is quadratic then $\mu_i = 1$.

Now a value of μ_i is chosen at each iteration so that $\mu_i s^T_i y_i$ is, in some sense, a better estimate than $s^T_i y_i$ of the true directional second derivative $s^T_i G s_i$. Biggs proposed calculating μ_i by representing $f(x+\lambda p)$ by a cubic model, so that

$$\mu_i = s^T_i y_i / [4g^T_{i+1} s_i + 2g^T_i s_i - 6(f_{i+1} - f_i)]$$

However, we now propose, on the assumption that the searches used along p_i (or direction s_i) are exact, and since μ_i must be positive, that the above equation can be rewritten as follows:

$$\mu_i = \left| \frac{s^T_i y_i}{2g^T_i s_i - 6(f_{i+1} - f_i)} \right|.$$

In practice, this simplified form μ_i is more effective than the original.

We now propose a different rank-two update in the QN class that uses non-quadratic properties of the objective function.

To simplify notation we use '*' for 'i+1' and nothing for 'i '. We are interested in the following general form in deriving our formulae:

$$H^* = H + \alpha u u^T + \beta(u v^T + v u^T) + \theta v v^T, \quad (1)$$

which is clearly symmetric, just as the Hessian or its inverse are. We also want to stress that the resulting formula should satisfy the modified Secant Equation "($H^* y = \mu s$). Thus,

$$\alpha\,(u^T y)\,u + \beta\,[(v^T y)\,u + (u^T y)\,v] + (\theta\, v^T y)\,v = \eta\, s - H y \tag{2}$$

which leads to a system of two equations with three unknowns whose solution (with a degree of freedom) determines the scalars α, β and θ ; i.e.,

$$\alpha u^T y + \beta v^T y = 1 \tag{3}$$

$$\beta u^T y + \theta v^T y = 1 \tag{4}$$

We propose two choices for β: $\beta = \dfrac{-1}{y^T H y}$ and $\beta = 0$, respectively. From the first choice, it follows that

$$\alpha = \frac{1}{u^T y} + \frac{v^T y}{\left(y^T H y\right)\left(u^T y\right)} \tag{5}$$

and

$$\theta = \frac{1}{v^T y}\left[1 + \frac{u^T y}{y^T H y}\right] \tag{6}$$

We also choose u and v as follows

$$u = \lambda\eta\, s \tag{7}$$
$$v = (1 - \lambda)\,\eta\, s - H y$$

Upon substitution of the above equations in (3) we obtain (for the choice λ = 1/2) [other choices generate other formulae]

$$H^* = H + \frac{hss^T}{2s^T y} + \frac{vv^T}{v^T y} + \frac{v^T y}{2(y^T H y)(s^T y)}\, hss^T - \frac{hsv^T + hvs^T}{2y^T H y} + \frac{hs^T y}{2(y^T H y)(v^T y)}\, vv^T \tag{8}$$

It can easily be verified that the above formula satisfies the modified Secant Equation.

The following theorem states a sufficiency condition on the sequence H_0 , H_1, H_2, ……, so that the updating defined by (8) will generate a sequence of conjugate direction vectors when it is applied to a quadratic.

Theorem

Given the quadratic function, $f(x) = 1/2\ x^T Ax + b^T x + a$, an initial point x_0, and an initial matrix H_0 , suppose that the sequence of matrices H_0 , H_1, H_2, ……satisfies the following conditions :

1) $g^T_{i+1} H_{i+1} y_i = 0$, where $y_i = g_{i+1} - g_i = As_i$;

2) $H_{i+1} y = \mu_i s_i$, where μ_i is a scalar,

then, the search directions p_0 , p_1, p_2, …… p_{i+1} generated by any QN algorithm satisfy the following relation:

$$p^T_k\, Gp_j = 0,\ j \neq k.$$

Proof:

The proof of this theorem is similar to that in (McDowell, 1983). The matrix H_{i+1}, defined in (8), can easily be shown to satisfy the assumptions (1) and (2) above, with exact line search. Moreover, we already know that the update defined in (8) satisfies $H_{i+1}\, y_i = \mu_i\, s_i$; thus $g^T_{i+1} H_{i+1} y_i = \mu_i\, g^T_{i+1} s_i = 0$ (exact line searches).

Again, the initial Hessian approximation B_0 is discussed in (Dennis & Schnabel, 1983; Al-Baali & Khalfan, 1996) and they recommend $B_0 = \| f(x_0) \| I$ so that

$$H_0 = \| f(x_0)^{-1} \| I. \qquad (9)$$

The obvious problem with simply setting $B_0 = H_0 = I$ is that it takes no account of the scale of $f(x)$. For this reason $\| B_0 \|$ may differ from $\| \nabla^2 f(x_0) \|$ by many orders of magnitude, which can cause the algorithm to perform an excessive number of iterations.

THE PCG METHOD

The numerical results encourage attempts to improve the new algorithm AH. Different PCG methods have been defined and developed by varying the number of vectors $[s_i, y_i]$ stored corresponding to choice of BFGS updating formula as preconditioner matrix.

Now we can define a new PCG algorithm using the new update as the preconditioner matrix; the new algorithm should be promising for general non-quadratic properties and the experimental evidence. The search direction generated by the new PCG algorithm is as follows:

1- set $p_0 = - H_0 g_0$, where H_0 is defined in (5).

$$2\text{-}\ p_i = -H_i g_i + \frac{y^T_{i-1} H_i g_i}{p^T_{i-1} y_{i-1}} p_{i-1},$$

where H_i defined in (8), is not computed explicitly .

The implementation of this algorithm is very similar to Nazareth's algorithm (Buckley, 1983), but here a different preconditioner matrix is used. Now we could use his "variable step" new update approximation as preconditioner; i.e., the mth updated matrix will be fixed after m steps, m set by the user according to available storage.

The algorithm requires the storage of some vectors and scalars though the cycle. When a restart is incorporated, we propose that H_0 is the identity matrix updated by one iteration of the new update formula in order to offer the probability that the first iteration will approximate the Newton direction rather than the steepest descent direction which is often poor.

Finally, Biggs' idea of basing the estimate of s^TGs on a cubic model of f along s may also improve on the efficient Buckley and Liner algorithm (Buckley & LeNir, 1983), for non-quadratic objective functions. In their algorithm, the m-step Biggs' BFGS formula is obtained from the QN part and then used as the preconditioner matrix in the CG-part. This is open to practical investigation.

NUMERICAL RESULTS

Table 1 summarizes our experimental results for the new update defined in (8) with the initial Hessian approximation defined in (9) and compared with published results (Tassopoulos & Storey, 1984; Hu & Storey, 1991; Luksan, 1994) for the original BFGS and the DFP as the E04DDF NAG -routine. All three routines use the same basic line-search, namely cubic fitting, and the common stopping criterion is $||g(x_{i+1})|| < 1.E\text{-}6$.

Over the restricted set of functions the new algorithm is clearly best, overall, on the higher dimensional problems (30% less computation than BFGS) whereas at very low dimensions DFP beats the new algorithm, although there are large individual variations. The higher dimensions have the greater computational significance.

TABLE 1

Comparison of Function and Gradient evaluations (FES)

Test functions	N	new update	BFGS	E04DDF (DFP)
Rosen	2	230	192	153
Rosen	48	7314	11417	12495
Cubic	2	168	225	159
*Cubic	40	2133	6970	12679
Powell	4	440	275	200
Powell	48	5292	7380	8628
Wood	4	263	440	660
Wood	48	4111	11417	28420
Miele	4	455	505	245
Miele	48	13395	13671	6323

The test functions and results under BFGS and E04DDF are those published in (Biggs, 1973) by Tassopoulos and Storey .

* See Tassopoulos Ph.D. thesis.

APPENDIX

Where the form of the test function is not explicitly given, a full description of the function may be found in (More, Garbow and Hillstrom, 1981), unless an alternative reference is provided. The notation **F** indicates that the initial estimate of the minimum x_0 is specified by a formula given in (More, Garbow and Hillstrom, 1981). x_j denotes the jth component of the starting-vector. Finally, the notation "([$\alpha,\beta,\ldots\omega$]*)" implies that the vector [$\alpha,\beta,\ldots,\omega$] is to be repeated as many times as necessary in order to give a starting vector of the required dimension.

(1) Discrete boundary value problem

(a) **F**; (b) $x_0 = (1,2,3,..,n)^T$.

(2) Extended Powell singular function:

(a) $x_0 = ([3,-1,0,1]^*)^T$; (b) x_0=$([2,2,3,-1]^*)^T$.

(3) Variably-Dimensioned function

(a) **F**; (b) $x_0 = ([10,5,-5,-10]^*)^T$.

(4) Penalty function I

(a) $x_0 = (1,2,..,n)^T$; (b) $x_0 = ([5,-5]^*)^T$.

(5) Modified Trigonometric function

$$f(x) \equiv \sum_{i=1}^{n} \{n - \sum_{j=1}^{n} \cos x_j + i(1-\cos x_i) - \sin x_i + e^{x_i} - 1\}^2 .$$

(a) **F**; (b) $x_0 = ([-1,1]^*)^T$.

(6) Penalty function II

(a) $x_0 = ([0.5]^*)^T$; (b) $x_0 = ([-0.2,-0.1]^*)^T$.

(7) Extended Rosenbrock function

(a) $x_0 = ([-1.2,1]^*)^T$; (b) $x_0 = ([-120,100]^*)^T$.

(8) Extended Wood function

(a) $x_0 = ([-300,-100]^*)^T$; (b) $x_0 = ([-30,10]^*)^T$.

(9) Watson's function

(a) $x_0 = ([-1,1]^*)^T$; (b) $x_0 = ([1,0]^*)^T$.

(10) Broyden Tridiagonal function

$x_0 = (-1,-1,..,-1)^T$.

(11) Oren and Spedicato power function

$x_0 = (1,1,..,1)^T$.

ACKNOWLEDGMENTS

The work done here was inspired by a similar approach followed in (Al Mahamad and Hutchinson, 1989).

REFERENCES

Al-Baali, M., and Khalfan, H.F., 1996. *Experimental Study of Globally and Super-Linearly convergent Self-Scaling QN Methods,* Technical Report, Dept. of Math and Computer Science, United Arab Emirates University.

Al Mahamad, N. and Hutchinson, 1989. *A New QN and PCG Algorithms based on Non-quadratic Properties*, Report 89.21, Univ. of Lyons.

Biggs, M.C. 1973. A note on minimization algorithms which make use of the non-quadratic properties of the objective function*, J. Inst. Math. Appl*. 12: 337-338.

Buckley , A. and LeNir, *A*. 1983. QN-like variable storage conjugate gradients*, Math. Prog.,* 27: 155-175.

Dennis, J.P. and Schnabel, B. 1983. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Englewood Cliffs, New Jersey.

Hu, Y.F., and Storey, C., 1991 *On Optimally and near-Optimally Conditioned Quasi-Newton Updates,* Report A141, Dept. of Math, University of Loughbourough.

J.J. More, B.S., Garbow and K.E. Hillstrom, 1981. Testing unconstrained optimization software, *ACM Transac. Math. Software*, 7: 17-41.

Luksan, L., 1994. Computational Experience with Variable-Metric Updates, *J. of Optimization Theory and Applications,* 83: 27-47.

McDowell ,D.G. 1983. Generalized conjugate directions for unconstrained func tion minimization, *JOTA*, 41: 523-532.

Tassopoulos, A. and Storey , C.A. 1984. A variable-metric method using a non-quadratic model, *JOTA*, 43: 383-393.