

The Diagonal Update for Unconstrained Optimization

Saad Shakir Mahmood

Ali Ibrahim Mansour,

&

Haydir Ali Hassan

Department of Mathematics / College of Education
Al-Mustansiriya University

Received
05 / 10 / 2010

Accepted
04 / 05 / 2011

الخلاصة

في هذا البحث ، قمنا بتقديم صنف جديد من طرق كواسي - نيوتن غير المقيدة حيث قمنا بتحديث عناصر القطر لمصفوفة هيسين ، وجعلنا العناصر التغير قطرية مساوية للصفر دائما ، وبذلك قد حافظنا على عدد الصفار الموجودة في مصفوفة هيسين القطرية ولذلك فان هذا التحديث سمي بالتحديث القطري.

ABSTRACT

In this research we introduced a new update of the Hessian matrix or we updating only the diagonal elements of Hessian matrix, and make the non-diagonal elements always equal to zero and in this case we can preserve the sparse property so called the Diagonal Update.

1. Introduction

Assume a priori that we know that some sparsity is present in the second derivative matrix of the objective function. In this case we would like to exploit this sparsity pattern so that the successive approximations of Hessian matrix, exhibit the same sparsity as the matrix they are approximating. This has two practical consequences:

1. the storage needed for the current approximation of Hessian matrix is drastically reduced, allowing much larger problems to be treated;
2. the rate of convergence of the method is improved because the approximations of Hessian matrix are better.

2. The diagonal Update

The problem is how to update the Hessian matrix with minimal cost flops, so by the quasi-Newton condition we have:

The Diagonal Update for Unconstrained Optimization.

$$B_{k+1}s_k = y_k \quad (2.1)$$

$$(B_k + C_k)s_k = y_k$$

Where C_k , the current correcting matrix, so we have

$$B_k s_k + C_k s_k = y_k$$

And hence $C_k s_k = y_k - B_k s_k$

If we set $r_k = y_k - B_k s_k$, then we have the following system

$$C_k s_k = r_k$$

By using matrix form, the last system can be write as follows

$$\begin{bmatrix} c_{11} & c_{12} & \cdot & \cdot & \cdot & c_{1n} \\ c_{21} & c_{22} & \cdot & \cdot & \cdot & c_{2n} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ c_{n1} & c_{n2} & \cdot & \cdot & \cdot & c_{nn} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \cdot \\ \cdot \\ \cdot \\ s_n \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ \cdot \\ \cdot \\ \cdot \\ r_n \end{bmatrix} \quad (2.2)$$

If we set $c_{ij} = 0$, for all $i \neq j$ in (2.2), then we have the following system

$$\begin{bmatrix} c_{11} & 0 & \cdot & \cdot & \cdot & 0 \\ 0 & c_{22} & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & \cdot & \cdot & \cdot & c_{nn} \end{bmatrix} \begin{bmatrix} s_1 \\ s_2 \\ \cdot \\ \cdot \\ \cdot \\ s_n \end{bmatrix} = \begin{bmatrix} r_1 \\ r_2 \\ \cdot \\ \cdot \\ \cdot \\ r_n \end{bmatrix} \quad (2.3)$$

In this case we have the following system of equations:

$$\begin{aligned} c_{11}s_1 &= r_1 \\ c_{22}s_2 &= r_2 \\ \cdot & \\ \cdot & \\ \cdot & \\ c_{nn}s_n &= r_n \end{aligned} \quad (2.4)$$

The solution of (2.4) is as follows:

$$\begin{aligned} c_{11} &= \frac{r_1}{s_1} \\ c_{22} &= \frac{r_2}{s_2} \\ \cdot & \\ \cdot & \\ \cdot & \\ c_{nn} &= \frac{r_n}{s_n} \end{aligned} \quad (2.5)$$

And in the general form the solution of (2.4), is as follows:

$$c_{ii} = \frac{r_i}{s_i}, \quad \text{for } i = 1, 2, \dots, n \quad (2.6)$$

3. Convergence Analysis

We discuss the Descent Property of the proposed algorithm. Let B_k be symmetric. and positive definite matrix

Since

$$B_k P_k = -g_k \quad \dots\dots\dots (3.1)$$

Then

$$P_k = -B_k^{-1} g_k \quad \dots\dots\dots(3.2)$$

Multiply both widest of (3.2) by g_k^T

$$g_k^T P_k = -g_k^T B_k^{-1} g_k$$

Since B_k is symmetric . and positive

$$\text{Then } -g_k^T B_k^{-1} g_k < 0$$

Hence P_k is descent direction for k .

1. The update is symmetric.
2. The update satisfies the quasi-Newton condition.
3. The Hessian matrix sparse.
4. In (2.1), if exist s_i equal to zero for some i , then the update is failed.
5. The convergent of the method is numerically by solving some standard problems.

4. The Diagonal update algorithm

1. choose an initial estimate x_0 , eps, set $k = 0$.
2. choose an initial estimate $B_k = I$.
3. Compute $\nabla f(x_k)$, and then evaluate the convergence of x_k . Either terminate, or proceed to the next step.
4. solve the linear equation $B_k p_k = -\nabla f(x_k)$.
5. Do line search in the direction of p_k to get λ_k , such that $f(x_k + \lambda_k p_k) < f(x_k)$.
6. set $x_{k+1} = x_k + \lambda_k p_k$.
7. compute $\nabla f(x_k)$, set $y_k = \nabla f(x_{k+1}) - \nabla f(x_k)$, and $s_k = \lambda_k p_k$.
8. set $r_k = y_k - B_k s_k$.

9. compute $C_k(i, j) = \begin{cases} r_k(i) & i = j \\ s_k(i) & i \neq j \\ 0 & i \neq j \end{cases}$
10. compute $B_{k+1} = B_k + C_k$
11. If $\|(\nabla f(x_{k+1}))^T \nabla f(x_{k+1})\| < \text{eps}$ then stop, else, set $k = k + 1$, and go to 3.

5. NUMERICAL RESULTS

In this test, solution of the following test problems has been attempted. The reason for their selection is that the problems appear to have been used in standard problems in most the literature. These functions represent a result of application in the branch of technology and industry.

In order to reduce the risk that unrepresentative results might be obtained on a particular problem by the choice of a particularly fortunate or unfortunate starting point, each problem was solved for a variety of starting point. The results are presented in Table 1. Convergence was assumed when a point was located where $\|\nabla f\|_2 < 10^{-8}$ in all cases. The entries in each column for the various methods denote the total flops required achieving convergence. All the starting points were chosen to comply with the requirement that the BFGS method, as implemented in our program, should converge from this point. Finally, an asterisk following the initial point indicates that method converged to a local minimum from that initial point.

Table 1: (Test Of the functions)

No	n	Starting points	Flops BFGS	Flops DU
1	10	(-1.2,1)	556441	364811
1	20	(-3.635,5.6213)	851094	598133
2	10	(-1.2,1)	1915344	1002218
2	10	(6.39,-0.221)	1824001	997742
3	30	(-1.2,1)	9035145	2843481
3	50	(9,12)	11856634	3687645
4	50	(30,30)	9834557	3132380
4	70	(1,1)	13109965	5367122
5	50	(-10,-0.5)	6897301	2151312
5	70	(100,100)	8244350	3797406
6	50	(0,1)	5500671	3291804
6	70	(0.01,5)	6312116	3718962
7	50	(3,-0.5)	5224813	2742658
7	70	(10,10,10)	6160365	3114127

From the above table clear that the total flops of diagonal update less than the total flops of the BFGS update.

6. CONCLUSIONS

Newton's method requires the evolution of the Hessian matrix $\nabla^2 f$ at each iteration, which can be very expensive. At each step Newton's method requires the solution of a linear system $\nabla^2 f(x_k)p_k = -\nabla f(x_k)$, to obtain p_k . When the initial estimate x_0 is poor, $\nabla^2 f(x_0)$ may be non positive definite. In this case the Hessian matrix is singular or nearly singular, accurate solution of the linear system may be very difficult or impossible. The development of Quasi-Newton methods lies in the attempt to eliminate the defects of Newton methods as follows:

1. Q-N methods do not evaluate the Hessian matrix; instead, they build approximations to the Hessian matrix by use of an update or correcting process.
2. Many of the update formulas produce positive definite matrices; provided that the initial matrix is positive definite and those other simple conditions (for example $s^T y > 0$) hold. This property is important because it ensures that the next search direction satisfies the Q-N condition.
3. If one of the B-version formulas is utilized, it is necessary (by analog of the Newton's method) to solve the following linear system of equations $B_k p_k = -\nabla f(x_k)$ to find the direction p_k .

In the general case, solving the above system requires $O(n^3)$ operations and this defect can be eliminated by updating the cholesky factors of the matrix B_k . If the cholesky factors B_k are available, the above system can be solved in $O(n^2)$ operations.

Direct updating of the LU factors of Hessian matrix is undertaken by computing the direction p_k from the equation $B_k p_k = -\nabla f(x_k)$, without computing the inverse of Hessian matrix at very iteration. The diagonal update, updating the Hessian matrix, preserves the symmetric property and satisfies the Q-N condition.

The diagonal update proposed as a new update based on the sparsty, and this update can solve an optimization problem with minimum cost flops.

Appendix

- 1) Rosenbrock's function

$$f(x) = \sum_{i=1}^{\frac{n}{2}} 100(x_{2i} - x_{2i-1}^2)^2 + (1 - x_{2i-1})^2$$

- 2) Dixon function [6]

$$f(x) = (1 - x_1)^2 + (1 - x_{10})^2 + \sum_{i=2}^9 (x_i^2 - x_{i+1})^2$$

- 3) Discrete integral equation function

$$f(x) = \sum_{i=1}^n \left[x_i + \frac{h}{2} \left((1-t_i) \sum_{j=1}^i t_j (x_j + t_j + 1)^3 + t_i \sum_{j=i+1}^n (1-t_j) (x_j + t_j + 1)^3 \right) \right]^2$$

$$\text{where } h = \frac{1}{n+1}, t_i = ih$$

The Diagonal Update for Unconstrained Optimization.

4) Penalty function I

$$f(x) = \sum_{i=1}^{n+1} f_i^2, \quad \text{Where } f_i = \sqrt{10^{-5}}(x_i - 1), \quad 1 \leq i \leq n \quad \text{and} \quad f_{n+1} = \sum_{j=1}^n (x_j^2) - \frac{1}{4}$$

5) Variably-Dimensioned function

$$f(x) = \sum_{i=1}^n (x_i - 1)^2 + f^2 + f^4$$

$$\text{Where } f = \sum_{j=1}^n j(x_j - 1)$$

6) The Chained Wood's function

$$f(x) = \sum_{i=1}^n [(100(x_{i+1} - x_1^2)^2 + (1 - x_i)^2 + 90(x_{i+3} - x_{i+2}^2)^2 + (1 - x_{i+2})^2 + 10(x_{i+1} + x_{i+3} - 2)^2 + 0.1(x_{i+1} - x_{i+3})^2)^2]$$

where $j = \{1, 3, 5, \dots, n-3\}$

7) Watson function

$$f(x) = \sum_{i=1}^n f_i^2(x)$$

$$\text{Where } f_i(x) = \sum_{j=2}^n (j-1)x_j t_i^{j-2} - \left(\sum_{j=1}^n x_j t_i^{j-1} \right)^2 - 1 \quad \text{and} \quad t_i = \frac{i}{29}$$

References

- 1) Chu, M. T, Funderlic, R. E and Gene H. G, (1997), "Rank Modifications Of Semi-definite Matrices With Applications To Secant Updates", <http://www.cs-tr.cs.cornell.edu>
- 2) Dennis, J. E and Schnabel, R. B, (1983), "Numerical Methods for Unconstrained Optimization and Nonlinear Equations", Prentice Hall, New Jersey.
- 3) Ford, J. A and Moghrabi, I. A, (1995), "Multi-Step Quasi-Newton Optimization Algorithms Which Utilize Curvature Information", <http://www.essex.ac.uk>.
- 4) Ford, J. A and Moghrabi, I. A, (1996), "Further Development Of Minimum Curvature Multi-Step Quasi-Newton Methods For Unconstrained Optimization", <http://www.essex.ac.uk>.
- 5) Ghandari, R. A, (1989), "On the Use of Function Values to Improve Quasi-Newton Methods", Ph.D. Thesis, University of Essex, Colchester.
- 6) Saad S., 2007, "Hart- Saad Update For Unconstrained optimization", Magistra Journal No. 47, TH. XV, pp.86-95, Widya Dharma University, Indonesia.
- 7) Xu, C. and Ma, X, (1996), "Factorized Updating Methods For Nonlinear Least Squares", Sea. Bull. Math, Vol., 20, no.3.