Practical Optimization Algorithms and Applications

Chapter VI: Quasi-Newton Methods

Lingfeng NIU

Research Center on Fictitious Economy & Data Science,
University of Chinese Academy of Sciences

niulf@ucas.ac.cn

Introduction

The main disadvantage of Newton's method, even when modified to ensure global convergence, is that the user must supply formulae from which the second derivative matrix can be evaluated. We want to derive methods closely related to Newton's method when only first derivatives are available.

- Finite Difference Newton Methods
- Quasi-Newton Methods

The quasi-Newton method is like Newton's method, except that the second derivative matrix is approximated by a symmetric positive definite matrix, which is corrected or updated from iteration to iteration.

Derivation

The quadratic model of the objective function at the current iterate x_k is

$$m_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p. \tag{1}$$

Here B_k is an $n \times n$ symmetric matrix.

When B_k is positive definite, the minimizer p_k of this convex quadratic model, which we can write explicitly as

$$p_k = -B_k^{-1} \nabla f_k, \tag{2}$$

is used as the search direction, and the new iterate is

$$x_{k+1} = x_k + \alpha_k p_k, \tag{3}$$

where the step length α_k is chosen to satisfy the Wolfe conditions.

Derivation

Instead of computing B_k afresh at every iteration, quasi-Newton Method proposed to update it in a simple manner to account for the curvature measured during the most recent step.

Derivation

Instead of computing B_k afresh at every iteration, quasi-Newton Method proposed to update it in a simple manner to account for the curvature measured during the most recent step.

Suppose that we have generated a new iterate x_{k+1} and wish to construct a new quadratic model, of the form

$$m_{k+1}(p) = f_{k+1} + \nabla f_{k+1}^T p + \frac{1}{2} p^T B_{k+1} p. \tag{4}$$

What requirements should we impose on B_{k+1}, based on the knowledge we have gained during the latest step?

Derivation - Secant Equation

The gradient of m_{k+1} should match the gradient of the objective function f at the latest two iterates x_k and x_{k+1},

$$\nabla m_{k+1}(-\alpha_k p_k) = \nabla f_{k+1} - \alpha_k B_{k+1} p_k = \nabla f_k.$$

Define

$$s_k = x_{k+1} - x_k = \alpha_k p_k, \qquad y_k = \nabla f_{k+1} - \nabla f_k, \tag{5}$$

we get

$$B_{k+1} s_k = y_k. \tag{6}$$

We refer this formula as the **secant equation**.

Derivation - Secant Equation

The gradient of m_{k+1} should match the gradient of the objective function f at the latest two iterates x_k and x_{k+1},

$$\nabla m_{k+1}(-\alpha_k p_k) = \nabla f_{k+1} - \alpha_k B_{k+1} p_k = \nabla f_k.$$

Define

$$s_k = x_{k+1} - x_k = \alpha_k p_k, \qquad y_k = \nabla f_{k+1} - \nabla f_k, \tag{5}$$

we get

$$B_{k+1} s_k = y_k. \tag{6}$$

We refer this formula as the **secant equation**.

Various possible ways exist of achieving this condition. In what follows the aims is to find something which is simple and involves only a small amount of computation, and yet which is effective.

Derivation - SR1

Perhaps the simplest possibility if to have

$$B_{k+1} = B_k + \sigma v v^T,$$

in which a symmetric rank one matrix $\sigma v v^T$ is added into B_k. Here, σ is either $+1$ or -1.
In order to make B_{k+1} satisfying the secant equation $y_k = B_{k+1} s_k$, we have

$$y_k = B_k s_k + [\sigma v^T s_k] v.$$

Since the term in brackets is a scalar, we deduce that v must be a multiple of $y_k - B_k s_k$, that is, $v = \delta(y_k - B_k s_k)$ for some scalar δ. By substituting this form of v into above equation, we obtain

$$(y_k - B_k s_k) = \sigma \delta^2 [s^T (y_k - B_k s_k)](y_k - B_k s_k). \tag{7}$$

Derivation - SR1

By reasoning in terms of B_k, we see that there are three cases:

- If $(y_k - B_k s_k)^T s_k \neq 0$, it is clear that equation (7) is satisfied if and only if we choose the parameters δ and σ to be

$$\sigma = \text{sign}[s^T(y_k - B_k s_k)], \qquad \delta = \pm[s_k^T(y_k - B_k s_k)]^{-1/2}.$$

  And there is a unique rank-one updating formula secant equation is given by

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}. \tag{8}$$

  satisfying the secant equation;

- If $y_k = B_k s_k$, then the only updating formula satisfying the secant equation is simply $B_{k+1} = B_k$;
- If $y_k \neq B_k s_k$ and $(y_k - B_k s_k)^T s_k = 0$, then there is no symmetric rank-one updating formula satisfying the secant equation.

Properties of SR1 Updating

Theorem

Choose any positive definite matrix as the initial matrix B_0 and use the line search method with symmetric rank-one updating formula (8) solve the following quadratic minimization problems:

$$\min_{x \in \Re^n} \frac{1}{2} x^T A x + b^T x + c.$$

If the updating formula (8) is well defined at each iteration, and if $s_0, s_1, \cdots, s_{n-1}$ are independent, then symmetric rank-one updating formula has the following hereditary property

$$H_{k+1} s_i = y_i, \qquad i = 0, \cdots, k$$

and the method terminates at most $n+1$ iterates with $B_n = A$.

Strategy to Prevent SR1 from Breaking Down

It has been observed in practice that SR1 performs well simply by skipping the update if the denominator is small. More specifically, the SR1 update is applied only if

$$|s_k^T(y_k - B_k s_k)| \geq r\|s_k\|\|y_k - B_k s_k\|, \tag{9}$$

where $r \in (0, 1)$ is a small number, say $r \in 10^{-8}$. If (9) does not hold, we set $B_{k+1} = B_k$. Most implementations of the SR1 method use a skipping rule of this kind.

In practice, $s_k^T(y_k - B_k s_k) \approx 0$ occurs infrequently, since it requires certain vectors to be aligned in a specific way. When it does occur, skipping the update appears to have no negative effects on the iteration, since the skipping condition implies that $s_k^T \bar{G} s_k \approx s_k^T B_k s_k$, where $\bar{G}$ is the average Hessian over the last step-meaning that the curvature of B_k along s_k is already correct.

Sherman-Morrison-Woodbury Formula

If the square nonsingular matrix A undergoes a rank-one update to become

$$\bar{A} = A + ab^T,$$

where $a, b \in \Re^n$, then if $\bar{A}$ is nonsingular, we have

$$\bar{A}^{-1} = A^{-1} - \frac{A^{-1}ab^TA^{-1}}{1 + b^TA^{-1}a}.$$

Furthermore, let U and V be matrices in $\Re^{n\times p}$ for some p between 1 and n. If we define

$$\hat{A} = A + UV^T,$$

then $\hat{A}$ is nonsingular if and only if $(I + V^TA^{-1}U)$ is nonsingular, and in this case we have

$$\hat{A}^{-1} = A^{-1} - A^{-1}U(I + V^TA^{-1}U)^{-1}V^TA^{-1}.$$

Derivation - SR1

The inverse of B_k, which we denote by $H_k = B_k^{-1}$, is useful in the implementation of the method, since it allows the search direction to be calculated by means of a simple matrix-vector multiplication. Using Sherman-Morrison-Woodbury formula, we can derive the following expression for the update of the inverse Hessian approximation H_k that corresponds to the update of B_k:

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k}. \tag{10}$$

We can see that SR1 method is **self-dual**, i.e. the inverse formula H_k can be obtained simply by replacing B, s and y by H, y and s, respectively.

Drawback of SR1 Updating

It is easy to see that even if B_k is positive definite, B_{k+1} may not have this property. (The same is, of course, true of H_k.) This suggests that

- rank-one updating does not provide enough freedom to develop a matrix with all the desired characteristics, and that a rank-two correction is required;
- the SR1 method is not suitable for the line search method.

Derivation - Curvature Equation

Given the displacement s_k and the change of gradients y_k, the secant equation requires that the symmetric positive definite matrix B_{k+1} map s_k into y_k. This will be possible only if s_k, y_k satisfy the **curvature condition**

$$s_k^T y_k > 0. \tag{11}$$

In fact, above condition is guaranteed to hold if we impose the Wolfe or strong Wolfe conditions on the line search.

Derivation - Rank Two Updates

A more flexible formula is obtained by following the correction to be of rank two, and such a formula can always be written

$$H_{k+1} = H_k + auu^T + bvv^T.$$

The secant equation must be satisfied, giving

$$s_k = H_{k+1}y_k = H_k y_k + auu^T y_k + bvv^T y_k.$$

Now u and v are no longer determined uniquely.

When the curvature condition is satisfied, the secant equation always has a solution B_{k+1}. In fact, it admits an infinite number of solutions, since there are $n(n+1)/2$ degrees of freedom in a symmetric matrix, and the secant equation represents only n conditions.

Derivation - DFP

An obvious choice is to try $u = s_k$ and $v = H_k y_k$. Then $au^T y_k = 1$ and $bv^T y_k = -1$ determine a and b. Thus

$$H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k}. \tag{12}$$

This formula is called the DFP updating formula, since it is the one originally proposed by Davidon in 1959, and subsequently studied, implemented, and popularized by Fletcher and Powell. This is the first quasi-Newton updating formula to be discovered.

Using Sherman-Morrison-Woodbury formula, we can derive the following expression for the update of the Hessian approximation B_{k+1} that corresponds to the DFP update of H_{k+1}

$$B_{k+1} = (I - \gamma_k y_k s_k^T) B_k (I - \gamma_k y_k s_k^T) + \gamma_k y_k y_k^T, \tag{13}$$

with $\gamma_k = 1/y_k^T s_k$.

Properties of DFP

Theorem

If $s_k^T y_k > 0$ for all k, then the DFP formula preserves positive definite matrices H_k.

The condition $s_k^T y_k > 0$ is realistic and always can be achieved. Notice that

$$s_k^T y_k = s_k^T \nabla f_{k+1} - s_k^T \nabla f_k$$

and $s_k^T \nabla f_k = \alpha_k p_k^T \nabla f_k < 0$, we have

- if an exact line search is used, $s_k^T \nabla f_{k+1} = 0$;
- if an inexact line search with the Wolfe condition is used,

$$s_k^T \nabla f_{k+1} \geq c_2 s_k^T \nabla f_k.$$

Properties of DFP

DFP has a number of important properties as follows:

- for quadratic functions (with exact line searches)
  - terminates in at most n iterations with B_n equal to the real Hessian matrix;
  - hereditary property: previous secant equations are preserved;
  - generates conjugate directions, and conjugate gradients when $H_0 = I$;
- for general functions
  - preserves positive definite matrices - hence the descent property holds;
  - requires $3n^2 + O(n)$ multiplications per iterations;
  - superlinear order of convergence;
  - global sonvergence for strictly convex functions (with exact line searches).

Derivation -BFGS

Another improtant formula was suggested by Broyden, Flectcher, Goldfab, and Shanno, and is know as the BGFS formula

$$H_{k+1} = (I - \rho_k s_k y_k^T) H_k (I - \rho_k s_k y_k^T) + \rho_k s_k s_k^T, \tag{14}$$

with $\gamma_k = 1/y_k^T s_k$. And

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}. \tag{15}$$

This resembles the DFP formula but with the interchanges $B \leftrightarrow H$ and $y \leftrightarrow s$ have been made. Formulae related in this way are said to be complementary or dual.

Properties of BFGS

- On the theoretical side, all the properties we mentioned for the DFP method also hold for the BFGS method.
  - In addition, global convergence of the BFGS method with inexact line searches which satisfy the Wolfe condtion has been proved, a result which has not yet been shown for the DFP method.
- The BFGS formula has been found to work well in practice, perhaps even better than the DFP formula.
  - The DFP formula is in a less satisfactory light when low accuracy line searches are carried out, and currently the DFP method is no longer preferred.

Algorithm 1: BFGS method

Given starting point x_0, convergence tolerance $\epsilon > 0$,
inverse Hessian approximation H_0;
$k \leftarrow 0$;
while $\|\nabla f_k\| > \epsilon$;
Compute search direction

$$p_k = -H_k \nabla f_k;$$

Set $x_{k+1} = x_k + \alpha_k p_k$ where α_k is computed from a line search procedure to satisfy the Wolfe conditions
Define $s_k = x_{k+1} - x_k$ and $y_k = \nabla f_{k+1} - \nabla f_k$;
Compute H_{k+1} by means of (BFGS);
$k \leftarrow k + 1$;
end (while)

Global Convergence of the BFGS Method

Theorem

Let B_0 be any symmetric positive definite initial matrix, and let x_0 be a starting point for which
(1) The objective function f is twice continuously differentiable.
(2) The level set $\mathcal{L} = \{x \in \Re^n | f(x) \leq (x_0)\}$ is convex, and there exist positive constants m and M such that

$$m\|z\|^2 \leq z^T \nabla^2 f(x) z \leq M\|z\|^2$$

for all $z \in \Re^n$ and $x \in \mathcal{L}$.
Then the sequence $\{x_k\}$ generated by Algorithm 1 with $\epsilon = 0$ converges to the minimizer x^* of f.

Superlinear Convergence of the BFGS Method

Theorem

Suppose that f is twice continuously differentiable and that the iterates generated by the BFGS algorithm converge to a minimizer x^* at which the Hessian matrix $\nabla^2 f$ is Lipschitz continuous at x^* that is,

$$\|\nabla^2 f(x) - \nabla^2 f(x^*)\| \leq L\|x - x^*\|,$$

for all x near x^*, where L is a positive constant. Suppose also that

$$\sum_{k=1}^{\infty} \|x_k - x^*\| < \infty. \tag{16}$$

holds. Then x_k converges to x^* at a superlinear rate.

Numerical Experiment Results

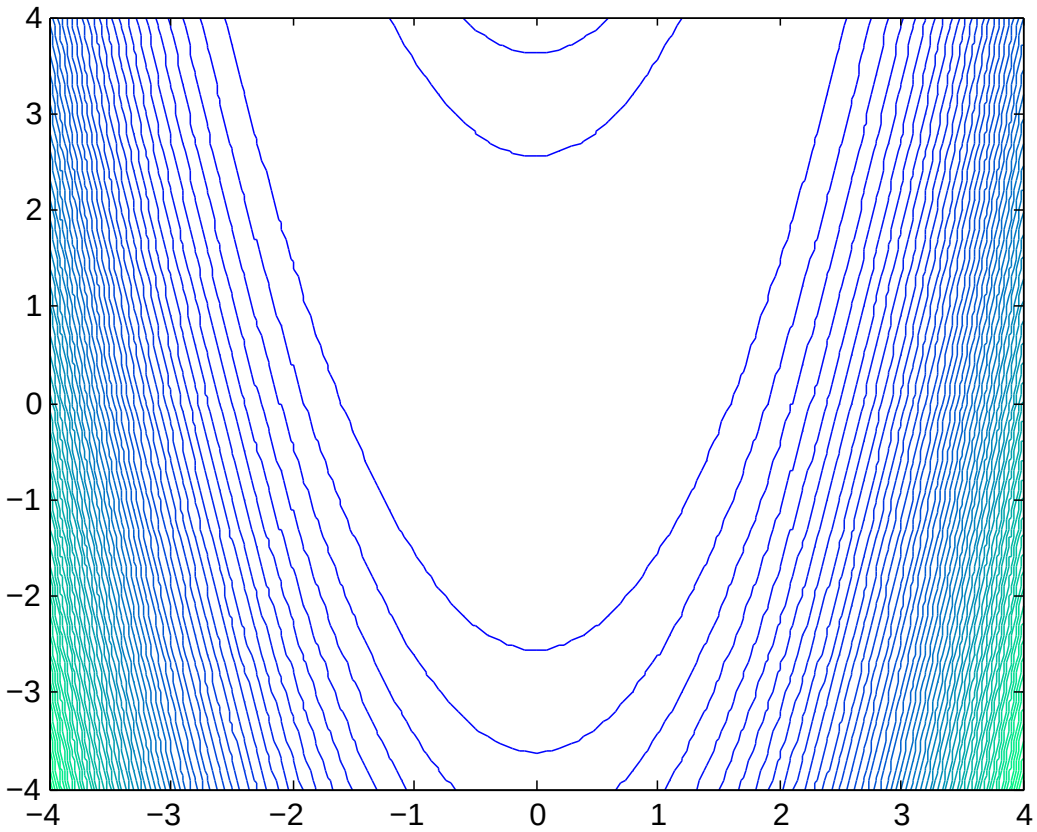
Rosenbrock's function

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$

Numerical Experiment Results

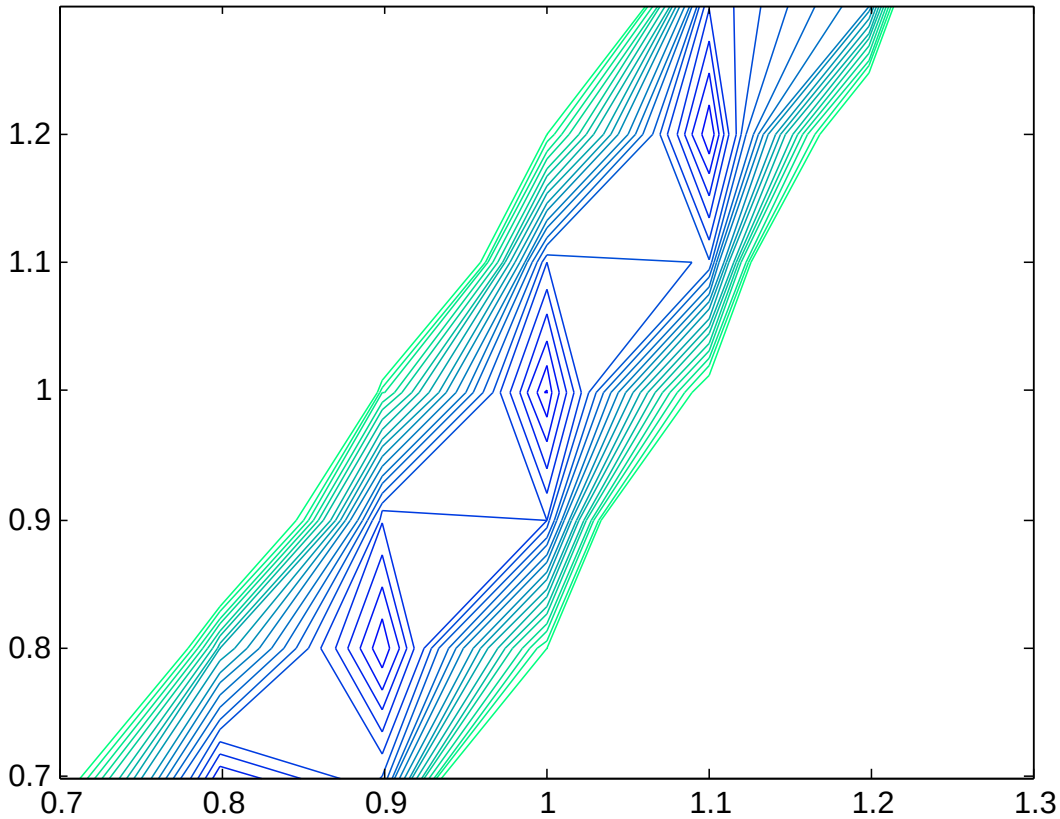
Rosenbrock's function

$$f(x) = 100(x_2 - x_1^2)^2 + (1 - x_1)^2$$



Numerical Experiment Results

The optimal solution is $x^* = (1, 1)^T$, and the corresponding optimal function value is $f(x^*) = 0$.



Numerical Experiment Results

Use the steepest descent, BFGS, and an inexact Newton method on this problem, respectively. The Wolfe conditions were imposed on the step length in all three methods. From the starting point $(1.2, 1)$, the steepest descent method required 5264 iterations, whereas BFGS and Newton took only 34 and 21 iterations, respectively to reduce the gradient norm to 10^{-5}.

steep. desc.	BFGS	Newton
1.827e-04	1.70e-03	3.48e-02
1.826e-04	1.17e-03	1.44e-02
1.824e-04	1.34e-04	1.82e-04
1.823e-04	1.01e-06	1.17e-08

The value of $\|x_k - x^*\|$ in last few iterations of the steepest descent, BFGS, and an inexact Newton method on Rosenbrock's function

Derivation - The Broyden Family

The formulae which has been introduced so far by no means exhaust all the possibilities. In fact one-parameter family of rank two formulae can be generated by taking

$$B_{k+1} = (1 - \phi_k)B_k^{BFGS} + \phi_k B_k^{DFP}. \tag{17}$$

This family includes of course the BFGS and DFP formulae, and also the rank one formula.

Broyden family can be specified by the following general formula:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y^T s_k} + \phi_k (s_k^T B_k s_k) v_k v_k^T, \tag{18}$$

where $v_k = [\frac{y_k}{y_k^T s_k} - \frac{B_k s_k}{s_k^T B_k s_k}]$. The last term in (18) is a rank-one correction. As we decrease ϕ_k, this matrix eventually becomes singular and then indefinite. A little computation shows that B_{k+1} is singular when ϕ_k has the value

$$\phi_k^c = \frac{1}{1 - \mu_k}, \tag{19}$$

where $\mu_k = \frac{(y_k^T B_k^{-1} y_k)(s_k^T B_k s_k)}{(y_k^T s_k)^2}$.

Properties of the Broyden Family

The Broyden family is important in that many of the properties of the DFP and BFGS formulae are common to the whole family.

- All members of the Broyden family satisfy the secant equation;
- By Cauchy-Schwarz inequality, $\mu_k \geq 1$ and therefore $\phi_C \leq 0$. Hence, if the initial Hessian approximation B_0 is symmetric and positive definite, and if $s_k^T y_k > 0$ and $\phi_k > \phi_k^c$ for each k, then all the matrices B_k generated by Broyden's formula (18) remain symmetric and positive definite.
- When the line search is exact, all methods in the Broyden class with $\phi_k \geq \phi_k^c$ generate the same sequence of iterates. However, the results would appear to be mainly of theoretical interest, since the inexact line searches used in practical implementations of Broyden-class methods (and all other quasi-Newton methods) cause their performance to differ markedly. Nevertheless, this type of analysis guided most of the development of quasi-Newton methods.

Thanks for your attention!