Chapter 6: Quasi-Newton Methods

Outline

1. The DFP Method
2. The BFGS Method
3. The SR1 Method
4. The Broyden Class
5. Convergence Analysis
6. The BB Method

Outline

1. The DFP Method
2. The BFGS Method
3. The SR1 Method
4. The Broyden Class
5. Convergence Analysis
6. The BB Method

Outline of Quasi-Newton Methods

- Originally proposed by W.C. Davidon in the mid 1950s
- Require only the gradient of the objective function to be supplied at each iterate. By measuring the changes in gradients, they construct a model of the objective function that is good enough to produce superlinear convergence.
- The improvement over steepest descent is dramatic, especially on difficult problems.
- Moreover, since second derivatives are not required, quasi-Newton methods are sometimes more efficient than Newton's method.

Derivation

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

$$m_k(p) = f_k + \nabla f_k^T p + \frac{1}{2} p^T B_k p. \tag{1.1}$$

  Here B_k is an $n \times n$ symmetric positive definite matrix that will be revised or updated at every iteration.

- The minimizer p_k of this convex quadratic model

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

  called as quasi-Newton step.

- The new iterate is

$$x_{k+1} = x_k + \alpha_k p_k, \tag{1.2}$$

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

Derivation

- Instead of computing B_k afresh at every iteration, W.C. Davidon 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

$$m_{k+1}(p) = f_{k+1} + \nabla f_{k+1}^T p + \frac{1}{2} p^T B_{k+1} p. \tag{1.3}$$

- 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}, then

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

Define

$$s_k = x_{k+1} - x_k = \alpha_k p_k, \qquad y_k = \nabla f_{k+1} - \nabla f_k, \tag{1.4}$$

we get

$$B_{k+1} s_k = y_k. \tag{1.5}$$

We refer this formula as the *secant equation*.

Derivation - Curvature Condition

- 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 and y_k satisfy the *curvature condition*

$$s_k^T y_k > 0. \tag{1.6}$$

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

Derivation

- If 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.
- To determine B_{k+1} uniquely, then, we impose the additional condition that among all symmetric matrices satisfying the secant equation, B_{k+1} is, in some sense, closest to the current matrix B_k. In other words, we solve the problem

$$\min_{B} \quad \|B - B_k\| \tag{1.7a}$$

$$s.t. \quad B = B^T, \qquad Bs_k = y_k, \tag{1.7b}$$

  where s_k and y_k satisfy the curvature condition (1.6) and B_k is symmetric and positive definite.

- Many matrix norms can be used in (1.7a), and each norm gives rise to a different quasi-Newton method.

Derivation

A norm that allows easy solution of the minimization problem (1.7a), and that gives rise to a scale-invariant optimization method, is the weighted Frobenius norm

$$\|A\|_W \equiv \| W^{1/2} A W^{1/2} \|_F. \tag{1.8}$$

The weight W can be chosen as any matrix satisfying the relation $Wy_k = s_k$.

Derivation

- If $W = \bar{G}_k^{-1}$ where $\bar{G}_k$ is the average Hessian defined by

$$\bar{G}_k = \int_0^1 \nabla^2 f(x_k + \tau\alpha_k p_k)\, d\tau.$$

- It follows from Taylor's theorem that $y_k = \bar{G}_k \alpha_k p_k = \bar{G}_k s_k$.
- With this weighting matrix and this norm, the unique solution of (1.7a) is

$$B_{k+1} = (I - \gamma_k s_k y_k^T) B_k (I - \gamma_k y_k s_k^T) + \gamma_k y_k y_k^T, \tag{1.9}$$

  with $\gamma_k = 1/y_k^T s_k$.
- Called as DFP updating formula
- It was originally proposed by Davidon in 1959, and subsequently studied, implemented, and popularized by Fletcher and Powell.

Derivation

- The computation of $-B_k^{-1} g_k$ needs to solve a system of linear equations
- The inverse of B_k, which we denote by $H_k = B_k^{-1}$, is useful in the implementation of the method, since it only involves a simple matrix-vector multiplication to calculate $-H_k g_k$.
- Using the Sherman-Morrison-Woodbury formula:

$$(A + ab^T)^{-1} = A^{-1} - \frac{A^{-1} ab^T A^{-1}}{1 + b^T A^{-1} a},$$

we can derive the following expression for the update of the inverse Hessian approximation H_k that corresponds to the DFP update of B_k

$$H_{k+1} = H_k - \frac{H_k y_k y_k^T H_k}{y_k^T H_k y_k} + \frac{s_k s_k^T}{y_k^T s_k}. \qquad (1.10)$$

Derivation

B_{k+1} and H_{k+1} are the rank-2 modifications of B_k and H_k, respectively. This is the fundamental idea of quasi-Newton updating:

Instead of recomputing the iteration matrices from scratch at every iteration, we apply a simple modification that combines the most recently observed information about the objective function with the existing knowledge embedded in our current Hessian approximation.

Outline

1. The DFP Method
2. The BFGS Method
3. The SR1 Method
4. The Broyden Class
5. Convergence Analysis
6. The BB Method

Derivation

- Considered to be the most effective of all quasi-Newton updating formulae
- Instead of imposing conditions on the Hessian approximations B_k, we impose similar conditions on their inverses H_k.
- The updated approximation H_{k+1} must be symmetric and positive definite, and must satisfy the secant equation (1.5), now written as

$$H_{k+1} y_k = s_k. \tag{2.1}$$

Derivation

The condition of closeness to H_k is now specified by

$$\min_{H} \quad \|H - H_k\| \tag{2.2}$$

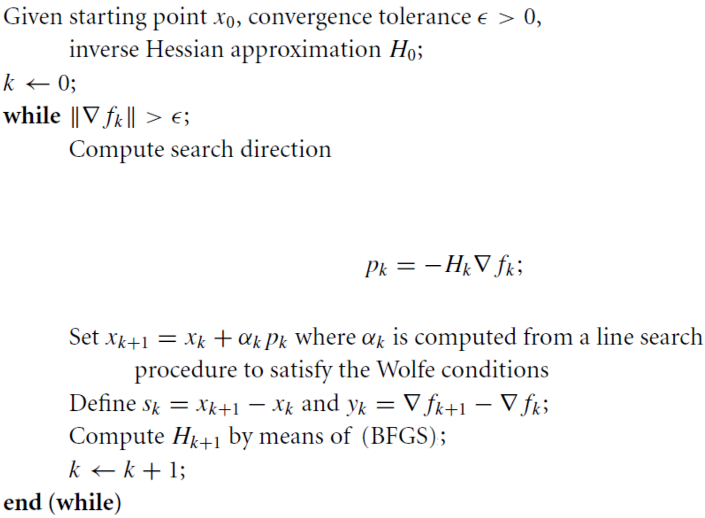
$$s.t. \quad H = H^T, \quad Hy_k = s_k. \tag{2.3}$$

The norm is again the weighted Frobenius norm described above, where the weight matrix W is now any matrix satisfying $Ws_k = y_k$. Assume again that W is given by the average Hessian $\bar{G}_k$. The unique solution H_{k+1} to (2.2) is given by

$$H_{k+1} = (I - \rho_k y_k s_k^T) H_k (I - \rho_k s_k y_k^T) + \rho_k s_k s_k^T, \tag{2.4}$$

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

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 $\alpha_k$ 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)
```

Computational complexity

- Each iteration can be performed at a cost of $O(n^2)$ arithmetic operations (plus the cost of function and gradient evaluations); there are no $O(n^3)$ operations such as linear system solves or matrix-matrix operations.
- The algorithm is robust, and its rate of convergence is superlinear, which is fast enough for most practical purposes. Even though Newton's method converges more rapidly (that is, quadratically), its cost per iteration is higher because it requires the solution of a linear system.
- A more important advantage for BFGS is, of course, that it does not require calculation of second derivatives.

Property of the BFGS Method

- In BFGS algorithm, H_{k+1} will be positive definite whenever H_k is positive definite and the curvature condition $s_k^T y_k > 0$;
- The BFGS quasi-Newton updating formula is invariant to changes in the variables;
- The BFGS formula has very effective *self-correcting properties*: If the matrix H_k incorrectly estimates the curvature in the objective function, and if this bad estimate slows down the iteration, then the Hessian approximation will tend to correct itself within a few steps.
- The self correcting properties of BFGS hold only when an adequate line search is performed. In particular, the Wolfe line search conditions ensure that the gradients are sampled at points that allow the quadratic model to capture appropriate curvature information.

Update on B_k

- We can derive a version of the BFGS algorithm that works with the Hessian approximation B_k rather than H_k. The update formula for B_k is obtained by simply applying the Sherman-Morrison-Woodbury formula to (2.4) to obtain

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k}. \tag{2.5}$$

- It is interesting to note that the DFP and BFGS updating formulae are duals of each other, in the sense that one can be obtained from the other by the interchanges $s \leftrightarrow y$, $B \leftrightarrow H$.
- Less expensive implementations of this variant are possible by updating Cholesky factors of B_k

Implementation - Initial Approximation H_0

- We can use specific information about the problem, for instance by setting it to the inverse of an approximate Hessian calculated by finite differences at x_0.
- The initial matrix H_0 often is set to some multiple βI of the identity, but there is no good general strategy for choosing β.
  - If β is "too large" so that the first step $p_0 = -\beta g_0$ is too long, many function evaluations may be required to find a suitable value for the step length α_0.
  - Some software asks the user to prescribe a value δ for the norm of the first step, and then set $H_0 = \delta \|g_0\|^{-1} I$ to achieve this norm.

Implementation - Initial Approximation H_0

A heuristic that is often quite effective is to scale the starting matrix after the first step has been computed but before the first BFGS update is performed. We change the provisional value $H_0 = I$ by setting

$$H_0 \leftarrow \frac{y_k^T s_k}{y_k^T y_k} I.$$

before applying the update to obtain H_1. This formula attempts to make the size of H_0 similar to that of $[\nabla^2 f(x_0)]^{-1}$ by approximating an eigenvalue of $[\nabla^2 f(x_0)]^{-1}$.

Implementation - Line Search

- The line search, which should satisfy either the Wolfe conditions or the strong Wolfe conditions, should always try the step length $\alpha_k = 1$ first, because this step length will eventually always be accepted (under certain conditions), thereby producing superlinear convergence of the overall algorithm.
- Computational observations strongly suggest that it is more economical, in terms of function evaluations, to perform a fairly inaccurate line search. The values $c_1 = 10^{-4}$ and $c_2 = 0.9$ are commonly used.
- The performance of the BFGS method can degrade if the line search is not based on the Wolfe conditions.

Outline

1. The DFP Method
2. The BFGS Method
3. The SR1 Method
4. The Broyden Class
5. Convergence Analysis
6. The BB Method

Derivation

- In the BFGS and DFP updating formulae, the updated matrix B_{k+1} (or H_{k+1}) differs from its predecessor B_k (or H_k) by a rank-2 matrix.
- In fact, as we now show, there is a simpler rank-1 update that maintains symmetry of the matrix and allows it to satisfy the secant equation.
- Unlike the rank-two update formulae, this *symmetric-rank-1*, or *SR1*, update does not guarantee that the updated matrix maintains positive definiteness. Good numerical results have been obtained with algorithms based on SR1.

Derivation

The symmetric rank-1 update has the general form

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

where σ is either $+1$ or -1, and σ and v are chosen so that B_{k+1} satisfies the secant equation $y_k = B_{k+1} s_k$. By substituting into this equation, we obtain

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

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

$$(y_k - B_k s_k) = \sigma \delta^2 [s^T (y_k - B_k s_k)](y_k - B_k s_k),$$

and it is clear that this equation is satisfied if (and only if) we choose the parameters δ and σ to be

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

Derivation

The only symmetric rank-1 updating formula that satisfies the secant equation is given by

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)(y_k - B_k s_k)^T}{(y_k - B_k s_k)^T s_k}. \tag{3.1}$$

By applying the Sherman-Morrison formula, we obtain the corresponding update formula for the inverse Hessian approximation H_k:

$$H_{k+1} = H_k + \frac{(s_k - H_k y_k)(s_k - H_k y_k)^T}{(s_k - H_k y_k)^T y_k}. \tag{3.2}$$

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.

Properties 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 observation was considered a major drawback in the early days of nonlinear optimization when only line search iterations were used.
- However, with the advent of trust-region methods, the SR1 updating formula has proved to be quite useful, and its ability to generate indefinite Hessian approximations can actually be regarded as one of its chief advantages.
- The main drawback of SR1 updating is that the denominator in (3.1) or (3.2) can vanish.

Properties of SR1 Updating

By reasoning in terms of B_k (similar arguments can be applied to H_k), we see that there are three cases:

- If $(y_k - B_k s_k)^T s_k \neq 0$, then the arguments above show that there is a unique rank-one updating formula 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

The last case clouds an otherwise simple and elegant derivation, and suggests that numerical instabilities and even breakdown of the method can occur. It 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**. This reasoning leads us back to the BFGS method, in which positive definiteness (and thus nonsingularity) of all Hessian approximations is guaranteed.

Why Interested in SR1 Updating?

- A simple safeguard seems to adequately prevent the breakdown of the method and the occurrence of numerical instabilities.
- The matrices generated by the SR1 formula tend to be good approximations to the true Hessian matrix often better than the BFGS approximations.
- In quasi-Newton methods for constrained problems, or in methods for partially separable functions, it may not be possible to impose the curvature condition $y_k^T s_k > 0$, and thus BFGS updating is not recommended. Indeed, in these two settings, indefinite Hessian approximations are desirable insofar as they reflect indefiniteness in the true Hessian.

Properties of SR1 Updating

We now introduce a strategy to prevent the SR1 method 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

$$|s_k^T(y_k - B_k s_k)| \geq r\|s_k\|\|y_k - B_k s_k\|, \tag{3.3}$$

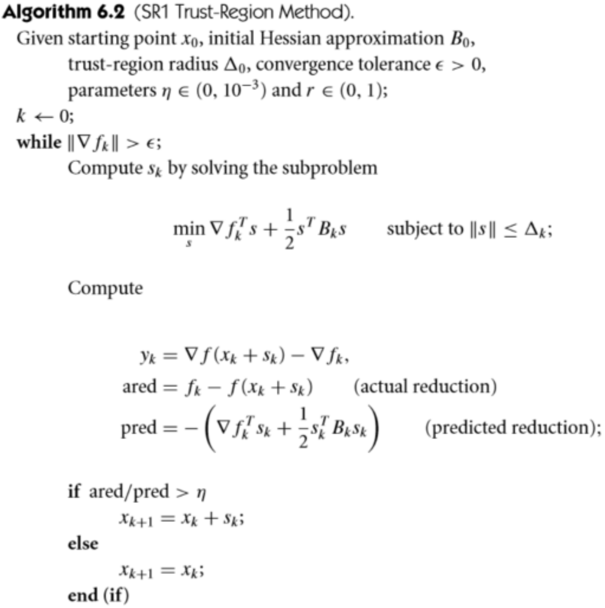
where $r \in (0,1)$ is a small number, say $r \in 10^{-8}$. If (3.3) does not hold, we set $B_{k+1} = B_k$. Most implementations of the SR1 method use a skipping rule of this kind.

Properties of SR1 Updating

Why do we advocate skipping of updates for the SR1 method, when in the previous section we discouraged this strategy in the case of BFGS?

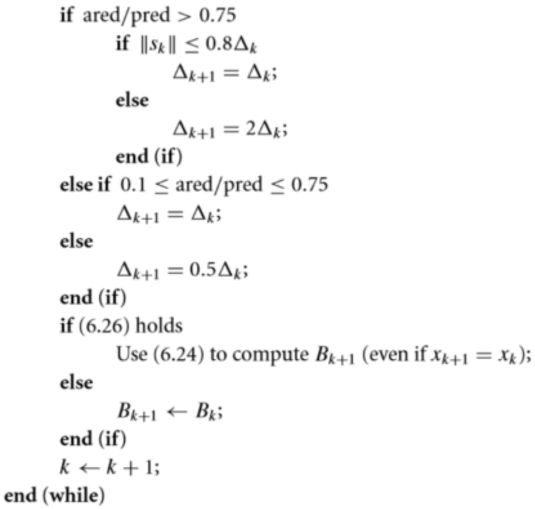
- $s_k^T(y_k - B_k s_k) \approx 0$ occurs infrequently, since it requires certain vectors to be aligned in a specific way. When it does occur, skipping the update appears to have no negative effects on the iteration, since the skipping condition implies that $s_k^T \bar{G} s_k \approx s_k^T B_k s_k$, where $\bar{G}$ is the average Hessian over the last step-meaning that the curvature of B_k along s_k is already correct.
- $s_k^T y_k \geq 0$ required for BFGS updating may easily fail if the line search does not impose the Wolfe conditions (e.g., if the step is not long enough), and therefore skipping the BFGS update can occur often and can degrade the quality of the Hessian approximation.

Algorithm 2: SR1 Trust-Region Method



```
Algorithm 6.2 (SR1 Trust-Region Method).
  Given starting point $x_0$, initial Hessian approximation $B_0$,
        trust-region radius $\Delta_0$, convergence tolerance $\epsilon > 0$,
        parameters $\eta \in (0, 10^{-3})$ and $r \in (0, 1)$;
  $k \leftarrow 0$;
  while $\|\nabla f_k\| > \epsilon$;
        Compute $s_k$ by solving the subproblem

            $\min_s \nabla f_k^T s + \frac{1}{2} s^T B_k s \qquad \text{subject to } \|s\| \le \Delta_k;$

        Compute

              $y_k = \nabla f(x_k + s_k) - \nabla f_k,$
            $\text{ared} = f_k - f(x_k + s_k) \qquad \text{(actual reduction)}$
            $\text{pred} = -\left(\nabla f_k^T s_k + \frac{1}{2} s_k^T B_k s_k\right) \qquad \text{(predicted reduction);}$

        if ared/pred $> \eta$
              $x_{k+1} = x_k + s_k$;
        else
              $x_{k+1} = x_k$;
        end (if)
```

Algorithm 2: SR1 Trust-Region Method (Cont.)



```
        if ared/pred $> 0.75$
                if $\|s_k\| \le 0.8\Delta_k$
                        $\Delta_{k+1} = \Delta_k$;
                else
                        $\Delta_{k+1} = 2\Delta_k$;
                end (if)
        else if $0.1 \le$ ared/pred $\le 0.75$
                $\Delta_{k+1} = \Delta_k$;
        else
                $\Delta_{k+1} = 0.5\Delta_k$;
        end (if)
        if (6.26) holds
                Use (6.24) to compute $B_{k+1}$ (even if $x_{k+1} = x_k$);
        else
                $B_{k+1} \leftarrow B_k$;
        end (if)
        $k \leftarrow k + 1$;
end (while)
```

Nonsymmetric Rank-1 Method

The rank-1 update has the form:

$$B_{k+1} = B_k + uv^T.$$

To guarantee the secant equations to hold for B_{k+1}, we obtain

$$y_k = B_k s_k + (v^T s_k)u.$$

So if $v^T s_k \neq 0$, $u = \frac{y_k - B_k s_k}{v^T s_k}$. Then

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

By setting $v = s_k$, we finally obtain the nonsymmetric rank-1 update formula

$$B_{k+1} = B_k + \frac{(y_k - B_k s_k)s_k^T}{s_k^T s_k}. \tag{3.4}$$

Nonsymmetric Rank-1 Method

- Nonsymmetric rank-1 update formula has much applications in solving nonlinear equations.
- B_{k+1} generated through (3.4) solves the following problem

$$\min \|B - B_k\|_2 \quad \text{s.t.} \quad Bs_k = y_k.$$

Outline

1. The DFP Method
2. The BFGS Method
3. The SR1 Method
4. The Broyden Class
5. Convergence Analysis
6. The BB Method

Derivation

Broyden class is a family of updates specified by the following general formula:

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k}{s_k^T B_k s_k} + \frac{y_k y_k^T}{y_k^T s_k} + \phi_k (s_k^T B_k s_k) v_k v_k^T, \tag{4.1}$$

where ϕ_k is a scalar parameter and

$$v_k = \left[\frac{y_k}{y_k^T s_k} - \frac{B_k s_k}{s_k^T B_k s_k} \right].$$

Derivation

The BFGS and DFP methods are members of the Broyden class-we recover BFGS by setting $\phi_k = 0$ and DFP by setting $\phi_k = 1$ in (4.1). We can therefore rewrite (4.1) as a "linear combination" of these two methods, that is,

$$B_{k+1} = (1 - \phi_k) B_{k+1}^{BFGS} + \phi_k B_{k+1}^{DFP}. \tag{4.2}$$

This relationship indicates that

- all members of the Broyden class satisfy the secant equation;
- members with $0 \leq \phi_k \leq 1$(*restricted Broyden class*) preserve positive definiteness of the Hessian approximations when $s_k^T y_k > 0$.

Properties of the Broyden Class

The last term in (4.1) 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

$$\phi_k^c = \frac{1}{1-\mu_k}, \tag{4.3}$$

where

$$\mu_k = \frac{(y_k^T B_k^{-1} y_k)(s_k^T B_k s_k)}{(y_k^T s_k)^2}. \tag{4.4}$$

By applying the Cauchy-Schwarz inequality to (4.4) we see that $\mu_k \geq 1$ and therefore $\phi_k^c \leq 0$. 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 (4.1) remain symmetric and positive definite.

Properties of the Broyden Class

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. This result applies to general nonlinear functions and is based on the observation that when all the line searches are exact, the directions generated by Broyden-class methods differ only in their lengths. The line searches identify the same minima along the chosen search direction, though the values of the line search parameter may differ because of the different scaling.

Properties of the Broyden Class

The Broyden class has several remarkable properties when applied with exact line searches to quadratic functions.

Theorem 1

Suppose that a method in the Broyden class is applied to a strongly convex quadratic function $f : \Re^n \to \Re$, where x_0 is the starting point and B_0 is any symmetric and positive definite matrix. Assume that α_k is the exact step length and the chosen value of ϕ_k did not produce a singular update matrix. Then the following statements are true.

Properties of the Broyden Class

Theorem 2

(i) The iterates converge to the solution in at most n iterations.

(ii) The secant equation is satisfied for all previous search directions, that is,

$$B_k s_j = y_j, \qquad j = k-1, \cdots, 1.$$

(iii) If the starting matrix is $B_0 = I$, then the iterates are identical to those generated by the CG method. In particular, the search directions are conjugate, that is,

$$s_i^T A s_j = 0 \text{ for } i \neq j,$$

where A is the Hessian of the quadratic function.

(iv) If the starting matrix B_0 is not the identity matrix, then the Broyden-class method is identical to the preconditioned CG method that uses B_0 as preconditioner.

(v) If n iterations are performed, we have $B_{n+1} = A$.

Properties of the Broyden Class

The results in the above theorem 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.

Outline

1. The DFP Method
2. The BFGS Method
3. The SR1 Method
4. The Broyden Class
5. Convergence Analysis
6. The BB Method

Convergence Analysis

- The fact that the Hessian approximations evolve by means of updating formulas makes the analysis of quasi-Newton methods much more complex.
- Although the BFGS and SR1 methods are known to be remarkably robust in practice, we will not be able to establish truly global convergence results for general nonlinear objective functions. That is, we cannot prove that the iterates of these quasi-Newton methods approach a stationary point of the problem from any starting point and any (suitable) initial Hessian approximation. In fact, it is not yet known if the algorithms enjoy such properties.
- In our global convergence analysis we will either assume that the objective function is convex.
- There are well known local, superlinear convergence results that are true under reasonable assumptions. These results apply to general nonlinear function.

Global Convergence of the BFGS Method

Theorem 3

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 converges to the minimizer x^* of f.

Global Convergence of the BFGS Method

- The above theorem has been generalized to the entire restricted Broyden class, except for the DFP method;
- An extension of the analysis shows that the rate of convergence of the iterates is linear. In particular, we can show that the sequence $\|x_k - x^*\|$ converges to zero rapidly enough that

$$\sum_{k=1}^{\infty} \|x_k - x^*\| < \infty. \tag{5.1}$$

Superlinear Convergence of the BFGS Method

Theorem 4

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

Convergence Analysis of the SR1 Method

Theorem 5

Suppose that the iterates x_k are generated by Algorithm 2. Suppose also that the following conditions hold:

(c1) The sequence of iterates does not terminate, but remains in a closed, bounded, convex set $\mathcal{D}$, on which the function f is twice continuously differentiable, and in which f has a unique stationary point x^*;

(c2) the Hessian $\nabla^2 f(x^*)$ is positive definite, and $\nabla^2 f(x)$ is Lipschitz continuous in a neighborhood of x^*;

(c3) the sequence of matrices $\{B_k\}$ is bounded in norm;

(c4) condition (3.3) holds at every iteration, where r is some constant in $(0,1)$. Then $\lim_{k\to\infty} x_k = x^*$, and we have that

$$\lim_{k\to\infty} \frac{\|x_{k+n+1} - x^*\|}{\|x_k - x^*\|} = 0.$$

Outline

1. The DFP Method
2. The BFGS Method
3. The SR1 Method
4. The Broyden Class
5. Convergence Analysis
6. The BB Method

The BB Method

- The secant equation:

$$B_k s_{k-1} = y_{k-1}.$$

- Barzilai and Borwein (BB) method:

$$x_{k+1} = x_k - D_k^{-1} g_k$$

  with $D_k \equiv \alpha_k I$.

- Solving

$$\min_{D=\alpha I} \|D s_{k-1} - y_{k-1}\|_2$$

  yields

$$\alpha_k^{BB} = \frac{s_{k-1}^T y_{k-1}}{s_{k-1}^T s_{k-1}}.$$

- For convex function $f(x) = \frac{1}{2} x^T H x + b^T x$,

$$\alpha_k^{BB} = \frac{g_{k-1}^T H g_{k-1}}{g_{k-1}^T g_{k-1}} = \frac{1}{\alpha_{k-1}^{SD}}.$$

The BB Method

- Nonmonotone property
- For strongly convex quadratic function, it has R-linear convergence.
- For general function f, line search is necessary:

$$f(x_k + \alpha d_k) \leq \max_{0 \leq i \leq \min\{k-1, m\}} f(x_{k-i}) + \delta \alpha d_k^T g_k,$$

Thanks for your attention!