Newton's Methods

1 Introduction

The method of steepest descent uses only first derivatives (gradients) in selecting a suitable search direction. If higher derivatives are used, the resulting iterative algorithm may perform better than the steepest descent method. Newton's method (sometimes called the Newton-Raphson method) uses first and second derivatives and indeed does perform better than the steepest descent method if the initial point is close to the minimizer.

Given a starting point, we construct a quadratic approximation to the objective function that matches the first and second derivative values at that point. We then minimize the approximate (quadratic) function instead of the original objective function. We use the minimizer of the approximate function as the starting point in the next step and repeat the procedure iteratively.

If the objective function is quadratic, then the approximation is exact, and the method yields the true minimizer in one step. If, on the other hand, the objective function is not quadratic, then the approximation will provide only an estimate of the position of the true minimizer.

2 Newton's Method

We can obtain a quadratic approximation to the twice continuously differentiable objection function $f : \mathbb{R}^n \to \mathbb{R}$ using the Taylor series expansion of f about the current point $\mathbf{x}^k$ neglecting terms of order three and higher. We obtain

$$f(\mathbf{x}) \approx f(\mathbf{x}^k) + (\mathbf{x} - \mathbf{x}^k)\mathbf{g}^k + \frac{1}{2}(\mathbf{x} - \mathbf{x}^k)^\top \mathbf{F}(\mathbf{x}^k)(\mathbf{x} - \mathbf{x}^k) \triangleq q(\mathbf{x}), \tag{1}$$

where, for simplicity, we use the notation $\mathbf{g}^k = \nabla f(\mathbf{x}^k)$. Applying the FONC to $q(\mathbf{x})$ yields

$$\mathbf{0} = \nabla q(\mathbf{x}) = \mathbf{g}^k + \mathbf{F}(\mathbf{x}^k)(\mathbf{x} - \mathbf{x}^k). \tag{2}$$

If $\mathbf{F}(\mathbf{x}^k) > 0$, then $q(\mathbf{x})$ achieves a minimum at

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \mathbf{F}(\mathbf{x}^k)^{-1}\mathbf{g}^k. \tag{3}$$

This recursive formula represents Newton's method.

3 Analysis of Newton's Method

Newton's method reaches the point $\mathbf{x}^*$ such that $\nabla f(\mathbf{x}^*) = 0$ in just one step starting from any initial point $\mathbf{x}^0$. To see this, suppose that $\mathbf{Q} = \mathbf{Q}^\top$ is invertible and

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} - \mathbf{x}^\top \mathbf{b}, \tag{4}$$

and

$$g(\mathbf{x}) = \nabla f(\mathbf{x}) = \mathbf{Q}\mathbf{x} - \mathbf{b} \tag{5}$$

and $\mathbf{F}(\mathbf{x}) = \mathbf{Q}$. Hence, given any initial point $\mathbf{x}^0$, by Newton's algorithm

$$\begin{aligned} \mathbf{x}^1 &= \mathbf{x}^0 - \mathbf{F}(\mathbf{x}^0)^{-1}\mathbf{g}^0, \\ &= \mathbf{x}^0 - \mathbf{Q}^{-1}[\mathbf{Q}\mathbf{x}^0 - \mathbf{b}] = \mathbf{Q}^{-1}\mathbf{b} = \mathbf{x}^*. \end{aligned} \tag{6}$$

Theorem 1. Let $\{\mathbf{x}^k\}$ be the sequence generated by Newton's method for minimizing a given objective function $f(\mathbf{x})$. If the Hessian $\mathbf{F}(\mathbf{x}^k) > 0$ and $\mathbf{g}^k = \nabla f(\mathbf{x}^k) \neq 0$, then the search direction

$$\mathbf{d}^k = -\mathbf{F}(\mathbf{x}^k)^{-1}\mathbf{g}^k = \mathbf{x}^{k+1} - \mathbf{x}^k, \tag{7}$$

from $\mathbf{x}^k$ to $\mathbf{x}^{k+1}$ is a descent direction for f in the sense that there exists an $\bar{\alpha} > 0$ such that for all $\alpha \in (0, \bar{\alpha})$,

$$f(\mathbf{x}^k + \alpha \mathbf{d}^k) < f(\mathbf{x}^k). \tag{8}$$

Proof. Let $\phi(\alpha) = f(\mathbf{x}^k + \alpha \mathbf{d}^k)$. Then, using the chain rule, we obtain

$$\phi'(\alpha) = \nabla f(\mathbf{x}^k + \alpha \mathbf{d}^k)^\top \mathbf{d}^k, \tag{9}$$

and at $\alpha = 0$, we have $\phi'(0) = \nabla f(\mathbf{x}^k)^\top \mathbf{d}^k$ which becomes negative if $\mathbf{d}^k = -\mathbf{F}(\mathbf{x}^k)^{-1}\mathbf{g}^k$, i.e.,

$$\phi'(0) = -\mathbf{g}^{kT}\mathbf{F}(\mathbf{x}^k)^{-1}\mathbf{g}^k < 0, \tag{10}$$

because $\mathbf{F}(\mathbf{x}^k)^{-1} > 0$ and $\mathbf{g}^k \neq 0$. Thus, there exists an $\bar{\alpha} > 0$ so that for all $\alpha \in (0, \bar{\alpha})$, $\phi(\alpha) < \phi(0)$. This implies that for all $\alpha \in (0, \bar{\alpha})$,

$$f(\mathbf{x}^k + \alpha \mathbf{d}^k) < f(\mathbf{x}^k), \tag{11}$$

which completes the proof. □

4 Levenberg-Marquardt Modification

If the Hessian matrix $\mathbf{F}(\mathbf{x}^k)$ is not positive definite, then the search direction $\mathbf{d}^k = -\mathbf{F}(x^k)^{-1}\mathbf{g}^k$ may not point in a descent direction. A simple technique to ensure that the search direction is a descent direction is to introduce the Levenberg-Marquardt modification of Newton's algorithm:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - (\mathbf{F}(\mathbf{x}^k) + \mu_k \mathbf{I})^{-1}\mathbf{g}^k, \tag{12}$$

where $\mu_k \geq 0$. In this case if we further introduce a step size α_k

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k(\mathbf{F}(\mathbf{x}^k) + \mu_k\mathbf{I})^{-1}\mathbf{g}^k, \tag{13}$$

then we are guaranteed that the descent property holds.

Consider a symmetric matrix $\mathbf{F}$, which may not be positive definite. Let $\lambda_1, \cdots, \lambda_n$ be the eigenvalues of $\mathbf{F}$ with corresponding eigenvectors $\mathbf{v}_1, \cdots, \mathbf{v}_n$. The eigenvalues are real, but may not all be positive. Next, consider the matrix $\mathbf{G} = \mathbf{F} + \mu\mathbf{I}$, where $\mu \geq 0$. Note that the eigenvalues of $\mathbf{G}$ are $\lambda_1 + \mu, \cdots, \lambda_n + \mu$. Indeed,

$$\mathbf{G}\mathbf{v}_i = (\mathbf{F} + \mu\mathbf{I})\mathbf{v}_i = \mathbf{F}\mathbf{v}_i + \mu\mathbf{I}\mathbf{v}_i = \lambda_i\mathbf{v}_i + \mu\mathbf{v}_i = (\lambda_i + \mu)\mathbf{v}_i, \tag{14}$$

which shows that for all $i = 1, \cdots, n$, $\mathbf{v}_i$ is also an eigenvector of $\mathbf{G}$ with eigenvalue $\lambda_i + \mu$. Therefore, if μ is sufficiently large, then all the eigenvalues of $\mathbf{G}$ are positive and $\mathbf{G}$ is positive definite. Accordingly, if the parameter μ_k in the Levenberg-Marquardt modification of Newton's algorithm is sufficiently large, then the search direction $\mathbf{d}^k = -(\mathbf{F}(\mathbf{x}^k) + \mu\mathbf{I})^{-1}\mathbf{g}^k$ always points in a descent direction. In this case if we further introduce a step size α_k

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha(\mathbf{F}(\mathbf{x}^k) + \mu_k\mathbf{I})^{-1}\mathbf{g}^k, \tag{15}$$

then we are guaranteed that the descent property holds.

The Levenberg-Marquardt modification of Newton's algorithm can be made to approach the behavior of the pure Newton's method by letting $\mu_k \to 0$. On the other hand, by letting $\mu_k \to \infty$, the algorithm approaches a pure gradient method with small step size. In practice, we may start with a small value of μ_k and increase it slowly until we find that the iteration is descent: $f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k)$.

5 Newton's Method for Nonlinear Least Squares

We now examine a particular class of optimization problems and the use of Newton's method for solving them. Consider the following problem:

$$\text{minimize} \sum_{i=1}^{m}(r_i(\mathbf{x}))^2, \tag{16}$$

where $r_i(\mathbf{x}) : \mathbb{R}^n \to \mathbb{R}$, $i = 1, \cdots, m$, are given functions. This particular problem is called a nonlinear least-squares problem.

Defining $\mathbf{r} = [r_1, \cdots, r_m]^\top$, we write the objective function as $f(\mathbf{x}) = \mathbf{r}(\mathbf{x})^\top\mathbf{r}(\mathbf{x})$. To apply Newton's method, we need to compute the gradient and the Hessian of f. The jth component of $\nabla f(\mathbf{x})$ is

$$\nabla f(\mathbf{x})_j = \frac{\partial f}{\partial x_j}(\mathbf{x}) = 2\sum_{i=1}^{m} r_i(\mathbf{x})\frac{\partial r_i}{\partial x_j}(\mathbf{x}). \tag{17}$$

Denote the Jacobian matrix of $\mathbf{r}$ by

$$\mathbf{J}(\mathbf{x}) = \begin{bmatrix} \frac{\partial r_1}{\partial x_1}(\mathbf{x}) & \cdots & \frac{\partial r_1}{\partial x_n}(\mathbf{x}) \\ \vdots & & \vdots \\ \frac{\partial r_m}{\partial x_1}(\mathbf{x}) & \cdots & \frac{\partial r_m}{\partial x_n}(\mathbf{x}) \end{bmatrix}. \tag{18}$$

Then, the gradient of f can be represented as $\nabla f(\mathbf{x}) = 2\mathbf{J}(\mathbf{x})^\top \mathbf{r}(\mathbf{x})$.

Next, we compute the Hessian matrix of f. The (k, j)th component of the Hessian is given by

$$\begin{aligned} \frac{\partial^2 f}{\partial x_k \partial x_j}(\mathbf{x}) &= \frac{\partial}{\partial x_k}\frac{\partial f}{\partial x_j}(\mathbf{x}) = \frac{\partial}{\partial x_k}\left(2\sum_{i=1}^m r_i(\mathbf{x})\frac{\partial r_i}{\partial x_j}(\mathbf{x})\right) \\ &= 2\sum_{i=1}^m \left(\frac{\partial r_i}{\partial x_k}(\mathbf{x})\frac{\partial r_i}{\partial x_j}(\mathbf{x}) + r_i(\mathbf{x})\frac{\partial^2 r_i}{\partial x_k \partial x_j}(\mathbf{x})\right). \end{aligned} \tag{19}$$

Letting $\mathbf{S}(\mathbf{x})$ be the matrix whose (k, j)th component is

$$\sum_{i=1}^m r_i(\mathbf{x})\frac{\partial^2 r_i}{\partial x_k \partial x_j}(\mathbf{x}), \tag{20}$$

we write the Hessian matrix as

$$\mathbf{F}(\mathbf{x}) = 2(\mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x}) + \mathbf{S}(\mathbf{x})). \tag{21}$$

Therefore, Newton's method applied to the nonlinear least-squares problem is given by

$$\mathbf{x}^{k+1} = \mathbf{x}^k - (\mathbf{J}(\mathbf{x})^\top \mathbf{J}(\mathbf{x}) + \mathbf{S}(\mathbf{x}))^{-1}\mathbf{J}(\mathbf{x})^\top \mathbf{r}(\mathbf{x}). \tag{22}$$

6 Conjugate Direction Methods

The class of conjugate direction methods can be viewed as being intermediate between the method of steepest descent and Newton's method. The conjugate direction methods have the following properties:

1. Solve quadratics of n variables in n steps.
2. The usual implementation, the conjugate gradient algorithm, requires no Hessian matrix evaluations.
3. No matrix inversion and no storage of an $n \times n$ matrix are required.

The conjugate direction methods typically perform better than the method of steepest descent, but not as well as Newton's method. As we saw from the method of steepest descent and Newton's method, the crucial factor in the efficiency of an iterative search method is the direction of search at each iteration. For a quadratic function of n variables $f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} - \mathbf{x}^\top \mathbf{b}$, $\mathbf{x} \in \mathbb{R}^n$, $\mathbf{Q} = \mathbf{Q}^\top > 0$, the best direction of search, as we shall see, is in the Q-conjugate direction. Basically, two directions $\mathbf{d}^1$ and $\mathbf{d}^2$ in $\mathbb{R}^n$ are said to be Q-conjugate if $\mathbf{d}^{1\top} Q \mathbf{d}^2 = 0$. In general, we have the following definition.

Definition 1. Let $\mathbf{Q}$ be a real symmetric $n \times n$ matrix. The directions $\mathbf{d}^0, \cdots, \mathbf{d}^m$ are Q-conjugate if for all $i \neq j$, we have $\mathbf{d}^{i\top}\mathbf{Q}\mathbf{d}^j = 0$.

Lemma 1. Let $\mathbf{Q}$ be a symmetric positive definite $n \times n$ matrix. If the directions $\mathbf{d}^0, \cdots, \mathbf{d}^k \in \mathbb{R}^n$, $k \leq n-1$, are nonzero and Q-conjugate, then they are linearly independent.

Proof. Let $\alpha_0, \cdots, \alpha_k$ be scalars such that

$$\alpha_0 \mathbf{d}^0 + \alpha_1 \mathbf{d}^1 + \cdots + \alpha_k \mathbf{d}^k = 0. \tag{23}$$

Pre-multiplying this equality by $\mathbf{d}^{j\top}\mathbf{Q}$, $0 \leq j \leq k$, yields

$$\alpha_j \mathbf{d}^{j\top}\mathbf{Q}\mathbf{d}^j = 0, \tag{24}$$

because all other terms $\mathbf{d}^{j\top}\mathbf{Q}\mathbf{d}^i = 0$, $i \neq j$, by Q-conjugacy. But $\mathbf{Q} = \mathbf{Q}^\top > 0$ and $\mathbf{d}^j \neq 0$; hence $\alpha_j = 0$, $j = 0, 1, \cdots, k$. Therefore, $\mathbf{d}^0, \cdots, \mathbf{d}^k, k \leq n-1$, are linearly independent. $\square$

7 The Conjugate Direction Algorithm

Consider minimizing the quadratic function of n variables

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^\top \mathbf{Q}\mathbf{x} - \mathbf{x}^\top \mathbf{b}, \tag{25}$$

where $\mathbf{Q} = \mathbf{Q}^\top > 0$, $\mathbf{x} \in \mathbb{R}^n$. Note that because $\mathbf{Q} > 0$, the function f has a global minimizer that can be found by solving $\mathbf{Q}\mathbf{x} = \mathbf{b}$.

7.1 Basic Conjugate Direction Algorithm

Given a starting point $\mathbf{x}^0$ and Q-conjugate directions $\mathbf{d}^0, \cdots, \mathbf{d}^{n-1}$; for $k \geq 0$,

$$\begin{aligned} \mathbf{g}^k = \nabla f(\mathbf{x}^k) &= \mathbf{Q}\mathbf{x}^k - \mathbf{b}, \\ \alpha_k &= \frac{-\mathbf{g}^{k\top}\mathbf{d}^k}{\mathbf{d}^{k\top}\mathbf{Q}\mathbf{d}^k}, \\ \mathbf{x}^{k+1} &= \mathbf{x}^k + \alpha_k \mathbf{d}^k. \end{aligned} \tag{26}$$

Theorem 2. For any starting point $\mathbf{x}^0$ the basic conjugate direction algorithm converges to the unique $\mathbf{x}^*$ (that solves $\mathbf{Q}\mathbf{x} = \mathbf{b}$) in n steps; that is, $\mathbf{x}^n = \mathbf{x}^*$.

Proof. Consider $\mathbf{x}^* - \mathbf{x}^0 \in \mathbb{R}^n$. Because the $\mathbf{d}^i$ are linearly independent, there exist constants β_i, $i = 0, \cdots, n-1$, such that

$$\mathbf{x}^* - \mathbf{x}^0 = \beta_0 \mathbf{d}^0 + \cdots + \beta_{n-1}\mathbf{d}^{n-1}. \tag{27}$$

Now pre-multiply both sides of this equation by $\mathbf{d}^{k\top}\mathbf{Q}$, $0 \leq k \leq n$, to obtain

$$\mathbf{d}^{k\top}\mathbf{Q}(\mathbf{x}^* - \mathbf{x}^0) = \beta_k \mathbf{d}^{k\top}\mathbf{Q}\mathbf{d}^k, \tag{28}$$

where the terms $\mathbf{d}^{k\top}\mathbf{Q}\mathbf{d}^i = 0$, $k \neq i$, by the Q-conjugate property. Hence,

$$\beta_k = \frac{\mathbf{d}^{k\top}\mathbf{Q}(\mathbf{x}^* - \mathbf{x}^0)}{\mathbf{d}^{k\top}\mathbf{Q}\mathbf{d}^k}. \tag{29}$$

Now, we can write $\mathbf{x}^k = \mathbf{x}^0 + \alpha_0\mathbf{d}^0 + \cdots + \alpha_{k-1}\mathbf{d}^{k-1}$. Therefore, $\mathbf{x}^k - \mathbf{x}^0 = \alpha_0\mathbf{d}^0 + \cdots + \alpha_{k-1}\mathbf{d}^{k-1}$. So writing $\mathbf{x}^* - \mathbf{x}^0 = (\mathbf{x}^* - \mathbf{x}^k) + (\mathbf{x}^k - \mathbf{x}^0)$ and pre-multiply the above by $\mathbf{d}^{k\top}\mathbf{Q}$, we obtain

$$\mathbf{d}^{k\top}\mathbf{Q}(\mathbf{x}^* - \mathbf{x}^0) = \mathbf{d}^{k\top}\mathbf{Q}(\mathbf{x}^* - \mathbf{x}^k) = -\mathbf{d}^{k\top}\mathbf{g}^k, \tag{30}$$

because $\mathbf{g}^k = \mathbf{Q}\mathbf{x}^k - \mathbf{b}$ and $\mathbf{Q}\mathbf{x}^* = \mathbf{b}$. Thus

$$\beta_k = \frac{-\mathbf{d}^{k\top}\mathbf{g}^k}{\mathbf{d}^{k\top}\mathbf{Q}\mathbf{d}^k} = \alpha_k, \tag{31}$$

and $\mathbf{x}^* = \mathbf{x}^n$, which completes the proof. □

Example 1. Find the minimizer of

$$f(x_1, x_2) = \frac{1}{2}\mathbf{x}^\top \begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix} \mathbf{x} - \mathbf{x}^\top \begin{bmatrix} -1 \\ 1 \end{bmatrix}, \quad \mathbf{x} \in \mathbb{R}^2, \tag{32}$$

using the conjugate direction method with the initial point $\mathbf{x}^0 = [0, 0]^\top$, and Q-conjugate directions $\mathbf{d}^0 = [1, 0]^\top$ and $\mathbf{d}^1 = [-3/8, 3/4]^\top$.

We have

$$\mathbf{g}^0 = \mathbf{Q}\mathbf{x}^0 - \mathbf{b} = [1, -1]^\top, \tag{33}$$

and hence

$$\alpha_0 = \frac{-\mathbf{g}^{0\top}\mathbf{d}^0}{\mathbf{d}^{0\top}\mathbf{Q}\mathbf{d}^0} = \frac{-[1, -1]\begin{bmatrix} 1 \\ 0 \end{bmatrix}}{[1, 0]\begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix}} = -\frac{1}{4}. \tag{34}$$

Thus

$$\mathbf{x}^1 = \mathbf{x}^0 + \alpha_0\mathbf{d}^0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} - \frac{1}{4}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} -1/4 \\ 0 \end{bmatrix}. \tag{35}$$

To find $\mathbf{x}^2$, we compute

$$\mathbf{g}^1 = \mathbf{Q}\mathbf{x}^1 - \mathbf{b} = \begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix}\begin{bmatrix} -1/4 \\ 0 \end{bmatrix} - \begin{bmatrix} -1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ -3/2 \end{bmatrix} \tag{36}$$

and

$$\alpha_1 = -\frac{\mathbf{g}^{1\top}\mathbf{d}^1}{\mathbf{d}^{1\top}\mathbf{Q}\mathbf{d}^1} = -\frac{[0, -3/2]\begin{bmatrix} -3/8 \\ 3/4 \end{bmatrix}}{[-3/8, 3/4]\begin{bmatrix} 4 & 2 \\ 2 & 2 \end{bmatrix}\begin{bmatrix} -3/8 \\ 3/4 \end{bmatrix}} = 2. \tag{37}$$

Therefore,

$$\mathbf{x}^2 = \mathbf{x}^1 + \alpha_1\mathbf{d}^1 = \begin{bmatrix} -1/4 \\ 0 \end{bmatrix} + 2\begin{bmatrix} -3/8 \\ 3/4 \end{bmatrix} = \begin{bmatrix} -1 \\ 3/2 \end{bmatrix}. \tag{38}$$

Because f is a quadratic function in two variables, $\mathbf{x}^2 = \mathbf{x}^*$.

8 Quasi-Newton Methods

The idea behind Newton's method is to locally approximate the function f being minimized, at every iteration, by a quadratic function. The minimizer for the quadratic approximation is used as the starting point for the next iteration. This leads to Newton's recursive algorithm

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \mathbf{F}(\mathbf{x}^k)^{-1}\mathbf{g}^k. \tag{39}$$

We may try to guarantee that the algorithm has the descent property by modifying the original algorithm as follows:

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \mathbf{F}(\mathbf{x}^k)^{-1}\mathbf{g}^k, \tag{40}$$

where α_k is chosen to ensure that $f(\mathbf{x}^{k+1}) < f(\mathbf{x}^k)$.

To avoid the computation of $\mathbf{F}(\mathbf{x}^k)^{-1}$, the quasi-Newton methods use an approximation to $\mathbf{F}(\mathbf{x}^k)^{-1}$ in place of the true inverse. This approximation is updated at every stage so that it exhibits at least some properties of $\mathbf{F}(\mathbf{x}^k)^{-1}$. To get some idea about the properties that an approximation to $\mathbf{F}(\mathbf{x}^k)^{-1}$ should satisfy, consider the formula

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k \mathbf{H}_k \mathbf{g}^k, \tag{41}$$

where $\mathbf{H}_k$ is an $n\times n$ real matrix and α is a positive search parameter. Expanding f about $\mathbf{x}^k$ yields

$$\begin{aligned} f(\mathbf{x}^{k+1}) &= f(\mathbf{x}^k) + \mathbf{g}^{k\top}(\mathbf{x}^{k+1} - \mathbf{x}^k) + \cdots \\ &= f(\mathbf{x}^k) - \alpha \mathbf{g}^{k\top}\mathbf{H}_k\mathbf{g}^k + \cdots . \end{aligned} \tag{42}$$

dominates the third. Thus, to guarantee a decrease in f for small α, we have to have

$$\mathbf{g}^{k\top}\mathbf{H}_k\mathbf{g}^k > 0. \tag{43}$$

A simple way to ensure this is to require that $\mathbf{H}_k$ be positive definite. In constructing an approximation to the inverse of the Hessian matrix, we should use only the objective function and gradient values. Thus, if we can find a suitable method of choosing $\mathbf{H}_k$, the iteration may be carried out without any evaluation of the Hessian and without the solution of any set of linear equations.

9 Approximating the Inverse Hessian

Let $\mathbf{H}_0, \mathbf{H}_1, \mathbf{H}_2, \cdots$ be successive approximations of the inverse $\mathbf{F}(\mathbf{x}^k)^{-1}$ of the Hessian. We now derive a condition that the approximations should satisfy.

To begin, suppose first that the Hessian matrix $\mathbf{F}(\mathbf{x})$ of the objective function f is constant and independent of $\mathbf{x}$. In other words, the objective function is quadratic, with Hessian $\mathbf{F}(\mathbf{x}) = \mathbf{Q}$ for all $\mathbf{x}$, where $\mathbf{Q} = \mathbf{Q}^\top$. Then,

$$\mathbf{g}^{k+1} - \mathbf{g}^k = \mathbf{Q}(\mathbf{x}^{k+1} - \mathbf{x}^k). \tag{44}$$

Let

$$\Delta \mathbf{g}^k \triangleq \mathbf{g}^{k+1} - \mathbf{g}^k, \tag{45}$$

and

$$\Delta \mathbf{x}^k \triangleq \mathbf{x}^{k+1} - \mathbf{x}^k. \tag{46}$$

Then, we may write $\Delta \mathbf{g}^k = \mathbf{Q}\Delta \mathbf{x}^k$.

We start with a real symmetric positive definite matrix $\mathbf{H}_0$. Note that given k, the matrix $\mathbf{Q}^{-1}$ satisfies

$$\mathbf{Q}^{-1}\Delta \mathbf{g}^i = \Delta \mathbf{x}^i, \quad 0 \le i \le k. \tag{47}$$

Therefore, we also impose the requirement that the approximation $\mathbf{H}_{k+1}$ of the Hessian satisfy

$$\mathbf{H}_{k+1}\Delta \mathbf{g}^i = \Delta \mathbf{x}^i, \quad 0 \le i \le k. \tag{48}$$

If n steps are involved, then moving in n directions $\Delta \mathbf{x}^0$, $\Delta \mathbf{x}^1$, $\cdots$, $\Delta \mathbf{x}^{n-1}$ yields

$$\mathbf{H}_n\Delta \mathbf{g}^0 = \Delta \mathbf{x}^0, \quad \mathbf{H}_n\Delta \mathbf{g}^1 = \Delta \mathbf{x}^1, \quad \cdots, \quad \mathbf{H}_n\Delta \mathbf{g}^{n-1} = \Delta \mathbf{x}^{n-1}. \tag{49}$$

This set of equations can be represented as

$$\mathbf{H}_n[\Delta \mathbf{g}^0, \Delta \mathbf{g}^1, \cdots, \Delta \mathbf{g}^{n-1}] = [\Delta \mathbf{x}^0, \Delta \mathbf{x}^1, \cdots, \Delta \mathbf{x}^{n-1}]. \tag{50}$$

Note that $\mathbf{Q}$ satisfies

$$\mathbf{Q}[\Delta \mathbf{x}^0, \Delta \mathbf{x}^1, \cdots, \Delta \mathbf{x}^{n-1}] = [\Delta \mathbf{g}^0, \Delta \mathbf{g}^1, \cdots, \Delta \mathbf{g}^{n-1}], \tag{51}$$

and

$$\mathbf{Q}^{-1}[\Delta \mathbf{g}^0, \Delta \mathbf{g}^1, \cdots, \Delta \mathbf{g}^{n-1}] = [\Delta \mathbf{x}^0, \Delta \mathbf{x}^1, \cdots, \Delta \mathbf{x}^{n-1}]. \tag{52}$$

Therefore, if $[\Delta \mathbf{g}^0, \Delta \mathbf{g}^1, \cdots, \Delta \mathbf{g}^{n-1}]$ is nonsingular, then $\mathbf{Q}^{-1}$ is determined uniquely after n steps, via

$$\mathbf{Q}^{-1} = \mathbf{H}_n = [\Delta \mathbf{x}^0, \Delta \mathbf{x}^1, \cdots, \Delta \mathbf{x}^{n-1}][\Delta \mathbf{g}^0, \Delta \mathbf{g}^1, \cdots, \Delta \mathbf{g}^{n-1}]^{-1}. \tag{53}$$

As a consequence, we conclude that if $\mathbf{H}_n$ satisfies the equations $\mathbf{H}_n\Delta \mathbf{g}^i = \Delta \mathbf{x}^i$, $0 \le i \le n-1$, then the algorithm $\mathbf{x}^{k+1} = \mathbf{x}^k - \alpha_k\mathbf{H}_k\mathbf{g}^k$, $\alpha_k = \underset{\alpha\ge 0}{argmin}\{f(\mathbf{x}^k - \alpha_k\mathbf{H}_k\mathbf{g}^k)\}$, is guaranteed to solve problems with quadratic objective functions in $n+1$ steps.

Specifically, quasi-Newton algorithms have the form

$$\begin{aligned} \mathbf{d}^k &= -\mathbf{H}_k\mathbf{g}^k, \\ \alpha &= \underset{\alpha\ge 0}{argmin}\{f(\mathbf{x}^k + \alpha\mathbf{d}^k)\}, \\ \mathbf{x}^{k+1} &= \mathbf{x}^k + \alpha_k\mathbf{d}^k, \end{aligned} \tag{54}$$

where the matrices $\mathbf{H}_0$, $\mathbf{H}_1$, $\cdots$ are symmetric. In the quadratic case these matrices are required to satisfy

$$\mathbf{H}_{k+1}\Delta \mathbf{g}^i = \Delta \mathbf{x}^i, \quad 0 \le i \le k, \tag{55}$$

where $\Delta \mathbf{x}^i = \mathbf{x}^{i+1} - \mathbf{x}^i = \alpha\mathbf{d}^i$ and $\Delta \mathbf{g}^i = \mathbf{g}^{i+1} - \mathbf{g}^i = \mathbf{Q}\Delta \mathbf{x}^i$.

10 Rank One Correction Formula

In the rank one correction formula, the correction term is symmetric and has the form $\alpha_k \mathbf{z}^k \mathbf{z}^{k\top}$, where $\alpha_k \in \mathbb{R}$ and $\mathbf{z}^k \in \mathbb{R}^n$. Therefore, the update equation is

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \alpha_k \mathbf{z}^k \mathbf{z}^{k\top} \tag{56}$$

Note that

$$rank(\mathbf{z}^k \mathbf{z}^{k\top}) = rank\left([z_1^k, \cdots, z_n^k]^\top [z_1^k, \cdots, z_n^k]\right) = 1 \tag{57}$$

Observe that if $\mathbf{H}_k$ is symmetric, then so is $\mathbf{H}_{k+1}$. Our goal now is to determine α_k and $\mathbf{z}^k$ given $\mathbf{H}_k$, $\Delta\mathbf{g}^k$, $\Delta\mathbf{x}^k$ so that the required relationship is satisfied; namely, $\mathbf{H}_{k+1}\Delta\mathbf{g}^i = \Delta\mathbf{x}^i$, $i = 1, \cdots, k$.

To begin, let us first consider the condition $\mathbf{H}_{k+1}\Delta\mathbf{g}^k = \Delta\mathbf{x}^k$. In other words, given $\mathbf{H}_k$, $\Delta\mathbf{g}^k$, $\Delta\mathbf{x}^k$, we wish to find α_k and $\mathbf{z}^k$ to ensure that

$$\mathbf{H}_{k+1}\Delta\mathbf{g}^k = (\mathbf{H}_k + \alpha_k \mathbf{z}^k \mathbf{z}^{k\top})\Delta\mathbf{g}^k = \Delta\mathbf{x}^k. \tag{58}$$

First note that $\mathbf{z}^{k\top}\Delta\mathbf{g}^k$ is a scalar. Thus,

$$\Delta\mathbf{x}^k - \mathbf{H}_k\Delta\mathbf{g}^k = (\alpha_k \mathbf{z}^{k\top}\Delta\mathbf{g}^k)\mathbf{z}^k, \tag{59}$$

and hence

$$\mathbf{z}^k = \frac{\Delta\mathbf{x}^k - \mathbf{H}_k\Delta\mathbf{g}^k}{\alpha_k(\mathbf{z}^{k\top}\Delta\mathbf{g}^k)}. \tag{60}$$

We can now determine

$$\alpha_k \mathbf{z}^k \mathbf{z}^{k\top} = \frac{(\Delta\mathbf{x}^k - \mathbf{H}_k\Delta\mathbf{g}^k)(\Delta\mathbf{x}^k - \mathbf{H}_k\Delta\mathbf{g}^k)^\top}{\alpha_k(\mathbf{z}^{k\top}\Delta\mathbf{g}^k)^2}. \tag{61}$$

Hence

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{(\Delta\mathbf{x}^k - \mathbf{H}_k\Delta\mathbf{g}^k)(\Delta\mathbf{x}^k - \mathbf{H}_k\Delta\mathbf{g}^k)^\top}{\alpha_k(\mathbf{z}^{k\top}\Delta\mathbf{g}^k)^2}. \tag{62}$$

The next step is to pre-multiply $\Delta\mathbf{x}^k - \mathbf{H}_k\Delta\mathbf{g}^k = (\alpha_k \mathbf{z}^{k\top}\Delta\mathbf{g}^k)\mathbf{z}^k$ by $\Delta\mathbf{g}^{k\top}$ to obtain

$$\Delta\mathbf{g}^{k\top}\Delta\mathbf{x}^k - \Delta\mathbf{g}^{k\top}\mathbf{H}_k\Delta\mathbf{g}^k = \Delta\mathbf{g}^{k\top}\alpha_k \mathbf{z}^k \mathbf{z}^{k\top}\Delta\mathbf{g}^k. \tag{63}$$

Observe that α_k is a scalar and so is $\Delta\mathbf{g}^{k\top}\mathbf{z}^k = \mathbf{z}^{k\top}\Delta\mathbf{g}^k$. Thus,

$$\Delta\mathbf{g}^{k\top}\Delta\mathbf{x}^k - \Delta\mathbf{g}^{k\top}\mathbf{H}_k\Delta\mathbf{g}^k = \alpha_k(\mathbf{z}^{k\top}\Delta\mathbf{g}^k)^2. \tag{64}$$

Taking this relation into account yields

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{(\Delta\mathbf{x}^k - \mathbf{H}_k\Delta\mathbf{g}^k)(\Delta\mathbf{x}^k - \mathbf{H}_k\Delta\mathbf{g}^k)^\top}{\Delta\mathbf{g}^{k\top}(\Delta\mathbf{x}^k - \mathbf{H}_k\Delta\mathbf{g}^k)}. \tag{65}$$

11 The DFP Algorithm

We now show that the DFP algorithm (due to Davidon, Fletcher and Powell) is a quasi-Newton method, in the sense that when applied to quadratic problems, we have $\mathbf{H}_{k+1}\Delta\mathbf{g}^i = \Delta\mathbf{x}^i$, $0 \le i \le k$.

Theorem 3. In the DFP algorithm applied to the quadratic with Hessian $\mathbf{Q} = \mathbf{Q}^\top$, we have $\mathbf{H}_{k+1}\Delta\mathbf{g}^i = \Delta\mathbf{x}^i$, $0 \le i \le k$.

Proof. We use induction. For $k = 0$, we have

$$\mathbf{H}_1\Delta\mathbf{g}^0 = \mathbf{H}_0\Delta\mathbf{g}^0 + \frac{\Delta\mathbf{x}^0\Delta\mathbf{x}^{0\top}}{\Delta\mathbf{x}^{0\top}\Delta\mathbf{g}^0}\Delta\mathbf{g}^0 - \frac{\mathbf{H}_0\Delta\mathbf{g}^0\Delta\mathbf{g}^{0\top}\mathbf{H}_0}{\Delta\mathbf{g}^{0\top}\mathbf{H}_0\Delta\mathbf{g}^0}\Delta\mathbf{g}^0 = \Delta\mathbf{x}^0. \tag{66}$$

Assume that the result is true for $k-1$; that is, $\mathbf{H}_k\Delta\mathbf{g}^i = \Delta\mathbf{x}^i$, $0 \le i \le k-1$. We now show that $\mathbf{H}_k\Delta\mathbf{g}^i = \Delta\mathbf{x}^i$, $0 \le i \le k$. First, consider $i = k$. We have

$$\mathbf{H}_{k+1}\Delta\mathbf{g}^k = \mathbf{H}_k\Delta\mathbf{g}^k + \frac{\Delta\mathbf{x}^k\Delta\mathbf{x}^{k\top}}{\Delta\mathbf{x}^{k\top}\Delta\mathbf{g}^k}\Delta\mathbf{g}^k - \frac{\mathbf{H}_k\Delta\mathbf{g}^k\Delta\mathbf{g}^{k\top}\mathbf{H}_k}{\Delta\mathbf{g}^{k\top}\mathbf{H}_k\Delta\mathbf{g}^k}\Delta\mathbf{g}^k = \Delta\mathbf{x}^k. \tag{67}$$

It remains to consider the case $i < k$. To this end,

$$\begin{aligned}\mathbf{H}_{k+1}\Delta\mathbf{g}^i &= \mathbf{H}_k\Delta\mathbf{g}^i + \frac{\Delta\mathbf{x}^k\Delta\mathbf{x}^{k\top}}{\Delta\mathbf{x}^{k\top}\Delta\mathbf{g}^k}\Delta\mathbf{g}^i - \frac{\mathbf{H}_k\Delta\mathbf{g}^k\Delta\mathbf{g}^{k\top}\mathbf{H}_k}{\Delta\mathbf{g}^{k\top}\mathbf{H}_k\Delta\mathbf{g}^k}\Delta\mathbf{g}^i \\ &= \Delta\mathbf{x}^i + \frac{\Delta\mathbf{x}^k}{\Delta\mathbf{x}^{k\top}\Delta\mathbf{g}^k}(\Delta\mathbf{x}^{k\top}\Delta\mathbf{g}^i) - \frac{\mathbf{H}_k\Delta\mathbf{g}^k}{\Delta\mathbf{g}^{k\top}\mathbf{H}_k\Delta\mathbf{g}^k}(\Delta\mathbf{g}^{k\top}\Delta\mathbf{x}^i). \end{aligned} \tag{68}$$

Now

$$\Delta\mathbf{x}^{k\top}\Delta\mathbf{g}^i = \Delta\mathbf{x}^{k\top}\mathbf{Q}\Delta\mathbf{x}^i = \alpha_k\alpha_i\mathbf{d}^{k\top}\mathbf{Q}\mathbf{d}^i = 0, \tag{69}$$

by the induction hypothesis. The same arguments yield $\Delta\mathbf{g}^{k\top}\Delta\mathbf{x}^i = 0$. Hence,

$$\mathbf{H}_{k+1}\Delta\mathbf{g}^i = \Delta\mathbf{x}^i, \tag{70}$$

which completes the proof. □

The update on the $\mathbf{H}$ matrix is

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{\Delta\mathbf{x}^k\Delta\mathbf{x}^{k\top}}{\Delta\mathbf{x}^{k\top}\Delta\mathbf{g}^k} - \frac{[\mathbf{H}_k\Delta\mathbf{g}^k][\mathbf{H}_k\Delta\mathbf{g}^k]^\top}{\Delta\mathbf{g}^{k\top}\mathbf{H}_k\Delta\mathbf{g}^k}. \tag{71}$$

12 The BFGS Algorithm

Recall that the updating formulas for the approximation of the inverse of the Hessian matrix were based on satisfying the equations

$$\mathbf{H}_{k+1}\Delta\mathbf{g}^i = \Delta\mathbf{x}^i, \quad 0 \le i \le k, \tag{72}$$

which were derived from $\Delta\mathbf{g}^i = \mathbf{Q}\Delta\mathbf{x}^i$, $0 \le i \le k$. We then formulated update formulas for the approximations to the inverse of the Hessian matrix $\mathbf{Q}^{-1}$.

An alternative to approximating $\mathbf{Q}^{-1}$ is to approximate $\mathbf{Q}$ itself. To do this let $\mathbf{B}_k$ be our estimate of $\mathbf{Q}$ at the kth step. We require $\mathbf{B}_{k+1}$ to satisfy

$$\Delta\mathbf{g}^i = \mathbf{B}_{k+1}\Delta\mathbf{x}^i, \quad 0 \le i \le k. \tag{73}$$

Notice that this set of equations is similar to the previous set of equations for $\mathbf{H}_{k+1}$ the only difference being that the roles of $\Delta\mathbf{x}^i$ and $\Delta\mathbf{g}^i$ are interchanged. Thus, given any update formula for $\mathbf{H}_k$, a corresponding update formula for $\mathbf{B}_k$ can be found by interchanging the roles of $\mathbf{B}_k$ and $\mathbf{H}_k$ and of $\Delta\mathbf{g}^k$ and $\Delta\mathbf{x}^k$.

Recall that the DFP update for the approximation $\mathbf{H}_k$ of the inverse Hessian is

$$\mathbf{H}_{k+1}^{DFP} = \mathbf{H}_k + \frac{\Delta\mathbf{x}^k\Delta\mathbf{x}^{k\top}}{\Delta\mathbf{x}^{k\top}\Delta\mathbf{g}^k} - \frac{\mathbf{H}_k\Delta\mathbf{g}^k\Delta\mathbf{g}^{k\top}\mathbf{H}_k}{\Delta\mathbf{g}^{k\top}\mathbf{H}_k\Delta\mathbf{g}^k}. \tag{74}$$

Using the complementarity concept, we can easily obtain an update equation for the approximation $\mathbf{B}_k$ of the Hessian:

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{\Delta\mathbf{g}^k\Delta\mathbf{g}^{k\top}}{\Delta\mathbf{g}^{k\top}\Delta\mathbf{x}^k} - \frac{\mathbf{B}_k\Delta\mathbf{x}^k\Delta\mathbf{x}^{k\top}\mathbf{B}_k}{\Delta\mathbf{x}^{k\top}\mathbf{B}_k\Delta\mathbf{x}^k}. \tag{75}$$

Now, to obtain the BFGS update for the approximation of the inverse Hessian, we take the inverse of $\mathbf{B}_{k+1}$ to obtain

$$\mathbf{H}_{k+1}^{BFGS} = \mathbf{B}_{k+1}^{-1} = \left(\mathbf{B}_k + \frac{\Delta\mathbf{g}^k\Delta\mathbf{g}^{k\top}}{\Delta\mathbf{g}^{k\top}\Delta\mathbf{x}^k} - \frac{\mathbf{B}_k\Delta\mathbf{x}^k\Delta\mathbf{x}^{k\top}\mathbf{B}_k}{\Delta\mathbf{x}^{k\top}\mathbf{B}_k\Delta\mathbf{x}^k}\right)^{-1}. \tag{76}$$

To compute $\mathbf{H}_{k+1}^{BFGS}$ by inverting the right-hand side of this equation, we apply the Sherman-Morrison formula.

Definition 2. Let $\mathbf{A}$ be a nonsingular matrix. Let $\mathbf{u}$ and $\mathbf{v}$ be column vectors such that $1 + \mathbf{v}^\top\mathbf{A}^{-1}\mathbf{u} \neq 0$. Then, $\mathbf{A} + \mathbf{u}\mathbf{v}^\top$ is nonsingular, and its inverse can be written in terms of $\mathbf{A}^{-1}$ using the following formula:

$$(A + \mathbf{u}\mathbf{v}^\top)^{-1} = \mathbf{A}^{-1} - \frac{(\mathbf{A}^{-1}\mathbf{u})(\mathbf{v}^\top\mathbf{A}^{-1})}{1 + \mathbf{v}^\top\mathbf{A}^{-1}\mathbf{u}}. \tag{77}$$

Applying Sherman-Morrison formula twice to $\mathbf{B}_{k+1}$ yields

$$\mathbf{H}_k^{BFGS} = \mathbf{H}_k + \left(1 + \frac{\Delta\mathbf{g}^{k\top}\mathbf{H}_k\Delta\mathbf{g}^k}{\Delta\mathbf{g}^{k\top}\Delta\mathbf{x}^k}\right)\frac{\Delta\mathbf{x}^k\Delta\mathbf{x}^{k\top}}{\Delta\mathbf{x}^{k\top}\Delta\mathbf{g}^k} - \frac{\mathbf{H}_k\Delta\mathbf{g}^k\Delta\mathbf{x}^{k\top} + (\mathbf{H}_k\Delta\mathbf{g}^k\Delta\mathbf{x}^{k\top})^\top}{\Delta\mathbf{g}^{k\top}\Delta\mathbf{x}^k}. \tag{78}$$