Optimization Theory and Applications

Kun Zhu (zhukun@nuaa.edu.cn)

November 28, 2018

Introduction

- Recall Newton's method

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - F\left(\mathbf{x}^{(k)}\right)^{-1}\mathbf{g}^{(k)}$$

and a modification for ensuring the descent property

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k F\left(\mathbf{x}^{(k)}\right)^{-1}\mathbf{g}^{(k)}$$

where

$$\alpha_k = \arg\min_{\alpha \geq 0} f\left(\mathbf{x}^{(k)} - \alpha F\left(\mathbf{x}^{(k)}\right)^{-1}\mathbf{g}^{(k)}\right)$$

- Property:
  - Fast convergence if we start close enough to solution
  - Requires computation of the Hessian inverse (which may be large)

Basic Idea

- ***Quasi-Newton methods***: approximate the Hessian inverse using only gradient information
- Use $\mathbf{H}_k$ to take place of the true Hessian inverse in Newton's algorithm
- Basic quasi-Newton algorithm

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \mathbf{H}_k \mathbf{g}^{(k)}$$

- The matrix $\mathbf{H}^{(k+1)}$ is updated at each iteration and is computed using $\mathbf{x}^{(k)}, \mathbf{x}^{(k+1)}, \mathbf{g}^{(k)}, \mathbf{g}^{(k+1)}$, and $\mathbf{H}_k$

Quasi-Newton Methods

- $\mathbf{H}_k$ is supposed to "mimic" $\mathbf{F}(\mathbf{x}^{(k)})^{-1}$
- What properties of $\mathbf{F}(\mathbf{x}^{(k)})^{-1}$ should it mimic?
  - At least $\mathbf{H}_k$ should be symmetric
  - Require $\mathbf{H}_k$ to be positive definite for ensuring the descent property
  - Another property that $\mathbf{H}_k$ should mimic is the "secant" property
- These are conditions that the serial of $\mathbf{H}_k$ should satisfy which are also the basis of the quasi-Newton method

Quasi-Newton Methods

- Select $\mathbf{H}_k$ to be positive definite to ensure descent property
- Illustration
  - Expanding f around $\mathbf{x}^{(k)}$ yields

$$\begin{aligned} f(\mathbf{x}^{(k+1)}) &= f(\mathbf{x}^{(k)}) + \mathbf{g}^{(k)T}(\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}) + o(\|\mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}\|) \\ &= f(\mathbf{x}^{(k)}) - \alpha \mathbf{g}^{(k)T}\mathbf{H}_k\mathbf{g}^{(k)} + o(\|\mathbf{H}_k\mathbf{g}^{(k)}\|\alpha) \end{aligned}$$

  - When α tends to 0, $\alpha \mathbf{g}^{(k)T}\mathbf{H}_k\mathbf{g}^{(k)}$ dominates $o(\|\mathbf{H}_k\mathbf{g}^{(k)}\|\alpha)$
  - Accordingly, to guarantee a decrease in f for small α, we need

$$\mathbf{g}^{(k)T}\mathbf{H}_k\mathbf{g}^{(k)} > 0$$

  - A simple way is to require $\mathbf{H}_k$ to be positive definite
  - An alternative way of showing this is to require $\phi'_\alpha(0) < 0$

Quasi-Newton Methods

- Accordingly, we can have the following proposition
- ***Proposition***: Let $f \in \mathcal{C}^1, \mathbf{x}^{(k)} \in \mathbb{R}^n, \mathbf{g}^{(k)} = \nabla f(\mathbf{x}^{(k)}) \neq 0$, and $\mathbf{H}_k$ an $n \times n$ real symmetric positive definite matrix. If we set

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - \alpha_k \mathbf{H}^{(k)} \mathbf{g}^{(k)}$$

where

$$\alpha_k = \arg\min_{\alpha \geq 0} f(\mathbf{x}^{(k)} - \alpha \mathbf{H}_k)$$

then we have $\alpha_k > 0$ and $f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$

Quasi-Newton Methods

- $\mathbf{H}_k$ mimics the "secant" property
- To explain this property, assume that f is quadratic, with Hessian $\mathbf{Q}$
- Note that $\mathbf{Q}$ satisfies

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

or

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

Quasi-Newton Methods

- Let

$$\Delta \mathbf{g}^{(k)} \triangleq \mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}$$
$$\Delta \mathbf{x}^{(k)} \triangleq \mathbf{x}^{(k+1)} - \mathbf{x}^{(k)}$$

- At any k, $\mathbf{Q}^{-1}$ satisfies:

$$\mathbf{Q}^{-1} \Delta \mathbf{g}^{(i)} = \Delta \mathbf{x}^{(i)}, \quad 0 \leq i \leq k$$

- To mimic $\mathbf{Q}^{-1}$, we want $\mathbf{H}_{k+1}$ to also satisfy

$$\mathbf{H}_{k+1} \Delta \mathbf{g}^{(i)} = \Delta \mathbf{x}^{(i)}, \quad 0 \leq i \leq k$$

- The above is called the quasi-Newton (or secant) condition

Quasi-Newton Methods

- Example: for $k = 3$, it would require

$$\mathbf{H}_4 \Delta \mathbf{g}^{(0)} = \Delta \mathbf{x}^{(0)},$$
$$\mathbf{H}_4 \Delta \mathbf{g}^{(1)} = \Delta \mathbf{x}^{(1)},$$
$$\mathbf{H}_4 \Delta \mathbf{g}^{(2)} = \Delta \mathbf{x}^{(2)},$$
$$\mathbf{H}_4 \Delta \mathbf{g}^{(3)} = \Delta \mathbf{x}^{(3)}$$

Quasi-Newton Methods

- Form of algorithm

$$\mathbf{d}^{(k)} = -\mathbf{H}_k \mathbf{g}^{(k)}$$

$$\alpha_k = \arg\min_{\alpha \geq 0} f(\mathbf{x}^{(k)} + \alpha \mathbf{d}^{(k)})$$

$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$$

  where the matrices $\mathbf{H}_0, \mathbf{H}_1, \ldots$ are symmetric

- In the quadratic case, the above matrices are required to satisfy

$$\mathbf{H}_{k+1} \Delta \mathbf{g}^{(i)} = \Delta \mathbf{x}^{(i)}, \qquad 0 \leq i \leq k$$

Quasi-Newton Methods

- Quasi-Newton method is a conjugate direction method
- ***Theorem***: Consider a quasi-Newton algorithm applied to a quadratic function with Hessian $\mathbf{Q} = \mathbf{Q}^T$ such that for $0 \leq k < n-1$

$$\mathbf{H}_{k+1}\Delta\mathbf{g}^{(i)} = \Delta\mathbf{x}^{(i)}, \qquad 0 \leq i \leq k$$

If $\alpha_i \neq 0,\ 0 \leq i \leq k$, then $\mathbf{d}^{(0)}, \ldots, \mathbf{d}^{(k+1)}$ are Q-conjugate

Quasi-Newton Methods

- ***Proof***:
  - We use induction
  - For $k = 0$, since $\alpha_0 \neq 0$, $\mathbf{d}^{(0)} = \Delta\mathbf{x}^{(0)}/\alpha_0$, and we have

$$\begin{aligned}
\mathbf{d}^{(1)T} Q \mathbf{d}^{(0)} &= -\mathbf{g}^{(1)T} \mathbf{H}_1 Q \mathbf{d}^{(0)} \\
&= -\mathbf{g}^{(1)T} \mathbf{H}_1 \frac{Q\Delta\mathbf{x}^{(0)}}{\alpha_0} \\
&= -\mathbf{g}^{(1)T} \frac{\mathbf{H}_1 \Delta\mathbf{g}^{(0)}}{\alpha_0} \\
&= -\mathbf{g}^{(1)T} \mathbf{d}^{(0)} \\
&= 0
\end{aligned}$$

    since $\alpha_0 > 0$ is the minimizer of $\phi(\alpha) = f(\mathbf{x}^{(0)} + \alpha\mathbf{d}^{(0)})$
  - Suppose the result is true for $k-1$ (i.e., $\mathbf{d}^{(0)}, \ldots, \mathbf{d}^{(k)}$ are Q-conjugate)
  - We now prove the result for k (i.e., $\mathbf{d}^{(0)}, \ldots, \mathbf{d}^{(k+1)}$ are Q-conjugate)

Quasi-Newton Methods

- ***Proof***: (Cont.)
  - It suffices to show that $\mathbf{d}^{(k+1)T} Q \mathbf{d}^{(i)} = 0, \quad 0 \leq i \leq k$
  - Given $i, 0 \leq i \leq k$, we have

$$\begin{aligned}
\mathbf{d}^{(k+1)} Q \mathbf{d}^{(i)} &= -\mathbf{g}^{(k+1)T} \mathbf{H}_{k+1} Q \mathbf{d}^{(i)} \\
&= -\mathbf{g}^{(k+1)T} \mathbf{H}_{k+1} \frac{Q \Delta \mathbf{x}^{(i)}}{\alpha_i} \\
&= -\mathbf{g}^{(k+1)T} \frac{\mathbf{H}_{k+1} \Delta \mathbf{g}^{(i)}}{\alpha_i} \\
&= -\mathbf{g}^{(k+1)T} \frac{\Delta \mathbf{x}^{(i)}}{\alpha_i} \\
&= -\mathbf{g}^{(k+1)T} \mathbf{d}^{(i)}
\end{aligned}$$

  - Since $\mathbf{d}^{(0)}, \ldots, \mathbf{d}^{(k)}$ are Q-conjugate by assumption, by the "expanding subspace" lemma, we have $\mathbf{g}^{(k+1)T} \mathbf{d}^{(i)} = 0$

Quasi-Newton Methods

- By the previous theorem, we conclude that if we apply a quasi-Newton algorithm to a quadratic, it terminates in n steps
- How do we generate the matrices $\mathbf{H}_k$ in such a way that it satisfies the quasi-Newton condition?
- There are several update formulas available for computing $\mathbf{H}_{k+1}$ based on $\mathbf{H}_k$, $\Delta\mathbf{g}^{(k)}$, and $\Delta\mathbf{x}^{(k)}$

Quasi-Newton Algorithm

- Methods for generating the $\mathbf{H}_k$
  - Rank one formula
  - DFP formula
  - BFGS formula
- All have the form

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \mathbf{U}_k$$

where $\mathbf{U}_k$ is an update (correction) term that depends on $\mathbf{H}_k$, $\Delta\mathbf{g}^{(k)}$, and $\Delta\mathbf{x}^{(k)}$

Rank One Correction Formula

- The rank one formula has the form

$$\mathbf{U}_k = \alpha_k \mathbf{z}^{(k)} \mathbf{z}^{(k)T}$$

  where $\alpha_k \in \mathbb{R}$ and $\mathbf{z}^{(k)} \in \mathbb{R}^n$

- Note that

$$\text{rank } \mathbf{z}^{(k)} \mathbf{z}^{(k)T} = \text{rank} \left(\begin{bmatrix} z_1^{(k)} \\ \vdots \\ z_n^{(k)} \end{bmatrix} \begin{bmatrix} z_1^{(k)} & \cdots & z_n^{(k)} \end{bmatrix} \right) = 1$$

  Hence the name *rank one* correction

- Note: if we start with a symmetric matrix $\mathbf{H}_0$, then the $\mathbf{H}_k$ remain symmetric

Rank One Correction Formula

- ***Question***: How to determine α_k and $\mathbf{z}^{(k)}$, such that given $\mathbf{H}_k, \Delta\mathbf{g}^{(k)}, \Delta\mathbf{x}^{(k)}$, the quasi-Newton condition is satisfied, that is

$$\mathbf{H}_{(k+1)}\Delta\mathbf{g}^{(i)} = \Delta\mathbf{x}^{(i)}, i = 1, \ldots, k$$

- Idea: begin with the condition $\mathbf{H}_{(k+1)}\Delta\mathbf{g}^{(k)} = \Delta\mathbf{x}^{(k)}$

Rank One Correction Formula

- ***Answer***: The quasi-Newton condition holds if and only if

$$\mathbf{U}_k = \frac{(\Delta\mathbf{x}^{(k)} - \mathbf{H}_k\Delta\mathbf{g}^{(k)})(\Delta\mathbf{x}^{(k)} - \mathbf{H}_k\Delta\mathbf{g}^{(k)})^T}{(\Delta\mathbf{x}^{(k)} - \mathbf{H}_k\Delta\mathbf{g}^{(k)})^T\Delta\mathbf{g}^{(k)}}$$

  which can be expressed as

$$\alpha_k = \frac{1}{(\Delta\mathbf{x}^{(k)} - \mathbf{H}_k\Delta\mathbf{g}^{(k)})^T\Delta\mathbf{g}^{(k)}}$$

$$\mathbf{z}^{(k)} = \Delta\mathbf{x}^{(k)} - \mathbf{H}_k\Delta\mathbf{g}^{(k)}$$

- Derivation is straightforward

Rank One Algorithm

- Set $k := 0$; select $\mathbf{x}^{(0)}$ and a real symmetric positive definite $\mathbf{H}_0$
- If $\mathbf{g}^{(k)} = 0$, stop; else, $\mathbf{d}^{(k)} = -\mathbf{H}_k \mathbf{g}^{(k)}$
- Compute

$$\alpha_k = \arg\min_{\alpha \geq 0} f(\mathbf{x}^{(k)} + \alpha \mathbf{d}^{(k)})$$
$$\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \alpha_k \mathbf{d}^{(k)}$$

- Compute

$$\Delta \mathbf{x}^{(k)} = \alpha_k \mathbf{d}^{(k)}$$
$$\Delta \mathbf{g}^{(k)} = \mathbf{g}^{(k+1)} - \mathbf{g}^{(k)}$$
$$\mathbf{H}_{k+1} = \mathbf{H}_k + \frac{(\Delta \mathbf{x}^{(k)} - \mathbf{H_k} \Delta \mathbf{g}^{(k)})(\Delta \mathbf{x}^{(k)} - \mathbf{H}_k \Delta \mathbf{g}^{(k)})}{\Delta \mathbf{g}^{(k)^T} (\Delta \mathbf{x}^{(k)} - \mathbf{H}_k \Delta \mathbf{g}^{(k)})}$$

- Set $k := k + 1$; go to step 2

Rank One Algorithm

- The rank one algorithm is based on satisfying the equation

$$\mathbf{H}_{k+1}\Delta\mathbf{g}^{(k)} = \Delta\mathbf{x}^{(k)}$$

- What about $\mathbf{H}_{k+1}\Delta\mathbf{g}^{(i)}$ for all $i = 0, \ldots, k$
- ***Theorem***: For the rank one algorithm applied to the quadratic with Hessian $Q = Q^T$, we have

$$\mathbf{H}_{k+1}\Delta\mathbf{g}^{(i)} = \Delta\mathbf{x}^{(i)}, 0 \leq i \leq k$$

- ***Proof***: Use induction
- Implication: the rank one algorithm is a quasi-Newton method

Example of Rank One Algorithm

- Example: Let

$$f(x_1, x_2) = x_1^2 + \frac{1}{2}x_2^2 + 3$$

Apply the rank one correction algorithm to minimize f. Use $\mathbf{x}^{(0)} = [1, 2]^T$ and $\mathbf{H}_0 = \mathbf{I}_2$

- We can represent f as

$$f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^T \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x} + 3$$

Thus

$$\mathbf{g}^{(k)} = \begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}^{(k)}$$

Because $\mathbf{H}_0 = \mathbf{I}_2$

$$\mathbf{d}^{(0)} = -\mathbf{g}^{(0)} = [-2, -2]^T$$

Example of Rank One Algorithm

- The objective function is quadratic, and hence

$$\alpha_0 = \arg\min_{\alpha \geq 0} f(\mathbf{x}^{(0)} + \alpha \mathbf{d}^{(0)}) = -\frac{\mathbf{g}^{(0)T}\mathbf{d}^{(0)}}{\mathbf{d}^{(0)T}Q\mathbf{d}^{(0)}}$$

$$= \frac{\begin{bmatrix} 2 & 2 \end{bmatrix}\begin{bmatrix} 2 \\ 2 \end{bmatrix}}{\begin{bmatrix} 2 & 2 \end{bmatrix}\begin{bmatrix} 2 & 0 \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 2 \\ 2 \end{bmatrix}} = \frac{2}{3}$$

and thus

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} + \alpha_0 \mathbf{d}^{(0)} = \left[-\tfrac{1}{3}, \tfrac{2}{3}\right]$$

We then compute

$$\Delta\mathbf{x}^{(0)} = \alpha_0 \mathbf{d}^{(0)} = [-4/3, -4/3]^T$$

$$\mathbf{g}^{(1)} = Q\mathbf{x}^{(1)} = [-2/3, 2/3]^T$$

$$\Delta\mathbf{g}^{(0)} = \mathbf{g}^{(1)} - \mathbf{g}^{(0)} = [-8/3, -4/3]^T$$

Example of Rank One Algorithm

- Because
$$\Delta \mathbf{g}^{(0)T}(\Delta \mathbf{x}^{(0)} - \mathbf{H}_0 \Delta \mathbf{g}^{(0)}) = [-8/3, -4/3][4/3, 0]^T = -\frac{32}{9}$$
We obtain
$$\mathbf{H}_1 = \mathbf{H}_0 + \frac{(\Delta \mathbf{x}^{(0)} - \mathbf{H}_0 \Delta \mathbf{g}^{(0)})(\Delta \mathbf{x}^{(0)} - \mathbf{H}_0 \Delta \mathbf{g}^{(0)})^T}{\Delta \mathbf{g}^{(0)T}(\Delta \mathbf{x}^{(0)} - \mathbf{H}_0 \Delta \mathbf{g}^{(0)})} = \begin{bmatrix} 1/2 & 0 \\ 0 & 1 \end{bmatrix}$$
Therefore
$$\mathbf{d}^{(1)} = -\mathbf{H}_1 \mathbf{g}^{(1)} = [1/3, -2/3]^T$$
and
$$\alpha_1 = -\frac{\mathbf{g}^{(1)T}\mathbf{d}^{(1)}}{\mathbf{d}^{(1)}Q\mathbf{d}^{(1)}} = 1$$
We now compute
$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} + \alpha_1 \mathbf{d}^{(1)} = [0, 0]^T$$
Note that $\mathbf{g}^{(2)} = 0$, and therefore $\mathbf{x}^{(2)} = \mathbf{x}^*$. Also, $\mathbf{d}^{(0)}$ and $\mathbf{d}^{(1)}$ are Q-conjugate

Drawbacks of Rank One Formula

- The $\mathbf{H}_k$ may not be positive definite, and hence $\mathbf{d}^{(k)} = -\mathbf{H}_k \mathbf{g}^{(k)}$ may not be a descent direction
- There may be numerical problems if

$$\Delta \mathbf{g}^{(k)T} \left(\Delta \mathbf{x}^{(k)} - \mathbf{H}_k \Delta \mathbf{g}^{(k)} \right) \approx 0$$

- We seek more sophisticated update formulas that avoid the above problems
- We study two other formulas: DFP and BFGS

The DFP Algorithm

- DFP update formula

$$\mathbf{U}_k = \frac{\Delta\mathbf{x}^{(k)}\Delta\mathbf{x}^{(k)T}}{\Delta\mathbf{x}^{(k)T}\Delta\mathbf{g}^{(k)}} - \frac{\mathbf{H}_k\Delta\mathbf{g}^{(k)}\Delta\mathbf{g}^{(k)T}\mathbf{H}_k}{\Delta\mathbf{g}^{(k)T}\mathbf{H}_k\Delta\mathbf{g}^{(k)}}$$

  And accordingly,

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

- Proposed by Davidon in 1950; popularized by Flecher and Powell in 1963
- Has two ”rank one” terms

The DFP Algorithm

- The DFP algorithm is a quasi-Newton method (satisfies the quasi-Newton condition)
- ***Theorem***: In the DFP algorithm applied to the quadratic with Hessian $\mathbf{Q} = \mathbf{Q}^T$, we have

$$\mathbf{H}_{k+1}\Delta\mathbf{g}^{(i)} = \Delta\mathbf{x}^{(i)}, \quad 0 \leq i \leq k$$

- ***Proof***: Use induction
- The DFP algorithm is a conjugate direction algorithm
- ***Theorem***: Suppose $\mathbf{g}^{(k)} \neq 0$. In the DFP algorithm, if $\mathbf{H}_k$ is positive definite, then so is $\mathbf{H}_{k+1}$

The DFP Algorithm

- DFP preserves the positive definiteness of $\mathbf{H}_k$
- DFP algorithm is superior to the rank one algorithm
- DFP algorithm may have problems in some cases (e.g., very large nonquadratic problems)

The BFGS Algorithm

- The BFGS update formula

$$\mathbf{U}_k = \left(1 + \frac{\Delta\mathbf{g}^{(k)T}\mathbf{H}_k\Delta\mathbf{g}^{(k)T}}{\Delta\mathbf{g}^{(k)T}\Delta\mathbf{x}^{(k)}}\right)\frac{\Delta\mathbf{x}^{(k)}\Delta\mathbf{x}^{(k)T}}{\Delta\mathbf{x}^{(k)T}\Delta\mathbf{g}^{(k)}} - \frac{\mathbf{H}_k\Delta\mathbf{g}^{(k)}\Delta\mathbf{x}^{(k)T} + (\mathbf{H}_k\Delta\mathbf{g}^{(k)}\Delta\mathbf{x}^{(k)T})^T}{\Delta\mathbf{g}^{(k)T}\Delta\mathbf{x}^{(k)}}$$

$$\mathbf{H}_{k+1} = \mathbf{H}_k + \mathbf{U}_k$$

- Proposed independently by Broyden, Fltcher, Goldfarb, and Shanno (BFGS) in 1970

The BFGS Algorithm

- The BFGS formula is derived from the DFP formula using ”complementarity”
- Consider the quasi-Newton condition, the approximation of the Hessian inverse should satisfy

$$\mathbf{H}_{k+1}\Delta\mathbf{g}^{(i)} = \Delta\mathbf{x}^{(i)}, \quad 0 \leq i \leq k$$

- Think of $\mathbf{B}_k$ as an approximation to the Hessian itself ($\mathbf{B}_k^{-1} = \mathbf{H}_k$), we require $\mathbf{B}_{k+1}$ to satisfy

$$\mathbf{B}_{k+1}\Delta\mathbf{x}^{(i)} = \Delta\mathbf{g}^{(i)}, \quad 0 \leq i \leq k$$

- The roles of $\Delta\mathbf{g}^{(i)}$ and $\Delta\mathbf{x}^{(i)}$ are interchanged
- Call the above the ”complementary quasi-Newton” condition

The BFGS Algorithm

- Recall the DFP update for approximation $\mathbf{H}_k$

$$\mathbf{H}_{k+1}^{DFP} = \mathbf{H}_k + \frac{\Delta\mathbf{x}^{(k)}\Delta\mathbf{x}^{(k)T}}{\Delta\mathbf{x}^{(k)T}\Delta\mathbf{g}^{(k)}} - \frac{\mathbf{H}_k\Delta\mathbf{g}^{(k)}\Delta\mathbf{g}^{(k)T}\mathbf{H}_k}{\Delta\mathbf{g}^{(k)T}\mathbf{H}_k\Delta\mathbf{g}^{(k)}}$$

- Using the complementarity concept, the DFP can be used, and the update for the approximation $\mathbf{B}_k$ is

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \frac{\Delta\mathbf{g}^{(k)}\Delta\mathbf{g}^{(k)T}}{\Delta\mathbf{g}^{(k)T}\Delta\mathbf{x}^{(k)}} - \frac{\mathbf{B}_k\Delta\mathbf{x}^{(k)}\Delta\mathbf{x}^{(k)T}\mathbf{B}_k}{\Delta\mathbf{x}^{(k)T}\mathbf{B}_k\Delta\mathbf{x}^{(k)}}$$

- This is the BFGS update of $\mathbf{B}_k$
- The above formula satisfies the complementary quasi-Newton condition

The BFGS Algorithm

- The previous formula for updating $\mathbf{B}_{k+1}$ is not immediately useful because what we need is the inverse Hessian

$$\mathbf{H}_{k+1}^{BFGS} = (\mathbf{B}_{k+1})^{-1}$$

- There exists some tricks for computing the inverse, as shown in the following Sherman-Morrison formula

The BFGS Algorithm

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

$$(\mathbf{A} + \mathbf{u}\mathbf{v}^T)^{-1} = \mathbf{A}^{-1} - \frac{(\mathbf{A}^{-1}\mathbf{u})(\mathbf{v}^T\mathbf{A}^{-1})}{1+\mathbf{v}^T\mathbf{A}^{-1}\mathbf{u}}$$

- Interpretation of the lemma: if $\mathbf{A}^{-1}$ is known, then this formula provides a "numerically cheap" way to compute the inverse of $\mathbf{A}$ corrected by the matrix $\mathbf{u}\mathbf{v}^T$

The BFGS Algorithm

- Note that the previous formula is of the form

$$\mathbf{B}_{k+1} = \mathbf{B}_k + \mathbf{u}_1\mathbf{v}_1^T + \mathbf{u}_2\mathbf{v}_2^T$$

- Hence

$$\mathbf{B}_{k+1}^{-1} = (\mathbf{B}_k + \mathbf{u}_1\mathbf{v}_1^T + \mathbf{u}_2\mathbf{v}_2^T)^{-1}$$

- Apply the lemma twice to $\mathbf{B}_{k+1}$ and replace $\mathbf{B}_k^{-1}$ by the symbol $\mathbf{H}_k$

$$\mathbf{H}_{k+1}^{BFGS} = \mathbf{H}_k + \left(1 + \frac{\Delta\mathbf{g}^{(k)T}\mathbf{H}_k\Delta\mathbf{g}^{(k)T}}{\Delta\mathbf{g}^{(k)T}\Delta\mathbf{x}^{(k)}}\right)\frac{\Delta\mathbf{x}^{(k)}\Delta\mathbf{x}^{(k)T}}{\Delta\mathbf{x}^{(k)T}\Delta\mathbf{g}^{(k)}} - \frac{\mathbf{H}_k\Delta\mathbf{g}^{(k)}\Delta\mathbf{x}^{(k)T} + (\mathbf{H}_k\Delta\mathbf{g}^{(k)}\Delta\mathbf{x}^{(k)T})^T}{\Delta\mathbf{g}^{(k)T}\Delta\mathbf{x}^{(k)}}$$

- This is the BFGS formula for updating $\mathbf{H}_k$

Summary of the BFGS Algorithm

- BFGS is the ”complementary” formula to DFP
- By the nature of complementarity, the BFGS formula inherits the properties of DFP
  - BFGS formula satisfies the quasi-Newton condition
  - BFGS has the conjugate directions property
  - BFGS inherits the positive definiteness property of the DFP, that is, if $\mathbf{g}^{(k)} \neq 0$ and $\mathbf{H}_k > 0$, then $\mathbf{H}_{k+1}^{BFGS} > 0$

Summary of the BFGS Algorithm

- The BFGS is reasonably robust when the line search is in-exact (could save time in line search part)
- BFGS is often far more efficient than the DFP formula
- For nonquadratic problems, quasi-Newton algorithms will not usually converge in n steps
- Some modifications may be necessary (e.g., reinitialize the direction vector to the negative gradient after every few iterations)
- Widely applied (e.g., in Matlab, Mathematical)