Optimization Methods
Lecture 7

Solmaz S. Kia
Mechanical and Aerospace Engineering Dept.
University of California Irvine
solmaz@uci.edu

Reading: page 285-297 from Ref[2].

Numerical solvers for unconstrained optimization

Unconstrained optimization:

$$x^\star = \underset{x \in \mathbb{R}^n}{\text{argmin}}\ f(x)$$

Iterative solution method $x_{k+1} = x_k + \alpha_k\, d_k$

Observations:

- Steepest descent algorithm can be very slow with lots of zig-zaging
- Newton method is faster but numerically is expensive due to information equipment associated with the evaluation, storage and inversion of Hessian.

Q: Is it possible to accelerate convergence with low numerical cost?

Some observations about successive descent methods

$$x_{k+1} = x_k - \alpha_k S_k g_k$$

where

- S_k is a symmetric $n \times n$ matrix and to guarantee that descent S_k should be positive definite
  - $S_k = (\nabla^2 f(x_k))^{-1}$: Newtown method
  - $S_k = I$: Steepest descent method
- α_k is chosen to minimize $f(x_{k+1})$.

Note: It is always a good idea to choose S_k as an approximation to the inverse of the Hessian

Some observations about successive descent methods

Rate of convergence of

$$x_{k+1} = x_k - \alpha_k S_k g_k, \text{ where } S_k > 0, \quad \alpha_k = \text{argmin } f(x_k - \alpha_k S_k g_k) \quad \text{(Al-1)}$$

solving the standard quadratic unconstrained optimization problem with cost $f(x) = \frac{1}{2}x^\top Qx - b^\top x$:

Note that $\alpha_k = \underset{\alpha>0}{\text{argmin}} f(x_k - \alpha S_k g_k) = \frac{g_k^\top S_k g_k}{g_k^\top S_k Q S_k g_k}$, where $g_k = \nabla f(x_k) = Qx_k - b$.

Let $x^\star$ be the unique minimum point of f, and define $E(x) = \frac{1}{2}(x - x^\star)^\top Q(x - x^\star)$. Then for the algorithm (A1-1) there holds at every step k

$$E(x_{k+1}) \leqslant \left(\frac{B_k - b_k}{B_k + b_k}\right)^2 E(x_k),$$

where b_k and B_k are, respectively, the smallest and largest eigenvalues of the matrix $S_k Q$.

Note: the observation above supports the idea that S_k should be chosen close to Q^{-1} (note that in this case b_k gets close to B_k and the rate improves substantially)

Fundamental idea of Quasi Newton Methods:

- Try to construct the inverse Hessian, or an approximation of it, using information gathered as the descent process progresses.
- The current approximation H_k is then used at each stage to define the next descent direction by setting $S_k = H_k$ in the modified Newton method.

The observations below gives us the guidelines to design H_k such that as k increases, H_k approximates the Hessian $(\nabla^2 f(x_k))^{-1}$

Let

- $g_k = \nabla f(x_k)$,
- $q_k = g_{k+1} - g_k$,
- $p_k = x_{k+1} - x_k$

then $g(x_{k+1}) = g(x_k + p_k) \approx g(x_k) + \nabla^2 f(x_k)^\top p_k$. Therefore,

$$q_k \approx \nabla^2 f(x_k)\, p_k$$

or

$$(\nabla^2 f(x_k))^{-1} q_k \approx p_k$$

Quasi Newton Methods

- We observe that if $(\nabla^2 f(x_k))$ was constant and equal to F and also $\{p_k\}_{k=0}^{n-1}$ was a set of n linearly independent directions, then we obtain

$$F = \begin{bmatrix} q_0 & q_1 & \cdots & q_{n-1} \end{bmatrix} \begin{bmatrix} p_0 & p_1 & \cdots & p_{n-1} \end{bmatrix}^{-1}$$

  This shows that for this special case, it is possible to construct the Hessian from the the information gathered as the descent process progresses!
- Consider again the case of constant Hessian $F = \nabla^2 f(x_k)$. In this case, we know $q_k \approx F\,p_k$ or equivalently $F^{-1}\,q_k \approx p_k$, for all k.

Based on the observations above, we set expect that H_k that wants to approximate $(\nabla^2 f(x_k))^{-1}$ satisfy

1. $H_{k+1} q_i = p_i, \quad i \in \{0, 1, \cdots, k\}$
2. H_k symmetric
3. $H_k > 0$

For the case of constant Hessian, after n linearly independent steps, then we have $H_n = F^{-1}$.

Quasi Newton Methods: Rank One Correction

The first quasi-Newton method is proposed as follows with $a_k \in \mathbb{R}$ and $z_k \in \mathbb{R}^n$ as design variables (**Rank One Correction**)

$$H_{k+1} = H_k + a_k z_k z_k^\top$$

Note that H_k is symmetric. a_k and z_k are designed such that $H_{k+1} q_k = p_k$, which results in

$$\boxed{H_{k+1} = H_k + \frac{(p_k - H_k q_k)(p_k - H_k q_k)^\top}{q_k^\top (p_k - H_k q_k)}}$$

Theorem Let F be a fixed symmetric matrix and suppose that $p_0, p_1, p_2, \cdots, p_k$ are given vectors. Define the vectors $q_i = F p_i$, $i \in \{0, 1, 2, \cdots, k\}$. Starting with any initial symmetric matrix H_0 let

$$H_{i+1} = H_i + \frac{(p_i - H_i q_i)(p_i - H_i q_i)^\top}{q_i^\top (p_i - H_i q_i)}$$

Then

$$p_i = H_{k+1} q_i \quad \text{for} \quad i \in \{0, 1, \cdots, k\}.$$

In **Rank One Correction**

- H_k is symmetric
- But not necessarily positive definite (we need $q_k^\top (p_k - H_k q_k) > 0$ which is not guaranteed at all times)

Quasi Newton Methods: Rank One Correction

The first quasi-Newton method is proposed as follows with $a_k \in \mathbb{R}$ and $z_k \in \mathbb{R}^n$ as design variables (**Rank One Correction**)

$$H_{k+1} = H_k + a_k z_k z_k^\top$$

Note that H_k is symmetric. a_k and z_k are designed such that $H_{k+1} q_k = p_k$, which results in

$$\boxed{H_{k+1} = H_k + \frac{(p_k - H_k\, q_k)(p_k - H_k\, q_k)^\top}{q_k^\top (p_k - H_k\, q_k)}}$$

Theorem Let F be a fixed symmetric matrix and suppose that $p_0, p_1, p_2, \cdots, p_k$ are given vectors. Define the vectors $q_i = F p_i$, $i \in \{0, 1, 2, \cdots, k\}$. Starting with any initial symmetric matrix H_0 let

$$H_{i+1} = H_i + \frac{(p_i - H_i\, q_i)(p_i - H_i\, q_i)^\top}{q_i^\top (p_i - H_i q_i)}$$

Then

$$p_i = H_{k+1} q_i \quad \text{for} \quad i \in \{0, 1, \cdots, k\}.$$

In **Rank One Correction**

- H_k is symmetric
- But not necessarily positive definite (we need $q_k^\top (p_k - H_k q_k) > 0$ which is not guaranteed at all times)

Davidon-Fletcher-Powell (DFP) method:
Initialization $k = 0$: start by $x_0 \in^n$ and any $H_0 > 0$
Step 1. Set $d_k = -H_k g_k$.
Step 2. obtain $\alpha_k =\subset \alpha > 0 \mathrm{argmin} f(x_k + \alpha d_k)$. Then obtain $x_{k+1} = x_k + \alpha d_k$ and $p_k = \alpha_k d_k$, and g_{k+1}.
Step 3. Set $q_k = g_{k+1} - g_k$ and

$$H_{k+1} = H_k + \frac{p_k p_k^\top}{p_k^\top q_k} - \frac{H_k q_k q_k^\top H_k}{q_k^\top H_k q_k}.$$

check the stoping condition. If not satisfied update k and return to Step 1.

- at each step the inverse Hessian is updated by sum of two symmetric rank one matrices (called often **Rank Two Procedure**)
- also referred at **Variable Metric Method**
- starting from a positive definite H_0, the subsequently generated H_k are positive definite

Quasi Newton Methods: Davidon-Fletcher-Powell (DFP) method

Davidon-Fletcher-Powell (DFP) method for a quadratic cost function

- generates the directions of the conjugate gradient method
- while constructing the inverse Hessian

Theorem: If cost function f is quadratic with positive definite Hessian F, then for the DFP method we have

- $p_i^\top F p_j = 0, \quad 0 \leqslant i < j \leqslant k$
- $H_{k+1} F p_i = p_i$ for $0 \leqslant i \leqslant k$.
