Optimization Methods		Date:	*09 March 2018*
Instructor: *Sujit Prakash Gujar*		Scribes:	Amit Kumar Madhur Modi Prachi Agrawal

Lecture 17: Non-Linear Programming - Finding roots of a differentiable function

1 Recap

- We consider *unconstrained non-linear program* of the form

$$\min_{x \in \mathbb{R}^n} f(x)$$

- *First order necessary condition* :- If $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable and x^* is a local minimum , then $\nabla f(x^*) = 0$
- *Second order necessary condition* :- If $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable and x^* is a local minimum , then $\nabla^2 f(x^*)$ is *positive semidefinite* .
- *Second order sufficient condition* :- If $f : \mathbb{R}^n \to \mathbb{R}$ is twice differentiable at x^* and
  - $\nabla f(x^*) = 0$
  - $\nabla^2 f(x^*)$ is *positive definite*

  then x^* is local minimum .
- For some functions, we can find its gradient but can we can solve it equating to 0. *eg* : $x^2 + y^2 siny + y^3 cos6y$.
- *Iterative descent method* :- We select a point x_0 and check if it satisfies critical point condition ($\nabla f(x_0) = 0$) . If not, we select another point x_k such that $f(x_{k+1}) < f(x_k)$. Since gradient is in the direction in which function increases, so we move opposite to gradient .

$$x_{k+1} = x_k - \alpha_k \frac{\nabla f(x_k)}{||\nabla f(x_k)||}$$

2 Bisection Method

If $f : \mathbb{R} \to \mathbb{R}$ is a real continuous function and $f(a) > 0$ and $f(b) < 0$, then there must exit at least one root between a and b .We pick another point c such that

$$c = \frac{a+b}{2}$$

We continue this step unless we find root of f .

Formally, this can written as :-

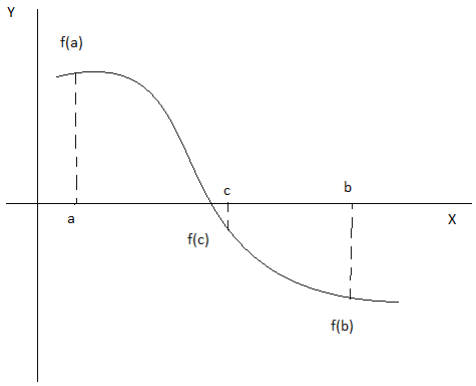

Figure 1: Bisection Method

Algorithm 1 Bisection Method

Input a and b
if $f(a) > 0$ and $f(b) < 0$ **then**
  $c = \dfrac{a+b}{2}$
end if
if $f(c) > 0$ **then**
  $a \leftarrow$c
  $b \leftarrow$b
end if
if $f(c) < 0$ **then**
  $a \leftarrow$a
  $b \leftarrow$c
else
  RETURN c
end if

After every iteration the size of the interval reduces by half. Thus after k iterations of the algorithm, the size of the interval becomes: $\frac{b-a}{2^k}$ .We stop once size of interval drops below ε .So we have :-

$$\frac{b-a}{2^k} < \varepsilon$$

$$2^k > \frac{b-a}{\varepsilon}$$

$$k > log_2 \frac{b-a}{\varepsilon} \tag{1}$$

3 Newton-Raphson Method

3.1 Introduction

Let $f : \mathbb{R} \to \mathbb{R}$ be a continuous, differentiable function. Let x_0 be the initial assumption of the root of the function $f(x) = 0$. The equation of tangent at x_0 is given by

$$f(x) = f(x_0) + (x - x_0)f(x_0)$$

The assumed root x_0 need not be the actual root of f. We take the point of intersection of the tangent with the x-axis as the next approximation to the root of $f(x) = 0$. Substituting $f(x) = 0$ in the above equation, we get

$$x = x_0 - \frac{f(x_0)}{f'(x_0)}$$

Therefore we keep on updating our guess using the above formulation until convergence. Thus,

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \tag{2}$$

This approach is called Newton's method.

Example 1: Given $f(x) = x^2 - 4x - 7$, find roots of $f(x) = 0$ near $x = 5$
$f'(x) = 2x - 4$
The update rule becomes,

$$x_{k+1} = x_k - \frac{(x^2 - 4x - 7)}{(2x - 4)}$$

With the given initial guess of $x_0 = 5$, we can produce the following values:

x_0	5
x_1	5.33333
x_2	5.31667
x_3	5.31662
x_4	5.31662

Therefore the final answer is 5.317.

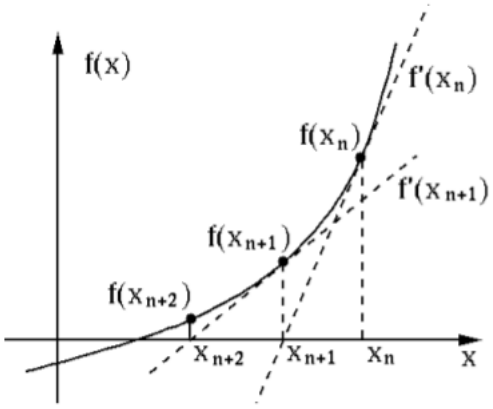

Figure 2: Convergence of Newtonś Method(source : Google Image(Quora))

3.2 Convergence of Newton's method

The Taylor series expansion of f around a point x_k gives

$$f(x) = f(x_k) + (x - x_k)f'(x_k) + \frac{(x - x_k)^2}{2!}f''(x_k) + ...$$

Using then exact expansion up to second order derivative we get

$$f(x) = f(x_k) + (x - x_k)f'(x_k) + \frac{(x - x_k)^2}{2!}f''(\zeta_k) \tag{3}$$

where $\zeta \epsilon [x...x_k]$ or $\zeta \epsilon [x_k...x]$

Let $f(\alpha) = 0$
Substituting x = α in equation (2) and equating it to 0, we get

$$f(x_k) + (\alpha - x_k)f'(x_k) + \frac{(\alpha - x_k)^2}{2!}f''(\zeta_k) = 0 \tag{4}$$

The term (α - x_k) in equation (3) denotes the error in Newton's approximation. Let it be ε_k for the k^{th} iteration.Therefore,

$$f(x_k) + (\alpha - x_k)f'(x_k) + \frac{\varepsilon_k^2}{2}f''(\zeta_k) = 0$$

$$\frac{f(x_k}{f'(x_k)}) + (\alpha - x_k) + \frac{\varepsilon_k^2}{2}\frac{f''(\zeta_k)}{f'(x_k)} = 0$$

$$\alpha - x_k + \frac{f(x_k}{f'(x_k)}) + \frac{\varepsilon_k^2}{2}\frac{f''(\zeta_k)}{f'(x_k)} = 0$$

$$\alpha - (x_k - \frac{f(x_k}{f'(x_k)})) + \frac{\varepsilon_k^2}{2}\frac{f''(\zeta_k)}{f'(x_k)} = 0 \tag{5}$$

From equation (1) we have

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Using equation (1) in equation (4), we get

$$\alpha - (x_{k+1}) + \frac{\varepsilon_k^2}{2}\frac{f''(\zeta_k)}{f'(x_k)} = 0$$

$$\alpha - (x_{k+1}) = -\frac{\varepsilon_k^2}{2}\frac{f''(\zeta_k)}{f'(x_k)}$$

Since $(\alpha - x_k)$ denoted by ε_k is the error at k^{th} iteration, we get

$$\varepsilon_{k+1} = -\frac{\varepsilon_k^2}{2}\frac{f''(\zeta_k)}{f'(x_k)}$$

$$\varepsilon_{k+1} = \varepsilon_k^2(\frac{-1}{2}\frac{f''(\zeta_k)}{f'(x_k)}) \tag{6}$$

Here ε_{k+1} is the error at $(k+1)^{th}$ iteration.
Let the initial error be ε_0, $\varepsilon < 1$
Therefore,

$$\varepsilon_{k+1} \leq \varepsilon_k^2 \tag{7}$$

Let say we want to check till $\varepsilon_{k+1} \leq \epsilon$
Error progresses as,

$$\varepsilon_0 \to \varepsilon_0^2 \to \varepsilon_0^4 \to ... \to \varepsilon_0^{2^k}$$

As per the termination condition,

$$\varepsilon_0^{2^k} \leq \epsilon \tag{8}$$

Taking log on both sides,

$$2^k log(\varepsilon_0) \leq log(\epsilon) \tag{9}$$

Notice that since $\epsilon < 1$ and $2^k \ \epsilon \ \mathbb{N}$, from equation (7) clearly

$$\varepsilon_0 < 1 \tag{10}$$

Therefore, clearly $\log(\varepsilon_0) < 0$ and $\log(\epsilon) < 0$
Using the above result in equation (8), we get

$$2^k > \frac{log(\epsilon)}{log(\varepsilon_0)}$$

$$k > \log_2 \frac{log(\epsilon)}{log(\varepsilon_0)} \tag{11}$$

In general, Newton-Raphson method is faster than Gradient descent.

3.3 Update rule and necessary condition

We aim to minimize $f(x)$. Let x^* be local minima of f, hence $\nabla f(x^*) = 0$. The newton raphson method is used to find roots of f. So we will solve $g(x) = \nabla f(x^*) = 0$. The update rule will be given by:

$$x_{k+1} = x_k - \frac{g(x_k)}{g^{'}(x_k)}$$

If the vector x is of n dimension then the update rule is given as:

$$x_{k+1} = x_k - (\nabla^2 f(x_k))^{-1} \nabla f(x_k) \tag{12}$$

As we know that

$$x_{k+1} = x_k + \alpha_k d_k$$

Choosing $\alpha_k = 1$,$d_k = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k)$

$$f(x_{k+1}) = f(x_k + \alpha_k d_k)$$

Using Taylor Expansion we get

$$f(x_{k+1}) = f(x_k) + \alpha_k d_k^T \nabla f(x_k)$$

As we are trying to reach our local minimum at each step so $f(x_{k+1}) < f(x_k)$. Thus we have the following inequality

$$d_k^T \nabla f(x_k) < 0$$

The class of algorithms which follow the above inequality are known as gradient descent methods. Since for Newton Raphson method $d_k = -(\nabla^2 f(x_k))^{-1} \nabla f(x_k)$

$$\implies -\nabla f(x_k)^T (\nabla^2 f(x_k))^{-1} \nabla f(x_k) < 0$$

$$\implies \nabla f(x_k)^T (\nabla^2 f(x_k))^{-1} \nabla f(x_k) > 0 \tag{13}$$

So, Newton raphson method will belong to the gradient descent class of algorithm if Hessian matrix is positive definite.

Example 2: Given

$$f(x) = (x-5)^2 - 2$$

$$x_0 = 0$$

Find x_1 using Newton Raphson's method

Let $g(x) = \nabla f(x)$

$$\implies g(x) = 2(x-5)$$

$$\implies g'(x) = 2$$

Using the update rule we can find x_1 which is given by the following equation:

$$x_1 = x_0 - \frac{g(x_0)}{g'(x_0)}$$

$$\implies x_1 = 0 - \frac{-10}{2}$$

$$\implies x_1 = 5$$

Since $f(x) = (x-5)^2 - 2$ has minima at point $x = 5$ so using newton raphson we reached the local

minima in this example in only one step.

Example 3: Given

$$f(x,y) = 2x^2 + y^2$$
$$x_0 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Find x_1 using Newton Raphson's method

$$\implies \nabla f(x,y) = \begin{pmatrix} 4x \\ 2y \end{pmatrix}$$
$$\implies \nabla^2 f(x,y) = \begin{pmatrix} 4 & 0 \\ 0 & 2 \end{pmatrix}$$

As per udpate rule x_1 will be given by the following equation:

$$x_1 = x_0 - (\nabla^2 f(x_0))^{-1} \nabla f(x_0)$$
$$\implies x_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix} - \begin{pmatrix} \frac{1}{4} & 0 \\ 0 & \frac{1}{2} \end{pmatrix} \begin{pmatrix} 4 \\ 2 \end{pmatrix}$$

$$\implies x_1 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$$

4 Additional Algorithms

As we know that

$$x_{k+1} = x_k + \alpha_k d_k$$

Let's assume that d_k is given by

$$d_k = -D_k \nabla f(x_k) \tag{14}$$

Modified Newton's Method

Newton Raphson Mehtod depend upon the computation of inverse of the Hessian matrix in every step and computation of inverse of matrix is computationally expensive. So to reduce computational complexity use same D_k for p steps where p>1.

Discretized Newton's Method

$D_k = (H(x_k))^{-1}$ where H is approximate of $\nabla^2 f(x_k)$ calculated using finite difference method.

Diagonally scaled steepest descent

In this method D_k is chosen as diagonal matrix, where j^{th} entry of the diagonal matrix is given by:

$$d_j = (\frac{\partial^2 f(x_k)}{\partial x_j^2})^{-1} \tag{15}$$