**10-725: Convex Optimization** **Fall 2013**

# Lecture 9: Newton Method

*Lecturer: Barnabas Poczos/Ryan Tibshirani* *Scribes: Wen-Sheng Chu, Shu-Hao Yu*

**Note**: *LaTeX template courtesy of UC Berkeley EECS dept.*

## 9.1 Motivation

Newton method is originally developed for finding a root of a function. It is also known as **Newton-Raphson method**. The problem can be formulated as, given a function $f : \mathbb{R} \to \mathbb{R}$, finding the point $x^\star$ such that $f(x^\star) = 0$.

Figure 9.1 illustrates another motivation of Newton method. Given a function $f$, we want to approximate it at point $x$ with a quadratic function $\widehat{f}(x)$. We move our the next step determined by the optimum of $\widehat{f}$.

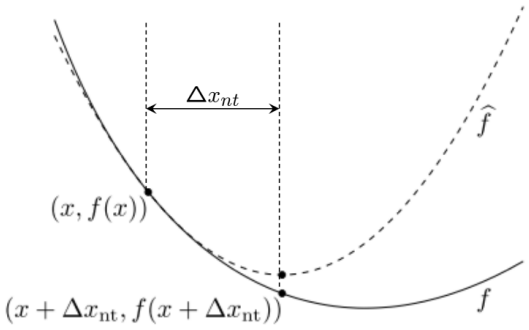



Figure 9.1: Motivation for quadratic approximation of a function.

## 9.2 History

Babylonian people first applied the Newton method to find the square root of a positive number $S \in \mathbb{R}_+$. The problem can be posed as solving the equation $f(x) = x^2 - S = 0$. They realized the solution can be achieved by applying the iterative update rule:

$$x_{n+1} = \frac{1}{2}(x_k + \frac{S}{x_k}) = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k^2 - S}{2x_k}. \tag{9.1}$$

This update rule converges to the square root of $S$, which turns out to be a special case of Newton method. This could be the first application of Newton method.

The starting point affects the convergence of the Babylonian's method for finding the square root. Figure 9.2 shows an example of solving the square root for $S = 100$. $x$- and $y$-axes represent the number of iterations and the variable, respectively. Given two starting points 0 and 50, the method converges to the true value 10. However, when a negative starting point is used, the method fail to converge to the true square root.

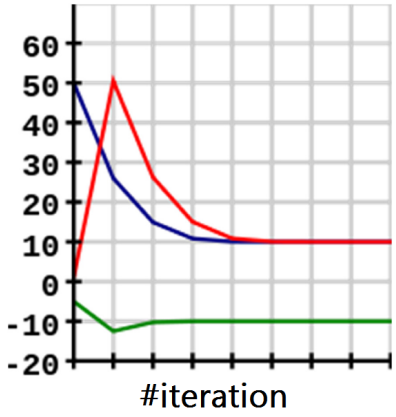



Figure 9.2: Convergence in different starting points when finding the square root

After the Babylonian's method, the formal Newton method began to evolve from **Isaac Newton** (1669) for finding roots of polynomials, **Joseph Raphson** (1690) for finding roots of polynomials, **Thomas Simpson** (1740) for solving general nonlinear equations, to **Arthur Cayley** (1879) for finding complex roots of polynomials.

## 9.3 Newton Method for Finding Roots

Recall our problem is to find a point $x^\star$ such that $f(x^\star) = 0$, given the function $f : \mathbb{R} \to \mathbb{R}$. From an arbitrary point $x$, we can use 1st order Taylor expansion for linearly approximating $f(x)$:

$$f(x + \Delta x) = f(x) + f'(x)\Delta x + o(|\Delta x|). \tag{9.2}$$

The last part, $o(|\Delta x|)$, can be neglected when approximating the function linearly. In order to find the root of this function, we set the linear approximation to 0:

$$\begin{aligned} & 0 \approx f(x) + f'(x)\Delta x, \\ \Rightarrow \quad & x^\star - x = \Delta x = -\frac{f(x)}{\phi'(x)}. \end{aligned} \tag{9.3}$$

Therefore, for each iteration, the iterative update rule becomes:

$$x_{n+1} = x_k - \frac{f(x)}{f'(x)}. \tag{9.4}$$

Figure 9.3 illustrates an example for finding the root of function $f(x)$ with a starting point $x_0$. Using the linear approximation at $x_0$, we get a line $\widehat{f}(x) = \widehat{f}(x_0) + f'(x_0)(x - x_0)$, and by solving $\widehat{f}(x) = 0$, we get a new point $x = x_0 + \Delta x_{NT}$. This process repeats with $x$ as a starting point for the next iteration until it converges to the truth of the nonlinear function. In the following we discuss two extensions of Newton method, including the generalization to multivariate functions and solving minimization problems.

### 9.3.1 Generalize to multivariate functions

Newton method can be easily generalized to multivariate functions. Denote the function $F : \mathbb{R}^n \to \mathbb{R}^m$ mapping a $n$-dimensional vector to a $m$-dimensional vector. Similar to the 1-D case, we can again apply the

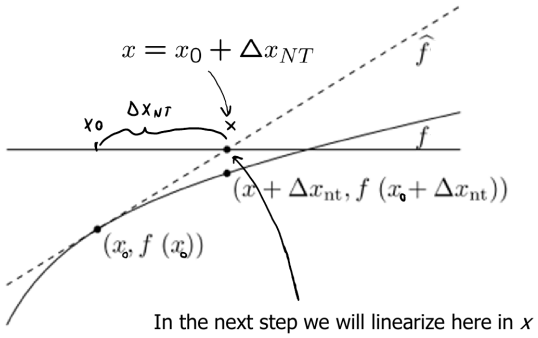



Figure 9.3: An updating step in Newton method

Taylor approximation:

$$0_m = F(x^\star) = F(x + \Delta x) = F(x) + \underbrace{\nabla F(x)}_{\mathbb{R}^{m\times n}} \underbrace{\Delta x}_{\mathbb{R}^n} + o(|\Delta x|), \tag{9.5}$$

where $0_m \in \mathbb{R}^m$ is a zero vector. Again the last term $o(|\Delta x|)$ can be neglected in the linear approximation. Therefore, we can obtain the similar iterations by:

$$\begin{aligned} & 0_m = F(x) + \nabla F(x)\Delta x, \\ \Rightarrow \quad & \Delta x = -[\nabla F(x)]^{-1} F(x). \end{aligned} \tag{9.6}$$

Note that pseudo inverse can be applied when the inverse of $\nabla F(x)$ does not exist. As a result, plugging in $\Delta x = x_{n+1} - x_k$, we have the update rule:

$$x_{k+1} = x_k - [\nabla F(x_k)]^{-1} F(x_k). \tag{9.7}$$

### 9.3.2 Generalize to minimization problems

Newton method can be also generalized for minimizing a function. Suppose a function $f : \mathbb{R}^n \to \mathbb{R}$ is differentiable, the goal is to find the minimum of the function, $\min_{x\in\mathbb{R}^n} f(x)$. The minimization problem can be converted into solving the root-finding problem $\nabla f(x) = 0_n$, where $\nabla f(x) : \mathbb{R}^n \to \mathbb{R}^n$. Applying Newton method again, we obtain $\nabla f(x) + \nabla^2 f(x)\Delta x = 0_k$. Then the update rule becomes $\Delta x = x_{k+1} - x_k = -[\nabla^2 f(x_k)]^{-1}\nabla f(x_k)$, which we term the *Newton step*.

## 9.4 Properties of Newton Method

**Lemma 9.1** *(Descent direction) If $\nabla^2 f \succ 0$, then Newton step is a descent direction.*

**Proof:** We know that if a vector has negative inner product with the gradient vector, then that direction is a descent direction. Recall that the Newton step is given by $\Delta x = x_{k+1} - x_k = -[\nabla^2 f(x_k)]^{-1}\nabla f(x_k)$. Because $\nabla^2 f$ is positive definite, we know $\nabla^2 f(x_k)$ is invertible. As a result, we have:

$$\nabla f(x_k)^\top \Delta x = -\nabla f(x_k)^\top [\nabla^2 f(x_k)]^{-1}\nabla f(x_k) < 0, \tag{9.8}$$

showing that the Newton step is a descent direction. ∎

**Useful properties:** There are some useful properties about Newton method:

1. Quadratic convergence in the neighborhood of a strict local minimum (under some conditions).
2. It can break down if $\nabla^2 f$ is degenerated (not invertible).
3. It can diverge.
4. It can be trapped in a loop.
5. It can converge to a loop (oscillating).

## 9.5 Convergence Rate

### 9.5.1 Review the rates

Here we review the convergence rates and some examples. To better explain the convergence rate, we define

$$\delta = \lim_{i\to\infty} \frac{|s_{i+1} - \bar{s}|}{|s_i - \bar{s}|}. \tag{9.9}$$

For a sequence $\{s_i\}$ and $\lim_{i\to\infty} s_i = \bar{s}$, we say the sequence exhibits a *linear* convergence rate by showing $\delta < 1$. When $\delta = 0$, the rate is *superlinear*. When $\delta = 1$, the rate is *sublinear*. When $\lim_{i\to\infty} \frac{|s_{i+1}-\bar{s}|}{|s_i-\bar{s}|^2} < \infty$, the rate is *quadratic*. Below we list a few examples with different convergence rates:

1. **Linear** convergence rate ($\delta < 1$):

$$\text{Example: } s_i = cq^i, 0 < q < 1, \bar{s} = 0,$$
$$\lim_{i\to\infty} \frac{|s_{i+1} - \bar{s}|}{|s_i - \bar{s}|} = \lim_{i\to\infty} \frac{cq^{i+1}}{cq^i} = q < 1.$$

2. **Superlinear** convergence rate ($\delta = 0$):

$$\text{Example: } s_i = \frac{c}{i!}, \bar{s} = 0,$$
$$\lim_{i\to\infty} \frac{|s_{i+1} - \bar{s}|}{|s_i - \bar{s}|} = \lim_{i\to\infty} \frac{ci!}{c(i+1)!} = \lim_{i\to\infty} \frac{1}{i+1} = 0.$$

3. **Sublinear** convergence rate ($\delta = 1$):

$$\text{Example: } s_i = \frac{c}{i^a}, a > 0, \bar{s} = 0,$$
$$\lim_{i\to\infty} \frac{|s_{i+1} - \bar{s}|}{|s_i - \bar{s}|} = \lim_{i\to\infty} \frac{ci^a}{c(i+1)^a} = \lim_{i\to\infty} (\frac{i}{i+1})^a = 1.$$

4. **Quadratic** convergence rate ($\lim_{i\to\infty} \frac{|s_{i+1}-\bar{s}|}{|s_i-\bar{s}|^2} < \infty$):

$$\text{Example: } s_i = q^{2^i}, 0 < q < 1, \bar{s} = 0,$$
$$\lim_{i\to\infty} \frac{|s_{i+1} - \bar{s}|}{|s_i - \bar{s}|^2} = \lim_{i\to\infty} \frac{q^{2^{i+1}}}{q^{2^i}} = \lim_{i\to\infty} q^{2^i} = 0.$$

### 9.5.2 Convergence rate for Newton method

Recall that the goal of Newton method is to find the optimal $x^\star$ such that $f(x^\star) = 0$, where $f : \mathbb{R} \to \mathbb{R}$. This section shows the *quadratic* convergence rate of Newton method.

Assume $f$ has continuous second derivative at $x^\star$. By the 2nd-order Taylor approximation, we know for a $\xi_k$ in between $x_k$ and $x^\star$:

$$0 = f(x^\star) = f(x_k) + \underbrace{\nabla f(x_k)}_{\text{grad. at } x_k}(x^\star - x_k) + \frac{1}{2}\underbrace{\nabla^2 f(\xi_k)}_{\text{Hess. at } \xi_k}(x^\star - x_k)^2. \tag{9.10}$$

Suppose $[\nabla^{-1} f(x_k)]$ exists and multiply $[\nabla^{-1} f(x_k)]$ to both sides of Eq. (9.10), we have

$$\begin{aligned} & 0 = [\nabla^{-1} f(x_k)]f(x_k) + (x^\star - x_k) + \frac{1}{2}[\nabla^{-1} f(x_k)]\nabla^2 f(\xi_k)(x^\star - x_k)^2 \\ \Rightarrow \quad & \underbrace{([\nabla^{-1} f(x_k)]f(x_k) - x_k)}_{-x_{k+1}} + x^\star = -\frac{1}{2}[\nabla^{-1} f(x_k)]\nabla^2 f(\xi_k)\underbrace{(x^\star - x_k)^2}_{\epsilon_k^2} \\ \Rightarrow \quad & \epsilon_{k+1} = -\frac{\nabla^2 f(\xi_k)}{2\nabla f(x_k)}\epsilon_k^2. \end{aligned} \tag{9.11}$$

Let $M = \sup_{x,y} \frac{|\nabla^2 f(x)|}{|2\nabla f(y)|} < \infty$ is a bounded quantity, then we can say Newton method has a quadratic convergence rate by showing:

$$|\epsilon_{k+1}| \leq M\epsilon_k^2. \tag{9.12}$$

Further if we assume $|\epsilon_0| = |x^\star - x_0| < 1$, we can say the error $\epsilon_k$ converges to 0 with quadratic rate. Hereby we complete the proof the convergence rate for Newton method. □

## 9.6 Problematic Cases

Newton method is in general very fast, but may fail in a few problematic cases. Below we illustrate a few examples showing the Newton methods can (1) be sensitive to initialization point, (2) get into cycles, (3) diverge, (4) converge only in linear time, or (5) difficult to minimize.

**Initialization points:** Table 9.1 shows an example of using Newton method with different initialization points $x_0$ for finding the roots of a polynomial $f(x) = x^3 - 2x^2 - 11x + 12$. With five $x_0$ differing in subtle values, Newton method converges to optimal points, showing the Newton method can be sensitive to initialization points.

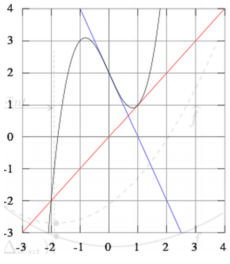
Table 9.1: Finding roots Ex#1

| $f(x) = x^3 - 2x^2 - 11x + 12$ | |
|---|---|
| $x_0$ | $x^\star$ |
| 2.352875270 | converges to 4 |
| 2.352841720 | converges to -3 |
| 2.352837350 | converges to 4 |
| 2.352836327 | converges to -3 |
| 2.352836323 | converges to 1 |

Table 9.2: Finding roots Ex#2

| $f(x) = x^3 - 2x + 2$ | |
|---|---|
| $i$th iteration | $x_i$ |
| 0 | 0 |
| 1 | 1 |
| 2 | 0 |
| 3 | 1 |
| 4 | 0 |

Figure 9.4: $f(x) = x^2 - 2x + 2$




**Cycles:** Another example is finding the roots for $f(x) = x^3 - 2x + 2$ with $x_0 = 0$, as illustrated in Figure 9.4. Table 9.2 shows the updated points in different iterations. The points $x$ cycles between 0 and 1 and never reaches the optimum, which also implies that Newton method is sensitive to the initialization point.

**Divergence:** The third example is finding roots for $f(x) = \sqrt[3]{x}$, where the first and second derivatives are $\nabla f(x) = \frac{1}{3}x^{-2/3}$ and $\nabla^2 f(x) = -\frac{2}{9}x^{-5/3}$. Then the Newton update becomes:

$$x_{k+1} = x_k - \frac{f(x_k)}{\nabla f(x_k)} = x_k - \frac{x_k^{\frac{1}{3}}}{\frac{1}{3}x_k^{\frac{-2}{3}}} = x_k - 3x_k = -2x_k. \tag{9.13}$$

This shows that except for $x_0 = 0$ (the root), the $x_k$ diverges no matter where it starts. One way to check the convergence is to check the expression of the limit of quadratic convergence. In this case:

$$\lim_{x\to 0} \frac{|\nabla^2 f(x)|}{|\nabla f(x)|} = \lim_{x\to 0} \frac{c}{|x|} = \infty, \tag{9.14}$$

which is unbounded.

**Converge only in linear time:** The example finds the roots of $f(x) = x^2$, where where the first and second derivatives are $\nabla f(x) = 2x$ and $\nabla^2 f(x) = 2$. The Newton update becomes:

$$x_{k+1} = x_k - \frac{f(x_k)}{\nabla f(x_k)} = x_k - \frac{x_k^2}{2x_k} = \frac{x_k}{2}. \tag{9.15}$$

showing that from any point, $x_k$ converges to zero but only with linear rate. Again we can check the expression of the limit of quadratic convergence:

$$\lim_{x\to 0} \frac{|\nabla^2 f(x)|}{|\nabla f(x)|} = \lim_{x\to 0} \frac{1}{|x|} = \infty, \tag{9.16}$$

which is unbounded.

**Difficult to minimize:** This examples aims to find the roots for $f(x) = 7x - \ln(x)$, where the first and second derivatives are $\nabla f(x) = 7 - \frac{1}{x}$ and $\nabla^2 f(x) = \frac{1}{x^2}$. The Newton update becomes:

$$x_{k+1} = x_k + (x_k - 7(x_k)^2) = 2x_k - 7x_k^2. \tag{9.17}$$

Figure 9.5 illustrates the function and Table 9.3 shows the update in each iteration with respect to different initial points. If we start at $x_0 = 1.0$, Newton method first fits a quadratic function at $x_0$, where the quadratic function badly estimates $f(x)$, and then leads to the next update to $x_1 = -5.0$. Newton method thus suffers from that $f(-5)$ is not even defined, ending up with divergence in this case. On the other hand, if we start at $x_0 = 0.1$ or $x_0 = 0.01$, Newton method converges to the true optimum. In this case the quadratic convergence only holds within a tiny interval from $x \approx 0$ to $x \approx 0.13$.

## 9.7 Generalization

So far, Newton method is defined in finite dimensional spaces. It can be generalized to infinite dimensional function spaces in the following ways:

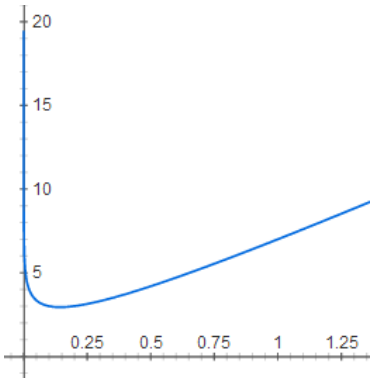
1) Define Newton method in Banach space and use Frechet derivatives
2) Define Newton method in curved manifolds (Eg#1: finding the minimum of a function defined on orthonormal matrices. Eg#2: Independent Component Analysis (ICA))
3) Define Newton method on complex number

This topic is not covered in this class.

Table 9.3: Update of each iteration

| $n$ | $x_n$ | $x_n$ | $x_n$ | $x_n$ |
|---|---|---|---|---|
| 0 | 1.0 | 0 | 0.1 | 0.01 |
| 1 | $-5.0$ | 0 | 0.13 | 0.0193 |
| 2 | $-185.0$ | 0 | 0.1417 | 0.03599257 |
| 3 | $-239,945.0$ | 0 | 0.14284777 | 0.062916884 |
| 4 | $-4.0302 \times 10^{11}$ | 0 | 0.142857142 | 0.098124028 |
| 5 | $-1.1370 \times 10^{24}$ | 0 | 0.142857143 | 0.128849782 |
| 6 | $-9.0486 \times 10^{48}$ | 0 | 0.142857143 | 0.1414837 |
| 7 | $-5.7314 \times 10^{98}$ | 0 | 0.142857143 | 0.142843938 |
| 8 | $-\infty$ | 0 | 0.142857143 | 0.142857142 |
| 9 | $-\infty$ | 0 | 0.142857143 | 0.142857143 |
| 10 | $-\infty$ | 0 | 0.142857143 | 0.142857143 |

Figure 9.5: Plot of $7x - \ln(x)$




## 9.8 Newton Fractals

Unlike gradient descent, Newton method becomes messy around the boundaries. Figures 9.6, 9.7, 9.8 and 9.9 illustrate the Newton fractals. Figures 9.6 and 9.7 runs Newton method on finding roots for $f(x) = x^4 - 1$, where the true roots are $-1, +1, -i, i$. Each point in the figures are colored by the convergence to different roots. That is, points with the same color converges to the same root. In Figure 9.7, the brighter color indicates that less iterations are needed. The closer to the boundaries, the more iterations are required. As can be observed in both figures, around the boundaries the plot becomes very complicated. Figures 9.8 and 9.9 show another two examples on the Newton fractals.

Figure 9.6: Newton fractal on $x^4 - 1$

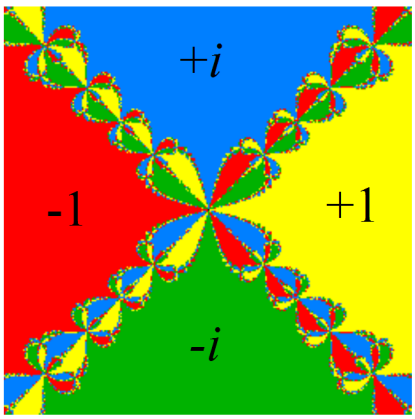
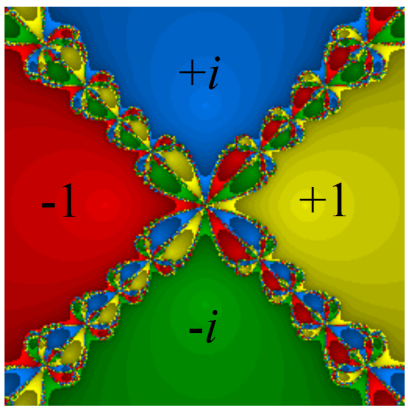



Figure 9.7: Newton fractal on $x^4 - 1$




## 9.9 Avoid Divergence (Damped Newton method)

One way to avoid divergence in Newton method is to use line search. In other words, instead of using a fixed step size 1 in each Newton update, we vary the step size by a line search approach (similar to what we do in gradient descent). This method is termed *Damped Newton method*. Specifically, for the $n$th Newton update, we search for the best step size $h_k$ $(0 < h_k \leq 1)$ using back tracking:

$$x_{k+1} = x_k - h_k[\nabla^2 f(x_k)]^{-1} \nabla f(x_k). \tag{9.18}$$

Figure 9.8: Newton fractal on $(x-1)^4(x+1)^4$

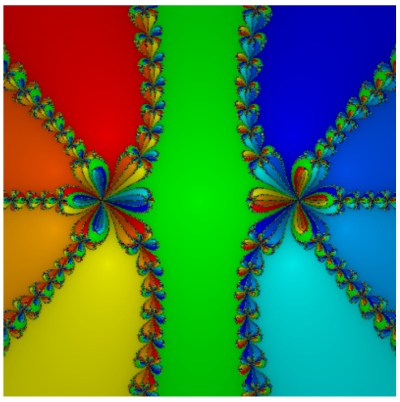
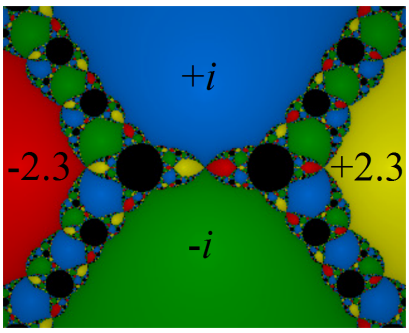


Figure 9.9: Newton fractal on a pure polynomial $f$ with the roots at $+i, -i, -2.3, +2.3$. Black indicates where Newton method fails to converge.




As long as $x_k$ is far from the local optimum, we do a line search for $h_k$. When the steps is close to the local optimum, $h_k$ will become 1 and thus a full Newton step is applied.

## References

[1] S. Boyd and L. Vandenberghe, "Convex Optimization," Chapter 9.5.

[2] Y. Nesterov, "Introductory Lectures on Convex Optimization."

[3] M. Bazaraa, H. Sherali and C. M. Shetty, "Nonlinear Programming: Theory and Algorithms."

[4] D. P. Bestsekas, "Nonlinear Programming."

[5] Wikipedia: http://en.wikipedia.org/wiki/Newton's_method.

[6] Fractals derived from Newton-Raphson iteration: http://www.chiark.greenend.org.uk/~sgtatham/newton/