Optimization Theory and Applications

Kun Zhu (zhukun@nuaa.edu.cn)

November 6, 2018

Introduction

- We consider the minimization of an objective function $f : \mathbb{R} \to \mathbb{R}$ (i.e., one-dimensional)
- The approach is to use an ***iterative search algorithm*** (or ***line-search*** method)
- Motivation: lower computation complexity comparing with solving FONC, SONC, and SOSC
- One-dimensional search methods are of interest
  - Special cases of search methods used in multivariable problems
  - Used as part of general multivariable algorithms

Introduction

- Iterative algorithm
  - Start with an ***initial candidate solution*** $x^{(0)}$
  - Generate a ***sequence of iterates*** $x^{(1)}, x^{(2)}, \ldots$
  - For each iteration, the next point $x^{(k+1)}$ depends on $x^{(k)}$, the objective function f, and/or derivative f', and/or second derivative f''

Introduction

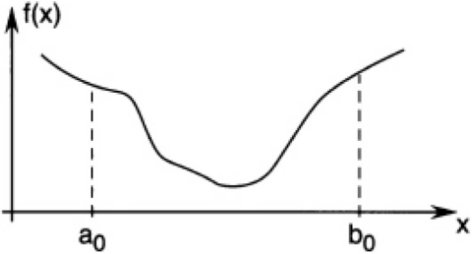
- Several iterative algorithms covered in this course
  - Golden section method (uses only f)
  - Fibonacci method (uses only f)
  - Bisection method (uses only f')
  - Secant method (uses only f')
  - Newton's method (uses f' and f'')

Golden Section Search

- Used to determine the minimizer of an objective function $f : \mathbb{R} \to \mathbb{R}$ over a closed interval $[a_0, b_0]$, without other constraints
- The only assumption of $f(x)$ is that it is ***unimodal***
  - $f(x)$ is unimodal if for some value x^*, it is monotonically decreasing (increasing) for $x \leq x^*$ and monotonically increasing (decreasing) for $x \geq x^*$
  - $f(x)$ has only one local minimizer, i.e., x^*

Golden Section Search

- An example of a unimodal function



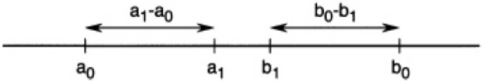
Golden Section Search

- Main idea
  - Evaluating the objective functions at different points within the interval $[a_0, b_0]$
  - Narrowing the interval by comparing the function values at these points
- Main goal
  - To narrow the range progressively until the minimizer is "boxed in" with sufficient accuracy
- Main issue
  - How to select appropriate points to reach the minimizer with as few evaluations as possible

Golden Section Search

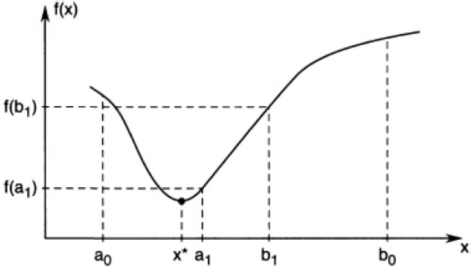
- Evaluate f at two intermediate points
- Choose the intermediate points to make the reduction in the range to be symmetric, i.e.,

$$a_1 - a_0 = b_0 - b_1 = \rho(b_0 - a_0), \quad \rho < \frac{1}{2}$$



Golden Section Search

- Evaluate f at intermediate points
- If $f(a_1) < f(b_1)$, the minimizer must lie in range $[a_0, b_1]$
- If $f(a_1) \geq f(b_1)$, the minimizer must lie in range $[a_1, b_0]$

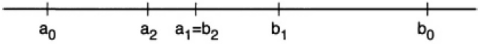


Golden Section Search

- With the above method, we narrow down the range
- Repeat the process and find two new intermediate points (e.g., a_2, b_2)
- Question: how to select intermediate points to minimize the number of objective function evaluations while reducing the width of the uncertainty

Golden Section Search

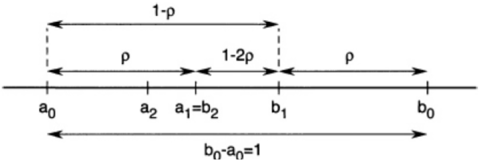
- Example: if $f(a_1) < f(b_1)$, then $x^* \in [a_0, b_1]$
- a_1 is an intermediate point in the new range $[a_0, b_1]$, and $f(a_1)$ is already evaluated
- Make a_1 coincide with b_2



- Thus only one new evaluation of f at a_2 is required

Golden Section Search

- Remind: $a_1 - a_0 = b_0 - b_1 = \rho(b_0 - a_0), \quad \rho < \frac{1}{2}$
- How to select ρ such that only one new evaluation of f can be achieved?



- Without loss of generality, assume $[a_0, b_0]$ is of unit length
- Since $b_1 - a_0 = 1 - \rho$ and $b_1 - b_2 = 1 - 2\rho$, we have
  $\rho(1-\rho) = 1 - 2\rho \ \rightarrow \ \rho^2 - 3\rho + 1 = 0 \ \rightarrow \ \rho = \frac{3-\sqrt{5}}{2} = 0.382$
  $1 - \rho = 0.618$: ***Golden ratio***

Golden Section Search

- We have $\frac{\rho}{1-\rho} = \frac{1-\rho}{1}$. Indicates that dividing a range in the ratio of ρ to $1-\rho$ has the ratio of the shorter segment to the longer equals the ratio of the longer to the sum of the two
- The rule was referred to by ancient Greek geometers as the ***Golden Section***
- With golden section, the objective function need only be evaluated at one new point
- The uncertainty range is reduced by the ratio $1-\rho = 0.618$ at each iteration
- N iterations reduces the range by the factor $(1-\rho)^N = 0.61803^N$

Golden Section Search

- Example: Suppose that we wish to use the golden section search method to find the value of x that minimizes

  $$f(x) = x^4 - 14x^3 + 60x^2 - 70x$$

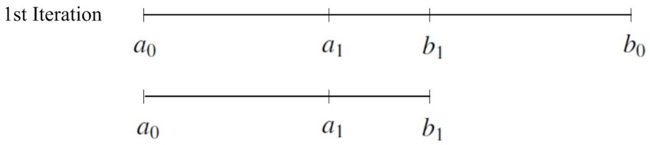
  in the interval $[0, 2]$. We wish to locate this value of x to within a range of 0.3

- Initial step, determine the number of iterations N

  $$2 \times (0.61803)^N = 0.3 \quad \rightarrow \quad N = 4$$

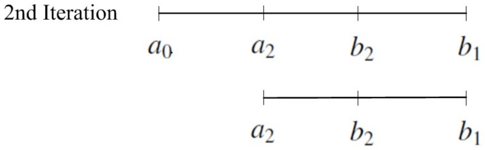
Golden Section Search

- First iteration
  - Select two intermediate points a_1 and b_1:
    $$a_1 = a_0 + \rho(b_0 - a_0) = 0.7639$$
    $$b_1 = a_0 + (1-\rho)(b_0 - a_0) = 1.236$$
  - Evaluate f at a_1 and b_1
    $$f(a_1) = -24.36$$
    $$f(b_1) = -18.96$$
  - Compare
    $f(a_1) < f(b_1)$, the range is narrowed to $[a_0, b_1] = [0, 1.236]$



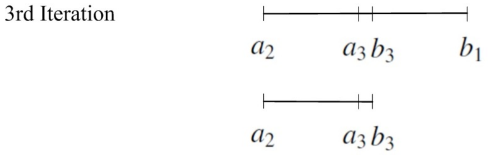
Golden Section Search

- Second iteration
  - We set $b_2 = a_1$
    $$a_2 = a_0 + \rho(b_1 - a_0) = 0.4721$$
  - Evaluate f at a_2 and b_2
    $$f(a_2) = -21.10$$
    $$f(b_2) = f(a_1) = -24.36$$
  - Compare
    $f(b_2) < f(a_2)$, the range becomes $[a_2, b_1] = [0.4721, 1.236]$



Golden Section Search

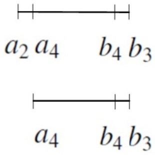
- Third iteration
  - We set $a_3 = b_2$
    $$b_3 = a_2 + (1-\rho)(b_1 - a_2) = 0.9443$$
  - Evaluate f at a_3 and b_3
    $$f(a_3) = f(b_2) = -24.36$$
    $$f(b_3) = -23.59$$
  - Compare
    $f(a_3) < f(b_3)$, the range becomes $[a_2, b_3] = [0.4721, 0.9443]$



Golden Section Search

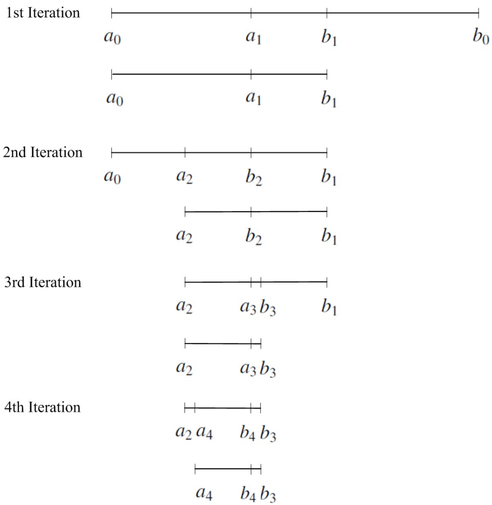
- Fourth iteration
  - We set $b_4 = a_3$
    $$a_4 = a_2 + \rho(b_3 - a_2) = 0.6525$$
  - Evaluate f at a_3 and b_3
    $$f(a_4) = -23.84$$
    $$f(b_4) = f(a_3) = -24.36$$
  - Compare
    $f(a_4) > f(b_4)$, the range becomes $[a_4, b_3] = [0.6525, 0.9443]$
  - Termination: $b_3 - a_4 = 0.292 < 0.3$

4th Iteration



Golden Section Search

- The entire process

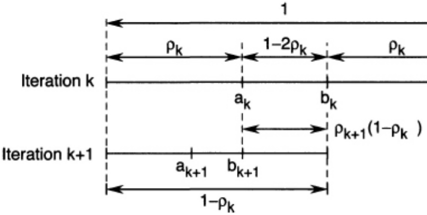


Fibonacci Method

- The golden section method uses the same value of ρ for all iterations
- A new search method: allows to vary the value ρ in different iterations. E.g., ρ_k for iteration k

Fibonacci Method

- The same goal as golden section method: select successive values of $\rho_k,\ 0 \le \rho_k \le 1/2$, such that only one new funcation evaluation is required at each stage



- From the figure, it is sufficient to choose ρ_k such that

$$\rho_{k+1}(1-\rho_k) = 1 - 2\rho_k$$

$$\rho_{k+1} = 1 - \frac{\rho_k}{1-\rho_k}$$

Fibonacci Method

- Many sequences $\rho_1, \rho_2, \ldots$ satisfy the above law of formation and the condition $0 \leq \rho_k \leq 1/2$
- After N iterations, the uncertainty range is reduced by a factor of
$$(1-\rho_1)(1-\rho_2)\ldots(1-\rho_N)$$
- Question: How to select sequences to minimize the reduction factor? The problem is formulated as a constrained optimization problem as follows

$$\begin{array}{rl} \text{minimize} & (1-\rho_1)(1-\rho_2)\cdots(1-\rho_N) \\ \text{subject to} & \rho_{k+1} = 1 - \dfrac{\rho_k}{1-\rho_k},\ k = 1, \ldots, N-1 \\ & 0 \leq \rho_k \leq \dfrac{1}{2},\ k = 1, \ldots, N. \end{array}$$

Fibonacci Method

- Before giving the solution to the above optimization problem, we introduce the ***Fibonacci sequence*** $F_1, F_2, F_3, \ldots$
  $F_{k+1} = F_k + F_{k-1}$, with $F_{-1} = 0$, and $F_0 = 1$
- Some values of elements in the Fibonacci sequence are

F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8
1	2	3	5	8	13	21	34

Fibonacci Method

- The solution to the above optimization problem can be represented by a Fibonacci sequence as

$$
\begin{aligned}
\rho_1 &= 1 - \frac{F_N}{F_{N+1}}, \\
\rho_2 &= 1 - \frac{F_{N-1}}{F_N}, \\
&\vdots \\
\rho_k &= 1 - \frac{F_{N-k+1}}{F_{N-k+2}}, \\
&\vdots \\
\rho_N &= 1 - \frac{F_1}{F_2},
\end{aligned}
$$

Fibonacci Method

- In the Fibonacci search method, the uncertainty range is reduced by the factor

$$(1-\rho_1)(1-\rho_2)\cdots(1-\rho_N) = \frac{F_N}{F_{N+1}}\frac{F_{N-1}}{F_N}\cdots\frac{F_1}{F_2} = \frac{F_1}{F_{N+1}} = \frac{1}{F_{N+1}}$$

- The reduction factor is less than that of golden section method
- Note that there is an anomaly in the final iteration of the Fibonacci search method, since

$$\rho_N = 1 - \frac{F_1}{F_2} = \frac{1}{2}$$

- Question: What is the problem with the final $\rho_N = 1/2$?

Fibonacci Method

- In each iteration, we need two intermediate points
- If $\rho = 1/2$, the two points coincide in the middle, we cannot further reduce the uncertainty range
- To avoid this problem, we perform new evaluation for the last iteration using $\rho = \rho_N - \varepsilon = \frac{1}{2} - \varepsilon$, where ε is a small number
- In this case, the reduction in the uncertainty range at the last iteration is

$$1 - (\rho_N - \varepsilon) = \frac{1}{2} + \varepsilon = \frac{1+2\varepsilon}{2}$$

- The reduction factor in the uncertainty range for the Fibonacci method is $\frac{1+2\varepsilon}{F_{n+1}}$

Fibonacci Method

- Example: Consider the function
  $$f(x) = x^4 - 14x^3 + 60x^2 - 70x$$
  Suppose that we wish to use the Fibonacci search method to find the value of x that minimizes f over the range $[0, 2]$, and locate this value of x to within the range 0.3
- Initial step: determine the number of iterations
  - We need to choose N such that
    $$\frac{1+2\varepsilon}{F_{N+1}} \leq \frac{\text{final range}}{\text{initial range}} = \frac{0.3}{2} = 0.15$$
  - Accordingly, we need $F_{N+1} \geq \frac{1+2\varepsilon}{0.15}$
  - For $\varepsilon \leq 0.1$, $N = 4$

Fibonacci Method

- Iterative 1
  - Determine ρ_1:

    $$\rho_1 = 1 - \frac{F_4}{F_5} = 1 - \frac{5}{8}$$

  - Determine the two intermediate points:

    $$a_1 = a_0 + \rho_1(b_0 - a_0) = \frac{3}{4}$$
    $$b_1 = a_0 + (1 - \rho_1)(b_0 - a_0) = \frac{5}{4}$$

  - Evaluate f at a_1 and b_1

    $$f(a_1) = -24.34 \qquad f(b_1) = -18.65$$

  - Compare

    $f(a_1) < f(b_1)$, the range is updated as $[a_0, b_1] = [0, \frac{5}{4}]$

Fibonacci Method

- Iterative 2
  - Determine ρ_2:

    $$\rho_2 = 1 - \frac{F_3}{F_4} = 1 - \frac{3}{5}$$

  - Determine the two intermediate points:

    $$a_2 = a_0 + \rho_2(b_1 - a_0) = \frac{1}{2}$$
    $$b_2 = a_1 = \frac{3}{4}$$

  - Evaluate f at a_2 and b_2

    $$f(a_2) = -21.69 \qquad f(b_2) = f(a_1) = -24.34$$

  - Compare

    $f(a_2) > f(b_2)$, the range is updated as $[a_2, b_1] = [\frac{1}{2}, \frac{5}{4}]$

Fibonacci Method

- Iterative 3
  - Determine ρ_3:

    $$\rho_3 = 1 - \frac{F_2}{F_3} = 1 - \frac{2}{3}$$

  - Determine the two intermediate points:

    $$a_3 = b_2 = \frac{3}{4}$$
    $$b_3 = a_2 + (1 - \rho_3)(b_1 - a_2) = 1$$

  - Evaluate f at a_3 and b_3

    $$f(a_3) = f(b_2) = -24.34 \qquad f(b_3) = -23$$

  - Compare

    $f(a_3) < f(b_3)$, the range is updated as $[a_2, b_3] = [\frac{1}{2}, 1]$

Fibonacci Method

- Iterative 4
  - Determine ρ_4:

    $$\rho_4 = 1 - \frac{F_1}{F_2} = 1 - \frac{1}{2}$$

  - Determine the two intermediate points (choose $\varepsilon = 0.05$):

    $$a_4 = a_2 + (\rho_4 - \varepsilon)(b_3 - a_2) = 0.725$$
    $$b_4 = a_3 = \frac{3}{4}$$

  - Evaluate f at a_4 and b_4

    $$f(a_4) = -24.27 \qquad f(b_4) = f(a_3) = -24.34$$

  - Compare

    $f(a_4) > f(b_4)$, the range is updated as $[a_4, b_3] = [0.725, 1]$

  - Termination

    $$b_3 - a_4 = 0.275 < 0.3$$

Bisection Method

- We still consider finding the minimizer of an objective function $f : \mathbb{R} \to \mathbb{R}$ over an interval $[a_0, b_0]$
- Besides the unimodal assumption of f, we further assume that f is continuously differentiable
- We can use values of the derivative of f (i.e., f') as a basis for reducing the uncertainty interval

Bisection Method

- The ***bisection method*** is a simple algorithm for successively reducing the uncertainty interval based on evaluations of the derivative

- Main idea of bisection method
  - Step 1: let $x^{(0)} = (a_0 + b_0)/2$ be the midpoint of the initial uncertainty interval
  - Step 2: evaluate $f'(x^{(0)})$
    - If $f'(x^{(0)}) > 0$, the minimizer x^* lies to the left of $x^{(0)}$, and the interval reduces to $[a_0, x^{(0)}]$
    - If $f'(x^{(0)}) < 0$, the minimizer x^* lies to the right of $x^{(0)}$, and the interval reduces to $[x^{(0)}, b_0]$
    - If $f'(x^{(0)}) = 0$, then $x^{(0)}$ is the minimizer

Bisection Method

- Two main features distinguish the bisection method from the golden section and Fibonacci methods
  - Instead of using values of f, the bisection method uses values of f'
  - At each iteration, the length of the uncertainty interval is reduced by a factor of $1/2$. Hence, the range is reduced by a factor of $(1/2)^N$ after N steps, which is smaller than in the golden method section and Fibonacci method

Bisection Method

- Example: We wish to find the minimizer of

$$f(x) = x^4 - 14x^3 + 60x^2 - 70x$$

  in the interval $[0, 2]$ to within a range of 0.3

- The golden section method requires $N = 4$ iterations

- For bisection method, we would choose N such that

$$2 \times (1/2)^N < 0.3 \quad \rightarrow \quad N = 3$$

Preliminaries for Newton's Method: Taylor's Series

- Taylor's formula is the basis for many numerical methods and models for optimization
- ***Taylor's Theorem***: Assume that a function $f : \mathbb{R} \to \mathbb{R}$ is m times continuously differentiable (i.e., $f \in \mathcal{C}^m$) on an interval $[a, b]$. Denote $h = [b - a]$, then

$$f(b) = f(a) + \frac{h}{1!}f^{(1)}(a) + \frac{h^2}{2!}f^{(2)}(a) + \ldots + \frac{h^{m-1}}{(m-1)!}f^{(m-1)}(a) + R_m,$$

(called ***Taylor's formula***) where $f^{(i)}$ is the ith derivative of f, and

$$R_m = \frac{h^m(1-\theta)^{m-1}}{(m-1)!}f^{(m)}(a + \theta h) = \frac{h^m}{m!}(a + \theta' h),$$

with $\theta, \theta' \in (0, 1)$

Newton's Method

- ***Newton's method*** (also known as the ***Newton - Raphson method***) have at least two applications
- Finding roots of equations
  - Finding successively better approximations to the roots of a real-valued function, by using the first-order derivative
- Optimization
  - Finding successively better approximations to the optimizers of a real-valued function, by using the first-order derivative and the second-order derivative

Newton's Method for Root Finding

- Objective:
  - Find x^* such that $g(x^*) = 0$
- Main idea:
  - Based on Taylor's formula, construct a new function to approximate the original function

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

  - $f(x)$ is an approximation to $g(x)$. Let $f(x) = 0$, we can have

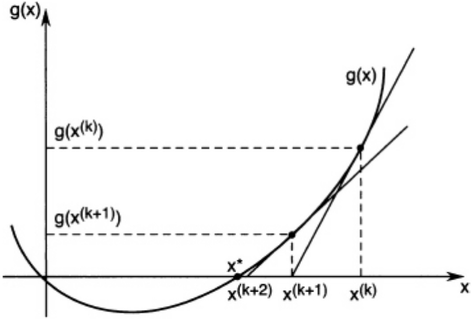
$$x = x_0 - \frac{g(x_0)}{g'(x_0)}$$

  - The process is repeated as

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

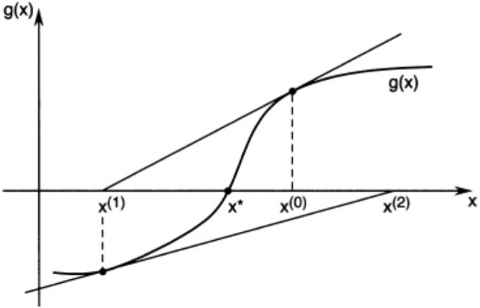
Newton's Method for Root Finding

- Geometrically, $(x_{k+1}, 0)$ is the intersection of the x-axis and the tangent of the graph of g at $(x_k, g(x_k))$



Newton's Method for Root Finding

- Note that Newton's method may fail if the first approximation to the root is such that the ratio $g(x_0)/g'(x_0)$ is not small enough
- An initial approximation to the root is very important



Newton's Method for Root Finding

- Example: We apply Newton's method to improve a first approximation, $x_0 = 12$, to the root of the equation

$$g(x) = x^3 - 12.2x^2 + 7.45x + 42 = 0$$

- We have $g'(x) = 3x^2 - 24.4x + 7.45$
- Performing two iterations yields
    - $x_1 = x_0 - \frac{g(x_0)}{g'(x_0)} = 12 - \frac{102.6}{46.65} = 11.33$
    - $x_2 = x_1 - \frac{g(x_1)}{g'(x_1)} = 11.33 - \frac{14.73}{116.11} = 11.21$

Newton's Method for Optimization

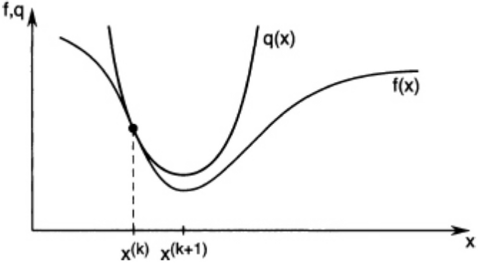
- We still consider the problem of minimizing a function f of a single real variable x
- We assume now that at each measurement point $x^{(k)}$ we can determine $f(x^{(k)})$, $f'(x^{(k)})$, and $f''(x^{(k)})$
- We try to construct a quadratic function to approximate $f(x)$ as

$$q(x) = f(x^{(k)}) + f'(x^{(k)})(x - x^{(k)}) + \frac{1}{2}f''(x^{(k)})(x - x^{(k)})^2$$

- $q(x)$ matches the first and second derivatives with that of $f(x)$
  - $q(x^{(k)}) = f(x^{(k)})$
  - $q'(x^{(k)}) = f'(x^{(k)})$
  - $q''(x^{(k)}) = f''(x^{(k)})$

Newton's Method for Optimization

- Instead of minimizing f, we minimize its approximation q
- An illustrative figure



Newton's Method for Optimization

- The first-order necessary condition for a minimizer q yields

$$q'(x) = f'(x^{(k)}) + f''(x^{(k)})(x - x^{(k)}) = 0$$

- Setting $x = x^{(k+1)}$, we obtain

$$x^{(k+1)} = x^{(k)} - \frac{f'(x^{(k)})}{f''(x^{(k)})}$$

Newton's Method for Optimization

- Example: Using Newton's method, find the minimizer of

  $$f(x) = \frac{1}{2}x^2 - \sin x$$

  Suppose that the initial value is $x^{(0)} = 0.5$, and the required accuracy is $\epsilon = 10^{-5}$, in the sense that we stop when $|x^{(k+1)} - x^{(k)}| < \epsilon$

- Initial step: Compute $f'(x) = x - \cos x,\ f''(x) = 1 + \sin x$

- First iteration:

  $$x^{(1)} = 0.5 - \frac{0.1 - \cos 0.5}{1 + \sin 0.5} = 0.7552$$

Newton's Method for Optimization

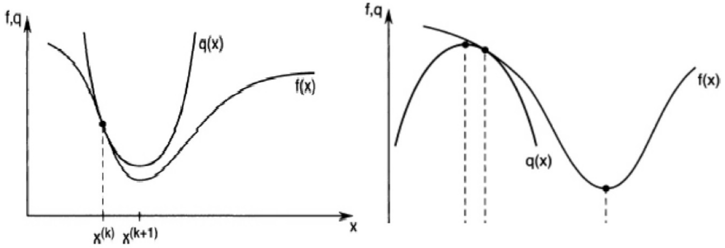
- Proceeding in a similar manner, we obtain

$$x^{(2)} = x^{(1)} - \frac{f'(x^{(1)})}{f''(x^{(1)})} = x^{(1)} - \frac{0.02710}{1.685} = 0.7391,$$
$$x^{(3)} = x^{(2)} - \frac{f'(x^{(2)})}{f''(x^{(2)})} = x^{(2)} - \frac{9.461 \times 10^{-5}}{1.673} = 0.7390,$$
$$x^{(4)} = x^{(3)} - \frac{f'(x^{(3)})}{f''(x^{(3)})} = x^{(3)} - \frac{1.17 \times 10^{-9}}{1.673} = 0.7390.$$

- $f'(x_4) = -8.6 \times 10^{-6} \approx 0$, and $f''(x_4) = 1.673 > 0$, this indicates that $x^* \approx x_4$ is a strict minimizer

Newton's Method for Optimization

- Note that Newton's method works well if $f''(x) > 0$ everywhere. However, if $f''(x) < 0$ for some x, Newton's method may fail to converge to the minimizer



Newton's Method for Optimization

- Question: any relationship between Newton's method for optimization and for root finding?
- Newton's method can also be viewed as a way to drive the first derivative of f to zero
- Accordingly, the optimization problem becomes the problem of finding the root of $f' = 0$

Secant Method

- Newton's method requires second derivatives of f

$$x^{(k+1)} = x^{(k)} - \frac{f'(x^{(k)})}{f''(x^{(k)})}$$

- If the second derivatives is not available, we may approximate it using first derivative information. Specifically,

$$f''(x^{(k)}) \approx \frac{f'(x^{(k)}) - f'(x^{(k-1)})}{x^{(k)} - x^{(k-1)}}$$

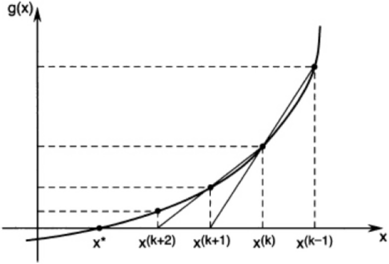
- Based on the above approximation, we can have the ***secant method*** as

$$x^{(k+1)} = x^{(k)} - \frac{x^{(k)} - x^{(k-1)}}{f'(x^{(k)}) - f'(x^{(k-1)})} f'(x^{(k)})$$

- Note that two initial points (i.e., $x^{(-1)}$, and $x^{(0)}$) are required

Secant Method

- Similar to Newton's method, the secant method can be interpreted as an algorithm for solving equations of the form $g(x) = f'(x) = 0$
- The secant algorithm for finding a root of $g(x) = 0$ takes the form
$$x^{(k+1)} = x^{(k)} - \frac{x^{(k)} - x^{(k-1)}}{g(x^{(k)}) - g(x^{(k-1)})} g(x^{(k)})$$
- Question: what is the difference with Newton's method?

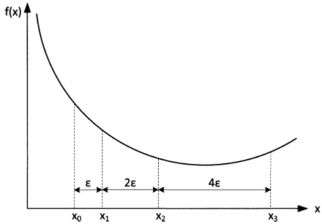


Bracketing

- An important assumption for previous methods
  - We have known an initial interval in which the minimizer is known to lie
- This interval is called a ***bracket***, and procedures for finding such a bracket are called ***bracketing*** methods
- Main idea to find a bracket $[a, b]$:
  - Find three points $a < c < b$ such that $f(c) < f(a)$ and $f(c) < f(b)$

Bracketing

- A simple bracketing procedure
  - Step 1: Pick three arbitrary points $x_0 < x_1 < x_2$
  - Step 2: Check:
    - If $f(x_1) < f(x_0)$ and $f(x_1) < f(x_2)$ then the desired bracket is $[x_0, x_2]$
    - If $f(x_0) > f(x_1) > f(x_2)$, then we pick a point $x_3 > x_2$ and check if $f(x_2) < f(x_3)$. If it holds, the desired bracket is $[x_1, x_3]$
    - Otherwise, continue this process until the function increases
- Typically, each new point chosen involves an expansion in distance between successive test points



Line Search in Multidimensional Optimization

- One-dimensional search methods play an important role in multidimensional optimization problem
  - iterative algorithm for multidimensional problems involve a line search at every iteration
- Let $f : \mathbb{R}^n \to \mathbb{R}$, iterative algorithms for finding a minimizer of f are of the form

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

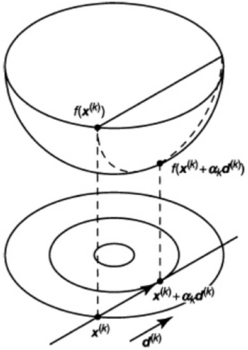
  where $\mathbf{x}^{(0)}$ is a given initial point and $\alpha_k \geq 0$ is chosen to minimize

  $$\phi_k(\alpha) = f(\mathbf{x}^{(k)} + \alpha \mathbf{d}^{(k)})$$

  The vector $\mathbf{d}$ is called the ***search direction*** and α_k is called the ***step size***

Line Search in Multidimensional Optimization

- The choice of α_k involves a one-dimensional minimization. This choice ensures that under certain conditions
$$f(\mathbf{x}^{(k+1)}) < f(\mathbf{x}^{(k)})$$
- Any of the one-dimensional methods discussed before can be used to minimize ϕ_k



Line Search in Multidimensional Optimization

- Some issues for practical uses of line search in multidimensional optimization problems
  - Determining the value of α_k that exactly minimizes ϕ_k may be computationally demanding, even worse, the minimization of ϕ_k may not exist
  - Practical experience suggests that it is better to allocate more computational time on iterating the multidimensional optimization algorithm rather than performing exact line searches

Line Search in Multidimensional Optimization

- These considerations lead to the development of **conditions for terminating** line-search algorithm
  - Lower-accuracy line searches
  - Still ensuring a sufficient decrease of f from one iteration to the next
- The basic idea is to ensure the step size is not **too small** or **too large**

Line Search in Multidimensional Optimization

- Some commonly used termination conditions: (let $\varepsilon \in (0,1)$, $\gamma > 1$, and $\eta \in (\varepsilon, 1)$ be given constants)
  - The Armijo condition ensures that α_k is not too large by requiring

    $$\phi_k(\alpha_k) \leq \phi_k(0) + \varepsilon \alpha_k \phi_k'(0)$$

    Further, it ensures that α_k is not too small by requiring

    $$\phi_k(\gamma\alpha_k) \geq \phi_k(0) + \varepsilon \gamma \alpha_k \phi_k'(0)$$

  - The Goldstein condition

    $$\phi_k(\alpha_k) \geq \phi_k(0) + \eta \alpha_k \phi_k'(0)$$

  - The Wolfe condition

    $$\phi_k'(\alpha_k) \geq \eta \phi_k'(0)$$

Line Search in Multidimensional Optimization

- Armijo backtracking algorithm:
  - A simple practical (inexact) line-search method
- Start with some candidate value for the step size α_k
- If α_k satisfies the prespecified termination condition (usually the first Armijo inequality), then we stop and α_k is used as the step size
- Otherwise, we iteratively decrease the value by some constant factor $\tau \in (0, 1)$ (typically $\tau = 0.5$) and re-check the condition
- The algorithm produces a value for the step size of the form $\alpha_k = \tau^m \alpha^{(0)}$ with m being the smallest value in $\{0, 1, 2, \ldots\}$ for which α_k satisfies the termination condition