Numerical Optimization 03: Bracket and Zoom

Qiang Zhu

University of Nevada Las Vegas

May 14, 2020

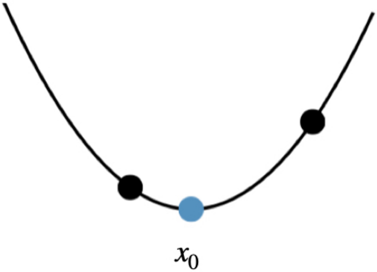
Overview

1. Bracketing Methods
2. Fibonacci Search
3. 0.618 Search
4. Interpolation
5. Summary

Bracketing

- identifying an interval in which a local minimum lies and then successively shrinking the interval.
- applied to a unimodal function

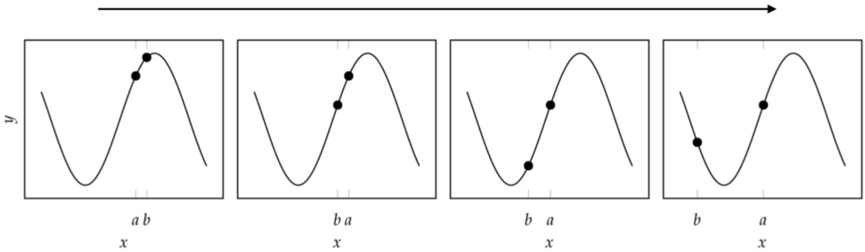
A unimodal function f is one where there is a unique x_0, such that f is monotonically decreasing for $x \leq x_0$ and monotonically increasing for $x \geq x_0$. It follows from this definition that the unique global minimum is at x_0, and there are no other local minima.



Initial Bracket

When optimizing a function, we often start by first bracketing an interval containing a local minimum.

- Starting at a given point, a trial move in the positive direction (1e-2)
- search in the downhill direction to find a new point that exceeds the lowest point.
- expand the step size by some factor of 2.



Fibonacci Search

Suppose we have a unimodal f bracketed by the interval $[a, b]$.

- Query f on the $1/3$ and $2/3$ points on the interval
- Query f on the center of the new interval
- Three queries ensures to shrink the interval by a factor of three
- $\cdots$

This actually follows

$$F_n = \begin{cases} 1 & \text{if } n \leq 2 \\ F_{n-1} + F_{n-2} & \text{otherwise} \end{cases}$$

Fibonacci Search Algorithm

Let's try to formulate the problem in a more rigorous way. Ideally, we want to shrink the interval $[a, b]$ within n iterations. For each update, we want to shrink by a factor of τ,

$$b_{k+1} - a_{k+1} = \frac{F_{n-k}}{F_{n-k+1}}(b_k - a_k)$$

$$F_0 = F_1 = 1,$$
$$F_{k+1} = F_k + F_{k-1} \qquad (k = 1, 2, \cdots),$$

Therefore,

$$\begin{aligned} b_n - a_n &= \frac{F_1}{F_2}(b_{n-1} - a_{n-1}) \\ &= \frac{F_1}{F_2}\frac{F_2}{F_3}\cdots\frac{F_{n-1}}{F_n}(b_1 - a_1) \\ &= \frac{1}{F_n}(b_1 - a_1) \end{aligned}$$

Solution of F_k

let $F_k = \tau^k$

$$\tau^2 = \tau + 1$$

$$\tau_{1,2} = \frac{1 \pm \sqrt{5}}{2}$$

$$F_k = A\tau_1^k + B\tau_2^k$$

Since $F_0 = F_1 = 1$,

$$F_k = \frac{1}{\sqrt{5}}(\tau_1^{k+1} - \tau_2^{k+1})$$

Fibonacci Algorithm

Suppose we have a unimodal f bracketed by the interval [a, b].

1. Query f on two points (λ_k, μ_k) on the interval $[a_k, b_k]$

$$\lambda_k = a_k + \left(1 - \frac{F_{n-k}}{F_{n-k+1}}\right)(b_k - a_k)$$

$$\mu_k = a_k + \frac{F_{n-k}}{F_{n-k+1}}(b_k - a_k)$$

2. if $f(\lambda_k) > f(\mu_k)$, go to step 3; otherwise, go to step 4
3. if $b_k - \lambda_k < \delta$, terminate. otherwise

$$a_{k+1} = \lambda_k, \quad b_{k+1} = b_k,$$

4. if $u_k - a_k \leq \delta$, terminate. Otherwise

$$a_{k+1} = a_k, b_{k+1} = \mu_k, \quad \mu_{k+1} = \lambda_k,$$

5. k $+=$ 1, go to step 2

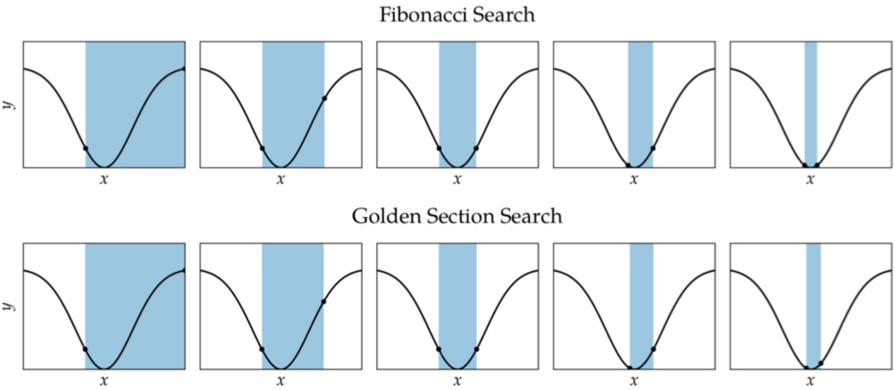
Golden Ratio Search

If we take the limit for large n, the ratio between successive values of the Fibonacci sequence approaches the golden ratio:

$$\lim_{n\to\infty} \frac{F_{k-1}}{F_k} = \frac{\sqrt{5}-1}{2} = 0.618$$

Therefore, we can always use 0.618 and 0.392 to check the two points within the updated interval.
Both Fibonacci and golden ratio searches have the property of linear scaling. Fibonacci is in principle the optimum search strategy for bracketing a unimodal function. However, golden ratio is more popular due to its simplicity.

Comparison



Homework: reproduce the above figure by yourself!

Interpolation with the help of gradient

Both Fibonacci and 0.618 searches do not need the gradient information. However, if the gradient is available, identifying the phase can be even faster. The idea is to approximate the target function in a analytic manner.

- Linear: bisection
- quadratic fit
- cubic interpolation

Bisection

The bisection method maintains a bracket $[a, b]$ in which at least one root is known to exist. If f is continuous on $[a, b]$, and there is some $y \in [f(a), f(b)]$, then the intermediate value theorem stipulates that there exists at least one $x \in [a, b]$, such that $f(x) = y$. It follows that a bracket [a, b] is guaranteed to contain a zero if $f(a)$ and $f(b)$ have opposite signs.

- Starting at $[a_1, b_1]$
- if $f`(a_k) \leq 0$ and $f`(b_k) \geq 0$, let $c_k = \frac{1}{2}(a_k + b_k)$
- if $f`(c_k) \geq 0$, let $a_{k+1} = a_k$, $b_{k+1} = c_k$, otherwise, $a_{k+1} = c_k$, $b_{k+1} = b_k$
- terminate if $(b_{k+1} - a_{k+1}) \leq \delta$ or it reaches the given number of iterations

Bisection method is commonly used to find roots of a function, or points where the function is zero.

Quadratic fit search

Given the bracketing point $a < b < c$, we want to find the coefficient p_1, p_2, p_3 which satisfies:

$$q(x) = p_1 + p_2 x + p_3 x^2$$

In matrix form, we have

$$y_a = p_1 + p_2 a + p_3 a^2$$
$$y_b = p_1 + p_2 b + p_3 b^2$$
$$y_c = p_1 + p_2 c + p_3 c^2$$

$$\begin{bmatrix} y_a \\ y_b \\ y_c \end{bmatrix} = \begin{bmatrix} 1 & a & a^2 \\ 1 & b & b^2 \\ 1 & c & c^2 \end{bmatrix} \cdot \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix}$$

$$q(x) = y_a \frac{(x-b)(x-c)}{(a-b)(a-c)} + y_b \frac{(x-a)(x-c)}{(b-a)(b-c)} + y_c \frac{(x-a)(x-b)}{(c-a)(c-b)}$$

We can solve the minimum by finding the derivative is zero:

$$x^* = \frac{1}{2} \frac{y_a(b^2-c^2) + y_b(c^2-a^2) + y_c(a^2-b^2)}{y_a(b-c) + y_b(c-a) + y_c(c-b)}$$

Summary

- Many optimization methods shrink a bracketing interval
- The step length is then selected by a zoom phase
- Fibonacci search and golden section search has the linear scaling of $\tau = 0.618$. They are derivative free.
- Root-finding methods like the bisection method can be used to find where the derivative of a function is zero.
- quadratic or cubic fit is more common choice in the zoom phase of line search