

# What is line search?

---

## Minimizing an objective

---

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

Starting from  $x_0$ ,  $k = 0$ , At the present iterate,  $x_k$ , a search direction  $d_k$ . Step size parameter  $\alpha$ .

A line search is a subroutine in the algorithm to choose a step size such that at the new iterate  $x_k + \alpha d_k$  the objective has a lower value, or in some sense, is a better point.

In the subroutine of line search, we are minimizing a univariate objective, i.e.,  $\phi(\alpha)$  for  $\alpha \in [l, u]$ .

## Finding the zero of a univariate function

---

Suppose that  $f(x)$  is twice continuously differentiable and  $x^*$  is its global minimizer, then  $f'(x^*) = 0$ . If  $f(x)$  is further convex, then a global minimizer of  $f(x)$  coincides with a zero of  $f'(x)$ .

Thus, we can find a solution to the optimization problem by solving

$$f'(x) = 0.$$

# Finding an inexact 'zero' of $f$

---

What is meant by 'finding' zero? Mathematically only analytical method can find it. For instance, line, quadratic, cubic polynomial, and some simple function such as  $\sin(x)$ . However, for polynomials whose order higher than 4, it is very hard to get theoretically solution (Gauss).

After the invention of computer, people more like to work with computers. Certain limitations makes it unrealistic to find an exact zero.

We are satisfied if an algorithm provides an interval  $[a, b]$  such that

$$f(a)f(b) < 0, \quad |a - b| \leq \delta$$

$\delta$  is some small tolerance.

An interval which contains  $x^*$  is called as the interval of uncertainty, the zero is said to be bracketed in an interval if  $f$  change sign in the interval.

# Bisection method

---

Bisection: systematically reducing the interval of uncertainty by function comparison.

Input  $a, b$  such that  $f(a)f(b) < 0$ .

Evaluate  $f$  at the midpoint and test its sign:

1 the point is zero, terminate;

2 if  $f((a+b)/2)f(b) < 0$ , then set  $a := \frac{a+b}{2}$ ;

3 if  $f(a)f((a+b)/2) < 0$ , then set  $b := \frac{a+b}{2}$ ;

Repeat the above procedure until  $b - a \leq \delta$ .

Total evaluations of  $f$  is about  $\log_2 \frac{b-a}{\delta}$ , using  $b^+ - a^+ = \frac{1}{2}(b - a)$ . Linear convergence!

Direct algorithm without taking account of the relative magnitudes of the values of  $f$  at various points. If  $f$  is sufficiently smooth, or well behaved, it is possible to use the derivatives of  $f$  to improve the performance of the line search.

# Newton's method

---

Approximate  $f$  by a new function, saying  $\hat{f}$ . A good candidate is the tangent line,  $\hat{f} = f(x_k) + f'(x_k)(x - x_k)$ . Hence one has

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

Example:  $f(x) = x^2 - a$  with  $a \geq 0$ , Newton iterate

$$x_{k+1} = \frac{1}{2} \left( x_k + \frac{a}{x_k} \right).$$

If  $a > 0$ , then globally convergent and locally quadratic convergent. Why?

If  $a = 0$ , global linear convergent, why?

Numerical difficulties occur when  $f'(x_k)$  is very small or zero.

Newton's method converges only locally and  $f$  is sufficiently smooth.

# Secant method

---

When  $f'$  is expensive, cumbersome to compute, we use another straight line that passes through the values of  $f$  at the most recent iterates; in essence, the derivative  $f'(x_k)$  in Newton method is replaced by the finite-difference approximation  $(f_k - f_{k-1})/(x_k - x_{k-1})$ , where  $f_k = f(x_k)$ .

Thus we get

$$x_{k+1} = x_k - \frac{x_k - x_{k-1}}{f_k - f_{k-1}} f_k.$$

If  $f'(x^*) \neq 0$  and  $x_0, x_1$  is sufficiently close to  $x^*$ , the secant method converges superlinearly with a rate  $r = 1.6180$ .

The method of false position: A modification of secant method. Using  $x_k, x_{k-1}$ , we obtain  $x_{k+1}$ , we can replace either  $x_k$  or  $x_{k-1}$  by  $x_{k+1}$ , depending on which function value agrees in sign with  $f_{k+1}$ . Improve the global convergence, but might be very slow.

# Safeguard zero-finding algorithms

---

The best methods available for zero-finding are the so-called safeguarded procedures. These algorithms are combinations of bisection and linear interpolation methods.

Assume that an interval of uncertainty  $[a, b]$  and two 'best' points are known. Using linear interpolation, one gets a new point  $u$ . We use certain requirements to keep the point  $u$  is 'good' in some sense.

# Univariate minimization

---

A univariate convex optimization problem can be solved via finding the zero of  $f'(x)$ . However, methods working on the original optimization problem is more efficient since we can further use information from the objective.

A univariate function is unimodal in  $[a, b]$  if there exists a unique  $x^* \in [a, b]$  such that, given any  $x_1, x_2 \in [a, b]$  for which  $x_1 < x_2$  :

if  $x_2 < x^*$  then  $f(x_1) > f(x_2)$ ;

if  $x_1 > x^*$  then  $f(x_1) < f(x_2)$ .

Reduce the interval of uncertainty for a unimodal function.

If  $f(x_1) < f(x_2)$  then reduce the interval to  $[a, x_2]$  or replacing  $b$  by  $x_2$ .

If  $f(x_1) > f(x_2)$ , then reduce the interval to  $[x_1, b]$  or replacing  $a$  by  $x_1$ .

If  $f(x_1) = f(x_2)$ , reduce the interval to  $[x_1, x_2]$ .

# Gold section search

---

We assume that the starting interval is  $[a, b] = [0, 1]$ . Choose  $x_1 < x_2 \in [0, 1]$ , then the reduced interval must be  $[0, x_2]$  or  $[x_1, 1]$ , this makes it needs only one point to add in next iteration. First we should choose  $x_2 = 1 - x_1$ , since otherwise the reducing rate of the interval is flexible.

Let  $x_1 = 1 - \tau, x_2 = \tau$ , the decreasing ratio is  $\tau$  at the first iteration. Assume that  $[0, \tau]$  is the new interval containing the minimizer. Since  $x_1 = 1 - \tau$  is a point in the new interval, the decreasing rate for next interval should be  $(1 - \tau)/\tau$ , this should equal to the first reducing rate  $\tau$ . Thus we get the following equation

$$\tau^2 + \tau - 1 = 0,$$

The unique solution of the above equation is

$$\tau = \frac{2}{1 + \sqrt{5}} \approx 0.618.$$

This is the gold section search.

Question: Why bisection is not good for optimization?

# Polynomial interpolation

---

Approximate  $f(x)$  by a simple function whose minimum can be easily evaluated, saying a quadratic function

$$\hat{f} = \frac{1}{2}ax^2 + bx + c,$$

where  $a > 0$ .

Linear approximation is not good in such sense.

Taylor-series expansion can be used, i.e.,

$$\hat{f} = f(x_k) + f'(x_k)(x - x_k) + \frac{1}{2}f''(x_k)(x - x_k)^2.$$

Thus, we obtain

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

We can replace  $f''(x_k)$  in the above formulae by

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

This is the secant method, one has

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

Again, safeguard method is the best choice.

## Computing the step length

$$\min \phi(\alpha) = f(x_k + \alpha d_k), \alpha > 0,$$

where  $f'(x_k)d_k < 0$ . The new point should decrease  $f$  'sufficiently'.

The Goldstein-Armijo principle:

$$0 < -\mu_1 \alpha_k f'(x_k)d_k \leq f_k - f_{k+1} \leq -\mu_2 \alpha_k f'(x_k)d_k,$$

where  $0 < \mu_1 \leq \mu_2 < 1$ . The upper and lower bounds in the above principle ensure  $\alpha_k$  'reasonable'.

Choose  $\alpha_0$  and  $0 < \rho < 1$ ,  $\alpha_{k+1} = \rho \alpha_k$ .

For enough large  $k$ ,  $\alpha_k$  satisfies the Goldstein-Armijo condition. However  $\alpha_k$  might be too small.

The choose  $\alpha_0$  is very important. For Newton method, to get quadratic convergence, one should try  $\alpha_0 = 1$  first. However, this might requires a lot of steps to get a step length satisfying Goldstein-Armijo condition if the Newton direction is not 'good'.

One should adjust  $\alpha_o$  according to the present iterate.

There are some variants of Goldstein-Armijo condition, such as Wolfe condition or strong Wolfe condition.