Project Proposal

Theme: Optimization methods of finding extrema value

Implementation and Comparison

1. Introduction

The problem of optimization is a problem of seeking the minimum (or maximum) of a function f(x) of several parameters x. According to the number of parameters and smoothness of the function of the problem can be complicated and sometimes very slow and one should adopt an appropriate method to obtain the best and fastest result.

2. Plan of Implementation

We will describe a few methods applicable in different situations: those applicable for one single parameter; those applicable when the function is smooth and as an example of what one can do in the situation when the function F is not smooth, we describe the Down-Hill Simplex method.

2.1 Golden Section Search

The first method we will implement is the Golden Section Method, which is applicable when we have only one parameter. In a sense is similar to the bisection method as a recursive function we learned in root finding.

The implementation works as follows:

(1) We start to form an interval [a, b], evaluate the function f (x) at two points inside x1 and x2 which are obtained by a constant 0 < α 2< 0.5.

$$X_1 = a + \alpha \cdot (b - a), X_2 = b - \alpha \cdot (b - a),$$

(2) Then we compare $f(x_1)$ to $f(x_2)$. If $f(x_1) < f(x_2)$ then we set $x = x_1$ and drop the point b and redefine $b' = x_2,\ a' = a$. Otherwise, we set $x = x_2$ and drop the point a and redefine $a' = x_1,\ b' = b$.

(3) We repeat this procedure several times, at each step the interval shrinks and generates new values x depending on which one contains a root.

(4) As we go through the iterations, we will approach a local minimum of the function f(x).

Even though the Golden Section Search is a one-dimensional method, we can still use it for more than one parameter. By optimizing the first with respect to the first variable, then with respect to the second and so on. This procedure has to be repeated many times and may eventually converge to a local minimum in the multi-dimensional space. Note that the convergence of this method is linear which is fairly slow.

2.2 Powell's method

The second method we look into is Powell's method, which is essentially a generalization of Newton's method we learned in root finding.

we can use Taylor expansion to approximate

$$F(\vec{x}_i) = F(\vec{x}_p) + \sum_i \partial_i F(\vec{x}_0) \cdot (x_i - x_{0,i})$$

(1) Looking for zeros of $\partial_i F(\vec{x})$

(2) Take $F(\vec{x})$ to be zero

(3) Linear expansion $F(\vec{x}_i) = F(\vec{x}_p) + \sum_i \partial_i F(\vec{x}_0) \cdot (x_i - x_{0,i})$

$$F_j(\vec{x}_i) = F_j(\vec{x}_p) + \sum_i \partial_i F_j(\vec{x}_0) \cdot (x_i - x_{0,i}), where \sum_i \partial_i F_j(\vec{x}_0) = A_{ji}$$

(4) $(x - x_0) = -A^{-1} \cdot F(\vec{x}_0)$

$$rescursion\ function: \vec{x}_n = \vec{x}_{n-1} - {A_{n-1}}^{-1} \cdot F(\vec{x}_{n-1})$$

$$where\ (A_{ji})_n = \partial_i F_j(\vec{x}_n)$$

This gives some approximation of $\vec{x}$. Then we can take this as a starting point for the next iteration, the same as in Newton's method:

$\vec{x}_n = \vec{x}_{n-1} - \frac{f(x_{n-1})}{f'(x_{n-1})}$ gives the generalization of Newton's method 1d case.

2.3 Down-Hill method

The third method is a multidimensional optimization method, which is applicable when the function is not differentiable or even cannot be efficiently approximated by the polynomial as in the previous methods.

For simplicity, we only talk about 2d case.

In the iterations of diagonalization of the Down Hill method, the implementation works as follows：

(1) We start with three points x_1, x_2, x_3, and make it to a triangle.

(2) Order the three points with $f(x_1) < f(x_2) < f(x_3)$, $x_0 = (x_1 + x_2)/2$

(3) We have four options to achieve the goal in the algorithm: Reflection, Expansion, Contraction and Shrinkage.

(4) Reflection: $x_r = x_0 + (x_0 - x_3)$

(5) If $f(x_1) \leq f(x_r) < f(x_2)$ then replace x_3 by x_r and go back to 2

(6) Expansion: If $f(x_r) < f(x_1)$, $x_e = x_0 + 2(x_0 - x_3)$

If $f(x_1) \leq f(x_e) < f(x_2)$ then replace x_3 by x_e and go back to 2

(7) Contracted: If $f(x_r) \geq f(x_2)$, $x_c = x_0 - {}^1\!/_2\,(x_0 - x_3)$

If $f(x_c) \geq f(x_3)$ then replace x_3 by x_c and go back to 2

(8) Shrink: if none of the above works, replace $x_i = x_1 + {}^1\!/_2\,(x_i - x_1)$

We continue in this way as we get closer and closer to the target minimum point. Eventually the triangles get so close.

3. Tests to do

In this project, we mainly use three examples to test the speed, accuracy and range of application of the three methods.

(1) Firstly, we use a simple one-dimension function to test the correctness of these three methods separately. Also, compare the accuracy and speed with which the three methods run this example.

(2) Second, we will apply a relatively complex function which is multiple dimensions and smooth in these three methods and check if all three methods can run. Meanwhile, comparing the speed and accuracy of the methods that can be run is necessary.

(3) Third, choosing a function which is neither differentiable nor efficiently approximated by the polynomial and applying it to methods. Then repeat the above steps to see which method works more accurately and faster.

Result Prediction: in an ideal situation, function (1) as a one-dimensional function will pass the test of the three methods smoothly. Since this is a very simple function, we predict that there will be no significant difference in the speed and accuracy of its implementation among the three methods. Function (2) is a two-dimensional smooth function. We predict it will go through Powell's method and Down-Hill Method. Since the golden section only works for 1D functions, we think that it would fail this test. Function (3) is a 2D function but lacks smoothness. The prediction result of this test is that only the Down-Hill method can pass.

4. Division of Work

Yiyuan is mainly responsible for the implementation of three method codes. Based on Yiyuan's implementation results above, Diemeng is responsible for subsequent tests and comparisons of speed, accuracy and application breadth of these three optimization methods.