FEDERAL STATE AUTONOMOUS EDUCATIONAL INSTITUTION

OF HIGHER EDUCATION

ITMO UNIVERSITY

Report

on the practical task No. 2

“Algorithms for unconstrained nonlinear optimization. Direct methods”

Performed by

Artyom Kusmin

J4134c

Accepted by

Dr Petr Chunaev

St. Petersburg

2021

**Goal:**

The use of direct methods (one-dimensional methods of exhaustive search, dichotomy, golden section search; multidimensional methods of exhaustive search, Gauss (coordinate descent), Nelder-Mead) in the tasks of unconstrained nonlinear optimization

**Formulation of the problem**:

1) Use the one-dimensional methods of exhaustive search, dichotomy and golden section search to find an approximate (with precision 𝜀 = 0.001) solution 𝑥: 𝑓(𝑥) → 𝑚𝑖𝑛 for the following functions and domains:

1.

2.

3.

Calculate the number of 𝑓-calculations and the number of iterations performed in each method and analyze the results. Explain differences (if any) in the results obtained.

2) Generate random numbers 𝛼 ∈ (0,1) and 𝛽 ∈ (0,1). Furthermore, generate the noisy data {𝑥k,𝑦k}, where 𝑘 = 0,…,100, according to the following rule:

𝑦k = 𝛼𝑥k + 𝛽 + 𝛿k, 𝑥k=k/100,

where 𝛿k~𝑁(0,1) are values of a random variable with standard normal distribution. Approximate the data by the following linear and rational functions:

by means of least squares through the numerical minimization (with precision 𝜀 =

0.001) of the following function:

To solve the minimization problem, use the methods of exhaustive search, Gauss and Nelder-Mead. If necessary, set the initial approximations and other parameters of the methods. Visualize the data and the approximants obtained in a plot separately for each type of approximant so that one can compare the results for the numerical methods used. Analyze the results obtained (in terms of number of iterations, precision, number of function evaluations, etc.).

**Brief theoretical part:**

Optimization problem is a wide-spread problem which is actual for other areas including Machine Learning. There are several types of optimization methods: direct, first-order, second-order…Direct methods use only values of function that needs optimization but not it’s derivatives. Thus, direct methods could be used for a wide range of functions (problems). However, it slowly converges to an optimal solution as it uses less information.

**Results:**

1. **Minimization of functions of one variable:**

Several methods were used to find minima of functions defined on specified intervals. These algorithms were compared between each other using calculated value, number of function calculations and number of iterations required for each method.

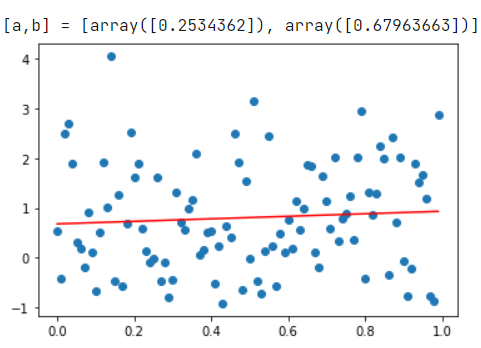
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Method | Function | x | Number of f-calculations | Number of iterations |
| Brute-Force | x3 | 0 | 1000 | 1000 |
| |x-0.2| | 0 | 1000 | 1000 |
| x\*sin(1/x) | -0.21729 | 1000 | 1000 |
| Dichotomy | x3 | 1.2057e-10 | 22 | 11 |
| |x-0.2| | 0.00010119 | 22 | 11 |
| x\*sin(1/x) | -0.21723 | 22 | 11 |
| Golden Section | x3 | 4.9257e-11 | 16 | 15 |
| |x-0.2| | 7.3313e-05 | 16 | 15 |
| x\*sin(1/x) | -0.21723 | 16 | 15 |

Brute-force method requires many calculations. In fact, number of iterations is the same as number of function calculations because on each iteration it calculates only one function using one argument value. However, minimum is quite precise.

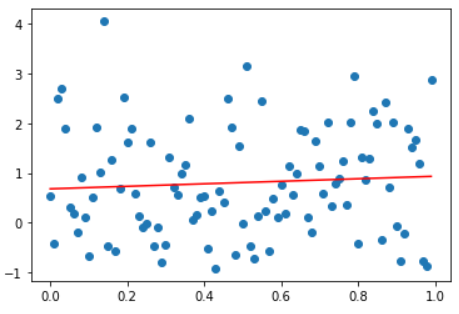
Dichotomy method requires less iterations as it converges faster. Number of calculations is 2 times more than number of iterations since on each iteration two values are used to calculate functions. Minima are also precise.

Golden Section method has a bit more iterations than dichotomy method but it is faster due to the number of f-calculations. Indeed, it calculates 2 values for function on first step and after that it calculates only one function value because the next step uses one previous point due to the choice of delta-value. Totally, it calculates N-1 function values on each iteration except the first and on the first iteration there are 2 calculations =>N+1 calculations.

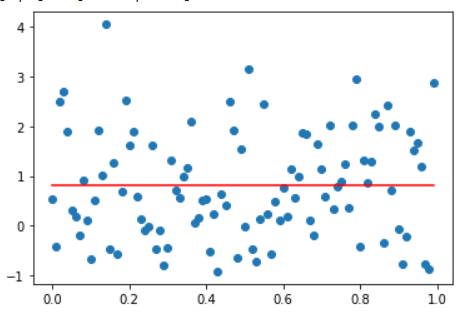
1. **Optimization of multivariate functions (least square):**
2. Visualization of LS-method for parameter searching (linear):



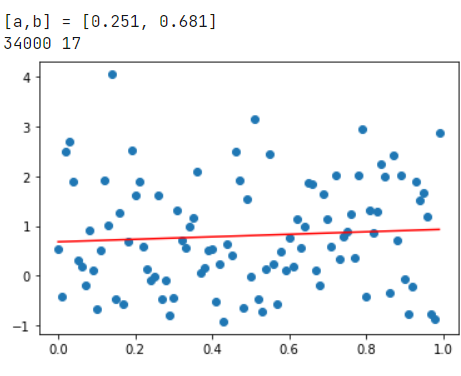
1. Visualization of Brute-Force method (linear approximation):



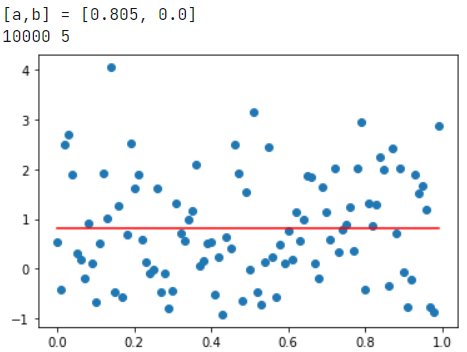
1. Visualization of Brute-Force method (rational approximation):



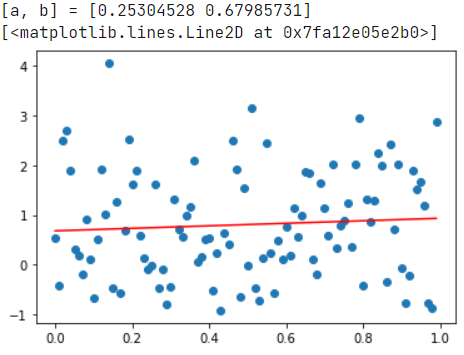
1. Visualization of Coordinate Descent Method (linear):



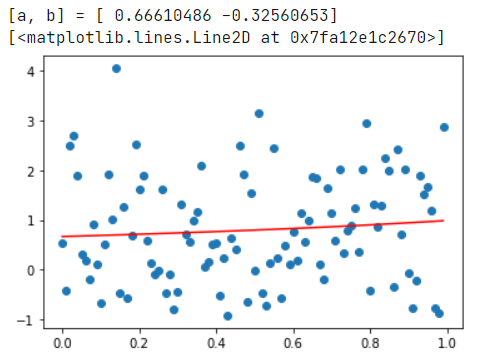
1. Visualization of Coordinate Descent Method(rational):



1. Visualization of Nelder-Mead Method(linear):



1. Visualization of Nelder-Mead Method(rational):



|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Method | Approximation | a | b | Number of f-calculations | Number of iterations |
| Brute-Force | Linear | 0.253 | 0.680 | 1000000 | 1000000 |
| Rational | 0.805 | 0.000 | 1000000 | 1000000 |
| Coordinate Descent | Linear | 0.251 | 0.681 | 34000 | 17 |
| Rational | 0.805 | 0.000 | 10000 | 5 |
| Nelder-Mead | Linear | 0.253 | 0.680 | 71 | 38 |
| Rational | 0.666 | -0.326 | 60 | 32 |

As it can be observed from the table, brute-force method is much time-consuming and requires lots of iterations and calculations but it outputs decent results which are quite close to reality. As for Gauss method, it is much better than brute-force in terms of number of iterations and f-evalutaions and parameters are well calculated as well. The best solution in terms of time is Nelder-Mead algorithm. Despite the bigger numbers of iterations than in Gauss Method, it performs better due to less calculations of the function values.

Really interesting values were obtained for Nelder-Mead method which are different from brute-force and coordinate descent. The reason behind it is that Nelder-Mead’s actualization is much more precise whilst brute-force and coordinate descent require more iterations to converge.

(!) For other set of values the minimal residual sums were obtained:

1. D = 120.34432458 – for Nelder-Mead
2. D = 121.1003674 – for brute-force (coordinate descent)

**Conclusions:**

Thus, different direct methods of optimization were considered. For one-dimensional case the best performance was highlighted for Golden Section method but Dichotomy method performed good too. Among multidimensional methods the best option is Nelder-Mead in terms of time and precision.

**Appendix:**

https://github.com/ShadowTechiner/Kusmin\_Homework.git