CS 268 Intro to Optimization Homework 1

Chen Li

October 6, 2015

1 Problem1 1D unconstrained optimization

1.1 1(a)

Implement the *Integrated Bracketing and Golden Section Algorithm*, with the corresponding stopping criteria. Apart from the stoping criteria, if the program iterate over enough iterations but still can't stop (In this case I set the limit to 10^6 iterations), it probably go to the wrong direction with no minimum, thus the program will stop and return the current value it get.

1.2 1(b)

The starting are draw from a Gaussian Distribution who's $\mu = 0$ and $\sigma = 1$.I choose $\epsilon_{abs} = 10^{-4}$, $\epsilon_F = 10^{-6}$ and $\epsilon_m = 10^{-15}$ and $EulerDistance$ as the distance measurement. Also, I set step size as 1.The Results are as follows:

Function	TestNum	$Time_{estimated}$	$Distance_{estimated}$	$Iteration_{estimated}$
$f(x) = x^2 - 2x + 1$	100	$2.341e^{-4} \pm 2.195e^{-5}$	$4.617e^{-1} \pm 6.736e^{-2}$	$0.22 \pm 4.86e^{-2}$
$f(x) = e^x - 5x$	100	$2.781e^{-4} \pm 2.042e^{-5}$	1.549 ± 0.201	0.18 ± 0.041
$f(x) = 5 + (x-2)^6$	100	$2.647e^{-4} \pm 1.376e^{-5}$	2.998 ± 0.983	0.35 ± 0.060

2 Problem2 Coordinate Descent

2.1 2(a)

I implement a 2-D *CoordinateDecentOptimizer* by repeatedly call *GoldenSection* Optimizer defined in Problem 1 on x and y. And I use the same stop policy for the 2-D Optimizer as mentioned in 1(a).

2.2 2(b)

I choose $\epsilon_{abs} = 10^{-4}$, $\epsilon_F = 10^{-6}$ and $\epsilon_m = 10^{-15}$ and $EulerDistance$ as the distance measurement. The Results are as follows:

Function	TestNum	$Time_{estimated}$	$Distance_{estimated}$	$Iteration_{estimated}$
$f(x,y) = x^2 + y^2$	100	$6.607e^{-3} \pm 6.584e^{-4}$	$3.211e^{-6} \pm 3.325e^{-7}$	5.15 ± 0.599
$f(x,y) = x^2 + 2y^2 + 2xy$	100	$8.565e^{-3} \pm 8.024e^{-4}$	$4.268e^{-6} \pm 4.323e{-}7$	5.41 ± 0.609

2.3 (1c)

The result analysis can be find in the ipython notebook called *ResultAnalysis.ipynb* in *results_analysis* folder.

3 Implementation

In *hw1.py*, the *GoldenSection*() method implement the 1-D unconstrained optimizer. *CoorDes*() method implement the Coordinate Descent by using *GoldenSection*() on each coordinate repeatedly. The *GSTest*() function the test for *GoldenSection*() method while the *CDTest*() performs the tests on *CoorDes*() method. The results can be find in the *test_results* folder.