Homework 5

Continuous optimization

(Max useful score: 100 - Available points: 370)

15-382: Collective Intelligence (Spring 2019)

Instructions

Homework Policy

Homework is due on Autolab by the posted deadline. As a general rule, you have a total of 6 late days. No credit will be given for homework submitted after the late days. After your 6 late days have been used you will receive 20% off for each additional day late.

If you find solutions in any source other than the material provided, you must mention the source.

Submission

Create a zipped archive including: a PDF file with the answers to the provided questions (they can be hand-written, but in this case you must have / use a "readable" handwriting), files that have been used for dealing with the questions that require programming, a README file that specifies how to use / run the programming files. The zipped archive should be submitted to Homework 5 on Autolab.

Contents

1 Optimization algorithms and models (160 points)

1.1 Function optimization using analytical methods (55 points)

Given the function

$$f(x_1, x_2, x_3) = x_1 + 2x_3 + x_2x_3 - x_1^2 - x_2^2 - x_3^2 :$$

1. (10 points) Find its critical points;

2. (20 points) Define whether their are points of minimum or maximum using algebraic methods (i.e., the Hessian matrix);

3. Determine whether the function is well conditioned or not by:

(a) (15 points) Considering the projections of the function on the planes $x_1 - x_2$, $x_1 - x_3$, $x_2 - x_3$, and studying the shape of the resulting isocontours;

(b) (20 points) Using the eigenvalues of the Hessian matrix;

1.2 Function optimization using gradient ascent (105 points)

The following optimization problem is given:

$$\max_{x_1,x_2} f(x_1, x_2) = 4x_1 + 6x_2 - 2x_1^2 - 2x_1x_2 - 2x_2^2.$$

1. (30 points) Find the optimum of the function using gradient ascent. The optimum occurs at $\left(\frac{1}{3}, \frac{4}{3}\right)$ and is reachable with good approximation in 6 iterations of the gradient algorithm starting from $(1, 1)$. Manually perform and report the details of the 6 iterations, and measure the final approximation error (for finding the step size at each iteration you can use a small program, if necessary).

2. (10 points) Show in a graph the progress of the algorithm reporting the performed steps and the isocontours (don't need to to it for all steps, at least the first three ones).

3. (25 points) Implement a programming code for the gradient ascent with *fixed step size*, and report in a table the behavior of the algorithm for three different step sizes at your choice. In the table report: the time to reach the optimum with the same approximation error as in the manual case, the number of steps, how many times the function didn't decrease monotonically during the iterations.

4. (30 points) Make use of the same implementation of the previous question using this time the knowledge of the *Hessian matrix to adaptively define the values of the step size.* Again, report in table: the time to reach the optimum with the same approximation error as in the manual case, the number of steps, how many times the function didn't decrease monotonically during the iterations.

5. (10 points) Explain why in this case, uniquely based on the properties of the function to optimize, it wouldn't make much sense to use a stochastic gradient ascent approach.

2 Weber point and stochastic gradients (90 points)

Consider the case of a Weber point problem. Inside a specified rectangular area n points (x_i, y_i), $i = 1, \ldots, n$ are given that represent location of facilities. The goal is to find the point in the area whose sum of the Euclidean distances from all given points is minimized. This point will be where the distribution center for the facilities will be built. Let the area be defined by the Cartesian product $[0, 100] \times [0, 100]$, and let $n = 10,000$.

1. Give the mathematical formulation of the optimization problem (which is a convex one).

2. (35 points) Write the code for a stochastic gradient descent algorithm with a fixed step size to find a numeric solution to the problem (with an approximation at your choice). Report and discuss test results.

3. (20 points) Write the code for a gradient descent algorithm with a fixed step size to find a numeric solution to the problem (with an approximation at your choice). Report and discuss test results.

4. (25 points) Write the code for a gradient descent algorithm with an adaptive step size (defined either by solving a line minization or using the Hessian) to find a numeric solution to the problem (with an approximation at your choice). Report and discuss test results.

5. (10 points) Discuss the pros and cons of the different approaches based on the observed results.

3 Use and analysis of PSO (120 points)

3.1 Solution of systems of equations (70 points)

The following system of equations is given:

$$\begin{cases} x_2 - x_1^2 = 0 \\ x_2 - x_1 = 2 \end{cases}$$

1. (20 points) Using an approach that could be applied to *any* other system of equations, transform the problem of finding the solutions of the system into an optimization problem;

2. (20 points) Explain how you would tackle the resulting optimization problem, either with PSO or with a gradient-based approach;

3. (30 points) the system has two solutions: discuss the characteristics that a PSO algorithm should have in order to be able to effectively find both these two solution (i.e., perform niching optimization). The core characteristics to be precisely (numerically / functionally) defined are:
   (a) number of particles;
   (b) topology of the social network;
   (c) acceleration coefficients;
   (d) possible ad hoc modifications of the basic algorithm that would favor to perform niching optimization.

3.2 Robotic PSO (50 points)

A contaminant from a point source is diffusing into an environment. Because of the disturbances in the environment, the diffusion process presents irregularities. However, a decreasing gradient of concentration for the contaminant can be measured between two nearby locations at increasing distances from the source. A fleet of n mobile robots is deployed in the environment to find the source of the contaminant. Robots can perform measurement of contaminant concentration in their neighborhood. Each robot is a "particle" in a robot PSO whose goal is to find the source in the shortest time.

1. (15 points) Define the social topology that seems the most appropriate for the task and explain your choice;
2. (20 points) Write down (formally, as it had to be implemented) the rule for a robot updating its velocity, accounting for the information available and the topology;
3. (15 points) The diffusion process is actually a dynamic one, that makes the concentration oscillating and possibly vanishing over time: which modifications would you make to the standard PSO in order to account for this variability?