Assignment #11

Due: 6:00pm Friday 16 December 2022
(Dean's Date)

Upload at: https://www.gradescope.com/courses/438130/assignments/2486355

Assignments in COS 302 should be done individually. See the course syllabus for the collaboration policy.

Problem 1 (20pts)

Solve the following problem using Lagrange multipliers:

$$\begin{aligned} \text{minimize:} \quad & x^2 + y^2 \\ \text{subject to:} \quad & x - y = 3 \end{aligned}$$

Problem 2 (25pts)

Your friend manages a coffee shop and is trying to figure out how to schedule their employees for the next week. The shop has two eight-hour shifts, seven days a week. There are ten employees and each of them has a productivity level ρ_i; you want to maximize the productivity level summed over all the shifts. No one should work double-shifts or work more than 40 hours per week. Every shift needs at least two people working. Nobody should work more than five out of seven days of the week.

You tell your friend that this problem can be solved using linear programming. Explain how to set that up by explaining what the decision variables would be, what the objective function is, and what the constraints are.

Problem 3 (25pts)

Determine whether the following functions are convex and explain your reasoning.

(A) $f(x) = e^x$

(B) $f(x) = x^3$

(C) $f(x) = |x|$

(D) $f(\boldsymbol{x}) = (\boldsymbol{x} - \boldsymbol{a})^T \boldsymbol{B}\boldsymbol{x}$ for $\boldsymbol{x} \in \mathbb{R}^d$, constant $\boldsymbol{a}$, and constant symmetric positive definite $\boldsymbol{B}$.

Problem 4 (28pts)

The Rosenbrock function

$$f(x, y) = 100(y - x^2)^2 + (1 - x)^2 \,,$$

is a classic test function for optimization routines. It has a global minimum of 0 at $(1, 1)$.

(A) Use `meshgrid` and `contourf` to make a contour plot of the function on the range $x \in [-2, 2]$, $y \in [-1, 3]$. You'll want to write code that looks kind of like this.

(B) Write a function that computes the gradient of the Rosenbrock function.

(C) Starting from $(-1, 2)$, perform gradient descent to minimize the function. You'll need to set the learning rate (probably to a very small number) and it may require quite a few steps (like thousands). Plot the path of your optimization on the contour plot from part (A).

(D) Now, minimize the problem using an off-the-shelf optimization tool. One powerful and widely-used method is the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm. Fortunately, `scipy.optimize.minimize` implements it (and many other methods). Check out the documentation, but you'll want to use it doing something along the lines of:

```
from scipy.optimize import minimize

steps = []
def cb(x):
  global steps
  steps.append(x)

result = minimize(func, x0, jac=grad_func, method='BFGS', callback=cb)
```

There are a few things going on here. `func` is the function you want to minimize. The argument `x0` is the initialization. The keyword argument `jac` takes the gradient function. The keyword argument `callback` is letting us keep track of the steps the algorithm takes, putting them into the variable `steps`. The structure `result` contains various things telling us how the optimization went and what minimum was found, if any.

As in (C), plot the path of the optimization (in `steps`). Explain what differences you see with the path from part (C).

Problem 5 (2pts)
Approximately how many hours did this assignment take you to complete?

My notebook URL: https://colab.research.google.com/XXXXXXXXXXXXXXXXXXXXXX

Changelog

- 6 December 2022 – Initial version.