

Assignment #11

Due: 6:00pm Friday 15 December 2023
(Dean's Date)

Upload at: [FIXME](#)

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:

$$\text{minimize: } x^2 + y^2$$

$$\text{subject to: } x - y = 3$$

Problem 2 (25pts)

An investment firm wants to allocate a budget of \$100,000 across four different investment options: stocks, bonds, real estate, and a startup venture. The expected annual return rates for these investments are 5% for stocks, 3% for bonds, 4% for real estate, and 10% for the startup venture. However, each investment comes with different risk factors and the firm wants to manage its risk exposure. The firm has established the following risk limits:

- No more than 40% of the total investment should be in high-risk options (stocks and startup venture).
- At least 30% of the total investment should be in low-risk options (bonds).
- The investment in real estate should not exceed 35% of the total investment.

Formulate a linear programming model to maximize the firm's expected annual return. Define the decision variables for the amounts to invest in each option, the objective function for maximizing return, and the constraints based on both the risk limits and the required properties for the percentages to make sense.

Problem 3 (25pts)

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

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

(B) $f(x) = \tanh(x)$

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

(D) $f(x) = (x - a)^T Bx$ for $x \in \mathbb{R}^d$, constant a , and constant symmetric positive definite 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/XXXXXXXXXXXXXXXXXXXXXXXXX>

Changelog

- 29 November 2023 – Initial version.