Homework 3

ORF522: Linear and Nonlinear Optimization

Instructor: Bartolomeo Stellato
AI: Irina Wang

Due on November 10

Problem 1

Either prove or find a counterexample for each of the following statement (you can assume all the functions are second order continuously differentiable):

1. if $f(x)$ is convex, $g(x)$ is convex, then $f(g(x))$ is convex.
2. If $f(x)$ is convex and nondecreasing, $g(x)$ is convex, then $f(g(x))$ is convex.
3. If $f(x)$ is concave and nonincreasing, $g(x)$ is convex, then $f(g(x))$ is convex.
4. If $f(x)$ is increasing and non-negative, then $xf(x)$ is convex on $x \geq 0$.

Problem 2

Consider the problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_j(x) \leq 0, \quad j = 1, \ldots, m, \end{array}$$

where f, g_j are continuously differentiable. Suppose that $x^\star$ is a local optimal solution, and $\nabla g_1(x^\star), \cdots, \nabla g_m(x^\star)$ are linearly independent. Prove that $x^\star$ is a global optimal solution to the following linear programming problem:

$$\begin{array}{ll} \text{minimize} & f(x^\star) + \nabla f(x^\star)^T(x - x^\star) \\ \text{subject to} & g_j(x^\star) + \nabla g_j(x^\star)^T(x - x^\star) \leq 0, \quad j = 1, \ldots, m. \end{array}$$

Problem 3

Consider the function

$$f(x, y, z) = 2x^2 + xy + y^2 + yz + z^2 - 6x - 7y - 8z - 9$$

1. By using the first-order necessary conditions, find the candidate minimum points of $f(x, y, z)$.

2. Verify using the second-order sufficient conditions whether those points are local minimum.

3. Find a global minimum point.

4. Use CVXPY to verify your results and include the code in the jupyter notebook. Here you may use the following form when inputting into CVXPY

$$f(x, y, z) = x^2 + \frac{1}{2}y^2 + \left(x + \frac{1}{2}y\right)^2 + \left(z + \frac{1}{2}y\right)^2 - 6x - 7y - 8z - 9.$$

Problem 4

We consider the approximation problem

$$\text{minimize} \quad \phi(Ax - b)$$

where $A \in \mathbf{R}^{m \times n}$ and $b \in \mathbf{R}^m$, the variable is $x \in \mathbf{R}^n$, and $\phi : \mathbf{R}^m \to \mathbf{R}$ is a convex penalty function that measures the quality of the approximation $Ax \approx b$.

We will consider the following choices of penalty function:

1. *Euclidean norm*

$$\phi(y) = \|y\|_2 = \left(\sum_{k=1}^{m} y_k^2\right)^{1/2}.$$

2. *ℓ_1-norm*

$$\phi(y) = \|y\|_1 = \sum_{k=1}^{m} |y_k|.$$

3. *Sum of the largest $m/2$ absolute values*

$$\phi(y) = \sum_{k=1}^{\lfloor m/2 \rfloor} |y|_{[k]}$$

where $|y|_{[1]}, |y|_{[2]}, |y|_{[3]}, \cdots$ denotes the absolute values of the components of y sorted in decreasing order.

4. *A piecewise-linear penalty.*

$$\phi(y) = \sum_{k=1}^{m} h(y_k), \quad h(u) = \begin{cases} 0 & |u| \le 0.2 \\ |u| - 0.2 & 0.2 \le |u| \le 0.3 \\ 2|u| - 0.5 & |u| \ge 0.3. \end{cases}$$

5. *Huber penalty*

$$\phi(y) = \sum_{k=1}^{m} h(y_k), \quad h(u) = \begin{cases} u^2 & |u| \le M \\ M(2|u| - M) & |u| \ge M \end{cases}$$

with $M = 0.2$.

6. *Log-barrier penalty.*

$$\phi(y) = \sum_{k=1}^{m} h(y_k), \quad h(u) = -\log(1 - u^2), \quad \mathbf{dom}\, h = \{u \mid |u| < 1\}.$$

Here is the problem. Generate data A and b as follows:

```
import numpy as np
m = 200
n = 100
A = np.random.randn(m,n)
b = np.random.randn(m)
b = b / (1.01 * np.max(np.abs(b)))
```

To compare the results, plot a histogram of the vector of residuals $y = Ax - b$, for each of the solutions x, using `matplotlib.pyplot.hist()`.

Hint: Use the CVXPY atoms: `norm`, `sum_largest`, `max`, `abs`, `sum`, `huber`.

Note: The normalization of b ensures that the domain of $\phi(Ax - b)$ is nonempty if we use the log-barrier penalty.