CSC 860 / MATH 795: Stochastic Optimization & Simulation Methodology
Fall 2022 Assignment 1

Mariia Aleksandrovych, Yaffa Atkins, Atoosa Lotfi

1 Exercise 1.3

Use a Taylor series expansion to show that for any descent direction $d(\theta)$ at a point θ of a twice continuously differentiable function, there exists $\epsilon_0 > 0$ such that

$$J(\theta + \epsilon d(\theta)) \leq J(\theta) \text{ for all } 0 \leq \epsilon \leq \epsilon_0.$$

(Answer)

Use Taylor Series expansion to show that for any descent direction d(θ), there exists some positive value where J(θ + ed(θ)) shrinks J(θ) for at least the first two derivatives.

1st derivative: descent direction d(θ) = -∇J(θ), since the gradient is negative, the first derivative is negative. Use hessian when multiple dimensions

2nd: Even if positive, its minimized because there's some tiny epsilon between 0 and 1 where epsilon*hessian > gradient

Taylor series: J(θ+e ∇J(θ)=J(θ)+ e.∇^2J(θ)+…

1.3

$$J(\theta+\varepsilon_0 d) = J(\theta) + \varepsilon_0 d^T \nabla J(\theta) + \frac{\varepsilon_0^2}{2} d^T \nabla^2 J(\theta) d$$

$\nabla^2 J(\theta)$ Hessian of $J(\theta)$, symmetric matrix $d \times d$ whose (i,j) entry is $\frac{\partial^2 f}{\partial x_i \partial x_j}$

if $J(\theta): \mathbb{R}^d \to \mathbb{R}$ such that J is twice-differentiable and has continuous derivatives in ~~an open~~ ball B ? around point $\theta \in \mathbb{R}^d$

Then for small enough $\Delta\theta = \varepsilon_0 d(\theta) \in \mathbb{R}^d$ such that

$\theta + \Delta\theta$ is also in $\mathbb{R}^d$ (ball B)

we have following:

$$\underset{(\theta+\Delta\theta)}{J(\theta+\varepsilon_0 d)} = J(\theta) + \varepsilon_0 d^T \nabla J(\theta) + \frac{\varepsilon_0^2}{2} d^T \nabla^2 J(\theta) d$$

Then for any step $\varepsilon \geq 0$ $0 \leq \varepsilon \leq \varepsilon_0$

when we at point $\theta' = \theta + \varepsilon d \in \mathbb{R}^d$, gradient decent ~~move~~ has to move in direction $\varepsilon d \in \mathbb{R}^d$ such that

$$J(\theta' + \varepsilon_0 d) < J(\theta')$$

This cold be done by choosing

$$\varepsilon_0 d^T \equiv \Delta\theta = -\eta \nabla J(\theta_\varepsilon^*) = -\eta \nabla J(\theta + \varepsilon d)$$

assuming that J is differ at $\theta_\varepsilon = \theta + \varepsilon d$

for any small enough $\varepsilon_0 d$

we have T. ε_0

$$J(\theta_\varepsilon + \varepsilon_0 d) = J(\theta_\varepsilon) + \varepsilon_0 d^T \nabla J(\theta_\varepsilon)$$

To get a maximum leverage out of moving along $\Delta\theta \equiv \varepsilon_0 d$, the $\Delta\theta$ shold be along along $-\nabla J(\theta_\varepsilon)$

Then if $\theta_{\varepsilon+1} = \theta_\varepsilon + \Delta\theta =$

$$= \theta_\varepsilon + (-\eta \nabla J(\theta_\varepsilon)) =$$

$$= \theta_\varepsilon - \eta \nabla J(\theta_\varepsilon)$$

Then

$$J(\theta_{\varepsilon+1}) = J(\theta_\varepsilon) - \eta\left((\nabla J(\theta_\varepsilon))^T \nabla J(\theta_\varepsilon)\right) < J(\theta_\varepsilon)$$

since $(\nabla J(\theta_\varepsilon))^T \nabla J(\theta_\varepsilon) > 0$. $\eta > 0$

2 Exercise 1.9

Let $J \in C^2$ be such that the gradient is a bounded and Lipschitz continuous function and consider the biased algorithm:

$$\theta_{n+1} = \theta_n - \epsilon_n(\nabla_\theta J(\theta_n) + \beta_n(\theta_n)),$$

where

$$\sum_{n=1}^{\infty} \epsilon_n = +\infty, \quad \sum_{n=1}^{\infty} \epsilon_n \|\beta_n(\theta_n)\| < \infty, \quad \sum_{n=1}^{\infty} \epsilon_n^2 < \infty.$$

If $\{\|\nabla J(\theta_n)\| : n \geq 0\}$ is bounded, then any limit point not achieved in finite time is a stationary point of J.

(Answer)

We use Taylor series to write cost function:

$$J(\theta_{n+1}) = J(\theta_n) + \nabla J(\theta_n)^T (\theta_{n+1} - \theta_n) + \frac{1}{2}(\theta_{n+1} - \theta_n)^T \nabla^2 J(\theta_n)(\theta_{n+1} - \theta_n) \quad (1)$$

Then substitute $(\theta_{n+1} - \theta_n)$

$$\theta_{n+1} = \theta_n - \varepsilon_n(\nabla J(\theta_n) + \beta_n(\theta_n)) \Rightarrow (\theta_{n+1} - \theta_n) = -\varepsilon_n(\nabla J(\theta_n) + \beta_n(\theta_n)) \quad (2)$$

Then the equation (1) becomes:

$$J\left(\theta_{n+1}\right)=J(\theta_n)-\varepsilon_n\left\|\nabla J(\theta_n)\right\|^2-\varepsilon_n\nabla J(\theta_n)^T\beta_n\left(\theta_n\right)+$$
$$+\frac{1}{2}\varepsilon_n^2\left(\nabla J\left(\theta_n\right)+\beta_n\left(\theta_n\right)\right)^T\nabla^2 J\left(\theta_n\right)\left(\nabla J\left(\theta_n\right)+\beta_n\left(\theta_n\right)\right)$$

Now we recall

$$\frac{1}{2}\varepsilon_n^2\left(\nabla J\left(\theta_n\right)+\beta_n\left(\theta_n\right)\right)^T\nabla^2 J\left(\theta_n\right)\left(\nabla J\left(\theta_n\right)+\beta_n\left(\theta_n\right)\right)=h_n$$

Then

$$\frac{1}{2}\varepsilon_n^2\left(\nabla J\left(\theta_n\right)+\beta_n\left(\theta_n\right)\right)^T\nabla^2 J\left(\theta_n\right)\left(\nabla J\left(\theta_n\right)+\beta_n\left(\theta_n\right)\right)\leq\frac{1}{2}\varepsilon_n^2 L\left\|\left(\nabla J\left(\theta_n\right)+\beta_n\left(\theta_n\right)\right)\right\|^2$$

$$h_n\leq\frac{1}{2}\varepsilon_n^2 L\left\|\left(\nabla J\left(\theta_n\right)+\beta_n\left(\theta_n\right)\right)\right\|^2$$

Let’s take a limit

$$\sum_n h_n\leq\sum_n\frac{1}{2}\varepsilon_n^2 L\left\|\left(\nabla J\left(\theta_n\right)+\beta_n\left(\theta_n\right)\right)\right\|^2\leq\frac{1}{2}L\left(\sum_n\varepsilon_n^2\left\|\nabla J\left(\theta_n\right)\right\|^2+\sum_n\varepsilon_n^2\left\|\beta_n\right\|^2\right)$$

Where $\sum_n\varepsilon_n^2\left\|\nabla J\left(\theta_n\right)\right\|^2<\infty$ because gradient is bounded, and $\sum_n\varepsilon_n^2<\infty$.

Also, $\sum_n\varepsilon_n^2\left\|\beta_n\right\|<\infty$

This means that h_n is summable, so we apply Lemma 1.2.

Which give us the next conditions:

i. $J\left(\theta_n\right)\to-\infty$

Or

ii. $J\left(\theta_n\right)$ converges to a finite value and $\sum_n\varepsilon_n\left(\left\|\nabla J(\theta_n)\right\|^2+\nabla J(\theta_n)^T\beta_n\left(\theta_n\right)\right)<\infty$ converges

If $\bar{\theta}$ the limit of the θ_n is not achieved in finite time, then we have the $\sum_n\varepsilon_n\left(\left\|\nabla J(\theta_n)\right\|^2+\nabla J(\theta_n)^T\beta_n\left(\theta_n\right)\right)<\infty$ and continuity of $\nabla J(\theta_n)$ so that
$\left\|\nabla J\left(\bar{\theta}\right)\right\|=\lim_{i\to\infty}\left\|\nabla J\left(\theta_{m_i}\right)\right\|=0$

This shows that $\bar{\theta}$ is stationary point.

Let’s prove it!

$\bar{\theta}$ is the limit of θ_n. Then using

$$\frac{1}{2}\varepsilon_n^2\left(\nabla J\left(\theta_n\right)+\beta_n\left(\theta_n\right)\right)^T \nabla^2 J\left(\theta_n\right)\left(\nabla J\left(\theta_n\right)+\beta_n\left(\theta_n\right)\right)\leq \frac{1}{2}\varepsilon_n^2 L\left\|\left(\nabla J\left(\theta_n\right)+\beta_n\left(\theta_n\right)\right)\right\|^2$$

The next equation

$$J\left(\theta_{i+1}\right)=J(\theta_i)-\varepsilon_i\left\|\nabla J(\theta_i)\right\|^2-\varepsilon_i\nabla J(\theta_i)^T\beta_i\left(\theta_i\right)+$$
$$+\frac{1}{2}\varepsilon_i^2\left(\nabla J\left(\theta_i\right)+\beta_i\left(\theta_i\right)\right)^T\nabla^2 J\left(\theta_i\right)\left(\nabla J\left(\theta_i\right)+\beta_i\left(\theta_i\right)\right)$$

Becomes

$$J\left(\theta_{i+1}\right)\leq J(\theta_i)-\varepsilon_i\left(\left(\left\|\nabla J(\theta_i)\right\|^2+_i\nabla J(\theta_i)^T\beta_i\left(\theta_i\right)\right)+\frac{1}{2}\varepsilon_i L\left\|\left(\nabla J\left(\theta_i\right)+\beta_i\left(\theta_i\right)\right)\right\|^2\right)$$

As $i\to\infty$ and $\varepsilon_i<1$

$\left(\left(\left\|\nabla J(\theta_i)\right\|^2+_i\nabla J(\theta_i)^T\beta_i\left(\theta_i\right)\right)+\frac{1}{2}\varepsilon_i L\left\|\left(\nabla J\left(\theta_i\right)+\beta_i\left(\theta_i\right)\right)\right\|^2\right)$ is positive so, i_0 exist such that $\left\{J\left(\theta_i\right):i\geq i_0\right\}$ is decreasing towards $J\left(\bar{\theta}\right)$, where $\bar{\theta}$ is a stationary point.

3 Exercise 1.11

Consider a single machine that can operate one piece at a time, and let the service times of the machine constitute a sequence of iid exponentially distributed random variables. Parts arrive to the machine according to a Poisson process with unit rate. In other words, the time between arrivals of parts is exponentially distributed with mean 1 and that interarrival times are mutually independent. In order for the system to be stable, assume that the expected service time is strictly less than one. Let $C(\theta)=1/\theta^2$ be the cost of operating the system at service mean θ. Let $P(\theta)$ denote the stationary probability that the queue length is larger than or equal to a threshold b.

(a) Find the solution to the constrained problem:

$$\min C(\theta),\ \text{s.t. } P(\theta)\leq\alpha.$$

Interpret the constraint qualifications, the second order condition, and the Lagrange multiplier. Hint: Use the fact that the probability that the stationary queue length equals n is given by $(1-\theta)\theta^n$, for $n\in\mathbb{N}$.

(Answer)

$\boxed{1.11}$ Say theres an assembly line. parts come at an average rate of 1 and pass through at a rate faster than 1. but since both rates are random, there could be pile ups.

$C(\theta)$ = cost of operating the system at service mean θ

Find $\min C(\theta)$ such that $P(\theta) \le \alpha$

$$P(\theta) = P(\text{queue} = \text{any } n \ge b)$$

$$P(\theta) = \sum_{n=b}^{\infty} p(\theta) = \sum_{n=b}^{\infty} (1-\theta)(\theta^n)$$

sum this formula as n goes from b to infinity

$$P(\theta) = \theta^b - \theta^\infty = \theta^b$$

$\theta^\infty = 0$ because θ = service time and theres a constraint that service time < 1 so $\theta < 1$

exponents on θ for $0 < \theta < 1$ get exponentially smaller

so θ^∞ is basically 0

$$\text{so } P(\theta) = \theta^b \le \alpha$$
$$= \theta \le \alpha^{1/b}$$
$$= \frac{\theta}{\theta} \le \frac{\alpha^{1/b}}{\theta}$$
$$= \alpha^{-1/b} \le \frac{1}{\theta}$$
$$= \alpha^{-2/b} \le \frac{1}{\theta^2}$$

we know $C(\theta) = \frac{1}{\theta^2}$

so $C(\theta) \ge \alpha^{-2/b}$

$\min C(\theta) = \alpha^{-2/b}$

Second order condition:

Cost function:

$$C(\theta) = 1/\theta^2$$

and two constrains:

1. $(\theta^b - \alpha) \le 0$
2. $\theta - 1 < 0$

Then unconstrained problem:

$$f(\theta) = 1/\theta^2 + \alpha_1\left(\theta^b - a\right)^2 + \alpha_2\left(\theta - 1\right)^2$$

Subject to minimization using second conditions:

F' = 0 -> stationary point,

F'' > 0 or F'' <0 -→ local minimum or maximum

Lagrange multipliers:

We have two constrains:

3. $(\theta^b - \alpha) \leq 0$
4. $\theta - 1 < 0$

Then the Lagrangian function:

$$L(\theta; \lambda_1, \lambda_2) = \frac{1}{\theta^2} + \lambda_1 \left(\theta^b - \alpha\right) + \lambda_2 (\theta - 1)$$

From definition 1.11 we have

$$\nabla_\theta \mathcal{L}(\theta^*, \lambda^*, \eta^*) = 0$$
$$\nabla_\lambda \mathcal{L}(\theta^*, \lambda^*, \eta^*) = g(\theta^*)^\top \leq 0, \lambda^* \geq 0, \textit{ and } \forall i : \lambda_i^* g_i(\theta^*) = 0$$
$$\nabla_\eta \mathcal{L}(\theta^*, \lambda^*, \eta^*) = h(\theta^*)^\top = 0;$$

Then we take partial derivative in relatively to each variable and multiplayer:

So, we have:

$$\frac{\partial L}{\partial \theta} = -\frac{2}{\theta^3} + \lambda_1 \left(b\theta^{b-1}\right) + \lambda_2 = 0 \;\Rightarrow \lambda_2 = \frac{2}{\theta^3} - \lambda_1 \left(b\theta^{b-1}\right) \quad (1)$$
$$\frac{\partial L}{\partial \lambda_1} = \left(\theta^b - \alpha\right) = g_1(\theta) \Rightarrow \lambda_1 g_1 = \lambda_1 \left(\theta^b - \alpha\right) = 0 \Rightarrow \left(\theta^b - \alpha\right) = 0 \text{ or } \lambda_1 = 0 \;\Rightarrow \left[\;\theta = \alpha^{1/b}\right]$$
$$\frac{\partial L}{\partial \lambda_2} = (\theta - 1) = g_2 \quad \Rightarrow \lambda_2 g_2 = \lambda_2 (\theta - 1) = 0 \quad \Rightarrow [\lambda_2 = 0]$$

And from one we have: $\lambda_1 b\theta^{b-1} = \dfrac{2}{\theta^3} \qquad \lambda_1 = \dfrac{2}{b\theta^{b-2}}$

Then with the $\alpha = 0.01$ and $b = 10$ $\theta^* = 0.6310$

(b) Program two different numerical methods to solve this problem for $\alpha = .01$ and $b = 10$. Plot the consecutive values of θ_n and discuss your results, comparing with the theoretical answer in part (a).

(Answer)

[GoogleColab_HW1](https://colab.research.google.com/drive/1u-SxxZ8fDp_VrruJm58eVr-tUfcB0usx?usp=sharing)

https://colab.research.google.com/drive/1u-SxxZ8fDp_VrruJm58eVr-tUfcB0usx?usp=sharing

Discuss results
Compare with part a

Gradient Descent Method:

Using theta of .25, our method converges at y = 0.9426 x = -1.0300 with thetaN = -1.03. Theta starts positive but over time it is minimized by alpha*derivative, so it slowly gets smaller and smaller.

In part a we found a positive theta* less than 1 , that the second constrain is in active and just the first constrain over beta is active.

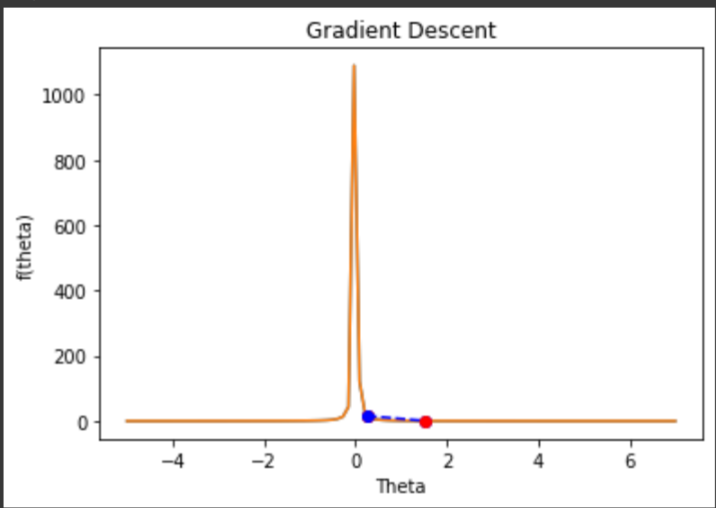
Gradient Descent

[8]

```
import numpy as np
import matplotlib.pyplot as plt
from matplotlib.ticker import MaxNLocator
from itertools import product


def C(theta):   #cost function
    return 1/(theta**2)

def cprime(theta):    #gradient cost function
    return -2/(theta**3)

def plotFunc(theta0):
    theta = np.linspace(-5, 7, 100)
    plt.plot(theta, C(theta))
    plt.plot(theta0, C(theta0),'ro')
    plt.xlabel('Theta')
    plt.ylabel('f(theta)')
    plt.title('Gradient Descent')

def plotPath(xs, ys, theta0):
    plotFunc(theta0)
    plt.plot(xs, ys, linestyle='--', marker='o', color='blue')
    plt.plot(xs[-1], ys[-1], 'ro')

theta0 = 0.25
plotFunc(theta0)
```

```python
def GradientDescent(function, cprime, theta0, alpha, beta):
    #setting the variables
    theta_K = theta0
    c_theta_k = C(theta_K)
    primeC_theta_k = -cprime(theta_K)
    x_axis = [theta_K] #store theta
    y_axis = [c_theta_k] #store c(theta)

    #P(theta) is suppposed to be less than or equal to alpha, P(theta) = theta^b
    while theta_K**beta <= alpha:
        theta_K = theta_K + alpha * primeC_theta_k
        c_theta_k = C(theta_K)
        primeC_theta_k = -cprime(theta_K)
        x_axis.append(theta_K) #plot theta
        y_axis.append(c_theta_k) #plot c(theta)
    # print results
    if theta_K**beta == alpha:
        print('Gradient descent does not converge.')
    else:
        print('Solution:\n  y = {:.4f}\n  x = {:.4f}'.format(c_theta_k, theta_K))
        print(theta_K)
    return x_axis, y_axis


xs, ys = GradientDescent(function, cprime, theta0, a, b)
plotPath(xs, ys, theta0)
```

```
Solution:
  y = 0.4272
  x = 1.5300
1.53
```



Also we tried the penalthy method but it