Homework10-Mathematical-Analysis

Tiberia-Giulia Farkas

19 December 2024

Problem Statement

Consider the quadratic function $f : \mathbb{R}^2 \to \mathbb{R}$ defined as

$$f(x, y) = \frac{1}{2}(x^2 + by^2),$$

where $b > 0$. The gradient descent method for finding the minimum of this function is given by

$$(x_{k+1}, y_{k+1}) = (x_k, y_k) - s_k \nabla f(x_k, y_k),$$

where $s_k > 0$ is the step size (also called the learning rate in machine learning literature).
For $b = 1, \frac{1}{2}, \frac{1}{5}, \frac{1}{10}$ plot the gradient descent iterations and some relevant contour lines of f to show convergence towards the minimum. What do you notice as b gets smaller?

Step Size Calculation

The step size s_k that minimizes the function

$$\varphi(s_k) = f((x_k, y_k) - s_k \nabla f(x_k, y_k))$$

is calculated as:

$$s_k = \frac{x_k^2 + b^2 y_k^2}{x_k^2 + b^3 y_k^2}.$$

Python Code for Gradient Descent Visualization

The following Python code demonstrates the gradient descent iterations and plots relevant contour lines of f for $b = 1, \frac{1}{2}, \frac{1}{5}, \frac{1}{10}$ to show convergence toward the minimum.

Listing 1: Gradient Descent Visualization

```
import numpy as np
import matplotlib.pyplot as plt

def function(x, y, b):
    return 0.5 * (x ** 2 + b * y ** 2)

def gradient_by_x(x):
    return x

def gradient_by_y(y, b):
    return b * y

def optimal_learning_rate(x, y, b):
    return (x ** 2 + b ** 2 * y ** 2) / (x ** 2 + b ** 3 * y ** 2) if x ** 2 + b ** 3 *
        y ** 2 != 0 else 0

def gradient_descent(x, y, b, max_iterations):
    iterations = 0
    steps = [(x,y)]

    while iterations < max_iterations:
```

```
        iterations += 1
        learning_rate = optimal_learning_rate(x, y, b)
        df_dx = learning_rate * gradient_by_x(x)
        df_dy = learning_rate * gradient_by_y(y, b)
        if df_dx == 0 or df_dy == 0:
            break

        x -= df_dx
        y -= df_dy
        steps.append((x,y))

    return steps, function(x, y, b), iterations

def plot_contour_lines():
    b_list = [1, 0.5, 0.2, 0.1]
    for b in b_list:
        x = np.linspace(-10, 10, 100)
        y = np.linspace(-10, 10, 100)
        X, Y = np.meshgrid(x, y)
        Z = function(X, Y, b)

        ax = plt.axes(projection='3d')

        path, result, iterations = gradient_descent(10, 10, b, 1000)
        steps_of_x = [p[0] for p in path]
        steps_of_y = [p[1] for p in path]
        steps_of_z = [function(x, y, b) for x, y in path]

        ax.plot_surface(X, Y, Z, rstride = 1, cstride = 1, alpha = 0.6, cmap = 'plasma',
             edgecolor = 'none')
        ax.plot3D(steps_of_x, steps_of_y, steps_of_z, 'green', label = 'Gradient Descent
             Path')
        ax.set_title(f"Contour lines for b = {b}", bbox = dict(boxstyle = 'round',
            facecolor = 'lavender', alpha = 0.8))

        text = (f"Learning rate: optimal\n"
                f"Minimum local value: {result}\n"
                f"Number of iterations: {iterations}")

        properties = dict(boxstyle = 'round', facecolor = 'aqua', alpha = 0.5)

        fig = plt.gcf()
        fig.text(0.02, 0.02, text, fontsize = 10, verticalalignment = 'bottom', bbox =
            properties)

        plt.legend(loc = 'upper right', bbox_to_anchor = (1.3, 1))
        plt.show()

plot_contour_lines()
```

Plots and Observations

As b becomes smaller, the shape of the function $f(x, y) = \frac{1}{2}(x^2 + by^2)$ becomes more stretched along the $y - axis$ compared to the $x - axis$. This means the function's "steepness" in the $y - direction$ is much less than in the $x - direction$. This stretch makes the function behave differently in the two directions, and the gradient descent method moves more slowly in the flatter direction (the $y - direction$ when b is small). The result is that it takes more steps to reach the minimum when b is smaller.

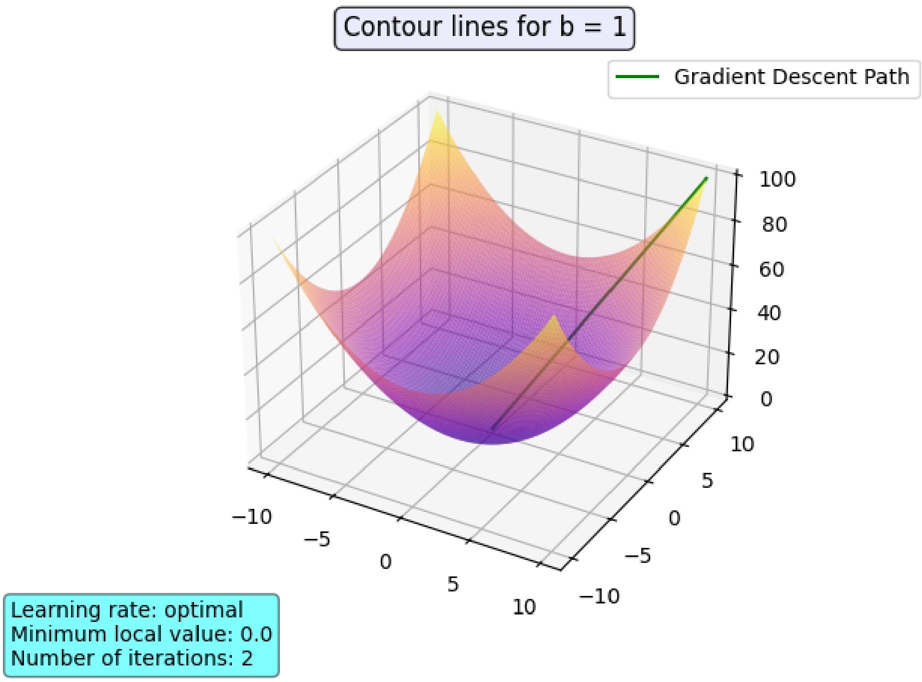


Figure 1: Contour lines for $b = 1$

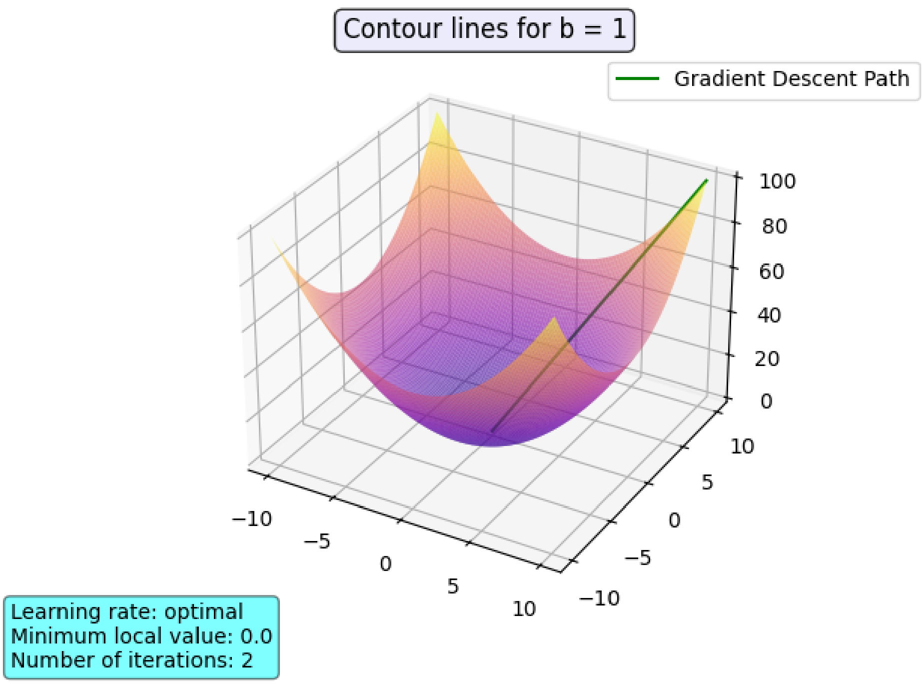


Figure 2: Contour lines for $b = 0.5$

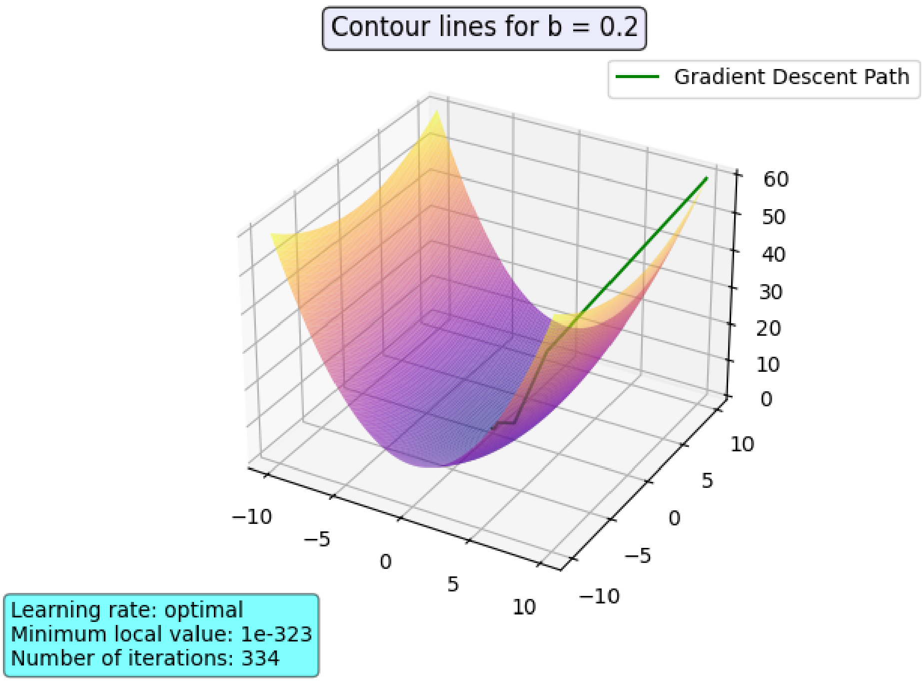


Figure 3: Contour lines for $b = 0.2$

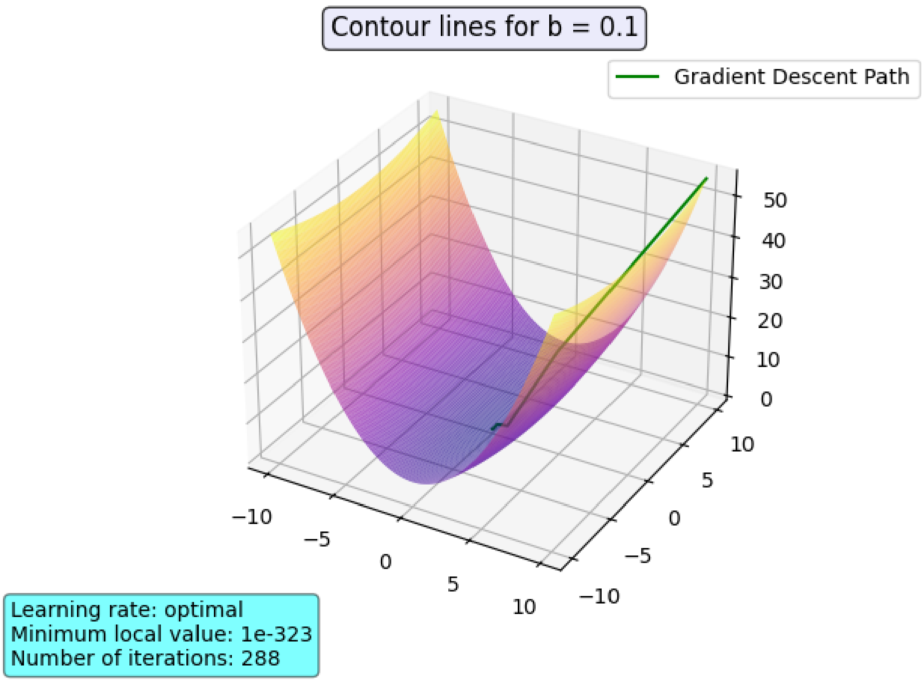
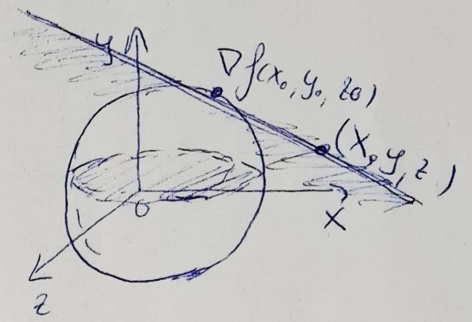


Figure 4: Contour lines for $b = 0.1$

Homework 10

1) The tangent plane to the unit sphere $x^2+y^2+z^2=1$ at (x_0, y_0, z_0)



$x^2+y^2+z^2=1$, $f(x,y,z) = x^2+y^2+z^2=1$

$\nabla f \perp$ level set

$\nabla f(x_0,y_0,z_0) \perp (x-x_0, y-y_0, z-z_0)$

$\nabla f(x_0,y_0,z_0)\cdot(x-x_0)(y-y_0)(z-z_0)=0$

$\frac{\partial f}{\partial x}(x_0 y_0 z_0)(x-x_0) + \frac{\partial f}{\partial y}(x_0, y_0, z_0)(y-y_0) + \frac{\partial f}{\partial z}(x_0, y_0, z_0)(z-z_0)=0$

$\frac{\partial f}{\partial x} = 2x$, $\frac{\partial f}{\partial y} = 2y$, $\frac{\partial f}{\partial z} = 2z$

$\Rightarrow$ Tangent : $2x_0(x-x_0) + 2y_0(y-y_0) + 2z(z-z_0)=0 \quad |:2$

$x_0(x-x_0) + \cancel{2}y_0(y-y_0) + z_0(z-z_0)=0$

$xx_0 - x_0^2 + yy_0 - y_0^2 + zz_0 - z_0^2 = 0$

$xx_0 + yy_0 + zz_0 - 1 = 0$

2) $x, y, f: \mathbb{R}^2 \to \mathbb{R}$, $f(x,y) = f(x(u,v), y(u,v))$

$\frac{\partial f}{\partial u} = \frac{\partial f \partial x}{\partial x \partial u} + \frac{\partial f \partial y}{\partial y \partial u}$ and $\frac{\partial f}{\partial v} = \frac{\partial f \partial x}{\partial x \partial v} + \frac{\partial f \partial y}{\partial y \partial v}$

We can take $F(u,v) = f[x(u,v), y(u,v))$

$\frac{\partial F}{\partial u} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial u} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial u}$ $\quad \frac{\partial F}{\partial v} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial v} + \frac{\partial f}{\partial y}\frac{\partial y}{\partial v}$

$u, v \xmapsto{g} x(u,v), y(u,v) \xmapsto{f} f(x(u,v), y(u,v))$

$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right)$ $\quad \nabla x = \left(\frac{\partial x}{\partial u}, \frac{\partial x}{\partial v}\right)$ $\quad \nabla y = \left(\frac{\partial y}{\partial u}, \frac{\partial y}{\partial v}\right)$

~~$D(f\circ g)(u) = D$~~

$$\Rightarrow Dg = \begin{pmatrix} \nabla x \\ \nabla y \end{pmatrix} = \begin{pmatrix} \frac{\partial x}{\partial u} & \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial u} & \frac{\partial y}{\partial v} \end{pmatrix}$$

$$D(f\circ g)(u) = D(f(g(u)))\, Dg(u) = \left[\frac{\partial f}{\partial x} g(u), \frac{\partial f}{\partial y} g(u)\right] \cdot$$

$$\cdot \begin{bmatrix} \frac{\partial x}{\partial u} \\ \frac{\partial y}{\partial u} \end{bmatrix} = \frac{\partial f}{\partial x} \cdot \frac{\partial x}{\partial u} + \frac{\partial f}{\partial y} \cdot \frac{\partial y}{\partial u}$$

$$\Rightarrow \frac{\partial f}{\partial u} = \frac{\partial f}{\partial x} \cdot \frac{\partial x}{\partial u} + \frac{\partial f}{\partial y} \cdot \frac{\partial y}{\partial u}$$

$$D(f\circ g)(v) = D(f(g(v)))\, Dg(v) = \left[\frac{\partial f}{\partial x} g(v), \frac{\partial f}{\partial y} g(v)\right] \cdot$$

$$\cdot \begin{bmatrix} \frac{\partial x}{\partial v} \\ \frac{\partial y}{\partial v} \end{bmatrix} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial v} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial v}$$

$$\Rightarrow \frac{\partial f}{\partial v} = \frac{\partial f}{\partial x} \frac{\partial x}{\partial v} + \frac{\partial f}{\partial y} \frac{\partial y}{\partial v}$$

3) $f: \mathbb{R}^2 \to \mathbb{R}$, $f(x,y) = \frac{1}{2}(x^2 + by^2)$, $b > 0$

gradient descent method for finding its minimum:

$$(x_{k+1}, y_{k+1}) = (x_k, y_k) - s_k \nabla f(x_k, y_k)$$

a) $s_k = ?$ s.t. it minimizes the function

$$\varphi(s_k) = f(x_{k+1}, y_{k+1}) = f((x_k, y_k) - s_k \nabla f(x_k, y_k))$$

$$\varphi'(s_k) = 0$$

$$\nabla f = \left(\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}\right) = (x, by)$$

$$\frac{\partial f}{\partial x} = x \quad , \quad \frac{\partial f}{\partial y} = by$$

Also $f(x,y) = \frac{1}{2}(x^2+by^2)$ has a unique global minimum at the origin $(0,0)$ for $b>0$

Starting from: $(x_{k+1}, y_{k+1}) = (x_k, y_k) - s_k \nabla f(x_k, y_k)$

we obtain:

$$x_{k+1} = x_k - s_k x_k = (1-s_k)\, x_k$$

$$y_{k+1} = y_k - s_k b y_k = (1-bs_k) y_k$$

$$\varphi(s) = f(x_{k+1}, y_{k+1}) = \frac{1}{2}\left\{(1-s_k)x_k)^2 + b[(1-bs_k)y_k]^2\right\}$$

$$\varphi(s) = \frac{1}{2}\left((1-s_k)^2 x_k^2) + b(1-bs_k)^2 y_k^2\right)$$

$$\varphi'(s) = 0 \Rightarrow \cancel{\frac{1}{2}[2(1-s_k)x_k^2 + 2(1-s_k)^2 x_k}$$

$$\Rightarrow \frac{1}{2}[2x_k^2(1-s_k)(-1) + 2by_k^2(1-s_k b)(-b)]$$

$$\varphi'(s) = -x_k^2(1-s_k) - b^2 y_k^2 (1-s_k b)$$

$$-x_k^2(1-s_k) - b^2 y_k^2 (1-s_k b) = 0$$

$$-x_k^2 + x_k s_k - b^2 y_k^2 + b^3 s_k y_k^2 = 0$$

$$x_k^2 s_k + b^3 s_k y_k^2 = x_k^2 + b^2 y_k^2$$

$$s_k(x_k^2 + b^3 y_k^2) = x_k^2 + b^2 y_k^2$$

$$s_k = \frac{x_k^2 + b^2 y_k^2}{x_k^2 + b^3 y_k^2}$$