Cálculo - Lista

Dyanna Cruz
Guilherme Camblor
Renato Campos

2 de dezembro de 2022

Exercício 1. *Ache os pontos de máximo e/ou de mínimo da função* $f(x, y) = x^4y^3$ *sejeito à* $x + y = 1$.

Resposta 1.

$$f(x, y) = x^4y^3 \to 4x^3y^3 \ e \ x^43y^2$$
$$g(x, y) = x + y - 1$$
$$\bigtriangledown f(x, y) =$$

$$\begin{cases} 4x^3y^3 = \lambda \\ x^43y^2 = \lambda \\ x + y - 1 = 0 \end{cases}$$

$$4x^3y^3 = \lambda \to 4x^3y^3 = x^43y^2 \to y = 4x^33x^4 \to y = \frac{3}{4}x$$
$$x + y - 1 = 0 \to x + \frac{3}{4}x = 1 \to \frac{7x}{4} = 1 \to 7x = 4 \to x = \frac{4}{7}$$
$$x + y - 1 = 0 \to \frac{4}{7} + y = 1 \to y = 1 \cdot \frac{-4}{7} \to \frac{7-4}{7} \to y = \frac{3}{7}$$

Portanto o ponto $P = \left(\frac{4}{7}, \frac{3}{7}\right)$ *e a função no ponto* $P = \frac{4^4 \cdot 3^3}{7^7}$.

Entretanto, dessa forma, não conseguimos observar se o ponto é de máximo ou mínimo. Vamos voltar e derivar uma segunda vez e analisar para onde o P tende.

Exercício 2. *Ache os pontos críticos da função* $f(x, y) = x^2 + y^2 + 3xy - x + y$ *e classifique-os (máximo, mínimo ou sela).*

Resposta 2.

$$f(x, y) = 2x + 3y - 1 \ e \ 2y + 3x + 1$$
$$\bigtriangledown f(x, y) = (2x + 3y - 1, 2y + 3x + 1) = (0, 0)$$
$$\begin{cases} 2x + 3y = 1 \to 2x + 3y - 1 = 0 \\ 2y + 3x = -1 \to 2y + 3x + 1 = 0 \end{cases}$$

$$2x + 3x + 3y + 2y = 0$$
$$5x + 5y = 0$$
$$5x = -5yx = \frac{-5y}{5}$$
$$x = -y$$

$$\begin{aligned}
2y + 3x &= -1 \\
2y + 3 \cdot (-y) &= -1 \\
-y &= -1 \rightarrow y = 1
\end{aligned}$$

$$\begin{aligned}
2x + 3y &= 1 \\
2x + 3 \cdot 1 &= 1 \\
2x + 3 &= 1 \\
2x &= 1 - 3 \\
2x = -2x &= \frac{-2}{2} \\
x &= -1
\end{aligned}$$

$P = (-1, 1)$. *Agora para poder classificar temos de encontrar o det da matriz Hessiana.*

$$(\text{Hess}\, f)_{ij} \equiv \frac{\partial^2 f}{\partial x_i \partial x_j} \rightarrow \text{Hess}\,(x_0, y_0) = \begin{bmatrix} \frac{\partial^2 f}{\partial x^2}(x_0, y_0) & \frac{\partial^2 f}{\partial x \partial y}(x_0, y_0) \\ \frac{\partial^2 f}{\partial x \partial y}(x_0, y_0) & \frac{\partial^2 f}{\partial y^2}(x_0, y_0) \end{bmatrix}$$

$$\begin{aligned}
f(x, y) &= x^2 + y^2 + 3xy - x + y \\
f'(x) &= 2x + 3y - 1 \\
f''(y) &= 3 \\
\frac{\partial f}{\partial x \partial y} &= 3 \\
\text{Hess} &= \begin{bmatrix} 2 & 3 \\ 3 & 2 \end{bmatrix} \\
det = (2 \cdot 2) - (3 \cdot 3) &= 4 - 9 = -5
\end{aligned}$$

Como o det é menor que zero, então podemos classificar que é um ponto de sela.
Para achar o ponto crítico da função, temos que:

$$f(-1, 1) = -1^2 + 1^2 + 3 \cdot (-1) \cdot (1) + 1 + 1 \rightarrow 1 + 1 - 3 + 1 + 1 \rightarrow 4 - 3 = 1$$

Logo o ponto crítico da função é (-1,1,1).

Exercício 3. *Ache o(s) ponto(s) do plano* $3x + y - z = 1$ *mais próximo de (1,1,1).*

Resposta 3.

Passo 1: Função objetiva:

$$\begin{aligned}
T(x, y, z) &= f(x_0, y_0) + f(x_0, y_0) \cdot (x - x_0) + f(x_0, y_0) \cdot (y - y_0) \\
f(x, y, z) &= x^2 + 1 - 2x + y^2 - 2y + 1 + z^2 - 2z + 1 \\
f(x, y, z) &= x^2 - 2x + y^2 - 2y + z^2 - 2z + 3 \\
g(x, y, z) &= 3x + y - z - 1 \rightarrow 3x + y - z = 1
\end{aligned}$$

Passo 2: Método dos multiplicadores de Lagrange:

$$\begin{cases} f(x, y, z) = x^2 - 2x + y^2 - 2y + z^2 - 2z + 3 \\ g(x, y, z) = 3x + y - z - 1 \end{cases}$$

$$\begin{aligned}
&\bigtriangledown f(x.y.z) = \lambda \bigtriangledown g(x, y, z) \\
&\frac{\partial f}{\partial x} = 2x - 2 \quad \frac{\partial f}{\partial x} = 2y - 2 \quad \frac{\partial f}{\partial x} = 2z - 2 \\
&\frac{\partial g}{\partial x} = 3 \quad \frac{\partial g}{\partial x} = 1 \quad \frac{\partial g}{\partial x} = -1 \\
&\begin{cases} 2x - 2 = 3\lambda \\ 2y - 2 = \lambda \\ 2z - 2 = -\lambda \end{cases}
\end{aligned}$$

$$2x - 2 = 3\lambda$$
$$2x = 3\lambda + 2$$
$$x = \frac{3\lambda + 2}{2}$$

$$2y - 2 = \lambda$$
$$2y = \lambda + 2$$
$$y = \frac{\lambda + 2}{2}$$

$$2z - 2 = -\lambda$$
$$2z = -\lambda + 2$$
$$z = \frac{-\lambda + 2}{2}$$

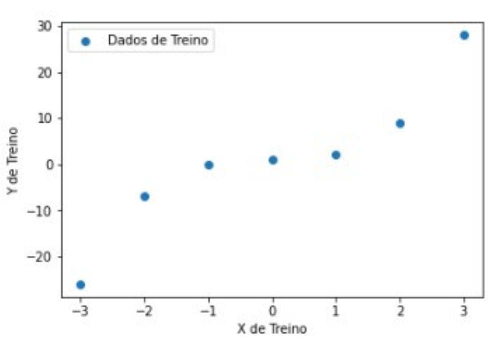
$$P(x, y, z) = P\left(\frac{3\lambda + 2}{2}, \frac{\lambda + 2}{2}, \frac{-\lambda + 2}{2}\right)$$
$$P(x, y, z) = P\left(\frac{5}{11}, \frac{9}{11}, \frac{13}{11}\right)$$

Exercício 4. *(Regressão "parabólica")*
Dado um conjunto de treinamento $\{(0,1),(1,2),(2,9),(3,28),(-1,0),(-2,-7),(-3,-26)\}$. *Ache a melhor parábola* $\hat{y} = w_1x^2x + b$ *de modo a minimizar a função de erro quadrático* $E(w_1, w_2, b)$. *Quanto seria y(5)?*

Resposta 4.

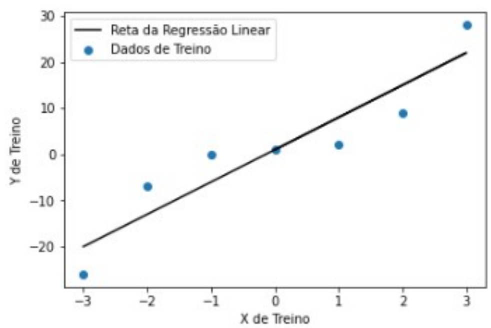
```python
# Bibliotecas
import numpy as np
import matplotlib.pyplot as plt

x_treino = np.array([0, 1, 2, 3, -1, -2, -3])
y_treino = np.array([1, 2, 9, 28, 0, -7, -26])
x_media = np.mean(x_treino)
y_media = np.mean(y_treino)

a = np.sum(x_treino * (y_treino - y_media)) / np.sum(x_treino *
                                                    (x_treino - x_media))
b = y_media - (a * x_media)

funcao = a * x_treino + b
R_quadrado = np.sum((funcao-y_media) ** 2) / np.sum((y_treino - y_media) ** 2)

print(f"R² é: {np.round(R_quadrado, 2)}\n")

plt.figure()
plt.scatter(x_treino, y_treino, label = "Dados de Treino")
plt.xlabel("X de Treino")
plt.ylabel("Y de Treino")
plt.legend()
plt.show()
```

```
R² é: 0.86
```



```python
# Bibliotecas
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import linregress

x_treino = np.array([0, 1, 2, 3, -1, -2, -3])
y_treino = np.array([1, 2, 9, 28, 0, -7, -26])
a2, b2, r2, p_value, std_err = linregress(x_treino, y_treino)
R_squared2 = r2**2

print(f"R² é: {R_squared2:.2f}\n")

plt.figure()
plt.plot(x_treino, a2 * x_treino + b2, 'k', label = "Reta da Regressão Linear")
plt.scatter(x_treino, y_treino, label = "Dados de Treino")
plt.xlabel("X de Treino")
plt.ylabel("Y de Treino")
plt.legend()
plt.show()
```

```
R² é: 0.86
```

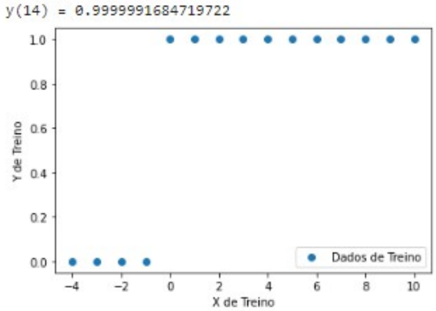


Exercício 5. *(Regressão "logística") Dado um conjunto de treinamento:*
$\{(0,1),(1,1),(2,1),(3,1),(4,1),(5,1),(6,1),(7,1),(8,1),(9,1),(10,1),(-1,1),(-2,0),(-3,0),(-4,0),(-5,0)\}$.
Ache a melhor sigmóide $\hat{y} = \frac{1}{1+e^{-(wx+b)}}$ *de modo a minimizar a função de erro quadrático* $E(w,b)$. *Quanto seria* $y(14)$ *e* $y(-7)$*?*

Resposta 5.

```python
# Bibliotecas
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import math

# Função
def sigmoid(x):
  return 1 / (1 + math.exp(-x))

# Conjunto de Treinamento
X = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, -1, -2, -3, -4]
Y = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0]
data = pd.DataFrame((zip(X, Y)), columns = ["X", "Y"])
X = np.array([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, -1, -2, -3, -4])
Y = np.array([1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0])

# Função
def normalizeData(data):

    for feature in data.columns:
        maxValue = data[feature].max()
        minValue = data[feature].min()

        data[feature] = (data[feature] - minValue) / (maxValue - minValue)

    return data

data = normalizeData(data)
x_media = np.mean(X)
y_media = np.mean(Y)

a = np.sum(X * (Y - y_media)) / np.sum(X * (X - x_media))
b = y_media - (a * x_media)
funcao = sigmoid(14)
R_quadrado = np.sum((funcao-y_media) ** 2) / np.sum((Y - y_media) ** 2)

print(f"R² é: {np.round(R_quadrado, 2)}\n")
print("y(14) =", funcao)

plt.figure()
plt.scatter(X, Y, label = "Dados de Treino")
plt.xlabel("X de Treino")
plt.ylabel("Y de Treino")
plt.legend()
plt.show()
```

```
R² é: 0.02

y(14) = 0.9999991684719722
```



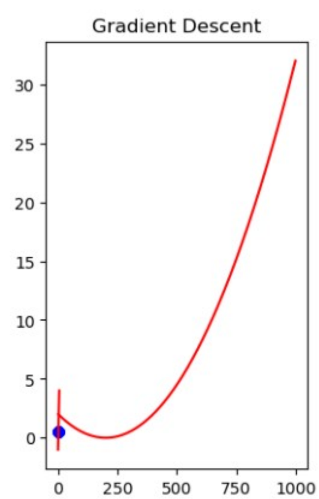
Exercício 6. *Ache os pontos de mínimo da função* $f(x, y) = x^2y^2$ *sejeito à* $x + y = 1$ *usando o algoritmo do gradiente descendente. Diga considere minimizar* $H(x, y, \lambda) = \bigtriangledown f(x, y) - \lambda \bigtriangledown g(x, y)$.

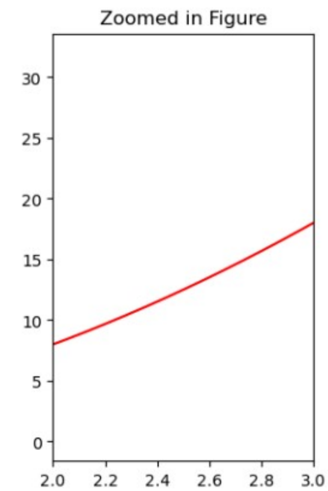
Resposta 6.

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

f_x = lambda x,y: (x**2)+(y**2)

x = np.linspace(-1,4,1000)
y = x

f_x_derivative = lambda x,y: 2*x+2*y

def plot_gradient(x, y, x_vis, y_vis):
    y = x
    plt.subplot(1,2,2)
    plt.scatter(x_vis, y_vis, c = "b")
    plt.plot(x, y, f_x(x, y), c = "r")
    plt.title("Gradient Descent")
    plt.show()
    plt.subplot(1,2,1)
    plt.scatter(x_vis, y_vis, c = "b")
    plt.plot(x,f_x(x, y), c = "r")
    plt.xlim([2.0,3.0])
    plt.title("Zoomed in Figure")
    plt.show()

def gradient_iterations(x_start, y_start,iterations, learning_rate):

    x_grad = [x_start]
    y_grad = [f_x(x_start, y_start)]
    for i in range(iterations):

        x_start_derivative = - f_x_derivative(x_start, x_start)

        x_start += (learning_rate * x_start_derivative)

        x_grad.append(x_start)
        y_grad.append(f_x(x_start, y_start))
    print ("Local minimum occurs at: {:.2f}".format(x_start, y_start))
```

```
    print ("Number of steps: ",len(x_grad)-1)
    plot_gradient(x, f_x(x, y) ,x_grad, y_grad)

gradient_iterations(0.5, 0.5, 100, 0.0001)
```

```
Local minimum occurs at: 0.48
Number of steps:  100
```





Exercício 7. *Com base nos dados históricos do último ano, monte um portifólio de risco mínimo envolvendo as ações: ITUB4, BBSA4, BBDC4 e BCSA34.*

Resposta 7.

```
!pip install yfinance
!pip install riskfolio-lib

# Bibliotecas
import numpy as np
import pandas as pd
import riskfolio as rp
from pandas_datareader import data as pdr
import yfinance as yf
from matplotlib import pyplot as plt
yf.pdr_override()

# Busca os preços das ações
assets = ['ITUB4.SA', 'BBAS3.SA', 'BBDC4.SA', 'BCSA34.SA']

# Data início
start = '2021-11-23'
end = '2022-11-25'

# Preços ajustados
prices = pdr.get_data_yahoo(assets, start = start, end = end)['Adj Close']

# Calcula os retornos
returns = prices.pct_change().dropna()

# Cria o objeto de portfolio
port = rp.Portfolio(returns = returns)
method_mu = 'hist'
method_cov = 'hist'

# Inputs do método de otimização
port.assets_stats(method_mu = method_mu, method_cov = method_cov, d = 0.94)

# Estima portfolio ótimo
model='Classic'
rm = 'MV'
obj = 'MinRisk'
w = port.optimization(model = model, rm = rm, obj = obj)
print(w * 100)
```

```
             weights
BBAS3.SA   24.463445
BBDC4.SA    0.203174
BCSA34.SA  22.240873
ITUB4.SA   53.092509
```