**沈思妤-10235501458-数据科学导论第五次作业**

**复习题：**

1. **PageRank的设计思想是什么？**

**答：**

链接分析：PageRank认为一个网页的重要性不仅取决于其内容，还取决于指向它的其他网页的质量和数量。如果许多重要的网页链接到某个网页，那么这个网页也被视为重要。

随机游走模型：PageRank模型可以看作是一个随机游走过程。想象一个用户在网络中随机浏览网页，每次随机选择一个链接，最终停留在某个网页的概率与该网页的PageRank值成正比。

迭代计算：PageRank通过迭代的方式计算每个网页的分数，直到分数稳定。在每次迭代中，网页的分数会根据其链接的网页的分数进行更新。

1. **贝叶斯定理的内容是什么？它又有哪些重要应用？**

**答：**贝叶斯定理是概率论中的一个基本概念，包括：

先验概率：在观察到某个证据之前，我们对事件发生的初步信念。这是基于以往的经验或知识的概率。

条件概率：当我们知道某个相关事件发生后，另一个事件发生的概率。例如，如果我们知道某个人有某种症状，基于这个信息来判断他是否得了某种疾病的概率。

后验概率：在获得新证据后，我们对事件发生概率的更新结果。这是我们根据新信息修正的信念。

贝叶斯定理的应用：在医疗领域，医生会根据病人的症状和体征来更新对某种疾病诊断的可能性。在机器学习中，贝叶斯定理被用来改进分类算法的准确性。在自然语言处理中，它也被用于文本分类和情感分析，帮助计算机更好地理解人类语言。

1. **试阐述蒙特卡罗方法的基本原理。**

**答：**

随机抽样：通过生成大量随机样本来近似计算某个值或概率。

估计：对所感兴趣的量（如积分、期望值等）进行估计，例如通过计算随机样本在特定区域内的比例来估计该区域的面积。

收敛性：随着随机样本数量的增加，估计结果会趋近于真实值，即统计收敛。

应用广泛：蒙特卡罗方法广泛应用于物理、金融、工程等领域，如风险评估、优化问题、模拟复杂系统等。

**5.梯度下降法的主要思想是什么？你能用通俗的语言解释出来吗？**

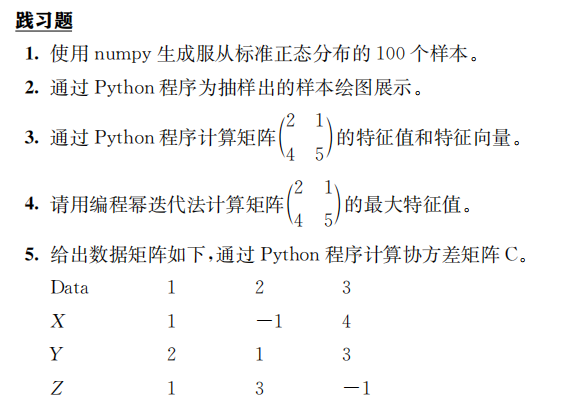
**答：**

方向选择：在多维空间中，梯度是一个向量，指向函数上升最快的方向。梯度的反方向则是函数下降最快的方向。

逐步更新：从一个初始点出发，计算该点的梯度，然后按照梯度的反方向移动一小步。这一小步的长度称为学习率。

迭代过程：重复上述步骤，直到达到预设的停止条件（如梯度非常小或经过的步数达到限制）。

通俗地说，梯度下降法就像在一座山上寻找最低点。你先朝着斜坡最陡的方向走，然后每次都选择一个小的步子向下走。随着你不断走，你会逐渐接近山谷的底部，最终找到最低点。

****

**1.答：**import numpy as np

samples = np.random.normal(loc=0, scale=1, size=100)

print(samples)

**2.答：**

import numpy as np

import matplotlib.pyplot as plt

samples = np.random.normal(loc=0, scale=1, size=100)

print(samples)

plt.hist(samples, bins=10, density=True, alpha=0.6, color='g')

xmin, xmax = plt.xlim()

x = np.linspace(xmin, xmax, 100)

p = np.exp(-0.5 \* x\*\*2) / np.sqrt(2 \* np.pi)

plt.plot(x, p, 'k', linewidth=2)

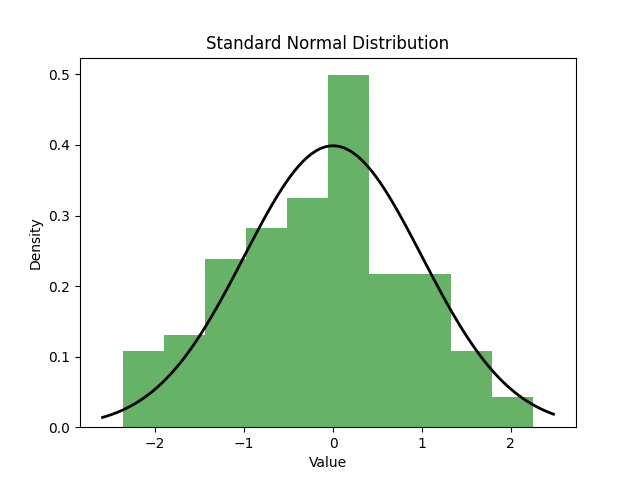
plt.title('Standard Normal Distribution')

plt.xlabel('Value')

plt.ylabel('Density')

plt.show()

图片如下：



**3.答：**import numpy as np

matrix = np.array([[2, 1],

[4, 5]])

eigenvalues, eigenvectors = np.linalg.eig(matrix)

print(eigenvalues)

print(eigenvectors)

运行后输出是：

[1. 6.]

[[-0.70710678 -0.24253563]

[ 0.70710678 -0.9701425 ]]

1. **答：**

import numpy as np

data = np.array([[1, -1, 4],

[2, 1, 3],

[1, 3, -1]])

cov\_matrix = np.cov(data)

print(cov\_matrix)

运行后输出是：

[[ 6.33333333 2.5 -5. ]

[ 2.5 1. -2. ]

[-5. -2. 4. ]]

**求解 f(×)=0.25\*(x-0.5)^2＋1的局部极小值，绘图展示梯度下降法的迭代过程**

**答：**代码是：

import numpy as np

import matplotlib.pyplot as plt

def f(x):

return 0.25 \* (x - 0.5)\*\*2 + 1

def df(x):

return 0.5 \* (x - 0.5)

learning\_rate = 0.1

x\_init = 3

iterations = 20

x\_vals = [x\_init]

f\_vals = [f(x\_init)]

x = x\_init

for i in range(iterations):

grad = df(x)

x = x - learning\_rate \* grad

x\_vals.append(x)

f\_vals.append(f(x))

x\_range = np.linspace(-1, 4, 400)

plt.figure(figsize=(8, 6))

plt.plot(x\_range, f(x\_range), label='f(x)', color='blue')

plt.scatter(x\_vals, f\_vals, color='red', zorder=5, label='梯度下降迭代点')

for i in range(len(x\_vals)):

plt.text(x\_vals[i], f\_vals[i], f'{i}', fontsize=9)

plt.title('梯度下降法求解 f(x) = 0.25\*(x-0.5)^2 + 1 的极小值')

plt.xlabel('x')

plt.ylabel('f(x)')

plt.legend()

plt.grid(True)

plt.show()

图片是：