第一次仿真实验报告

冯海桐-522031910557

2024 年 11 月 10 日

1 仿真实验题1

1.1 实验描述

通过仿真实验验证高维数据的集中分布。利用高斯分布随机函数生成维数d（$d = 1000, 500, 100, 3$）四组样本数据（样本个数 $N = 10000$），分别计算每一个样本与原点的距离，估计每一组样本数据的其距离的分布函数，并画出相应分布函数的波形曲线。

1.2 实验分析

设高斯分布随机函数为$X \sim N(0,1)$，则d维高斯分布随机函数为$X \sim N(0, I_d)$，其中I_d为d维单位矩阵。设$X = (X_1, X_2, \cdots, X_d)$，$Y = |X|$，$Z = Y^2$，则$Z$服从自由度为$d$的卡方分布，即$Z \sim \chi^2(d)$。

Z的概率密度函数为

$$f_Z(z) = \frac{1}{2^{d/2}\Gamma(d/2)} z^{d/2-1} e^{-z/2} \tag{1}$$

则Z的分布函数为

$$F_Z(z) = \int_0^z f_Z(t)dt = 1 - \frac{\gamma(d/2, z/2)}{\Gamma(d/2)} \tag{2}$$

其中$\gamma(a, x) = \int_0^x t^{a-1} e^{-t} dt$为不完全伽马函数。

则Y的分布函数为

$$F_Y(y) = F_Z(y^2) = 1 - \frac{\gamma(d/2, y^2/2)}{\Gamma(d/2)} \tag{3}$$

则Y的概率密度函数为

$$f_Y(y) = \frac{dF_Y(y)}{dy} = \frac{y^{d-1} e^{-y^2/2}}{2^{d/2-1}\Gamma(d/2)} \tag{4}$$

其概率密度最大点为$y = \sqrt{d-1}$。

1.3 实验代码

利用python编写代码如下：

```python
import numpy as np
import matplotlib.pyplot as plt
from scipy.special import gammaln

# Set the dimensions and the number of samples
```

```
dimensions = [1000, 500, 100, 3]
N = 10000

plt.figure(figsize=(12, 8))

for d in dimensions:
    # Generate random samples
    data = np.random.randn(N, d)

    # Calculate the distances
    distances = np.linalg.norm(data, axis=1)

    # Plot the histogram of the distances
    plt.hist(distances, bins=100, density=True, alpha=0.6, label=f'd={d}')

    # Plot the theoretical distribution
    y = np.linspace(1e-10, np.max(distances), 1000)
    log_pdf = (d-1) * np.log(y) - y**2 / 2 - (d/2 - 1) * np.log(2) - gammaln(d/2)
    pdf = np.exp(log_pdf)
    plt.plot(y, pdf, linewidth=2)

# Show the plot
plt.title('Distribution of distances from the origin')
plt.xlabel('Distances')
plt.ylabel('Density')
plt.legend()
plt.show()
```

先生成d维高斯分布随机函数，计算每一个样本与原点的距离，然后画出相应分布函数的波形曲线。由于 $d = 1000, 500$ 时，指数计算过于庞大，故取对数计算，最后再取指数。

1.4 实验结果

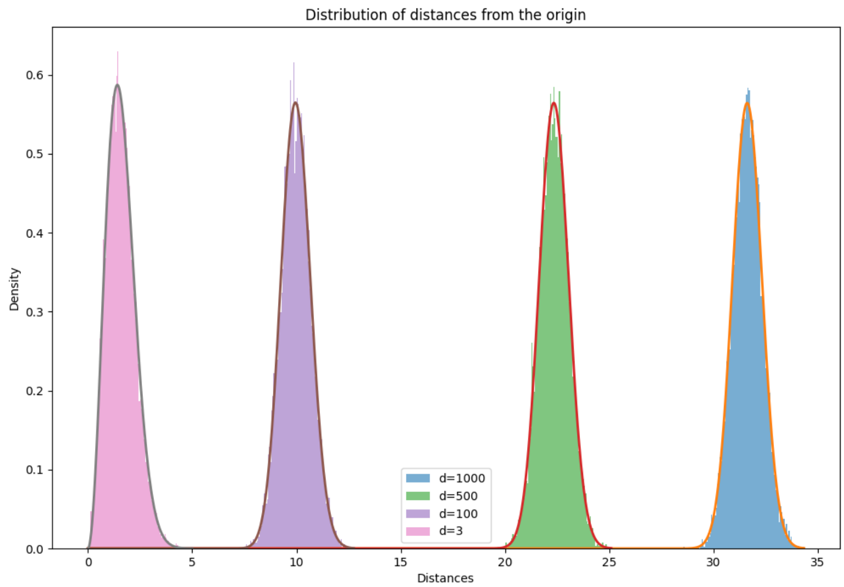

图 1: 高维数据的集中分布

由此可见，仿真实验结果与理论分析基本吻合。

2 仿真实验题2

2.1 实验描述

设 $f(x) = x_1x_2x_3 + x_3^3$ 是一个三个自变量的函数。考虑统计模型 $y = f(x) + \varepsilon, x \in R^n$，其中 ε 服从零均值，单位方差的高斯分布。分别对不同维数自变量 $n = 5, 10, 20, 30$，进行采样 $N = 1000$ 个样本数据。利用三次多项式进行回归，分别计算不同维数回归模型的方差，并讨论模型方差随着自变量维数增加时的变化规律。

2.2 实验分析

由于 $f(x) = x_1x_2x_3 + x_3^3$ 是一个三个自变量的函数，随着自变量维数的增加，回归模型的方差理应不变。但是由于采样数据的误差，回归模型随着自变量维数（不相关特征）的增加，容易产生过拟合现象，导致模型方差增大。

2.3 实验代码

利用python编写代码如下：

```
import numpy as np
from sklearn.preprocessing import PolynomialFeatures
```

```python
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

# Set the dimensions and the number of samples
dimensions = [5, 10, 20, 30]
N = 1000

# Store the variances
variances = []

for d in dimensions:
    X = np.random.randn(N, d)
    y = X[:, 0] * X[:, 1] * X[:, 2] + X[:, 2]**3 + np.random.randn(N)

    # Split the data into training and testing sets
    X_train, X_test, y_train, y_test = train_test_split(X, y,
        test_size=0.3, random_state=42)

    # Fit the polynomial regression model
    poly = PolynomialFeatures(degree=3)
    X_train_poly = poly.fit_transform(X_train)
    X_test_poly = poly.transform(X_test)

    model = LinearRegression()
    model.fit(X_train_poly, y_train)

    # Calculate the variance
    y_pred = model.predict(X_test_poly)
    variance = np.var(y_pred - y_test)
    variances.append(variance)

# Print the variances
for d, var in zip(dimensions, variances):
  print(f"Model variance in dimension {d}: {var}")
```

先生成 n 维高斯分布随机函数，然后进行多项式回归，计算模型方差。

2.4 实验结果

进行三次实验，得到不同维度下的模型方差如表1所示。

维度	第一次	第二次	第三次
5	1.126	1.100	1.053
10	2.058	2.190	2.326
20	3.086	3.007	3.226
30	3.183	3.287	4.576

表 1: 不同维度下的模型方差

由此可见，随着自变量维数增加，模型出现过拟合现象，模型方差增大。