BoostStrap主成分分析

朱金秋 1220086621

In [1]:

```
import pandas as pd
import numpy as np
```

数据处理

注：因为给的数据python读取是字符型，所以有数据处理阶段

In [2]:

```
data = pd.read_csv("D:/Data/mardia.dat",header=None)
f = open("D:/Data/mardia.dat")
L =[]
for lines in f.readlines():
    a =lines.split()
    L.append(a)
data = pd.DataFrame(L)  #去除分隔符，导入到数据框中

for i in range(0,5):
    data[i]=pd.to_numeric(data[i])#转成数值型
```

函数编写

In [3]:

```
"""
设置随机抽样子函数
输入：抽样数量
输出：抽处的样本
"""
def random_sample(n):
    rd = list(n*np.random.rand(n))#U（0，n）随机数
    index = []
    for a in rd:
        index.append(int(a))
        xb=data.iloc[index]
    return xb
```

In [15]:

```
"""
主函数
输出结果

"""
if __name__ == '__main__':
    n = data.shape[0] #样本量
    cov_data = data.cov()#协方差
    data_val,data_xvec = np.linalg.eig(np.array(cov_data))#特征值,特征向量
    theta = data_val.max()/data_val.sum()#计算最大特征值/总特征值和
    print("原始数据的特征值:\n",data_val)
    print("原始数据的特征向量:\n",data_xvec)
    print("原始数据最大theta:\n",theta)
    Nr =1000
    dataval_Nr = []#建立空列表用于存储特征根数据
    theta_Nr = []#建立空列表用于存储theta值
    for i in range(Nr):  #循环200次
        xb = random_sample(n)#调用子函数
        cov_xb = xb.cov()
        xb_val,xb_vec = np.linalg.eig(np.array(cov_xb))
        xb_theta = xb_val.max() / xb_val.sum()
        dataval_Nr.append(list(xb_val))
        theta_Nr.append(xb_theta)

    mean_theta =sum(theta_Nr)/Nr  #计算theta均值
    var_theta =sum(np.array(theta_Nr)**2)/Nr-mean_theta**2 #计算方差
    SE_theta = np.sqrt(var_theta)

    print("1000次循环后的theta均值:\n",mean_theta)
    print("1000次循环后的theta方差:\n",var_theta)
    print("1000次循环后的SE_theta:\n",SE_theta)
```

```
原始数据的特征值:
 [686.98981044 202.11107121  32.15328545  84.63044329 103.74731228]
原始数据的特征向量:
 [[-0.50544565 -0.74874751  0.07939388 -0.29618426 -0.29978884]
 [-0.36834859 -0.20740314  0.18887639  0.78288817  0.41559003]
 [-0.34566119  0.07590813 -0.92392015  0.00323634  0.14531817]
 [-0.45112258  0.30088849  0.28552169 -0.51813972  0.59662645]
 [-0.53465013  0.54778205  0.15123239  0.17573202 -0.60027584]]
原始数据最大theta:
 0.619115038421291
1000次循环后的theta均值:
 0.6195260527235017
1000次循环后的theta方差:
 0.0021713093177880283
1000次循环后的SE_theta:
 0.046597310199066516
```

In [ ]: