实验 PCA

姓名: 李昀哲 学号: 20123101

1 题目和数据：

题目： PCA

- 通过对给定数据进行主成分分析来了解 PCA 算法

数据描述:

一组生成的二维数据;

10.235186 11.321997
10.122339 11.810993
9.190236 8.904943
9.306371 9.847394
8.330131 8.340352
10.152785 10.123532
10.408540 10.821986
9.003615 10.039206
9.534872 10.096991
9.498181 10.825446
9.875271 9.233426
10.362276 9.376892
10.191204 11.250851
7.720499 6.476300

2 算法：

主成分分析是设法将原来众多具有一定相关性（比如 P 个指标），重新组合成一组新的互相无关的综合指标来代替原来的指标。

主成分分析，是考察多个变量间相关性一种多元统计方法，研究如何通过少数几个主成分来揭示多个变量间的内部结构，即从原始变量中导出少数几个主成分，使它们尽可能多地保留原始变量的信息，且彼此间互不相关.通常数学上的处理就是将原来 P 个指标作线性组合，作为新的综合指标。

主成分分析的原理是设法将原来变量重新组合成一组新的相互无关的几个综合变量，同时根据实际需要从中可以取出几个较少的总和变量尽可能多地反映原来变量的信息的统计方法叫做主成分分析或称主分量分析，也是数学上处理降维的一种方法。主成分分析是设法将原来众多具有一定相关性（比如 P 个指标），重新组合成一组新的互相无关的综合指标来代替原来的指标。通常数学上的处理就是将原来 P 个指标作线性组合，作为新的综合指

标。最经典的做法就是用 F1（选取的第一个线性组合，即第一个综合指标）的方差来表达，即 Va（rF1）越大，表示 F1 包含的信息越多。因此在所有的线性组合中选取的 F1 应该是方差最大的，故称 F1 为第一主成分。如果第一主成分不足以代表原来 P 个指标的信息，再考虑选取 F2 即选第二个线性组合，为了有效地反映原来信息，F1 已有的信息就不需要再出现再 F2 中，用数学语言表达就是要求 Cov（F1,F2）=0，则称 F2 为第二主成分，依此类推可以构造出第三、第四，……，第 P 个主成分

3 代码及结果

数据集读入

```
def loadDataSet(fileName, delim = '\t'):
    """
       函数说明：加载数据集
       parameters:
           fileName -数据集名称
           delim -分隔符
       return:
       mat(datArr) -数据矩阵
    """
    fr = open(fileName)
    stringArr = [line.strip().split(delim) for line in fr.readlines()]
    datArr = [list(map(float,line)) for line in stringArr]
    return mat(datArr)  # 转换为二维的数组便于运算
```

PCA 原理实现

```
def pca(dataMat, topNfeat = 9999999):
    """
    函数说明：PCA算法实现
    parameters:
        dataMat -用于进行PCA操作的数据集
        topNfeat -应用的N个特征
    return:
        lowDataMat -将维后的数据集
        reconMat -重构的数据集
"""
    meanVals = mean(dataMat,axis=0)
    meanRemoved = dataMat - meanVals

    covMat = cov(meanRemoved,rowvar=0)
    eigVals,eigVects = linalg.eig(mat(covMat))
    eigValInd = argsort(eigVals)
    eigValInd = eigValInd[:-(topNfeat+1):-1]
    redEigVects = eigVects[:,eigValInd]
    lowDDataMat = meanRemoved * redEigVects
    reconMat = (lowDDataMat * redEigVects.T) + meanVals
    return lowDDataMat,reconMat
```

画图

```
def drawDataSet(dataMat, reconMat):
    """
    函数说明：绘制数据集
    parameters:
        dataMat -原始数据集
        reconMat -重构数据集
    return:
        A picture
    """
    fig = plt.figure()
    ax = fig.add_subplot(111)
    ax.scatter(dataMat[:,0].flatten().A[0],dataMat[:,1].flatten().A[0],marker='^',s=90)
    ax.scatter(reconMat[:,0].flatten().A[0],reconMat[:,1].flatten().A[0],marker='o',s=50,c='red')
    plt.show()
```

测试结果

降维前的结果以及降维后的结果