

浙江大学

本科实验报告

Principal Component Analysis

课程名称： 人工智能实验

姓 名：

学 院： 信息与工程学院

专 业： 信息工程

学 号：

指导老师： 胡浩基、魏准

2023 年 6 月 4 日

一、 实验题目

1. 实验 4-1

利用 PCA 函数, 对 testSet.txt 中数据做降维分析: (1) 可视化 topNfeat 分别等于 1、2 时, PCA 的输出数据; (2) 通过与原始数据对比, 讨论 topNfeat 分别等于 1、2 时的降维效果;

2. 实验 4-2

利用 PCA, 对 secom.data 数据降维, (1) 讨论 topNfeat 取值对降维数据的影响; (2) 找到降维后恢复的数据与原始数据相对误差小于 9% 的 topNfeat

二、 实验代码

1. pca.py

pca.py

```
1 from numpy import *
2 '''
3 函数的map函数是用的python2标准, python3无法正常使用。
4 '''
5 def loadDataSet(fileName, delim='t'):
6     fr = open(fileName)
7     stringArr = [line.strip().split(delim) for line in fr.readlines()]
8     datArr=[]
9     for line in stringArr:
10         data=[]
11         for j in line:
12             data.append(float(j))
13         datArr.append(data)
14     return mat(datArr)
15
16
17 def pca(dataMat, topNfeat=9999999):
18     meanVals = mean(dataMat, axis=0)
19     meanRemoved = dataMat - meanVals #remove mean
20     covMat = cov(meanRemoved, rowvar=0)
21     eigVals,eigVects = linalg.eig(mat(covMat))
22     eigValInd = argsort(eigVals) #sort, sort goes smallest to largest
23     eigValInd = eigValInd[:-(topNfeat+1):-1] #cut off unwanted dimensions
24     redEigVects = eigVects[:,eigValInd] #reorganize eig vects largest to smallest
25     lowDDataMat = meanRemoved * redEigVects #transform data into new dimensions
26     reconMat = (lowDDataMat * redEigVects.T) + meanVals
27     return lowDDataMat, reconMat
28
29 def replaceNanWithMean():
30     datMat = loadDataSet('secom.data', '')
31     numFeat = shape(datMat)[1]
```

```
32 for i in range(numFeat):
33     meanVal = mean(datMat[nonzero(~isnan(datMat[:,i].A))[0],i]) #values that are not
        NaN (a number)
34     datMat[nonzero(isnan(datMat[:,i].A))[0],i] = meanVal #set NaN values to mean
35 return datMat
```

2. 实验 4-1

实验 4-1

```
1 from numpy import *
2 import pca
3 import matplotlib.pyplot as plt
4
5 dataMat = pca.loadDataSet('testSet.txt')
6 lowDMat, reconMat = pca.pca(dataMat, 2)
7 shape(lowDMat)
8 fig = plt.figure(1)
9 ax = fig.add_subplot(1,1,1)
10 ax.scatter(dataMat[:,0].tolist(), dataMat[:,1].tolist(), marker = '^', s = 90)
11 ax.scatter(reconMat[:,0].tolist(), reconMat[:,1].tolist(), marker = 'o', s = 50, c = 'red')
12 plt.savefig("4-1-topNfeat(2).png", dpi = 400)
13 plt.show()
```

3. 实验 4-2

实验 4-2

```
1 import numpy as np
2 import pca
3 import matplotlib.pyplot as plt
4
5 dataMat = pca.replaceNaNWithMean()
6 meanVals = np.mean(dataMat, axis = 0)
7 meanRemoved = dataMat - meanVals
8 covMat = np.cov(meanRemoved, rowvar = 0)
9 eigVals, eigVects = np.linalg.eig(np.mat(covMat))
10
11 # 计算降维后恢复的数据与原始数据相对误差小于9%的topNfeat
12 topNfeat = 0
13 Re_err = 0
14 for i in range(999):
15     lowDMat, reconMat = pca.pca(dataMat, i)
16     Err = np.linalg.norm(dataMat - reconMat) / np.linalg.norm(dataMat)
17     if Err < 0.09:
18         topNfeat = i
19         Re_err = Err
20         break
21
```

```
22 print('降维后恢复的数据与原始数据相对误差小于9%的topNfeat为:',topNfeat)
23 print('此时相对误差为: %0.5f%%' % (Re_err*100) )
```

三、 实验结果

1. 实验 4-1

topNfeat = 1:

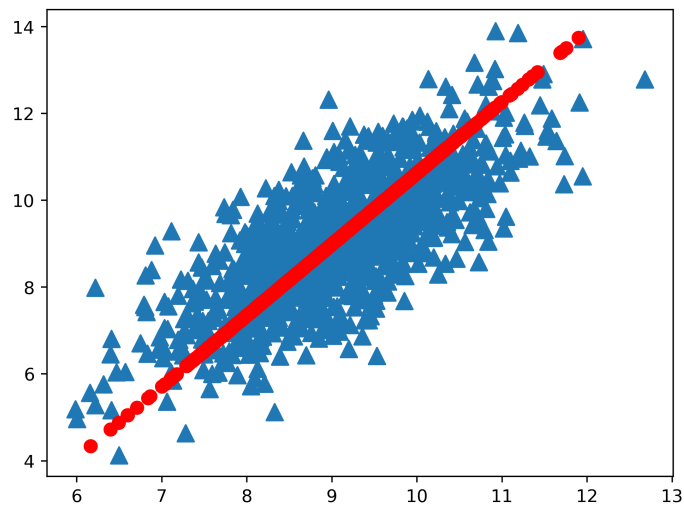
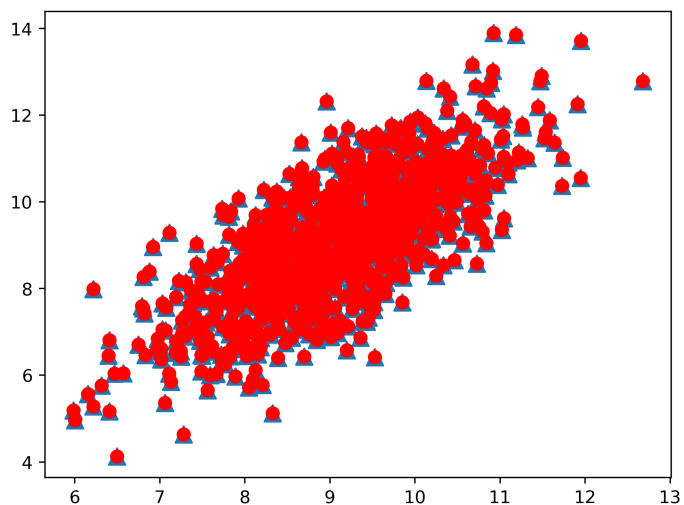


图 1: topNfeat = 1 可视化

topNfeat = 2:

图 2: $\text{topNfeat} = 2$ 可视化

从可视化图可以看出, $\text{topNfeat} = 1$ 时, 降维效果比较显著, 而 $\text{topNfeat} = 2$ 时, 数据和原始数据重合, 降维效果不好, 这是因为原始数据只有两个特征, 而 $\text{topNfeat} = 2$ 并没有剔除任何特征。

2. 实验 4-2

topNfeat 的取值影响了降维后数据与原始数据的误差, 当 topNfeat 越小时, 降维越大, 但是误差也随之增大。

经过实验找到降维后恢复的数据与原始数据相对误差小于 9% 的 topNfeat 为 8。