实验一 主成分分析的原理与实现

一、实验目的

1.掌握主成分分析算法原理

2.熟练运用主成分分析算法对多维数据进行降维

二、实验要求

1．独立完成实验

2．书写实验报告书

三、实验内容

1.在 NumPy 中实现 PCA

1.1 实验背景：

主成分分析(或称主分量分析，principal component analysis)由皮尔逊(Pearson,1901)首先引入，后来被霍特林(Hotelling,1933)发展了。

主成分分析是一种通过降维技术把多个变量化为少数几个主成分(即综合变量)的统计分析方法。这些主成分能够反映原始变量的绝大部分信息，它们通常表示为原始变量的某种线性组合。

主成分分析的一般目的是：a)变量的降维；b)主成分的解释。

1.2 实验步骤：

将数据转换成前 N 个主成分的伪码大致如下：

(1).去除平均值

(2).计算协方差矩阵

(3).计算协方差矩阵的特征值和特征向量

(4).将特征值从大到小排序

(5).保留最上面的 N 个特征向量

(6).将数据转换到上述 N 个特征向量构建的新空间中

建立一个名为 pca.py 的文件并将下列代补全用于实现 PCA 降维。

```
from numpy import *

def loadDataSet(fileName, delim='\t'):
    fr = open(fileName) # 打开文件
    stringArr = [line.strip().split(delim) for line in fr.readlines()]
    datArr = [list(map(float,line)) for line in stringArr] # 利用list构建字符矩阵和数据矩阵
    return mat(datArr)

def pca(dataMat, topNfeat=9999999): # 第一个参数是用于进行PCA操作的数据集，第二个参数topNfeat则是一个可选参数，即应用的N个特征。
    meanVals = mean(dataMat, axis=0)

    # 去平均值
    covMat = cov(meanRemoved, rowvar=0)
    eigVals,eigVects = linalg.eig(mat(covMat))
    eigValInd = argsort(eigVals)


    # 从小到大对N个值排序，去除不必要的维数

    # 将数据转化到新空间
    reconMat = (lowDDataMat * redEigVects.T) + meanVals
    return lowDDataMat, reconMat
```

代码包含了通常的 NumPy 导入和 loadDataSet()函数。对于 pca()函数的两个参数，如果不指定 topNfeat 的值，那么函数就会返回前 9999999 个特征，或者原始数据中全部的特征。

首先计算并减去原始数据集的平均值。然后，计算协方差矩阵及其特征值，接着利用 argsort()函数对特征值进行从小到大的排序。根据特征值排序结果的逆序就可以得到 topNfeat 个最大的特征向量。这些特征向量将构成后面对数据进行转换的矩阵，该矩阵则利用 N 个特征将原始数据转换到新空间中。

1.3 实验要求：

最终提交完成后的代码，并以 pca.py 命名，需成功构建 pca 算法并用于后续实验的具体应用中去。

1.4 实验总结：

本实验要求学生掌握 PCA 算法实现的具体流程，并能运用代码实现。

2.利用 PCA 对半导体制造数据降维

2.1 实验背景：

半导体是在一些极为先进的工厂中制造出来的。工厂或制造设备不仅需要花费上亿美元，而且还需要大量的工人。制造设备仅能在几年内保持其先进性，随后就必须更换了。单个集成电路的加工时间会超过一个月。在设备生命期有限，花费又极其巨大的情况下，制造过程中的每一秒钟都价值巨大。如果制造过程中存在瑕疵，我们就必须尽早发现，从而确保宝贵的时间不会花费在缺陷产品的生产上。

一些工程上的通用解决方案是通过早期测试和频繁测试来发现有缺陷的产品，但仍然有一些存在瑕疵的产品通过了测试。如果机器学习技术能够用于进一步减少错误，那么它就会为制造商节省大量的资金。

我们将考察面向上述任务中的数据集，具体地讲，它拥有 590 个特征。该数据包含很多的缺失值。这些缺失值是以 NaN（Not a Number 的缩写）标识的。在 590 个特征下，几乎所有样本都有 NaN，我们用平均值来代替缺失值，平均值根据那些非 NaN 得到。

2.2 实验步骤：

将下列代码添加到 pca.py 文件中。

```
def replaceNanWithMean():
    datMat = loadDataSet('secom.data', ' ')
    numFeat = shape(datMat)[1]
    for i in range(numFeat):
        meanVal = mean(datMat[nonzero(~isnan(datMat[:,i].A))[0],i]) # 计算所有非NaN的平均值
        datMat[nonzero(isnan(datMat[:,i].A))[0],i] = meanVal  # 将所有NaN置为平均值
    return datMat
```

上述代码首先打开了数据集并计算出了其特征的数目，然后再在所有的特征上进行循环。对于每个特征，首先计算出那些非 NaN 值的平均值。然后，将所有 NaN 替换为该平均值。

接下来在该数据集上应用 PCA。要求利用 PCA 算法将 590 维数据降至 20 维，并统计其方差贡献率。

2.3 实验要求：

最终提交的代码需实现利用 PCA 对其进行降维，保存至 pca_semi.py 并呈现其方差贡献率的排名情况，保存在 pca_var.py。

2.4 实验总结：

本实验是要求学生在学会 PCA 算法的原理后，能熟练运用在具体问题上，提升学生的动手能力。

备注：数据集见附件。