# 作业二：PCA

实验目的：将 UCI 手写数据集中的 32×32 的二值图像'3'的集合用二维表示。

实验过程：

当从 UCI 数据集中获取数据矩阵之后得到的是一个 199*1024 的矩阵，其中行代表样例，共 199 个样例，列代表特征，共 1024 个特征。

第一步分别求每一维特征的平均值，然后对于所有的样例，都减去对应的均值。

```
average = np.mean(dataMatrix,axis=0)
m, n = np.shape(dataMatrix)
data_adjust = []
avgs = np.tile(average, (m, 1))
data_adjust = dataMatrix - avgs
```

第二步求特征协方差矩阵，因为数据特征是 32*32 维的，所以协方差矩阵是 1024 维。

```
#计算协方差矩阵
covX = np.cov(data_adjust.T)
```

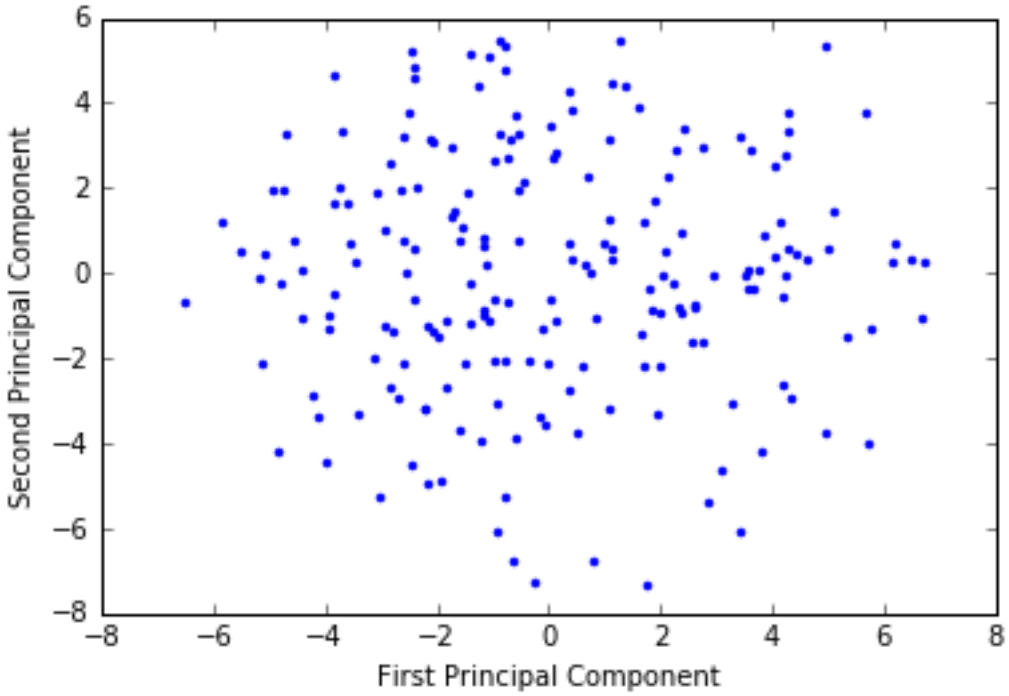
第三步，利用 numpy 库的 linalg.eig()方法求协方差的特征值和特征向量。

第四步，将特征值按照从大到小的顺序排序，选择其中最大的 k 个，然后将其对应的 k 个特征向量分别作为列向量组成特征向量矩阵。

第五步，将样本点投影到选取的特征向量上，在该实验中投影后的数据为 199*2 维，这样就将原始数据进行了有效降维。

```
#按照 featValue 进行从大到小排序
index = np.argsort(-featValue)
finalData = []
if k > n:
    print "k must lower than feature number"
    return
else:
   #注意特征向量时列向量，而 numpy 的二维矩阵(数组)a[m][n]中，a[1]表示第 1 行值，所以这里需要进行转置
    selectVec = np.matrix(featVec.T[index[:k]])
    finalData = data_adjust * selectVec.T
    reconData = (finalData * selectVec) + average
```

实验结果：




实验小结：PCA 是一种分析、简化数据集的技术，它的主要思想是将 n 维特征映射到 k 维上（k<n），这 k 维是全新的正交特征，经常用于减少数据集的维数，同时保持数据集中的对方差贡献最大的特征。其方法主要是通过对协方差矩阵进行特征分解，以得出数据的主成分（即特征向量）与它们的权值（即特征值）。