|  |
| --- |
| **Github账号：https://github.com/ZmA0/Coursera\_machine\_learing\_exercise** |
| **实验题目:K均值算法以及PCA算法在降低数据维度中的应用** |
| **实验摘要：**   1. 实验目的：   利用matlab或者octave，应用K-means算法来进行图片像素颜色的压缩，同时还要求应用PCA算法对面部识别中数据的降维处理和恢复。   1. 实验要求：   利用matlab或者octave的数学编程语言进行书写，按照代码文件中的要求编写并提交文件，使得程序运行通过并且得到足够的分数即为通过。   1. 代码文件要求：   需要编写的文件为  findClosestCentroids.m,  computeCentroids.m,  pca.m,  projectData.m  recoverData.m  需要填写但不会有相应分数的文件：  KMeansInitCentroids.m |
| **题目描述**   1. findClosestCentroids.m   本题instruction部分告诉我们这题是需要进行无监督学习中聚类中K均值算法的第二步的编写，也就是为每个训练实例寻找与之对应最近的聚类中心。在无监督学习中，实例是没有样本标签的，也就是我们需要为其划分类别定义样本的中心点。需要注意的是我们应该遍历每个样本，计算出与之对应的距离最近的聚类中心，其中聚类中心是我们提前第一步初始化后的值。这里需要我们进行两层循环，来得到每个样本对应的聚类中心，并将其索引值存入一列向量，便于后面使用。  2.computeCentroids.m：  这道题Instruction部分告诉我们承接上一题进行K均值算法的第三步，也就是进行聚类中心到已有集合点的平均值，即计算每个聚类中心到其所有包括的样本的点距离的平均值。这道题需要我们上一题中的存索引值的结果向量，不过这一次我们需要对每个聚类中心进行遍历，利用idx向量找出属于其的数据全体然后进行平均值的计算。这道题属于K均值算法中较为关键的一步，通过这里我们后续会将聚类中心改变到计算出的均值，然后重复之前的步骤才能完成算法结果的不断优化。  3.pca.m  这道题instructions告诉我们这一题计算样本的协方差矩阵，并且返回协方差矩阵的特征向量矩阵U和对角线为特征值的矩阵S。这一题是主要成分分析算法PCA的第一步，通过这一步我们有了包含n维方向向量的矩阵U，之后我们可以根据需要选取K维作为我们投射标准。代码编写较为简单，利用公式即可完成。   1. projectData.m   这道题需要我们进行数据的投射也就是数据压缩降维的过程。需要我们将上一题中U矩阵的前K列提取出来，这样一来代表我们需要压缩到K维。之后利用公式将我们的原来的X矩阵进行一个坐标变换，变换方法利用公式即可。之后这组样本的x就被化为K维的了，将变换后的数据存入结果向量Z中即可。部分代码示例已给。   1. recoverData.m   这道题与上一题过程相反，将已经投射过的数据恢复到K维，具体实现方法与上一题刚好相反，由于这里涉及的都是线性运算，是存在可逆性的，因此我们只要按照上一题的步骤反写出来即可。   1. KMeansInitCentroids.m   这一个文件我们从pdf中就可以直接看到代码，这个文件虽然不需要我们编写，但同样重要，这个文件的意义就是随机化生成聚类中心的值，这是K均值算法的第一步，为后面的实现提供了初始值。 |
| **实验过程**  1.findClosestCentroids.m  通过上面的描述这道题要我们进行两个遍历运算，外层遍历每个样本示例，内层遍历每个聚类中心到该样本的距离值并且要进行大小比较，选出最小值。这题的结构可以知道是两个for循环，内层一个if判断的样子。同时要设定有一个比较变量min，这是我们在用if判断时的比较标准，通过不断比较，将min值始终保持在以遍历过的聚类中心距离值的最小值，距离计算公式计算出距离来，通过与min进行比较来完成样本寻找中心的过程。这里需要注意的是还有一个索引向量idx，这个向量的维度是mx1维，它代表m个样本中每个样本所对应的聚类中心的索引值，因此我们在一旦if满足条件后，需要将当前内层循环的索引赋给它，而它本身的索引值是随外层循环而变的，这点编写时容易出错。综上，代码入下：  for i=1:length(X)  min=1e10;  for j=1:K  temp=sum((X(i,:)-centroids(j,:)).^2);  if temp<min  min=temp;  idx(i)=j;  end  end  end  2.computeCentroids.m  这道题根据上面的描述可以知道这道题要利用上面题目的结果同样进行两层遍历和一层逻辑上的if，这次需要对聚类中心进行外层的遍历，而样本点进行内层遍历，if的作用是编程逻辑上需要利用idx向量判断当前样本对应的索引值是否为当前外层遍历的值，若是则进行累加，最后当前样本中心的所有下属的样本都进行累加后，计算平均值。其中，累加是因为centroids和X向量的列数都相同可以直接相加，这样一来就可以直接对centroids进行平均值的运算，在将这个值赋给自己即可，算法实现起来比较简单。这道题大体结构与上题相同，不过在上一题的循环比较换成了计算平均值。注意的是计算平均值是要在内层完成一轮循环后进行。综上，代码如下：  for i=1:K  sum=0; |
| for j=1:m  if(idx(j)==i)  centroids(i,:)=centroids(i,:)+X(j,:);  sum=sum+1;  end  end  centroids(i,:)=centroids(i,:)/sum;  end  3.pca.m  这道题通过上面的描述可以知道我们需要进行协方差矩阵的运算以及方向向量矩阵U和特征值矩阵S的运算，这二者都可以直接利用公式得到。计算协方差矩阵的公式，在matlab中直接编写即可，之后计算两个矩阵的公式可以直接利用matlab的内置函数svd即可，这个函数的返回值前两个恰好就是我们这题需要的结果，我们直接编写即可。综上所述，代码如下：  Sigma=(X' \* X)./m;  [U,S,V]=svd(Sigma);   1. projectData.m   这道题通过我们上面的描述可以知道通过矩阵U来选取我们想要的K维向量，作为我们的投射基准，其实是相当于线性代数中的另一个坐标系中的基的概念，而整个PCA也是基于线性代数中的坐标变化，也就是从一个n维坐标系变换到K维坐标系，其中就需要K维坐标系下的一组基向量完成变换，而变换公式其实也是线代中的坐标变换的知识，因此这里结合一点编程知识就不难解决了。而部分代码示例已经在题目里给出，我们所做的就是对X的行进行遍历，将每组样本都与新的U矩阵进行上面的运算，就得到了新的K维的特征向量，再将其存入我们的结果矩阵中去，就可以完成题目。综上，代码如下：  U\_reduce=U(:,1:K);%n x K  for i=1:size(X,1)  x=X(i,:)';% n x 1  projection\_k=x' \* U\_reduce;%1 x n n x K  Z(i,:)=projection\_k;%1 x K  End   1. recoverData.m   这道题与上面是一个互逆的过程，因为都是利用线性代数的知识，运算 |
| 都是一些线性向量进行运算，从理论上说是存在这种对称性和互逆性的，因此这一题的编写步骤就是将上一题的代码反写。不过需要注意的是矩阵的维度问题，这里参与运算的U\_reduce矩阵需要进行一个转置运算才能得到正确结果，因为这里的维度都是K维，而U矩阵并没有变化，所以要进行一个转置操作。综上，代码如下：  for i=1:size(Z,1)  v=Z(i,:)'; % K X 1  recovered\_j=v' \* U(:,1:K)';% 1 x K K x n  X\_rec(i,:)=recovered\_j;%1 x n  end  **实验总结**  本次编程实验的文件分为两个部分，前一部分是应用K均值算法来对图片进行压缩。在运行ex7.m文件后可以看到通过不断的循环迭代，聚类中心的值越来越偏于某一个具体位置而不发生改变。之后程序会对一张图片进行压缩，原理就是从传统的RGB格式成千上万中颜色只提取16中颜色进行表示，而这16种颜色就是我们的聚类中心。通过这一方法，我们就可以将一个拥有多个样本的数据集只用很少一部分来代表大类从而达到了数据压缩的目的。编程过程中总体不难，只要理解了K均值算法的目的和过程就能完成。第二部分是PCA的应用，运行ex7\_pca.m后，我们可以看到程序先对一些数据进行投射向量的选取和均值误差的运算，之后程序还演示了对人脸识别的降维和恢复的过程。降维后我们可以看到人脸只有模糊的主体部分，但是恢复后可以看到恢复的人脸基本和原图一样。这里编程中比较难的地方就是线性代数的坐标变化的理解和基座标的理解。如果对这里的线代知识掌握熟练，编程就没有什么困难了。 |
| **参考文献**  **<https://wenku.baidu.com/view/f328b62b69dc5022abea0068.html>**  （参考了这个笔记） |