编程练习7：

*K*-均值聚类与主成分分析

机器学习

# 介绍

在本练习中，您将实现k-means聚类算法并将其应用于压缩图像。在第二部分中，您将使用主成分分析来寻找人脸图像的低维表示。在开始编程练习之前，我们强烈建议观看视频讲座并完成相关主题的复习问题。

要开始练习，您需要下载起始代码并将其内容解压缩到要完成练习的目录。如果需要，请在开始本练习之前使用octave/matlab中的cd命令更改到此目录。

您也可以在课程网站的“环境设置说明”中找到安装octave/matlab的说明。

## 本练习中包含的文件

ex7.m-octave/matlab脚本，用于k-means的第一个练习。

ex7\_pca.m-octave/matlab脚本，用于pca 的第二个练习。

ex7data1.mat -pca的数据集

ex7data2.ma -k-means的数据集

ex7 faces.mat脸部数据集

birdsmall.png-示例图像

displaydata.m-显示存储在矩阵绘图线中的二维数据。

drawLine.m-在现有的图形plotdatapoints上绘制一条线。plotdataPoints.m-初始化k-means的聚类中心

plotprogresskmeans.m-绘制k-means在运行时的每一步。

runkMeans.m-运行k-means算法

submit.m-提交脚本，将解决方案发送给oUR服务器

[]pca.m-执行主成分分析*？*

[]projectdata.m-将数据集投影到低维空间中*？*

[]recoverdata.m-从投影中恢复原始数据*？*

[]findclosestmentroids.m-查找最近的聚类中心（用于k-means）*？*

[]compute centroid.m-计算聚类中心的平均值（用于k-平均值）*？*

[]kmeansinitcentroids.m-k-聚类中心的初始化*？*

*？*指示需要完成的文件

在整个练习的第一部分中，您将使用脚本ex7.m，在第二部分中，您将使用ex7这些脚本为问题设置数据集，并调用将要编写的函数。您只需要按照本分配中的说明修改其他文件中的函数。

## 在哪里寻求帮助

本课程的练习使用Octaveor Matlab，这是一种高级编程语言，非常适合数值计算。如果您没有安装八度音阶或matlab，请参考课程网站“环境设置说明”中的安装说明。[〔1〕](" \l "_ftn1" \o ")

在octave/matlab命令行中，键入帮助，然后键入函数名，显示内置函数的文档。例如，帮助图将显示用于绘制的帮助信息。有关倍频程函数的更多文档可以在倍频程文档页面中找到。matlab文档可以在matlab文档页面中找到。

我们还强烈鼓励使用在线讨论与其他学生讨论练习。但是，不要查看其他人编写的任何源代码，也不要与其他人共享源代码。



# 1             *K*-表示聚类

在本练习中，您将实现k-means算法并将其用于图像压缩。您将首先从一个示例2d数据集开始，该数据集将帮助您获得k-means算法如何工作的直觉。之后，您将使用k-均值算法进行图像压缩，将图像中出现的颜色数量减少到该图像中最常见的颜色。这部分练习将使用ex7.m。

## 1.1           实施-手段*K*

K均值算法是一种将相似数据实例自动聚类的方法。具体地说，你得到了一个训练集x（1），…，x（m）（其中x（i）r），并希望将数据分组为几个有凝聚力的“集群”。k-means背后的直觉是一个迭代过程，它从猜测初始质心开始，然后通过重复地将示例分配给它们最近的质心，然后根据这些分配重新计算质心来改进这种猜测。*n*

k均值算法如下：

|  |
| --- |
| %初始化质心质心=kmeansinitcentroid（x，k）；对于iter=1：迭代  %群集分配步骤：将每个数据点分配给  %最近的质心。idx（i）对应于c\_（i），指数  %分配给示例i idx=findclosestcentroids的形心（x，形心）；  %移动质心步骤：基于质心的计算方法  %赋值中心=计算（x，idx，k）；  结束 |

算法的内环重复执行两个步骤：（i）将每个训练示例x（i）分配给其最近的质心；（ii）使用分配给它的点重新计算每个质心的平均值。对于质心，k-均值算法总是收敛到某个最终的均值集。请注意，收敛解可能并不总是理想的，它取决于形心的初始设置。因此，在实际应用中，K均值算法通常会在不同的随机初始化下运行几次。在不同随机初始化的不同解决方案之间进行选择的一种方法是选择具有最低成本函数值（畸变）的解决方案。

在下一节中，您将分别实现k-means算法的两个阶段。

### 1.1.1          寻找最近的质心

在k-means算法的“聚类分配”阶段，该算法在给定质心当前位置的情况下，将每个训练示例x（i）分配给其最近的质心。具体来说，对于我们设置的每个示例

*C*（i）：=j，使x（）i−\_j 2最小化，

其中c（i）是最接近x（i）的质心的指数，而μj是第j个质心的位置（值）。注意c（i）对应于起始代码中的idx（i）。

您的任务是完成findclosestcentroids.m中的代码。此函数获取数据矩阵x和所有质心在质心内的位置，并应输出一个一维数组idx，该数组包含每个训练示例的最近质心的索引（值在{1，…，k}，其中k是质心总数）。

你可以在每个训练例子和每个质心上使用一个循环来实现这一点。

完成findclosestcentroids.m中的代码后，脚本ex7.m将运行您的代码，您将看到与前3个示例的质心指定相对应的输出[1 3 2]。

*现在您应该提交解决方案。*

### 1.1.2          计算质心平均值

如果将每个点指定给质心，算法的第二阶段将为每个质心重新计算分配给它的点的平均值。具体来说，对于我们设置的每个质心k



其中，ck是分配给质心k的一组示例。具体来说，如果两个示例（例如x（3）和x（5）分配给质心k=2，则应更新

现在您应该在computecentroid.m中完成代码。您可以使用质心上的循环来实现这个函数。您也可以在示例上使用循环；但是如果您可以使用不使用此类循环的向量化实现，那么您的代码可能运行得更快。

在computecentroid.m中完成代码后，脚本ex7.m将运行代码并在kmeans的第一步后输出质心。

*现在您应该提交解决方案。*

## 1.2           *K*-示例数据集的含义

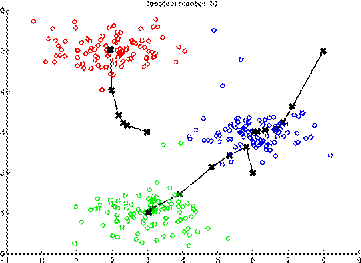


图1：预期输出。

完成这两个函数（findclosestcentroids和computecentroids）后，ex7.m中的下一步将在玩具2d数据集上运行k-means算法，以帮助您了解k-means是如何工作的。函数是从runkmeans.m脚本内部调用的。我们鼓励您查看函数以了解其工作原理。注意，代码调用了在循环中实现的两个函数。

当您运行下一步时，k-means代码将生成一个可视化，它将在每次迭代中引导您完成算法的进度。多次按Enter键查看k-means算法的每个步骤如何更改质心和簇分配。最后，您的图应该如图1所示。

## 1.3           随机初始化

ex7.m中示例数据集的质心的初始分配是这样设计的，这样您将看到与图1中相同的图。在实践中，初始化质心的一个好策略是从训练集中选择随机的例子。

在本部分练习中，应使用以下代码完成函数kmeansinitcentroid.m：

%将质心初始化为随机示例

%随机重新排序示例的索引randidx=randperm（大小（x，1））；%以前k个示例为质心，质心=x（randidx（1:k），：）；

上面的代码首先随机排列示例的索引（使用randperm）。然后，根据指数的随机排列选择前k个例子。这样就可以随机选择示例，而不会有两次选择同一示例的风险。

*您不需要为此部分练习提交任何资料。*

## 1.4           使用-平均值的图像压缩*K*



图2:128x128的原始图像。

            在本练习中，您将应用k-means进行图像压缩。在一个

简单的24位图像颜色表示，每个像素表示为三个8位无符号整数（范围从0到255），指定红色、绿色和蓝色强度值。这种编码通常被称为RGB编码。我们的图像包含数千种颜色，在这部分练习中，您将把颜色的数量减少到16种颜色。[〔2〕](" \l "_ftn2" \o ")

通过进行这种压缩，可以有效地表示（压缩）照片。具体地说，您只需要存储16种选定颜色的rgb值，对于图像中的每个像素，您现在只需要存储该位置的颜色索引（其中仅需要4位来表示16种可能性）。

在本练习中，您将使用k-means算法选择16种颜色，这些颜色将用于表示压缩图像。具体来说，您将把原始图像中的每一个像素作为一个数据示例，并使用k-means算法来查找16种颜色，它们最适合将像素分组（群集）到三维RGB空间中。计算完图像上的簇质心后，将使用16种颜色替换原始图像中的像素。

### 1.4.1          *K*-像素上的平均值

在Octave/Matlab中，可以按如下方式读取图像：

|  |
| --- |
| %加载128x128彩色图像（bird小PNG）  A=imread（&apos;鸟小PNG’；  %您需要安装映像包才能使用  %未读。如果未安装映像包，则  %应改为将以下行改为  %  %负载（鸟small.mat&apos;）；%将图像加载到变量a中 |

这将创建一个三维矩阵A，其前两个索引标识像素位置，最后一个索引表示红色、绿色或蓝色。例如，A（50，33，3）给出第50行和第33列像素的蓝色强度。

ex7.m中的代码首先加载图像，然后对其进行整形，以创建一个m×3像素颜色矩阵（其中m=16384=128×128），并对其调用k-means函数。

找到顶部k=16色来表示图像后，现在可以使用findclosestcentroids函数将每个像素位置指定给其最近的质心。这允许您使用每个像素的质心指定来表示原始图像。请注意，您已经显著减少了描述图像所需的位数。原始图像需要128×128像素位置中的每一个24位，因此总大小为128×128×24=393216位。新的表示需要以16色字典的形式进行一些开销存储，每个字典需要24位，但图像本身则只需要每个像素位置4位。因此，最终使用的位数是16×24+128×128×4=65920位，相当于将原始图像压缩约6倍。

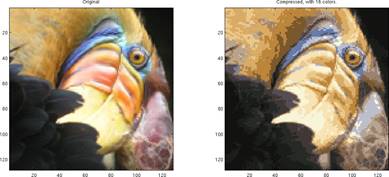
1 

图3：原始图像和重建图像（使用K-均值压缩图像时）。

最后，您可以通过仅基于质心指定重构图像来查看压缩的效果。具体地说，您可以用分配给每个像素的质心的平均值替换每个像素位置。图3显示了我们得到的重建。即使生成的图像保留了原始图像的大部分特征，我们也会看到一些压缩伪影。

*这部分练习不需要提交任何材料。*

## 1.5           可选（未分级）练习：使用自己的图像

在本练习中，修改我们提供的代码，以便在您自己的图像上运行。请注意，如果您的图像非常大，那么k-means可能需要很长时间才能运行。因此，我们建议您在运行代码之前将图像大小调整为可管理的大小。您还可以尝试改变k以查看对压缩的影响。



# 2             主成分分析

在本练习中，您将使用主成分分析（PCA）来执行维度缩减。您将首先使用一个示例2d数据集进行实验，以获得PCA如何工作的直观性，然后将其用于一个更大的5000张人脸图像数据集的数据集。

提供的脚本ex7 pca.m将帮助您完成练习的前半部分。

## 2.1           示例数据集

为了帮助您了解PCA的工作原理，您将首先从二维数据集开始，该数据集具有一个大变化方向和一个小变化方向。脚本ex7 pca.m将绘制训练数据（图4）。在本练习的这一部分中，您将可视化使用PCA将数据从二维减少到一维时发生的情况。例如，在实践中，您可能希望将数据从256维减少到50维；但是在本示例中使用低维数据可以使我们更好地可视化算法。

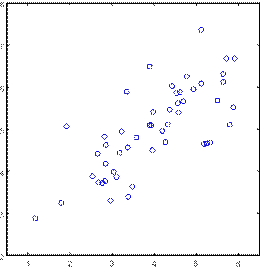


图4：示例数据集1

## 2.2           实施PCA

在这部分练习中，您将实现pca。主成分分析包括两个计算步骤：首先，计算数据的协方差矩阵。

然后，使用octave/matlab的svd函数计算特征向量u1，u2，…，un。这些将对应于数据变化的主要组成部分。

在使用PCA之前，首先通过从数据集中减去每个特征的平均值来规范化数据，并缩放每个维度以使它们在相同的范围内是很重要的。在提供的脚本ex7 pca.m中，已经使用featurenormalize函数为您执行了此规范化。

规范化数据后，可以运行PCA来计算主要组件。您的任务是完成pca.m中的代码，以计算数据集的主要组件。首先，您应该计算数据的协方差矩阵，它由以下公式给出：



其中x是以行为单位的数据矩阵，m是示例数。注意∑是一个n×n矩阵，而不是求和算子。

在计算协方差矩阵之后，可以对其运行svd来计算主成分。在Octave/Matlab中，可以使用以下命令运行SVD:[U，S，V]=SVD（sigma），其中U将包含主要组件，S将包含对角矩阵。

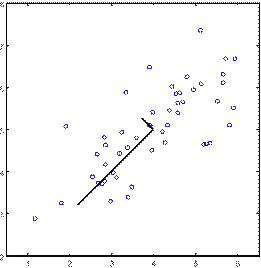


图5：数据集的计算特征向量

完成pca.m之后，ex7pca.m脚本将在示例数据集上运行pca，并绘制找到的相应主组件（图5）。该脚本还将输出找到的顶部主成分（特征向量），您应该会看到大约[-0.707-0.707]的输出。（由于U1和−U1对第一个主分量同样有效，因此可能是倍频程/matlab输出负的。）

*现在您应该提交解决方案。*

## 2.3           PCA降维

在计算了主要组件之后，您可以使用它们来减少数据集的特征维度，方法是将每个示例投影到较低的维度空间，x（i）→z（i）（例如，将数据从二维投影到一维）。在本部分练习中，您将使用PCA返回的特征向量，并将示例数据集投影到一维空间中。

在实践中，如果您使用的是学习算法，如线性回归或神经网络，那么现在可以使用投影数据而不是原始数据。通过使用投影数据，可以更快地训练模型，因为输入中的维度较少。

### 2.3.1          将数据投影到主要组件上

现在，您应该完成projectdata.m中的代码。具体来说，您将得到一个数据集x、主要组件u以及要减少到k的所需维度数。您应该将x中的每个示例投影到u中的k个顶部组件上。请注意，u中的k个顶部组件是由u的前k列给出的，即u reduce=u（：，1:k）。

完成projectdata.m中的代码后，ex7 pca.m将把第一个示例投影到第一个维度上，您应该看到大约1.481的值（如果您得到的是-u1而不是u1，则可能是-1.481）。

*现在您应该提交解决方案。*

### 2.3.2          重建数据的近似值

将数据投影到低维空间后，您可以通过将数据投影回原始高维空间来大致恢复数据。您的任务是完成recoverdata.m，将z中的每个示例投影回原始空间，并在x rec中返回恢复的近似值。

完成recoverdata.m中的代码后，ex7 pca.m将恢复第一个示例的近似值，您将看到一个约为[-1.047-1.047]的值。

*现在您应该提交解决方案。*

### 2.3.3          可视化投影

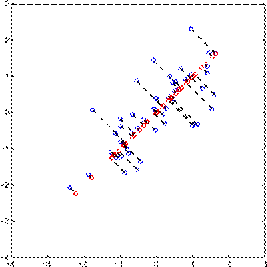


图6：PCA后的标准化和预测数据。

在完成项目数据和恢复数据之后，ex7 pca.m现在将同时执行投影和近似重建，以显示投影如何影响数据。在图6中，原始数据点用蓝色圆圈表示，而投影数据点用红色圆圈表示。投影有效地只保留U1给出的方向上的信息。

## 2.4           人脸图像数据集

在本部分练习中，您将在人脸图像上运行PCA，以了解如何在实践中使用PCA来减小尺寸。dataset ex7faces.mat包含人脸图像的datasetx，每个图像的灰度为32×32。x的每一行对应于一个面图像（长度为1024的行向量）。ex7的下一步[〔3〕](" \l "_ftn3" \o ") pca.m将加载并可视化这些人脸图像的前100个（图7）。



图7：人脸数据集

### 2.4.1          人脸主成分分析

要在人脸数据集上运行PCA，我们首先通过从数据矩阵X中减去每个特征的平均值来规范化数据集。脚本ex7 pca.m将为您执行此操作，然后运行PCA代码。运行PCA之后，您将获得数据集的主要组件。请注意，u（每行）中的每个主要组件都是长度为n的向量（对于面数据集，n=1024）。结果表明，我们可以通过将这些主要组件重新整形为与原始数据集中的像素对应的32×32矩阵来可视化它们。脚本ex7pca.m显示了描述最大变化的前36个主要成分（图8）。如果需要，还可以更改代码以显示更多的主要组件，以查看它们如何捕获越来越多的详细信息。

### 2.4.2          降维

既然已经计算了面数据集的主要组件，那么就可以使用它来减少面数据集的维度。这允许您使用输入尺寸较小（例如100维）的学习算法，而不是原来的1024维。这有助于加快您的学习算法。



图8：面数据集上的主要组件



图9：仅从前100个主要组件重建的面和面的原始图像。

ex7 pca.m中的下一部分将面数据集投影到前100个主要组件上。具体地说，每个人脸图像现在用一个向量z（i）∈r来描述。100

要了解在维度缩减中丢失了什么，可以仅使用投影数据集恢复数据。在ex7 pca.m中，执行数据的近似恢复，并并排显示原始和投影的人脸图像（图9）。从重建过程中，可以观察到人脸的总体结构和外观保持不变，而精细细节却丢失了。这是数据集大小的显著减少（超过10倍），有助于显著加快学习算法的速度。例如，如果你正在训练一个神经网络来执行人脸识别（在人脸图像中，预测人脸的识别度），你可以只使用100维的降维输入，而不是原始像素。

## 2.5           可选（未分级）练习：用于可视化的PCA

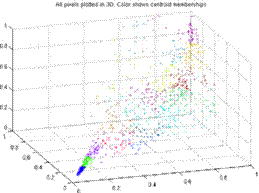


图10：三维原始数据

在早期的k-means图像压缩练习中，您在三维RGB空间中使用了k-means算法。在Ex7的最后一部分在pca.m脚本中，我们提供了使用scatter3函数在这个3D空间中可视化最终像素分配的代码。每个数据点都根据分配给它的集群进行着色。您可以在图形上拖动鼠标来旋转和检查三维数据。

事实证明，将数据集可视化为三维或更大的维度可能很麻烦。因此，即使以丢失某些信息为代价，也往往只希望以二维形式显示数据。在实践中，PCA经常被用来降低数据的维数，以便于可视化。在ex7的下一部分m，脚本将把你的pca实现应用到三维数据中，将其减少到二维，并在二维散点图中显示结果。PCA投影可以被认为是一个旋转，它选择的视图可以最大化数据的传播，这通常对应于“最佳”视图。

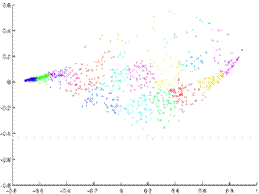


图11：使用PCA生成的二维可视化

# 提交和评分

完成任务的各个部分后，请务必使用提交功能系统将您的解决方案提交给我们的服务器。下面是这个练习的每个部分的得分情况。

|  |  |  |
| --- | --- | --- |
| **部分** | **提交的文件** | **点** |
| 查找最近的质心 | find关闭质心.m | 30点 |
| 计算质心平均值 | 计算机entroid.m | 30点 |
| 主成分分析 | 主成分分析 | 20点 |
| 项目数据 | 项目数据.m | 10点 |
| 恢复数据 | 恢复数据.m | 10点 |
| 总分 |  | 100点 |

您可以多次提交解决方案，我们只考虑最高分数。

[〔1〕](" \l "_ftnref1" \o ")八度是matlab的免费替代品。对于编程练习，您可以自由使用八度或matlab。

[〔2〕](" \l "_ftnref2" \o ")本练习中使用的提供的照片属于Frank Wouters，并经其允许使用。

[〔3〕](" \l "_ftnref3" \o ")此数据集基于野生数据集中已标记面的裁剪版本。