|  |
| --- |
| **Github账号：https://github.com/ZmA0/Coursera\_machine\_learing\_exercise** |
| **实验题目:异常检测算法以及协同过滤算法的应用** |
| **实验摘要：**   1. 实验目的：   利用matlab或者octave，应用异常检测算法来进行数据中异常数据的甄别以及利用协同过滤算法进行一个简易电影推荐系统的编写。   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  estimateGuassian.m  (为高斯分布模型估计合适的参数)  selectThreshold.m  (为异常检测算法选择合适的阈值)  cofiCostFunc.m  (计算协同过滤算法的代价函数) |
| **题目描述**  1.estimateGuassian.m  本题instruction部分告诉我们需要计算数据集的均值和方差。这道题是关于异常检测中高斯分布模型参数的计算。其中，传入数据矩阵X，是一个mxn维矩阵，每n维的数据都存储在一行，我们需要返回两个向量，mu和sigma2,均为nx1的列向量。这二者就代表我们后续进行高斯分布模型的均值和方差向量。需要注意的是我们的均值和方差是对于每个特征而言的，也就是对一个特征的m个样本进行运算。  2.selectThreshold.m：  这道题Instruction部分告诉我们为异常检测算法选择合适的阈值。我们知道异常检测的大致过程就是利用一个阈值，在样本成高斯分布的时候利用这个阈值去甄别异常数据。同时题目还告诉我们选择阈值的根据是利用F1-score（和准确率和召回率）来进行判断，所以这题的过程就是对于每个可能的参数阈值epsilon，都进行一遍这个阈值所对应的预测结果和实际样例之间所得的F1-score，来标识这个阈值的结果如何，我们最终要选取最大的那个F1作为结果，F1越大代表阈值的拟合效果越好。  3.cofiCostFunc.m  这道题是要进行协同过滤算法的代价函数的编写，之前的许多练习我们都知道如果要进行代价函数编写的话，无非是一个J，一个J的梯度，之后或许会有正则化的操作。我们这道题也是一样的，不过是添加了一个标识矩阵R。题目的instruction部分告诉我们，这是关于电影推荐系统的代价函数，R矩阵的内容是关于用户是否对某电影评分，如果评分则为1，否则为0。也就是说我们这道题中并不是所有的数据都能在编写有实际的用途，因为我们只对用户参与了评分的样例才能进行计算。在计算J之后，我们还要计算出J关于X和theta的导数值。这道题的最后有一部是要我们再对J和两个导数值进行正则化的操作。大体过程我们已经非常熟悉，不同与以往的一个是R矩阵的引入以及J关于X矩阵的导数，在以往我们的代价函数是不会对X进行梯度运算的。 |
| **实验过程**  1.estimateGuassian.m  通过上面的描述这道题只需要我们计算两个列向量，计算方法我们都熟悉，就是普通的均值和方差的计算，根据公式和方差公式直接进行计算即可。但我们需要注意的就是这一题的遍历是进行列遍历，也就是我们的操作对象是列向量，之后要将每一列计算的值存入相应位置的结果向量中去。所以综上，这题的结构就是一个循环遍历，计算出均值之后利用公式再计算出方差，存入向量结束。因此源代码为  for i=1:n  mu(i)=sum(X(:,i))/m;  sigma2(i)=sum((X(:,i)-mu(i)).^2)/m;  end  2.selectThreshold.m  这道题根据上面的描述可以知道整道题的过程就是循环，计算F1，比较大小，最终选择最优解。其中观察题目源代码可知，题目已经将循环和比较的部分写好了，也就是这道题仅需要我们计算准确率召回率和F1-score即可。其中准确率的计算公式我们知道是true positive/(true positive + false positive),召回率的计算公式是true positive/(true positive + false negative),之后利用这二者计算F1-score=2\*precision\*recall/(precision + recall)。对于这道题的编写来说我们需要做的工作就是先将数据集进行一个筛选，得到我们的预测结果向量，之后利用上面的计算公式将这三者计算出来即可。其中具体到matlab的知识可以知道与操作&（一个&）才能对矩阵进行运算，这样一来可以大大简化代码的编写。之后将与的结果用sum函数相加，就可以得到相应的tp，fp等。综上，源代码如下：  predictions = (pval<epsilon);  tp=sum((predictions == 1)&(yval==1));  fp=sum((predictions == 1)&(yval==0));  fn=sum((predictions == 0)&(yval==1)); |
| prec=tp/(tp+fp);  rec=tp/(tp+fn);  F1=2\*prec\*rec/(prec+rec);     1. cofiCostFunc.m   这道题通过我们上面的描述可以知道这道题总共分三个部分，基本代价函数的编写，两个梯度值的运算，三个正则化操作。首先，对于代价函数的编写我们可以利用公式  即可得到，但是我们需要注意的是要引入R矩阵，也就是说我们的h函数需要与R矩阵进行一个运算，来将没有用途的那些数据置零，这样才符合题意。其中根据题目描述矩阵的维度，将h直接点乘R即可满足题意，最后再将这个矩阵平方后展开累加就是我们的代价函数。之后计算梯度值，同样的利用公式进行计算即可。只不过同样要注意的是公式中的前一项也是经过R矩阵处理过的，并不是我们的原始数据直接进行的操作。最后我们要进行这三者的正则化。对J来说正则化的项是theta和X的累加，而不是我们之前学的只对theta的累加。对两个梯度值来说，正则项是对自己的一个惩罚。因此直接按照公式进行编写即可。综上可得，源码如下：  h=X\*Theta'-Y;  H=(h.\*R).^2;  J=1/2 \* sum(H(:)); |
| X\_grad=(h.\*R) \* Theta;  Theta\_grad=(h.\*R)' \* X;  reg\_J=lambda/2 \* sum(Theta(:).^2)+lambda/2 \* sum(X(:).^2);  J=J+reg\_J;  X\_grad=X\_grad+lambda \* X;  Theta\_grad=Theta\_grad+lambda \* Theta;  **实验总结**  本次编程实验的文件分为两个部分，前一部分是应用异常检测算法，运行ex8.m可以看见利用我们前面编写的两个文件，分别得到合适的高斯分布模型和筛选阈值，题目文件为我们画出了数据成高斯分布的一个等值线的图，之后通过不断进行F1的比较选出最合适的阈值，进行异常数据的检测，图上之后明显标出了数据低于这个阈值的，大体上完成了对异常检测算法的应用。第二部分是电影推荐系统的简单编写，运行ex8\_cofi.m可以看到程序在不断进行代价函数的迭代运算，还给出了我们用户对电影的评分图，之后给了推荐的电影名单。这是最后一次编程作业，代码中并没有涉及到新的知识点，而是对前面的东西进行了改编和扩充，在编写过程第三个文件的时候对R矩阵如何操作遇到困难，后来也是看了pdf之后才明白这个矩阵的运算意义。 |
| **参考文献**  **<https://wenku.baidu.com/view/f328b62b69dc5022abea0068.html>**  （参考了这个笔记）  <http://www.ilovematlab.cn/thread-56379-1-1.html>  （matlab的&和&&的区别） |