|  |
| --- |
| **Github账号：https://github.com/ZmA0/Coursera\_machine\_learing\_exercise** |
| **实验题目:支持向量机算法的编写及其关于邮件过滤器的简单应用** |
| **实验摘要：**   1. 实验目的：   利用matlab或者octave，首先对SVM算法中边界划分，高斯核函数等进行编写，对题目文件中给予的数据进行划分；之后利用支持向量机算法（SVM）来进行邮件过滤器的简单编写，实现对邮件内容的简单过滤。   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  gaussianKernel.m,  dataset3Params.m,  processEmail.m,  emailFeatures.m |
| **题目描述**   1. gaussianKernel.m   本题instruction部分告诉我们这里需要进行高斯核函数的编写，其中函数的参数是关于x1，x2两个列向量，和常数sigma；返回值为函数的计算结果（也应是以列向量）。这个文件是SVM算法中初步的一个算法，利用带有高斯核函数的SVM可以对较复杂的数据集进行边界划分。   1. dataset3Params.m：   这道题Instruction部分告诉我们进行SVM中代价函数和核函数的C和sigma的最优选择的函数的编写，同时还要计算对应参数下的交叉验证集中的错误代价即Error值的计算。这个文件需要我们先进行C和sigma的选择，也就是需要我们自己先声明一个待选数据集，之后通过已有函数得到模型的计算结果，用得到的模型和交叉验证集计算出预测结果，再得到Error值。其中我们可以利用svmTrain和svmPredict来进行模型的选择和预测结果的计算。  3.processEmail.m  这道题instructions部分很多，但是大体来讲需要我们进行邮件内容词汇的标记。意思也就是我们已有一个单词集，即相当于敏感词汇，之后我们有经过初步整理的邮件内容，我们需要对内容和单词集进行一个遍历，如果在内容中出现了单词集的词汇，我们要将出现的词汇在单词集中的索引存入一个行向量中，代表这个单词出现过。这样一来，我们就对邮件内容进行了一个初步的筛选和标记。   1. emailFeatures.m   这道题承接上一题，上一题中利用一个行向量来进行词汇的标记，标记内容是对应词汇的索引值，这题中也是要进行词汇的标记，不过这次的标记方式是二元标记，即出现标1，没出现标0。这样一来，我们所需要的结果数组就是和单词集的等长数组，利用上一题的结果，来进行这一题的标记。 |
| **实验过程**   1. gaussianKernel.m   通过上面的描述这道题就是要根据高斯核函数的公式进行编写即可，只不过这里的地标L换成了x2，不过道理都是一样，而且这里的x1，x2题目题头已经全部展成了列向量，我们直接按照公式编写即可，无需进行形式的变化。利用点乘即可对每个元素操作。综上，代码如下：  q=sum((x1-x2).^2);  sim=exp(-q/(2\*sigma^2));   1. Dataset3Params.m   这道题根据上面的描述可以知道这道题需要有三个步骤来进行。第一步，利用已有函数来进行模型的选择，之后利用该模型计算交叉验证集的预测结果；第二步根据计算出的预测结果来进行Error值的计算，也就是错误误差；第三步选择不同的参数C和sigma重复前两步得到最小的Error值，此时的C和sigma就是我们需要的最优值。  首先，对于第一步的编写我们直接调用之前提过的两个函数即可，函数参数的书写以及返回值在各自的文件中都有，按照方式来即可。注意的是在利用svmPredict函数时要满足题意对Xval进行运算。其次Error值的计算在题目note部分中也有，也就是用了 ~= 判断是否等于以及mean函数，这样一来，一步就可以得到错误误差的值。然后就是对Error值的比较来选择不同的C和sigma，这里需要我们对两个参数有一个待选集，同时还要讲Error，C，sigma初值都设为较大的值，这样一来在编程实现时，才能保证第一次循环是正确的。  具体实现方法我们用两层for和最内层的if来构造，保证对每对参数都有判断验证过程。综上，代码如下：（待选集iteration中的数据由ex6中提供）  iteration=[0.01,0.03,0.1,0.3,1,3,10,30];  Cmin=1e10;  sigma\_min=1e10; |
| error\_min=1e10;  len=length(iteration);  for i = 1:len  for j = 1:len  C1=iteration(i);  sigma1=iteration(j);  model=svmTrain(X,y,C1,@(x1,x2) gaussianKernel(x1,x2,sigma1));  predictions=svmPredict(model,Xval);  error\_val=mean(double(predictions ~= yval));  if(error\_val<error\_min)  Cmin=C1;  sigma\_min=sigma1;  error\_min=error\_val;  end  end  end  C=Cmin;  sigma=sigma\_min;   1. processEmail.m   这道题通过上面的描述可以知道我们需要进行一个遍历操作，在这样一个for循环中标记出出现的单词，并存入数组。其实题目已经给了我们很多的提示，我们可以知道利用strcmp就可以满足我们的需要。这里对结果数组的扩展也给了我们格式。综上代码如下：  len=length(vocabList);  for i=1:len  if(strcmp(vocabList(i),str))  word\_indices=[word\_indices,i];  end  end   1. emailFeatures.m   这道题比上面的题还要简单一点，由于已经有了标记的结果，我们这一题需要进行二元标记对上一题的结果数组进行操作。对这一题的结果数组的长度进行遍历，将上一题结果数组的内容作为这个结果数组的索引，并且将值置1即可。其余未标记的置0即可。综上，代码入下： |
| len=length(word\_indices);  for i=1:len  x(word\_indices(i))=1;  end  **实验总结**  第六次编程所有文件编写后运行ex6.m,题目开始给出一些数据，利用我们编写的带有高斯核函数的SVM对其进行边界划分并且给我们将图形表示出来了。而且在给出一个复杂边界的数据时，我们通过图形可以发现我们的SVM仍在很好的工作，并且将边界也划分出来了。之后运行ex6\_spam.m，这是关于邮件过滤器的编写。运行之后我们发现它首先给出了具有标记索引的结果数组，之后利用数字化后的特征应用于SVM上来判断是不是一封垃圾邮件。这次实验确实体会到利用SVM的方便，本次在编写过程中没有对于思维要求很高的题目，不过第二题中确实要求结果很多，需要一步步理清之间的先后关系。总体而言本次试验的应用性和趣味性比技术性要大一点。 |
| **参考文献**  **<https://wenku.baidu.com/view/f328b62b69dc5022abea0068.html>**  （参考了这个笔记） |