《机器学习》读书笔记

1. 算法概述（算法名称及原理）

EM算法

EM算法是一种迭代优化策略，通常被用来计算参数的极大似然估计或在Bayes统计框架下计算参数后验分布的众数。

为什么使用EM算法？

概率模型有时既包含观测变量，也包含隐变量(或潜在变量)

如果模型只包含观测变量，则我们可以通过最大化对数似然 l(θ) 来得到 θ



如果模型包含隐变量，则(1)式变化为：



最后一个等式的成立利用到了边缘分布的概率密度函数公式做了一个转换，从而将隐含变量Zi显示出来，它的含义为P(xi)的边缘概率密度就是联合分布p(xi,zi)中的zi取遍所有可能取值后，联合分布的概率密度之和

该式的极大似然概率是隐变量和观测变量联合概率求和的对数，很难求解，下面是EM的解决方案，在EM算法中，zi是标准类别归属的变量，例如在身高的例子当中，它有两个可能的取值，男生和女生

EM不是直接求解，而是使用一个迭代过程来不断的逼近 l(θ)的局部最大值(EM算法的解和初值选择是有关系的)，从而得到θ的近似解。

概率论相关知识：

知识点1

对于P(Y,Z|θ)，如果观测变量Y和隐变量Z已知，即只有一个未知变量θ，那么我们可以通过极大似然概率的最大化来得到θ的解。

知识点2

对于P(Z|Y,θ)，如果观测变量Y和模型参数θ已知，那么这个概率值也是可以计算出来的，即我们会得到p(z1),p(z2),...,p(zn),(zi∈Z)。

下面是EM算法迭代求解 θ的过程：

1、选择模型参数的初值θ(0)，开始迭代。

2、E步：

记θ(i)为第i次迭代参数θ的估计值，则第i+1次迭代的E步(求解期望值)，计算：



上式是在已知观测变量Y和模型的估计参数θ(i)条件下求解log P(Y,Z|θ)的期望，根据知识点2可以得到隐变量Z的概率分布，所以(3)式可以进一步转化为：



(4)式的含义：首先在第i次求得观测变量Y和模型参数θ(i)的条件下，得到了隐变量Z的概率分布，现在Z可以认为是已知的了，这个表达式现在只有一个未知变量θ，然后便可以根据知识点1来得到θ的值

3、M步：

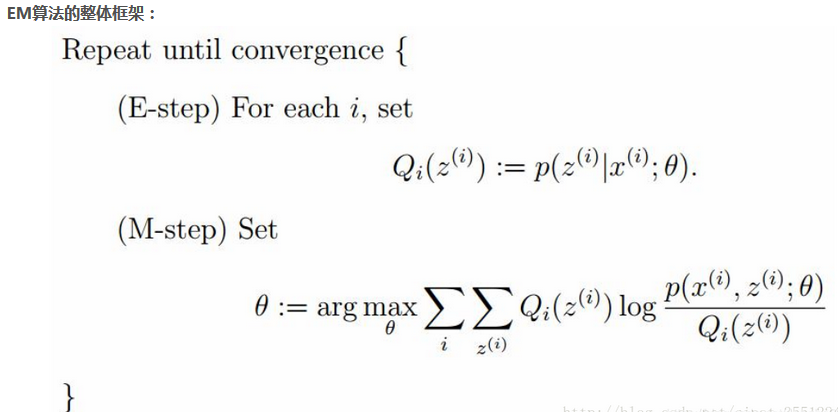
在算法第2步中，我们可以使用事实1来得到θ的值，这个解就是第i+1次的模型参数θ(i+1)的估计值：



即利用第i次迭代的模型参数估计值得到了第i+1次模型参数的估计值。

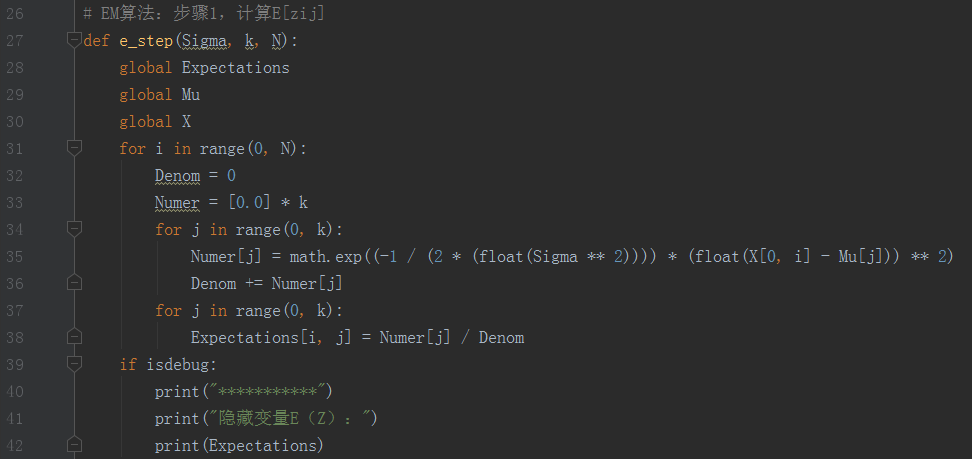
4、接下来就是不断重复算法的第2步和第3步，直到θ的估计值收敛。

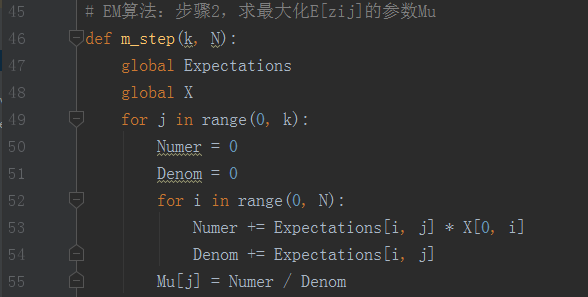
1. 算法设计（流程图及主要分段代码，附详细代码注释）

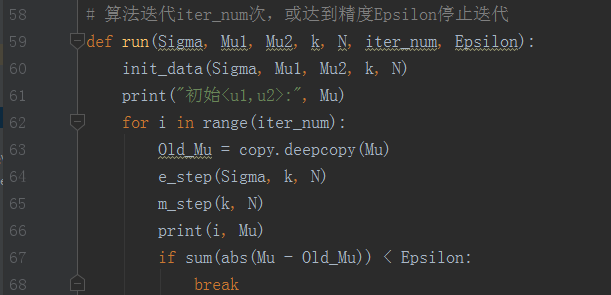


其中是样本i关于类别z的概率分布

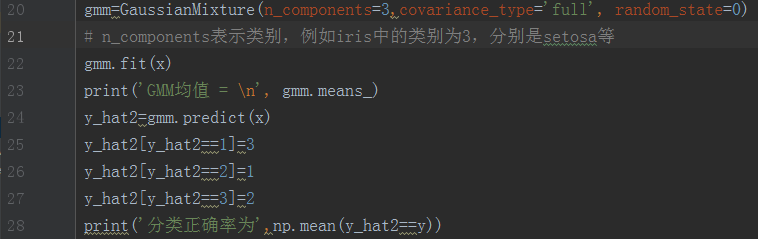
EM算法的代码







基于EM算法的GMM算法的实现（调包实现）



1. 选用数据（数据集描述，包括来源，行数，列数，格式等）

数据集一：

iris鸢尾花，行数：150，列数：4

列属性：

1. sepal length in cm
2. sepal width in cm
3. petal length in cm
4. petal width in cm

类别：  
Iris Setosa ，Iris Versicolour ，Iris Virginica

数据集二：

car，行数：1728，列数：6

列属性及取值：  
1) buying: vhigh, high, med, low.   
2) maint: vhigh, high, med, low.   
3) doors: 2, 3, 4, 5more.   
4) persons: 2, 4, more.   
5) lug\_boot: small, med, big.   
6) safety: low, med, high.

类别：

unacc, acc, good, vgood

数据集三：

wine，行数：178，列数：13

属性：

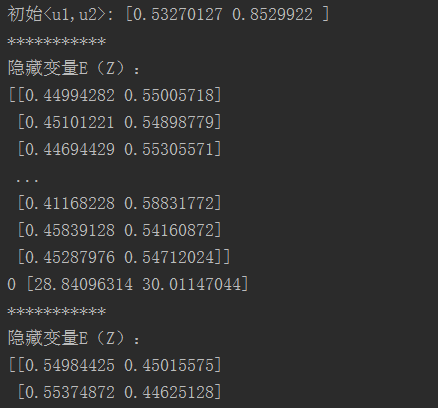
1. Alcohol
2. Malic acid
3. Ash
4. Alcalinity of ash
5. Magnesium
6. Total phenols
7. Flavanoids
8. Nonflavanoid phenols
9. Proanthocyanins
10. Color intensity
11. Hue
12. OD280/OD315 of diluted wines
13. Proline

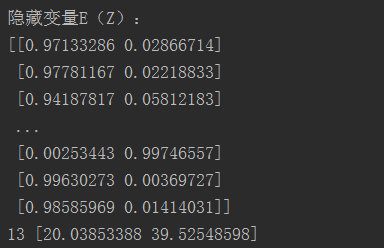
类别：

Alcohol 1，2，3

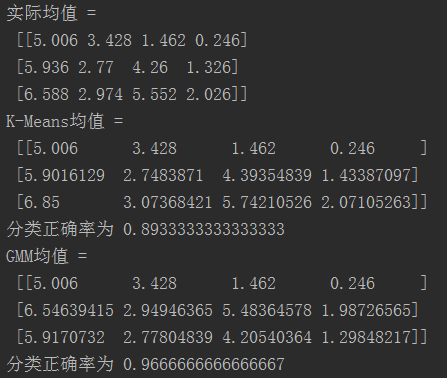
1. 评价方法（说明训练集和测试集分配方法及评价指标）
2. 实验结果截图

EM算法实验的迭代过程及结果截图

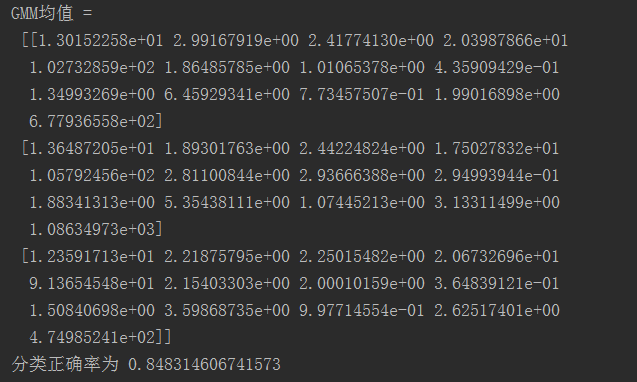




GMM分类的结果（iris）



GMM分类结果（wine）



1. 实验结果分析及比较

本周所学习的EM算法是一种优化算法，我们可以基于EM算法并结合其他算法（如GMM、K-Means）进行聚类

之前学习的算法都是监督学习算法，而GMM算法是一种聚类算法，它属于无监督学习算法

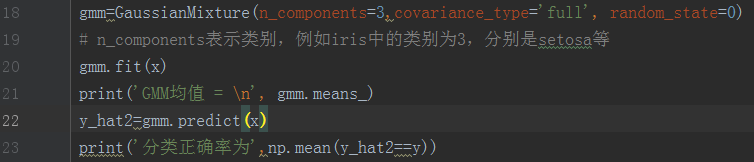
1. 遇到的问题及解决方法，实践心得

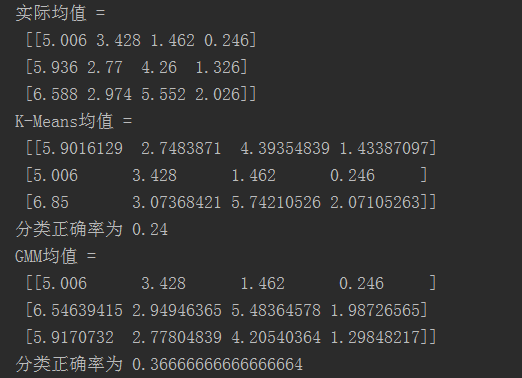
问题一：关于准确率

在iris实验中，K-Means给出的第一行似乎和实际的第二行很接近，第二行和实际的第一行很接近。同样，GMM给出的均值矩阵也有同样的问题，第二行和第三行似乎对调了。上网百度发现这不是算法的问题，算法只起到聚类的作用，并不保证标签一样，三个类别六种标签算法只能随机一种

解决办法：把预测的结果的标签改一下

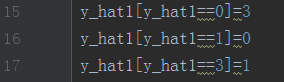
原始代码



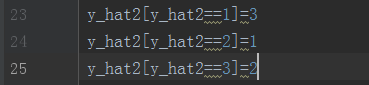


手动改变类别标签

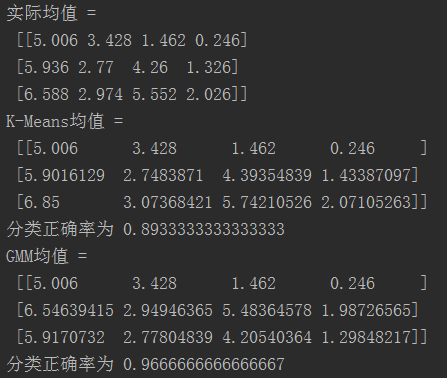
K-means



GMM

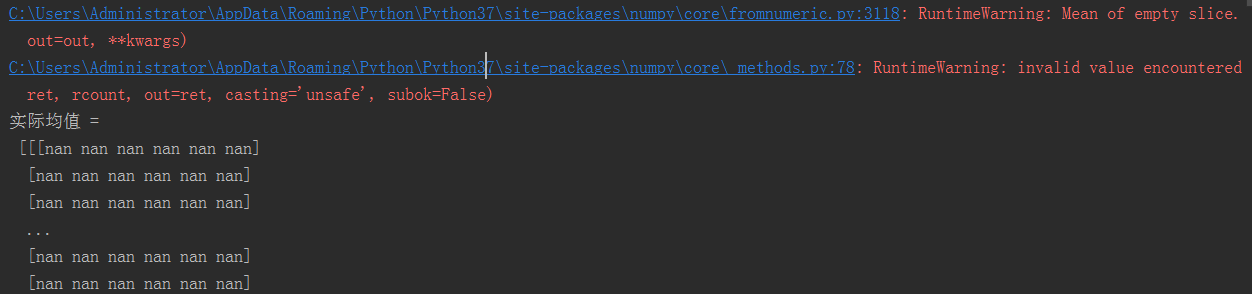


结果的正确率很明显增加了



问题二：代码问题

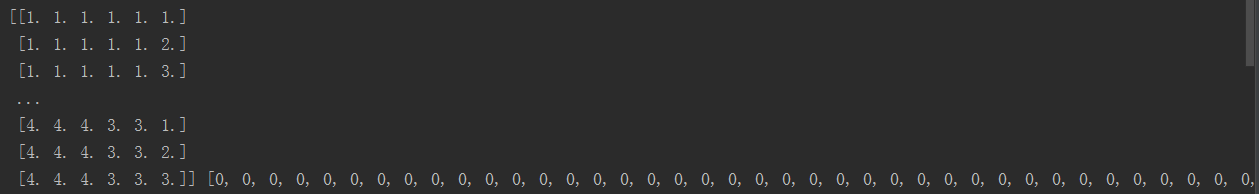
在引入car数据集的时候出现了下列问题



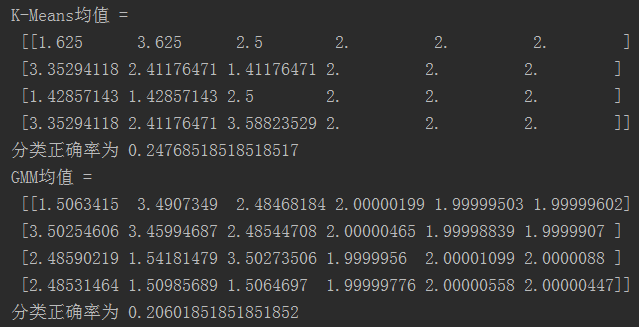
由于实际均值是调包实现



所以不太了解问题出在哪里,我觉得可能是数据集的呈现方式问题，数据集输出如下：



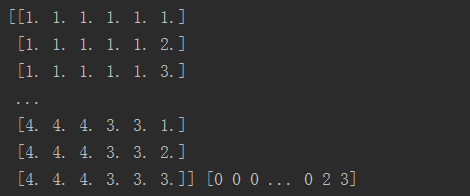
不知道实际均值就没有办法调整标签的顺序，实验的结果正确率非常低

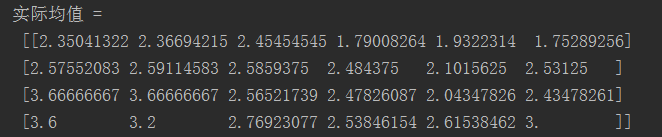


解决办法：

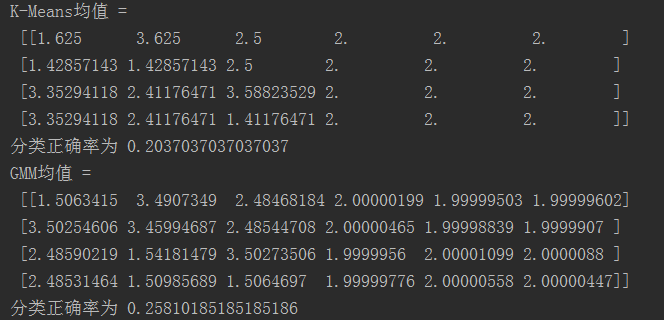
修改输出的数据集的格式







再手动修改标签，得到正确率如下



发现相比于iris和wine，car的正确率非常小，所以可能是该数据集不适用于聚类算法

会议总结：

EM算法是一种优化算法，主要用于参数估计确定参数（比如前面的knn中的k，贝叶斯高斯模型中的k等）

EM算法是一种自收敛的算法（具体证明利用到了联合概率密度和边缘概率密度的关系公式+杰森不等式+单调有界准则），它能够很好的起到聚类的效果，算法比较高效稳定

下次实验之前建议用图的形式将数据集的分布展示出来，在充分了解算法原理之前可以加深对于算法的理解（比如说这个算法用于解决什么类型的数据分布问题），并且在了解算法的原理之后可以对不同的数据集进行预处理从而得到一个更加客观的实验结果，最后可以用图的形式展示实验结果

对于数据集的预处理问题，目前接触到的方法有归一化处理，一般连续性数据在实验之前要进行归一化处理

当属性很多时，考虑降维方法，比如决策树算法（基于信息熵的属性选择），pca成分分析法等

十次十折交叉验证是评估分类算法的一种度量，不适用于聚类算法，聚类评价没有统一规范

关于代码问题

在鸢尾花实验中，由于刚开始没有更换类别的顺序导致实验正确率很低，解决办法是手动修改类别，下面用代码解决类别不匹配问题

from sklearn.metrics.pairwise import pairwise\_distances\_argmin

order = pairwise\_distances\_argmin(m, gmm.means\_, axis=1, metric='euclidean')

（详细代码见week6-EM1）

在画图的时候若是不能够将多维的图形展示出来的话可以将属性进行两两分类组合进行实验，可以参考现在的论文，里面有多维属性的展示