assignment 3

张海斌*

2020 年 6 月

目 录

1 实现

在代码中，我实现了多个函数：计算聚类结果的准确度 `getClusterAccuracy` ，用于生成二维正态分布数据的 `getNorm` ，生成随机协方差矩阵的 `getRandCovs` ，用于计算多维正态分布概率密度及其对数值的 `norm` 和 `log_norm` 。协方差矩阵是通过先随机生成标准差和相关系数，然后计算出协方差矩阵得到。输入数据使用 sklearn 的 iris 鸢尾花数据集，同时也可以用多个正态分布构成的数据集，这个自定义的数据集可通过调用 `getDataset` 得到。

高斯混合模型 `GMM` 中有一个 `fit` 函数用来对数据进行多次拟合获取最优解。而 `fit_once` 函数是对数据进行一次拟合。有些时候计算数据可能偏离比较厉害导致 `np.exp` 指数计算结果为无穷大（溢出），对于这种情况我会检查无效的计算结果并且直接退出拟合，并进行下一轮的拟合。在 `fit_once` 函数中，开始会随机初始化一些参数，如不同正态分布的模型权重、各个正态分布的均值和协方差矩阵，之后就开始 EM 算法。EM 算法中，每一轮 EM 迭代根据课程 PPT 中的公式在 E 步固定权重 `weights` ，计算 γ `gamma` ，在 M 步根据 γ 计算更新权重 `weight` ，各正态分布的均值 `means` 和协方差矩阵 `covs_list` 。一轮 EM 迭代的最后，计算似然估计值 `p` ，并与上一轮计算的似然估计值比较，如果基本相同，则认为结果已经收敛，可以结束 EM 算法。

每一轮 `fit_once` 结束后，在 `fit` 中都会比较似然估计值，并选取似然估计值最大的一轮拟合结果作为模型的拟合结果保存到模型中。并且用该轮的结果计算得到数据的聚类分析结果保存在模型的 `pred` 变量中。

通过 `run` 函数可以绘制原始数据集，并对模型进行训练，得到并绘制出结果。这个函数有几个参数， `normalize` 可以控制是否对数据进行预处理使数据标准化， `n` 可以指定在模型中拟合的次数。

最后对结果计算了精度。

*学号 17307130118

2 结果

通过 `python source.py` 就可以直接运行代码。

iris 数据集见Figure 1，代码结果见Figure 2和Figure 3前者是写代码过程中得到的一个较好的结果，后者是最终代码跑出的结果。最终精度大约为 0.9733。

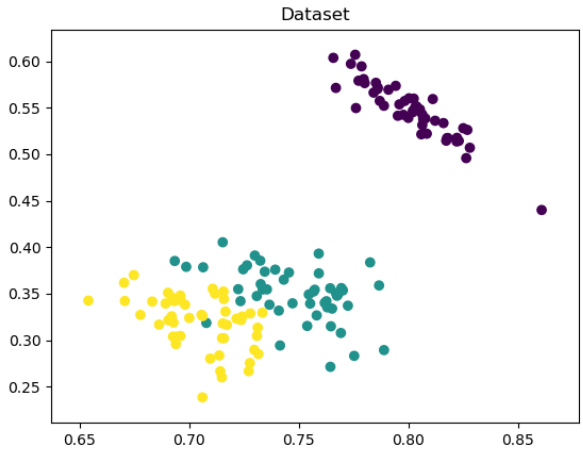

Figure 1: iris 数据集

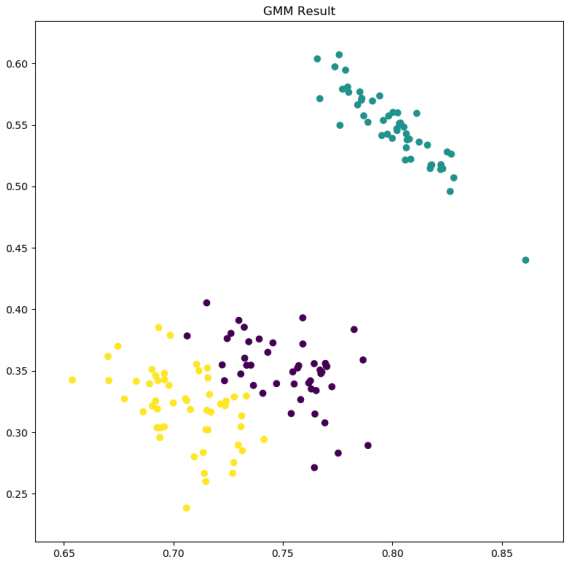

Figure 2: iris 聚类结果 1

自定义正态分布数据集见Figure 4，聚类结果见Figure 5。结果也很不错，精度有 0.963。

3 感想

开始的时候我没有对 `norm` 和 `log_norm` 的结果确定其有效性，即使得到 `INF` 的值仍继续执行下去，然后程序执行中就会报大量的错误，如除以 0，或者求 0 的对数等等常见的数学上的异常计算。

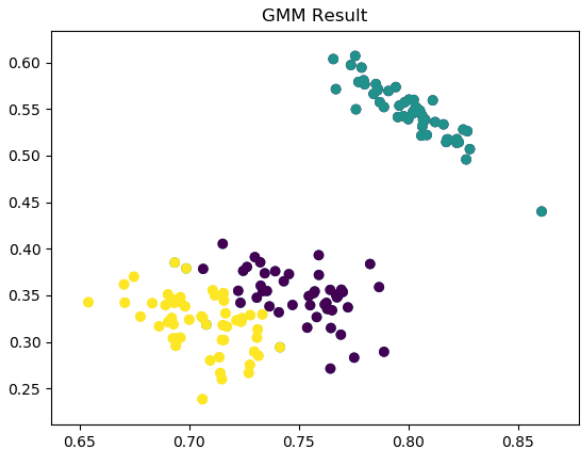

Figure 3: iris 聚类结果 2

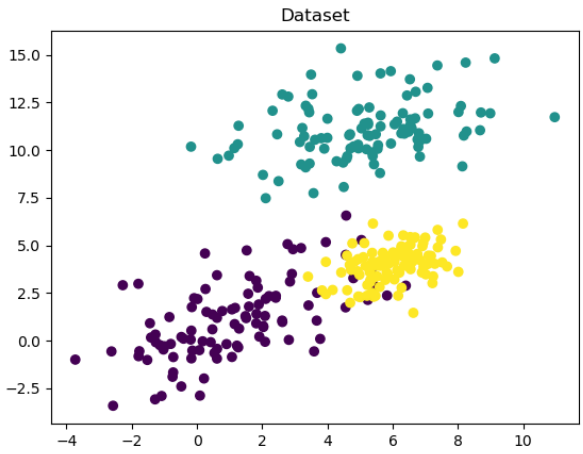

Figure 4: 自定义数据集

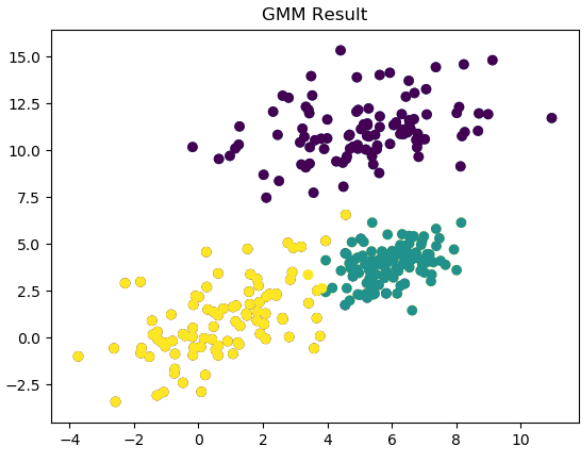

Figure 5: 自定义数据集聚类结果

为了 debug 这些错误花费了很多时间，甚至还尝试使用了 `scipy.stats.multivariate_normal` 来计算正态分布的概率密度和其对数值，但这些都没有很大的作用，而且有时甚至会让程序的运行时间变得很长。最后我发现只要能确保正态密度概率函数的结果是有效的，就能避免后续的很多错误。总之，我觉得模型计算中会涉及很多计算数据范围的问题，如果数据大小超出范围就无法再进行后续的计算，这些是在平时用公式时无法注意到的，只有实际通过代码进行计算才可能遇到。

4 参考

- Stanislas Morbieu –Accuracy: from classification to clustering evaluation