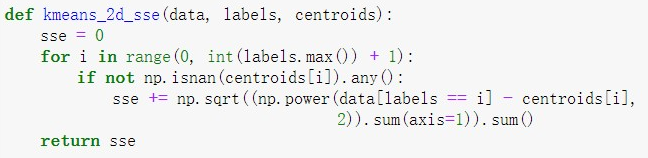
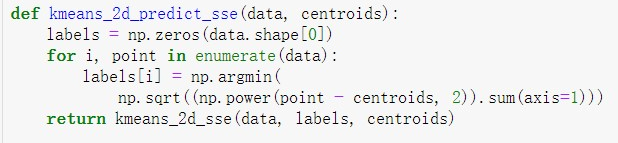
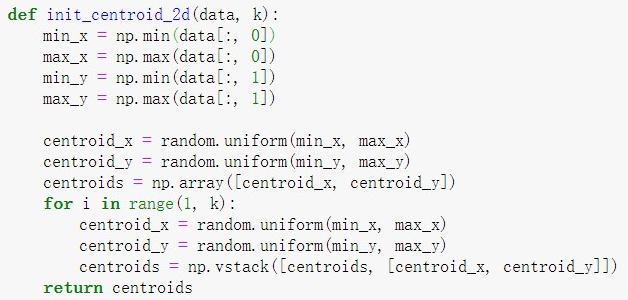
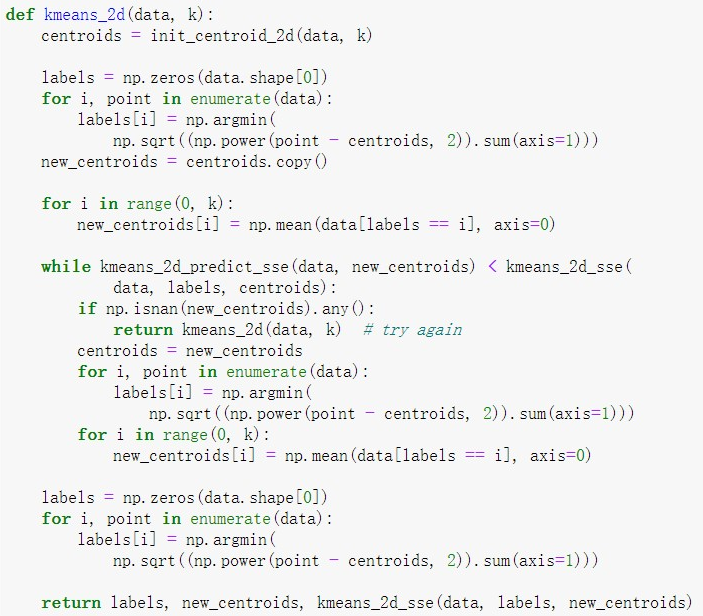
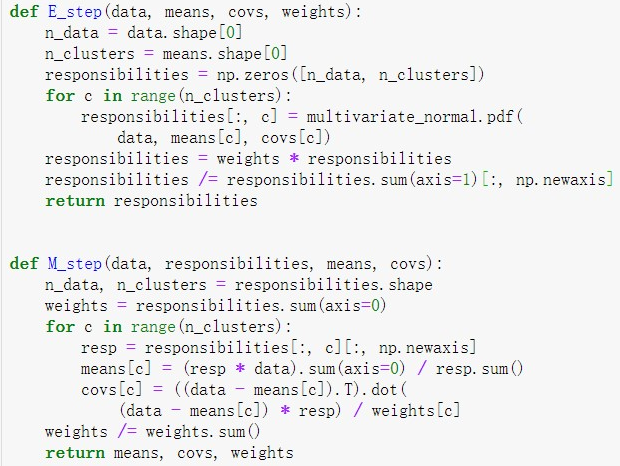
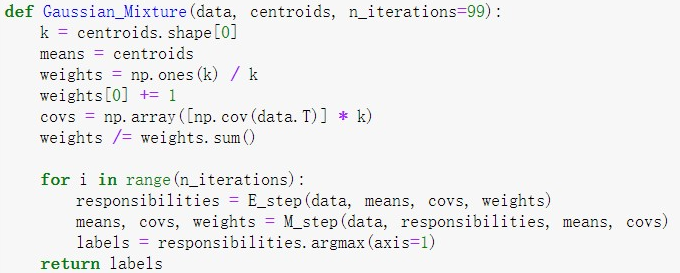
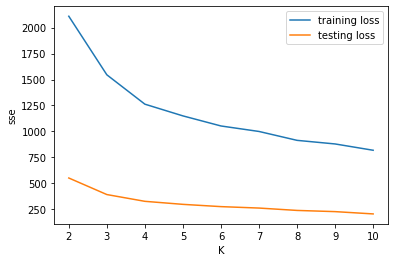
机器学习任务1实验报告

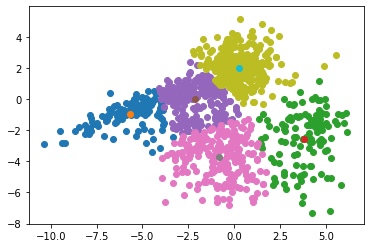
2017211416 张志博

1. 任务定义：目标是将二维数据集cluster.dat分为几个集群。
2. 完成任务（共60分）
   1. 方法1：K-means（20分）
   2. 方法1：额外内容（10分）
   3. 方法2：高斯混合模型（Gaussian Mixture Model）（30分）
3. 输入输出
   1. 输入：二维数据集
   2. 输出：数据分类结果（以及相关过程量）
4. 方法描述
   1. 方法1：K-means
      1. 目标：在cluster.dat上实现K-means方法并尝试不同数量的集群。
      2. 算法
         1. SSE计算函数**kmeans\_2d\_sse**
            1. 功能：SSE计算功能，输入数据集、分类标签及中心点，计算并输出SSE
            2. 
         2. 预测函数**kmeans\_2d\_predict\_sse**
            1. 功能：输入数据集和中心点，预测并输出分类标签结果
            2. 
         3. 中心点随机生成函数**init\_centroid\_2d**
            1. 功能：输入数据集和数量k，随机生成k个中心点，也用于GMM
            2. 
         4. 函数**kmeans\_2d**
            1. 功能：输入数据集和分类数量k，输出分类标签结果、中心点和用于评估的结果损失
            2. 
   2. 方法2：高斯混合模型（Gaussian Mixture Model）
      1. 目标：在cluster.dat上对混合的高斯实施EM拟合并尝试使用不同数量的混合。
      2. 算法
         1. E步和M步
            1. 功能：E步通过数据集、均值、协方差矩阵、权重计算出responsibilities，M步通过E步计算出的数据集、均值、协方差矩阵和新的responsibilities再计算出新的均值、协方差矩阵和权重。
            2. 
         2. 高斯混合
            1. 功能：输入数据集、初始的中心点（由其他方法，得出可提升性能，故没有把随机生成中心点的过程嵌入函数）和迭代次数，初始化EM算法所需的参数并执行n次EM迭代，计算并输出分类结果。
            2. 
5. 结果分析（性能评价）
   1. 方法1：K-means
      1. 不同k时分类结果和中心点

|  |  |
| --- | --- |
| K = 2 |  |
| K = 3 |  |
| K = 4 |  |
| K = 5 |  |

* + 1. SSE和K的关系图：



* + 1. 结果评价：几乎不存在最佳的K
       1. 光是从这组数据本身其实还是很难看出这组数据的结构的，因为数据点本身分布密集程度就很高，即使是把它们全分成1类也是合理的，因此，我认为在聚类关系不明显时使用强行聚类有时并不能正确地揭示数据地结构特性。但如果一定要选的话，k=5时分成的5组直观上更好，同时训练和测试SSE都比较小，且更高的k对SSE的提升效果也变更加弱。
       2. 
  1. 方法2：高斯混合模型（Gaussian Mixture Model）
     1. 迭代次数99次不同k时分类结果和中心点

|  |  |
| --- | --- |
| K = 2 |  |
| K = 3 |  |
| K = 4 |  |
| K = 5 |  |

* + 1. 结果评价：尽管不同的初始中心点对分类结果由影响，但是每次4个高斯混合的效果几乎都是最好的，它可以清楚地分出4类，同时更多的高斯也不能让分类效果更好。

1. 编程和实验的软硬件环境
   1. 软件环境
      1. OS：Windows 10
      2. IDE：PyCharm Professional + Jupyter Notebook
      3. Python：3.7
   2. 硬件环境
      1. CPU：Intel(R) Core(TM) i7-9750H
2. 代码
   1. 详见Task1\_Clustering.ipynb（内附实验输出结果）