暑期学校实验项目：高考志愿填报助手

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 小组名称 | aiB组 | | | | | | |
| 姓 名 | 张政 | 专业 | 人工智能 | 班级 | 615182 | 学号 | 61518207 |
| 实验时间 | 2020.8.31-2020.9.23 | | 指导教师 | 孔祥龙 | | 成绩 |  |
| **一、实验背景和目的**  高考是我国人才选拔的重要途经，而高考志愿填报则对于每一名考生来说又是除考试外最重要的一项事务。本次实验以2017-2019三年的各大高校录取分数和各省市的分数段数据为基础，尝试开发一套基于历史数据的高考志愿填报助手的系统。 | | | | | | | |
| **二、小组任务和个人任务**  本小组的具体工作是进行聚类算法的设计与实现，并和其他小组进行对接。我们的聚类算法需要对多年的分数、排名及高校进行分析，得出分段分布模型，以及专业的分布模型。我的个人任务是完成聚类算法的主体框架，并能通过数据调试至可正常运行。 | | | | | | | |
| 1. **个人任务需求分析**   我的主要任务是构建聚类函数，函数的输入包含两项，一项是分数列表，里面的每一个元素是包含分数和相应人数的列表；另一项输入是聚类中心的个数（是一个超参量）。函数的输出是一个列表，该列表长度与不同分数的数量等长，每一个元素记录了该对应分数最终被分到的聚类中心的序号。函数的主体部分包含两个部分，一部分是计算数据点与簇中心的距离，并根据最小距离更新数据点的归属；第二部分是实现簇中心的自我更新（根据每一次迭代后属于当前簇中心的所有点的加权值来更新）。 | | | | | | | |
| 1. **实验过程（需附上关键代码及相关说明）**   聚类算法的思想和步骤我们曾在人工智能专业课程内学习过，于此不在赘述。只说明一些关于我在实践过程中遇到的问题以及相应的解决措施。  起初，我确定距离计算可使用简单的欧式距离计算公式，考虑到不同的分数所对应的人数很可能不同，所以仅计算欧式距离并不能体现这一特征，为处理这一问题，我在根据每一分数对应的人数创建了对应人数个的数据点来代表每一个个体。然而这一措施引来了另一个问题，那就是算法的时间复杂度过大，因为不同的分数的个数并不很多，数量级在10^2，但是考生的数量很可能会达到10^4，那么用一个数据点去代表一个考生的方式必定导致计算量很大。为解决时间复杂度的问题，我最终改变了簇中心迭代时的计算方法，改为用人数去加权欧式距离的计算方法，那么数据点就可以仅根据不同分数的个数创建。  IMG_256 | | | | | | | |
| 1. **实验结果与分析**   2020925_pic3  上图为实验结果的一个示例，横轴为分数，纵轴为对应分数的人数，图中相同颜色的部分是表示属于相同一个聚类中心的所有分数。而图中所示含有k=5个聚类中心，k值是由其他组员通过拐点法确实的。观察聚类结果，每一个类别所包含的分数都是“连续”的，这一点是符合逻辑的，并且也大致符合直观上的分类（低，中低，中，中高，高）。综上，算法功能正确实现。 | | | | | | | |
| **六、实验总结与心得体会**  通过本次实验，我更多的是熟悉了码云上团队开发项目时任务和代码管理的操作，相比于以往我们通过私人的方式（QQ等）互相传送代码时更加有条理，并且同一组的所有成员都可以通过码云的仓库管理看到其他组员做了何种改动。  另一方面，我在本次实验过程中更加体会了函数封装的重要性。只有把函数封装做好了，代码在给其他人使用以及在自己后续做更改时才能更加高效、快捷，并且，函数封装更适合与小组内不同成员的功能函数的对接。当然，这也就需要大家在写代码之前商量好彼此的函数输入和输出的模式。 | | | | | | | |

2020年9月制