暑期学校实验项目：高考志愿填报助手

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 小组名称 | AI算法B组(第七组) | | | | | | |
| 姓 名 | 吉中旭 | 专业 | 吴健雄学院 | 班级 | 615182 | 学号 | 61518210 |
| 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**  每年高考志愿的填报对于考生来说都是一个很大的难题，许多考生因为对相关信息不了解而错过了自己合适的学校和专业。本次实验项目拟开发一个高考志愿填报助手，希望能够通过考生的分数，兴趣等信息给考生的志愿填报以合适的意见。 | | | | | | | |
| **二、小组任务和个人任务**  小组任务:  通过对各分段的特征信息进行聚类，需要对多年的分数，分数的排名，专业进行分析，得出恰当的分段分布模型和专业分布模型，辅助推荐系统的构建。  个人任务:   1. 聚类算法具体结构的设计，需要使用聚类算法来对每年的分数进行预处理。以获得每个学校大致的分数区间 2. 数据库信息的处理,对从数据库中读取到的信息进行过滤,获得需要各省的分数分布以及排名等方便用于整合的数据 | | | | | | | |
| **三、个人任务需求分析**  (1)聚类算法的设计:  这一模块主要实现的功能是进行每个省分段的划分，这样做的大致原因是:不同分段对应的学校数目不同，高分段的可选学校数相对较少，而中低分段的可选学校数相对较多。对不同分段的考生, 需要采取不同的择校策略。因此，为了确定考生所考分数所在的分段，需要利用聚类算法将不同省份的分数段进行聚类，将分数划分为高分段，中分段,低分段等等。具体的做法大致拟采用带权值的K-means，即将分数作为聚类点，将每个分数对应的考生人数作为聚类中心。在聚类时，会产生一个具体的问题，即如何确定所聚的类数。如果所聚的类数过多，划分的分段过细，起不到简化问题的效果。而如果所聚的类数过少，各个分段的特征又不能被展现出来。为了获得理想的效果, 采用以”误差平方和”为损失函数, 同时加以”肘部法则”判决的做法。具体方法是:画出聚类误差和聚类数之间的函数图像，然后找出变化率由陡峭到平缓的转折点。希望由此确定聚类数这一参数。  (2)数据库信息的处理:  这一模块主要的目的是对数据库中已有数据通过上面的聚类算法进行处理。获得每个省的聚类中心点(各分段的中心点)，在读取时需要能够对异常的信息加以过滤。同时,因为我们有三年的考分数据，需要对三年的数据加以整合。希望能够利用这些数据将学校,聚类信息,考生排名信息相结合，实现”分类针对性推荐”的功能。 | | | | | | | |
| **四、实验过程（需附上关键代码及相关说明）**  (1)聚类算法的设计:  首先是聚类算法的编写，函数中变量的基本定义如下    输入的是数据data和聚类数k.输入同时创建一个簇中心集合cluster(长度为聚类数)和描绘分数隶属情况的数组belong(长度与输入的data相同)。data的数据类型如图所示,其中包含n个分数数据，每个数据是一个列表，形式为[分数, 人数]。  接着是数据点所属类的更新(数组belong的更新)    通过循环找出数据点到各个聚类中心的距离,并将该数据点归类到距离其最近的聚类中心中。  最后是聚类中心数组的更新(cluster的更新)    计算每个类中数据的权值(人数\*分数)，并加权求平均,得到更新后的中心点值  为了求出最佳聚类数，采用的均方误差判别准则如下:    通过已经获得的聚类中心(cluster)和隶属中心(belong)数组,计算每个点到其聚类中心点的距离并求和。    设置聚类数为2-8，利用上面的函数计算不同聚类数下的均方误差，并找出拐点。  (2)数据库信息的处理  原始从数据库中获取到的数据信息较为繁杂，如下图所示,可以看到不同省份对应的数据不同，同时还有省份数据缺失，直接进行处理很不方便，需要加以修改。    修改方法为:    通过将这些数据进行聚类,如果数据的总数较少(这里设定阈值为5),就将这一数据剔除。这样最终获得的是一个各省分段后的聚类中心的列表。 | | | | | | | |
| **五、实验结果与分析**  聚类结果:  首先先展示确定聚类数的方法，通过计算不同聚类数对应的误差平方和,绘制的图形为    发现拐点大致在聚类数为5这一位置,因此选择5作为聚类中心数。  之后对各省高考分段聚类，获得江苏省2019高考分数的数据分布大致如下所示    可以看出数据分布比较平均，大致服从正态分布，和中心极限定理是相吻合的。同时分数的划分也较为合理，证明了之前聚类方法的正确性。  同时,对各个省的聚类数据而言,处理后会获得一个距离中心的列表，借由这个列表可以完成学校推荐的功能，列表的部分数据形式大致如下所示。前面的编号为”省份+年份+科类”，后面的5个数据为5个聚类中心点。 | | | | | | | |
| 通过上面获得的数据,结合推荐程序,最终所输出的推荐学校如下: | | | | | | | |
| **六、实验总结与心得体会**  在本次实验中，通过程序的编写和调试，以及组内成员的交流。主要有以下的几点体会:   1. 算法的理论可用和具体实现之间的距离:虽然这次实验中采用的是一个简单的聚类算法，理论上应当不需要做太多的调整, 但实际应用到对分数进行分类时还是会出现各种各样的问题,如分的类过少，原本应当分为5类的数据最后只分了4类;分的类不符合实际情况,有的类中数据过多，有的类中数据过少；读取聚类数据时,因为数据不完全导致程序报错等等。虽然其中有很多是技术上实现的问题，但从另一个侧面也反应出实际构建应用系统时的困难之处。从数据读取，到处理，到算法构建，到最终的结果呈现。每一步都需要我们仔细检查和审视。单单是了解算法本身，而不了解这些各个模块交互的知识，是没有办法做出方便,好用，稳定的系统的。 2. 团队成员之间协作和交流的重要:因为这次的实验需要设计到不同组之间的交互，就算在组内也涉及到不同成员之间的交流。因此在编写程序时,就格外需要预先确定程序输入输出的类别规格。在本次实验时，我因为这一方面没有事先和其他成员商量完善，做出的输出起初并不适宜他人使用，而自己也花费了许多精力，造成了时间上的浪费。之后的课程中需要对此更加注意。 3. 实际性能的检验:因为这次实验中我们使用的是一个开放性的算法，即使编译通过,也无法确定我们算法的工作效果。需要结合具体实例进行大量的工作检验。但我们最开始却遗漏了这一点。起初我们采用了处于中间分段的数据进行检验发现效果尚可，就认为我们的算法已经可以满足功能。但后来有别的同学采用高分段数据检验,发现推荐的学校和分数明显不匹配。经过反复检查后，我们发现是因为我们犯了一个较为严重的错误，只采用了聚类中心处的学校做推荐。之后我们给推荐程序加上了数据到聚类中心的权值这一操作，最终取得了较好的效果。从这个经验中我们意识到，在检查时一定要采用大量的数据,最好是多个人同时检查，这样才有可能更好的检测出程序的漏洞。 | | | | | | | |

2020年9月制