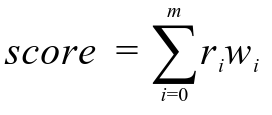
基于多决策向量的词条质量评价方法－说明文档

1. 方法原理

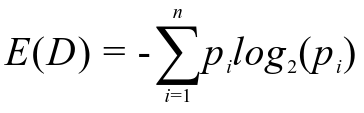
多决策向量模型属于共同决策模型，之所以称之为“多决策”，是由于词条的质量决定于多个评价项。词条若有m个属性，词条向量为m维向量，此时共同决策分类器的个数也等于m，假设词条向量X为(x1,x2,...xm)，训练分类器组Classfieri，其中有i属于1-m。对于每一个词条向量X，存在唯一的向量R与之对应，有：R = (r1,r2,...rm)，ri∈{0,1}。由上式可知，R向量是由m个分类器分别对m个属性分类得到的结果，ri由0或1的值组成，其中标注１代表高质量分类，标注0代表低质量。

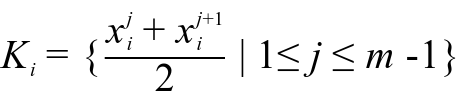
通过训练多个CART决策树，将原本的词条向量中的属性数据转化为基于该属性的质量分类，每一项质量分类结果ri都由对应的分类器根据对应属性值得到。由于词条的质量基本无法根据单个属性区分，所以ri无法作为词条质量评价的参考结果，但是不同的属性对于词条质量的区分度是不相同的，如以百度百科为例，图片数与参考数多的词条更有可能被认为是高质量词条，所以以图片数或参考数区分词条质量，要比以其他属性区分的准确率高。由此可知，每一个属性都对应一个权重wi，反映该属性在所有属性中对词条质量的影响程度，在此基础上借鉴人工评审模型，提出了词条质量评分的计算公式：



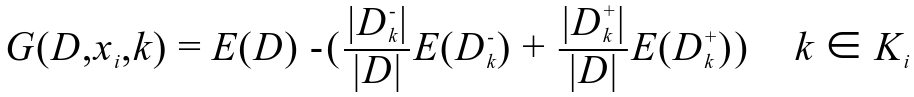
权重确立：

E(x)为D的总信息量，对于某一个连续属性xi，需要将连续的取值划分成n个离散取值。常用的离散化策略是二分法，假设xi在D上出现了m个不同的取值，将取值从小到达排列为D$Y70]_N[{7E)0)K@QUY6TP，存在划分点k将D划分为D-与D+，其中D-为Xi不大于k的取值集合，其中D+为Xi大于k的取值集合。由此可知，连续属性Xi对应有m-1个划分点{k1,k2,...km-1}，其中：





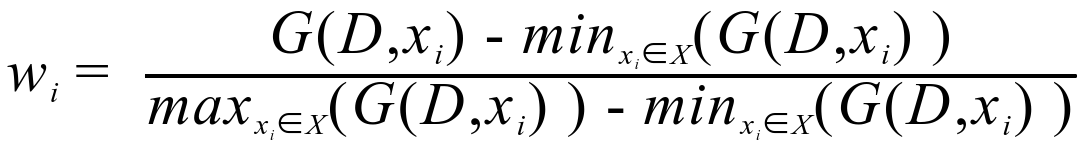
计算相应属性的信息增益集合。由于连续属性xi实质上也是离散化的，所以将两个相邻值的中间值作为划分值。划分完成后，原来的系统被划分成两个系统D-与D+，对应的信息增益可以表示为：



计算信息增益量最大值。选择不同划分值中信息增益的最大值作为词条属性xi的信息增益，即：



max-min归一化得到权重。其中G(D,xi)表示xi属性的信息增益，在得出信息增益值之后，对{G(D,x1),G(D,x2),...,G(D,x3)}进行Max-Min归一化处理，将G(D,xi)转化成权重，其中wi为词条属性xi的权重，具体如公式-所示：



1. 模型训练

训练数据为词条以及对应编辑者的属性数据，以.csv格式保存，分为正例（标注优质词条）和负例（标注普通词条）两个csv文件，具体格式如下表所示。

表１ 百度百科词条自身相关属性

|  |  |
| --- | --- |
| 编号 | 属性名称 |
| 1 | 摘要字数 |
| 2 | 内容字数 |
| 3 | 一级标题个数 |
| 4 | 二级标题个数 |
| 5 | 三级标题个数 |
| 6 | 图片数 |
| 7 | 参考文献数 |
| 8 | 总浏览数 |
| 9 | 分享数 |
| 10 | 赞同数 |
| 11 | 编辑次数 |
| 12 | 标签数 |
| 13 | 义项数 |

表２ 词条编辑者自身相关属性

|  |  |
| --- | --- |
| 编号 | 属性名称 |
| 1 | 编辑者提交的优质版本数 |
| 2 | 编辑者提交通过总数 |
| 3 | 编辑者等级 |
| 4 | 编辑者提交通过的特色词条数目 |
| 5 | 编辑者创建词条通过数 |
| 6 | 编辑者提交总数 |
| 7 | 编辑者经验 |
| 8 | 编辑者通过率 |

配置根目录下myConfig.py的参数，分别有普通词条数据路径self.normal\_data\_path，特色词条数据路径self.characteristic\_data\_path，测试集划分比self.test\_size = 0.1，每一列训练次数self.train\_time，模型保存地址self.model\_save\_path，多决策向量保存路径self.tensor\_save\_path = "../data/"。

全部参数配置完成且正确，且确保正确配置了python3.5的环境变量后，在根目录下运行python3 train.py，开始训练。

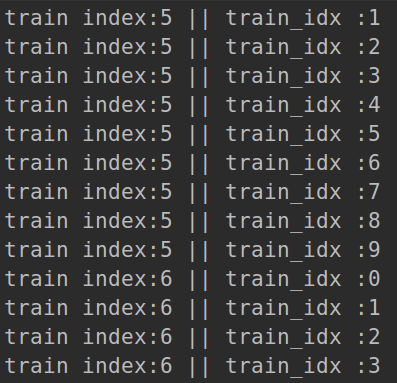


图1 训练过程图

训练完成后，根据配置的地址生成对应模型的.pickle二进制文件和对应测试集的多决策向量文件MDT.csv。

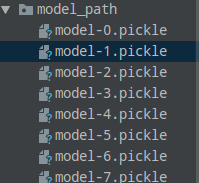


图2 生成二进制模型

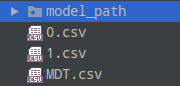


图3 生成多决策向量MDT.csv

上述模型与数据全部生成后，训练完毕。

1. 模型测试

切换到gain目录下，运行python3 information-gain.py，计算多决策向量权重，等待输出结果。

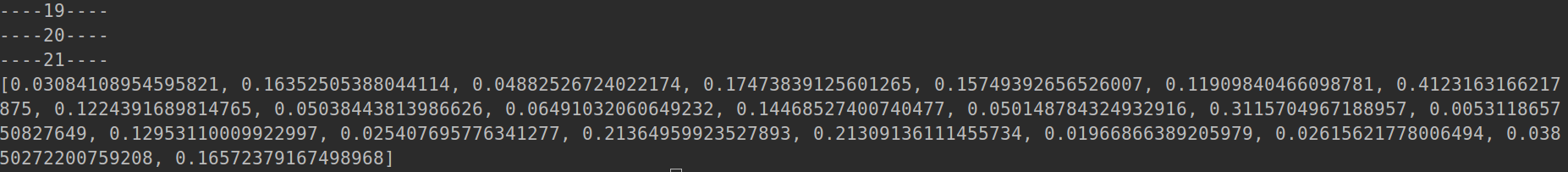


图４ 权重计算结果

将权重复制到myConfig.py下的self.weight，运行python3 predict.py。生成质量评分排序柱状图。

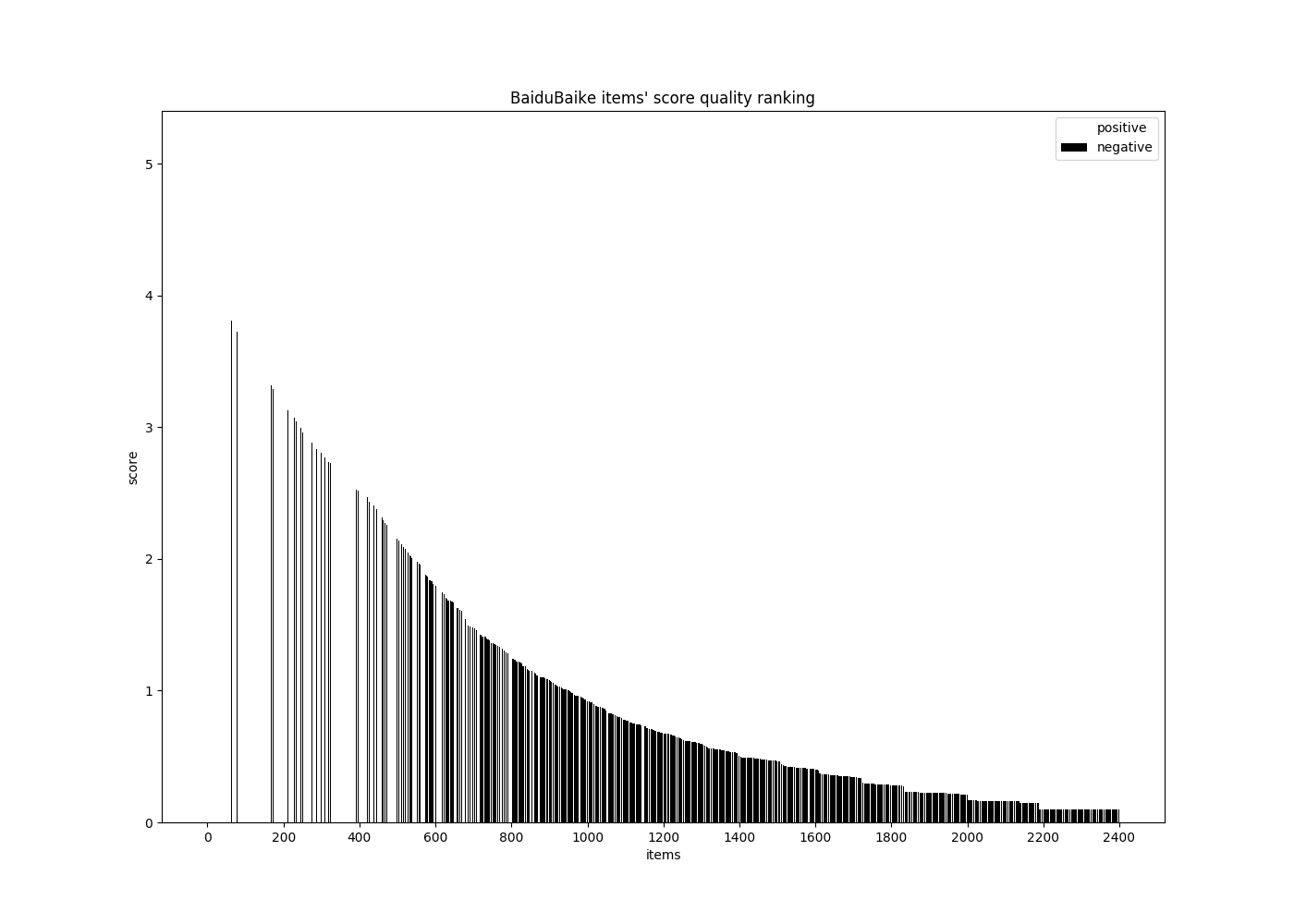


图５ 标注为普通词条的评分分布

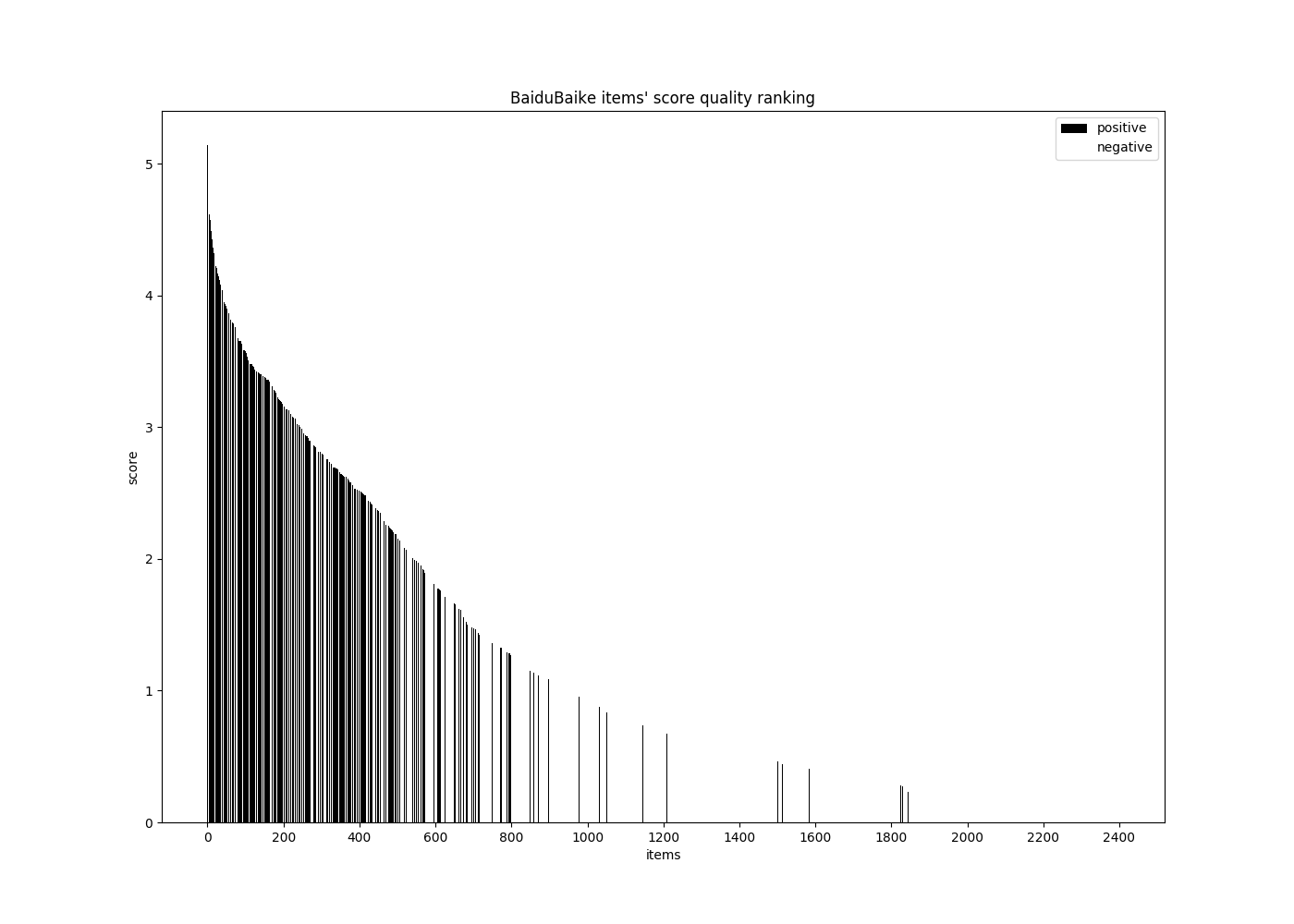


图６　标注为特色词条的评分分布

在默认阈值百分比为20%，即筛选前百分之20%回归高质量词条时，模型评估参数如下所示：

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| precision\_pos | precision\_neg | recall\_pos | recall\_neg | F1\_pos | F1\_neg |
| 0.87167 | 0.70239 | 0.77793 | 0.90781 | 0.96591 | 0.93596 |

1. 模型应用

命令行输入python3 runner.py url，url为百度百科网页地址，如周杰伦的百度百科网址为https://baike.baidu.com/item/%E5%91%A8%E6%9D%B0%E4%BC%A6/129156?fr=aladdin

运行如下命令：

python3　runner.py　<https://baike.baidu.com/item/%E5%91%A8%E6%9D%B0%E4%BC%A6/129156?fr=aladdin>

得到输出如下图所示，可见得分为3.11846，根据前10%为高质量词条的原则，该词条被评为优质词条，根据网页查看，该词条也确实为优质词条。

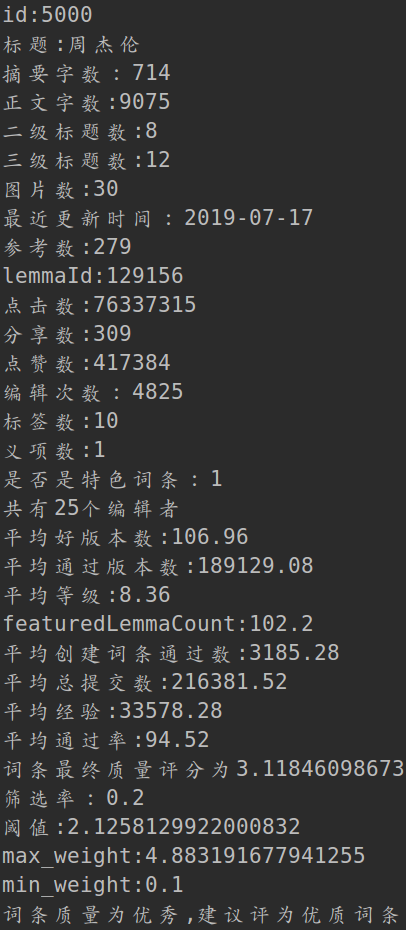


图7　词条质量评价输出结果