决策树

1. 思路
2. 处理数据
3. 分析数据

首先拿到数据集，分析得共两个数据集，



第一个数据集包括785列50000行，第一列是标签，即数字。后面的784个数字表示组成手写数字的像素点，但这些数字并不是有则为1，无则为0的填充，而是无则为0，有则大于0。



第二个数据集包括785列10000行，格式与上述格式相同。

1. 处理数据

每个维度上的值若大于128测存1，否则就不管，相当于简化计算

1. 处理结果

处理完的结果跟处理前的差不多，只是大于128的数就被处理成1了。这里处理完之后每个数字的含义就有这样的规律：0和1的意义完全不同，而1和2、1和128的意义差不多。

1. 算法的相关函数

此次代码参考李航的第二版《统计机器学习》，代码分为很多个函数，以ID3算法为例：

|  |  |  |
| --- | --- | --- |
| 函数名 | 功能 | 函数细节 |
| loadData | 加载并处理数据 | 第一列数字保留到标签列表；  第二列数字处理后保存到数据列表 |
| createTree | 构建树 | 决策树的核心部分：构建树  根据统计机器学习P77算法：  首先设置阈值  然后构建子节点  若所有实例均为同一类，则将该类作为节点标记，返回只有一个节点的树。  若A为空，则T为单点树，将D中实例数最大的类Ck作为类标记，返回树。  Ag为信息增益最大特征，EG为最大信息增益，通过calcBestFeature函数获得。  比较最大信息增益和阈值大小，若Ag的信息增益小于阈值  置T为单节点树，并将D中实例数最大的类Ck作为类标记，返回树  否则，初始化一个树，用字典保存  迭代建树 |
| calcBestFeature | 获取最佳特征 | 初始化最大信息增益为-1  初始化最大特征为-1  循环计算信息增益  开始更新最大信息增益和特征 |
| mainClass | 找主要类别 | 相当于用KNN的方法找一个类别字典中数量最多的字典 |
| calc\_H\_D | 计算经验熵 | 先将H(D)初始化为0  保存所有类型的特征（利用集合的非重复性）  计算Pi及其对应的熵 |
| calc\_D\_A | 计算经验条件熵 | 初始化H(D|A)  计算|Di|/|D|·H(Di)并累加 |
| getsubDataArr | 更新数据数组（使用更新后的数组进行建树的迭代） | 初始化数据和标签数组  若Ag=ai  则更新数据数组（一个变两个，以Ag分开）和标签数组 |

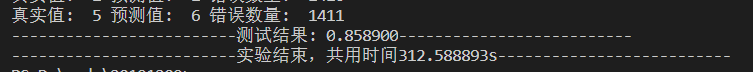
1. 实验

ID3算法：

参数设置：阈值Epsilon = 0.1

评价标准：准确率accur = 1- errorCnt/len(testDataList)

实验结果：



C4.5算法：

ID3算法：

参数设置：阈值Epsilon = 0.1

评价标准：准确率accur = 1- errorCnt/len(testDataList)

实验结果：

没写出来，计算HaD时会出现0的情况