**Softmax及CART决策树**

卢立仁

**摘要：**本文会介绍softmax分类器和CART决策树的原理及实现过程，以及作者的学习过程和收获。

**关键词：**softmax,多分类,神经网络,机器学习,CART决策树

1. **softmax分类器**
   1. **任务简介**

用一个类别为10的数据集训练一个softmax分类器，使其尽可能在测试集中达到较高的准确度。

* 1. **原理的说明及公式的解释**
* 模型：本模型是一个以softmax函数为激活函数的单层神经网络
* 激活函数：softmax函数可以将多分类的输出值范围转换到[0,1]中。在多分类任务中，将样本的特征输入到以softmax函数为激活函数的神经网络（以只有一层输出层为例），得到由该样本在各类别的概率所形成的向量，选取概率最高的值为预测结果。
* 矩阵的设置：

设数据集共有M个样本，每个样本共有n个特征。其中，对样本（不含类别标签）矩阵的设置如下：

对单个样本，，对整个数据集，.

对类别标签矩阵（采用one-hot编码）的设置如下：

对单个样本：，对整个数据集：.

对预测值矩阵的设置如下：

对单个样本，，

对整个数据集，.

* 损失函数：采用交叉熵作为损失函数。
* 权重的更新：经过推导可以得出，
* 细节的处理：

1. 特征的归一化：由于本数据集的某些特征取值与大部分特征取值相差过大，且其本身数值过大，容易影响损失函数的收敛效果，且可能造成输出值的溢出等后果，所以需要在训练前对特征的取值进行归一化处理。本文采用的归一化方式为standardization:.
2. 防止数值溢出：由于指数函数的增长率过高，使用softmax函数时容易造成数值溢出的后果，所以需要对原始的函数进行优化。本文在使用softmax函数时，将输入函数每一个输出值z减去输出值中的最大值max(Z)以达到防止数值溢出的效果。具体公式如下：
   1. **训练流程**

数据预处理及偏置、权重的初始化

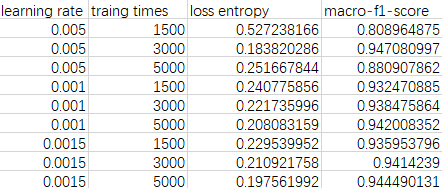
设置学习率与迭代次数并开始迭代

计算模型的评价指标，并调整参数进行下一轮训练

* 1. **模型的评价指标及试验结果**
* 评价指标：

macro-f1-score:,

* 试验结果：



1. **CART决策树**
   1. **任务简介**

理解cart决策树的原理，并用代码复现。

* 1. **原理及流程**

cart决策树分为分类树与回归树两种，二者的主要区别在于预测值的类型不同，前者预测的是样本的一个种类（离散值），后者预测的是一个数值（多为连续值）。Cart决策树的构建可分为决策树的生成与决策树的剪枝两步。

* + 1. 决策树的生成
* 主要流程：

继续划分

分

* 分类树与回归树的差别：分类树和回归树在决策树生成阶段的大致流程是相同的，不同点在于对最优属性的选择。

分类树采用基尼指数选择最优属性与最优划分点：如，数据集D在特征A取某一个取值a时的基尼指数为.遍历所有特征的所有取值，基尼指数最小的即为最佳特征的最优划分点。

回归树采用平方误差最小化准则选择最优属性与最优划分点：采用启发式的方法，选第j个特征和它的某个取值s，作为切分变量和切分点，定义两个区域

和，

然后寻找最优切分变量j和最优切分点s，

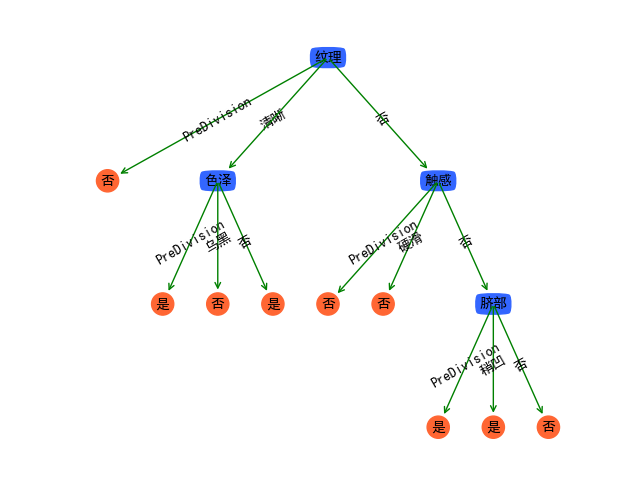
，

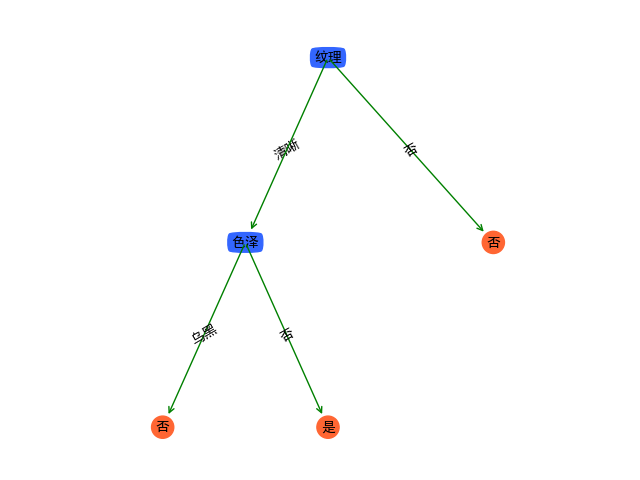
通过求导易得，.

* + 1. 决策树的剪枝
* 主要流程：

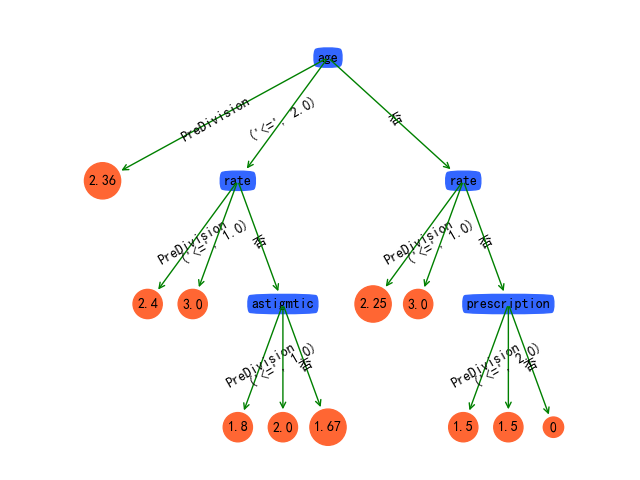
继续剪枝

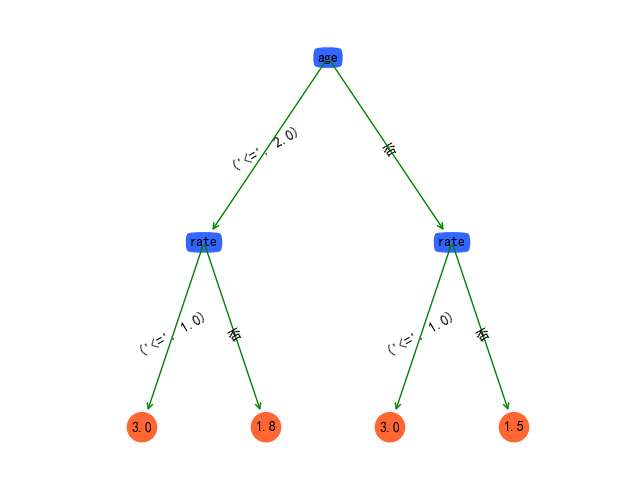
* α的计算：，其中为该结点作为单结点树时的损失，为该结点作为根结点时的子树的损失，为该结点作为根结点时的子树的叶结点数。
  1. **效果**
* 分类树：

****

****

* 回归树：

****

****

1. **学习过程与收获**

刚看到softmax的内容时感到无从下手，式子能看懂，但不知道里面的输出值是怎么来的。我于是继续查找资料，发现资料的图片都有个共同点，就是有多个结点，这令我想到了神经网络，然后我通过看视频和博客就了解了大致的原理。开始打代码时又出现了很多问题，如偏置和权重的初始化、特征的归一化，与对标签进行独热编码等问题，其中最困扰我的是怎么进行偏置的更新，在多个结点与多个会影响偏置结点的值的样本的输入下更新参数是我没有试过的，这在资料上也找不到。于是，我便自己推了一下公式，最终能将整个过程都用矩阵形式运算。我在训练分类器中收获到的是应该带着问题去找答案，这样效率会很高。

CART决策树应该是我所见过的最麻烦的一个算法了，数学公式都不算难，难就难在过程的繁琐。我在最初的时候没有参考过别人的算法，自己也没有对整个框架的构建进行思考，就一股脑的写了下去，这导致我的代码十分冗杂，bug也频出不穷，且我的决策树规定的数据集还是dataframe格式，这些都极大地拖慢了我的进度。我在此过程中深刻感受到了打代码也应该先进行整体的规划的道理。