机器学习第三次实践心得体会

一 介绍

本次机器学习实践是随机森林，共有7个文件，其中train(树叶分类器数据集）是本次使用的训练集，共192个特征，160个样本，16个类别：



1.主函数randomforest.m，完成训练集数据的预处理，对特征和样本进行抽样，生成决策树模型和随机森林，以及计算模型在训练集上的准确率。

2.函数buildtree.m构造决策树模型，使用抽样和特征选择后的训练集子集递归地构造决策树。

3.函数en\_Entropy.m作用是计算某一列矩阵的熵。

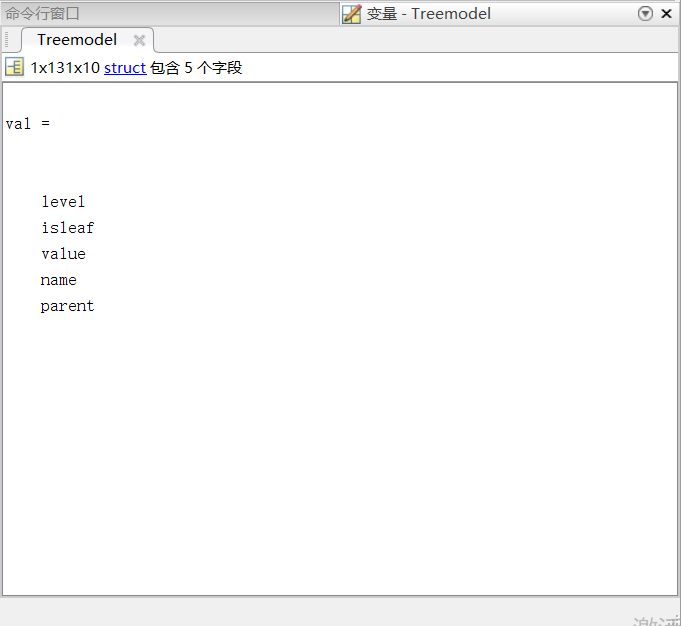
4.函数featurechoise.m作用是选择当前属性中增益最大的属性。

5.函数newdata.m作用是分裂数据集，分裂出下一节点需要使用的数据集。

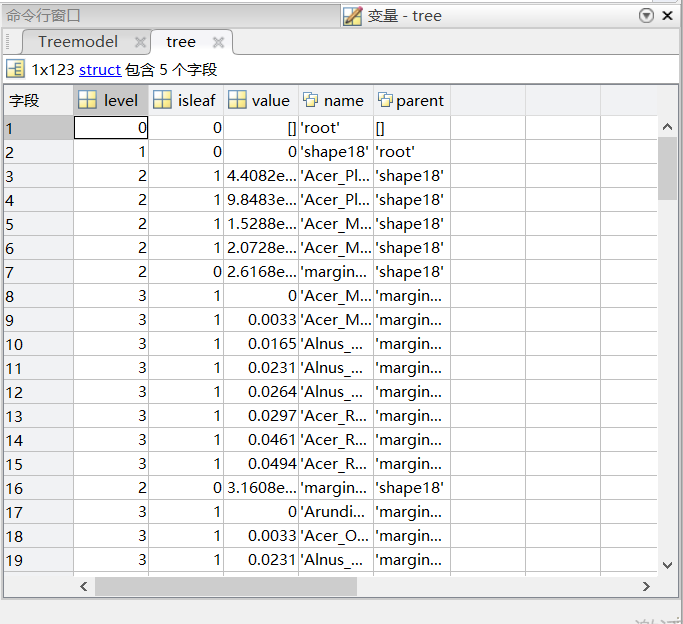
6.函数treeclassfisher作用是对输入的数据进行分类。

二 结果

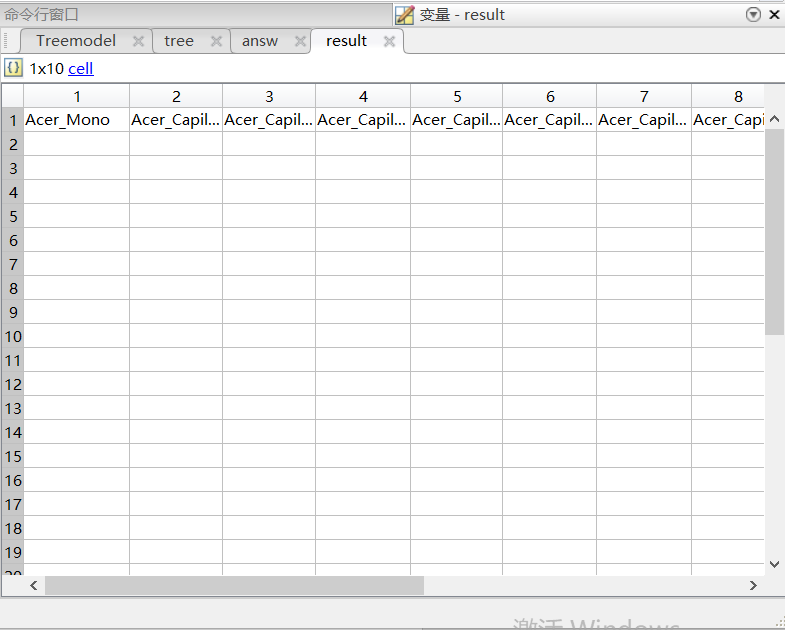
所构造的随机森林共有十颗决策树，训练集共有192个特征，每棵树有放回地随机选择96个特征构建决策树：



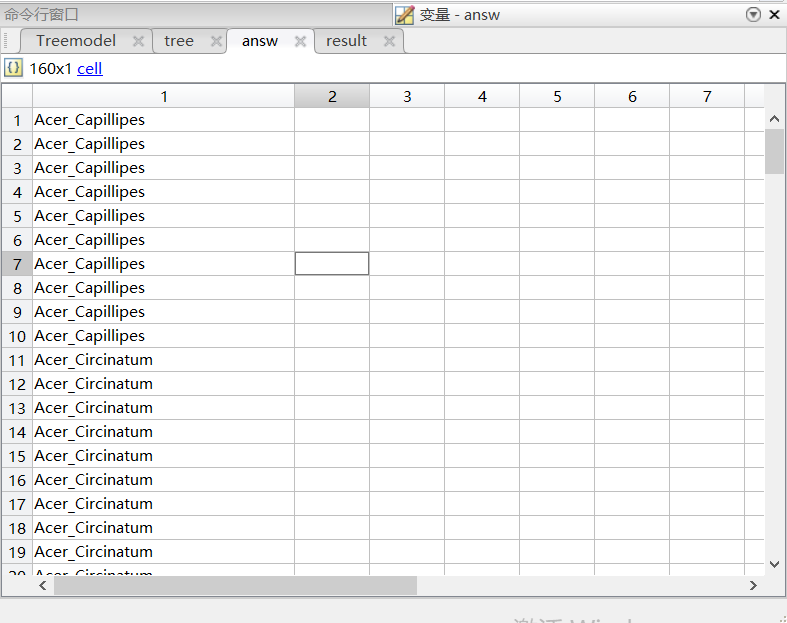
其中某一棵决策树如下，parent是父节点姓名，name是当前节点姓名，value是分支的属性值，isleaf代表是否为叶节点，level为当前层数，存储方式为前序遍历：



随机森林对某一样本进行分类，第一棵树分类结果为'Acer\_Mono'，其余九棵树分类结果为'Acer\_Capillipes'，投票结果为'Acer\_Capillipes'：



对训练集所有样本进行分类与投票，结果如下：



随机森林在训练集上的准确率：



三 遇到的问题以及解决方法

1. 随机森林的存储：

本次实践是基于第二次实践的决策树算法实现的，搭建的十棵树存储在一个三维矩阵中，但是由于每棵树所使用的特征与样本不同，每棵树的节点数也不同，因此会造成维数不一致的问题。解决方法是预先构造150个空节点，当生成决策树时，用新的节点覆盖空节点。在使用决策树时，从随机森林中取出决策树，再剪去无效的空节点。

1. 样本与数量的选择：

每颗决策树由随机有放回抽取的特征与样本构造，当某棵树选择样本时，由于样本的选择问题，可能缺失某些特征的属性值，当对具有该属性值的样本进行分类时，这棵决策树将无法分类出该样本，处理方法是对于此类情况，该决策树的分类结果为“unknown”，由其他树对该样本进行分类。

特征选择方面，由于选择的树叶数据集是图片信息，分别是8×8边缘图像、8×8形状图像与8×8纹理图像，选择特征过少会造成图像信息的大量缺失，因此对于每种图像，随机选择了32维特征进行决策树的构造。

四 可以改进的地方

1.本次完成的决策树算法没有进行剪枝。

2.没有画出决策树。

3.程序的运行速度较慢，还有需要优化的地方