机器学习之决策树

2011011239 计算机系 13 班 王庆

目录

一. 实验设计 1
二. 实验数据 2
三. 实验分析 4
四. 参考网址 6
五. 感谢篇 6

一. 实验设计

任务一：使用 C4.5 算法（与实际算法稍微不同）构造决策树

输入:训练数据集 D,特征集 A,阈值 threshold(如果实例数少于 threshold，则特殊处理)

输出：决策树 T

1. 如果 D 中所有的实例属于同一类 label,则置 T 为单节点树,并将该 label 作为该节点的类，返回 T
2. 如果 A 为空，则置 T 为单节点数，并将 D 中实例数中最大的类（比例所占最多）label 作为该节点的类，返回 T
3. 如果 D 中实例的数目少于 threshold，则置 T 为单节点数，并将 D 中实例数中最大的类（比例所占最多）label 作为该节点的类，返回 T
4. 否则，按照以下算式计算 A 中各特征对 D 的信息增益比，选择信息增益比最大的特征 Ai 作为节点

5. 对 Ai 这个特征的每一个可能的值 i 作为该节点的子节点，根据 value=i 将 D 分割成若干非空子集 Di，将 Di 中实例数作为训练集，A-Ai 作为特征集递归建树

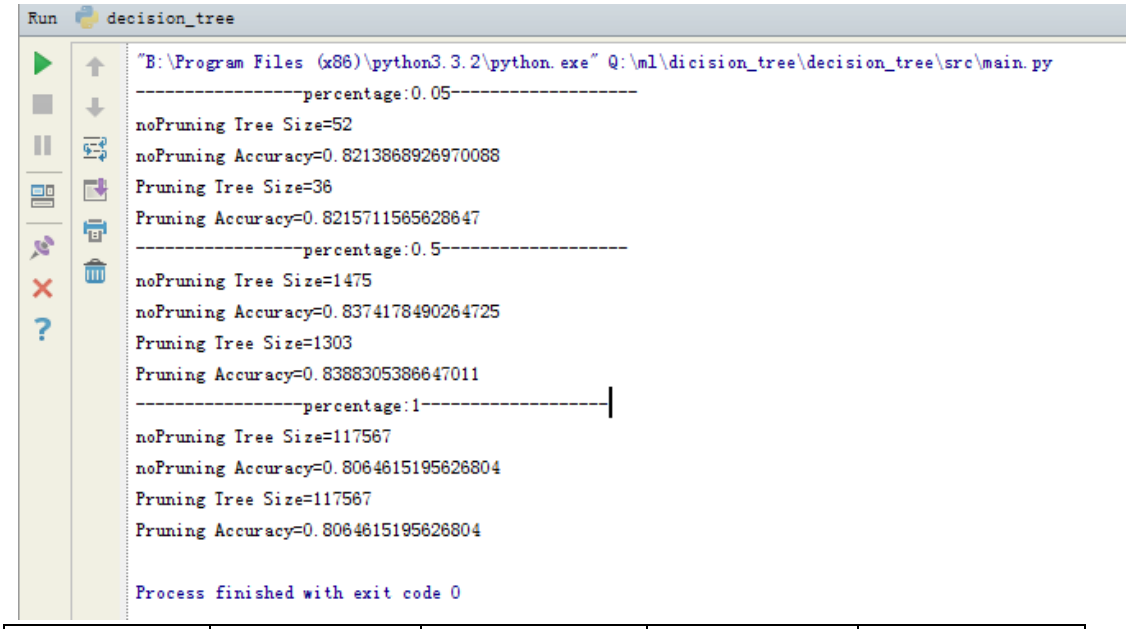
任务二：后剪枝算法，贪婪剪枝

1. 将训练数据分为训练集和测试集
2. 判断点的子节点是否都为叶子结点，如果是，则直接把该点改为测试集 label 概率最大的叶子结点，如果拿该叶子结点测试测试集的概率〉原节点测试测试集的概率，

   则剪掉该点，即将该节点换成测试集 label 概率最大的叶子结点。
3. 如果点的子节点不全是叶子结点，则测试集沿着该点分离出相关数据，递归剪枝。

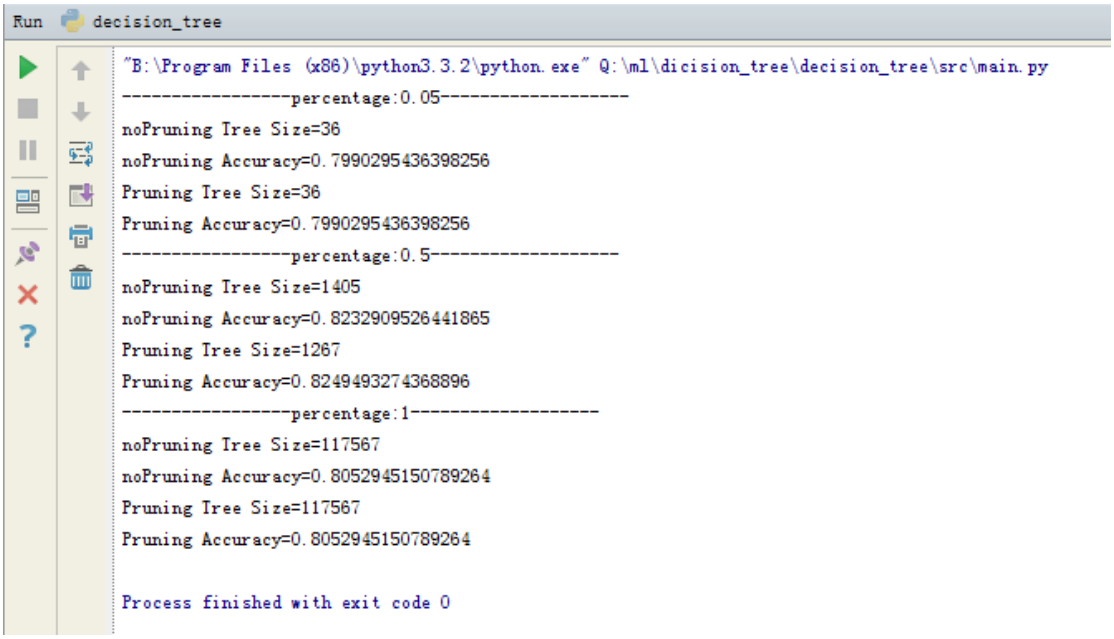
二. 实验数据

第一次数据：



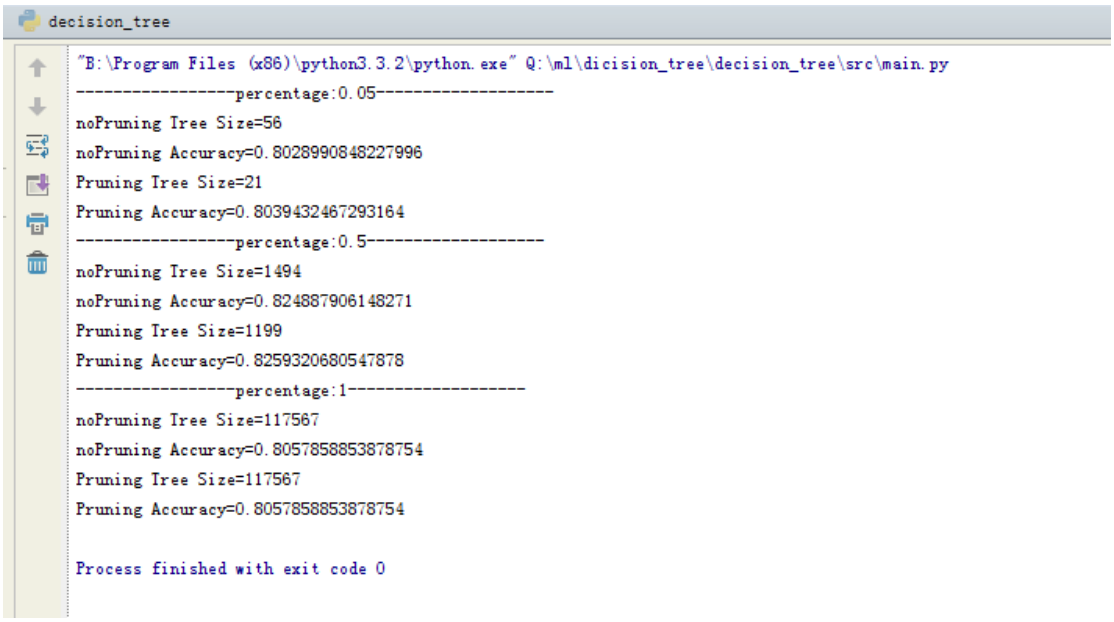
训练集	未剪枝-树点	未剪枝-准确率	后剪枝-树点	后剪枝-准确率
5%	52	0.8213868997	36	0.8215711566
50%	1475	0.8374178490	1303	0.8388305387
100%	117567	0.8064615196	117567	0.8064615196

第二次数据：



训练集	未剪枝-树点	未剪枝-准确率	后剪枝-树点	后剪枝-准确率
5%	36	0.7990295436	36	0.7990295436
50%	1405	0.8232909526	1267	0.8249493274
100%	117567	0.8052945151	117567	0.8052945151

第三次数据：



训练集	未剪枝-树点	未剪枝-准确率	后剪枝-树点	后剪枝-准确率
5%	56	0.8028990848	21	0.8039432467
50%	1494	0.8248879061	1199	0.8259320681
100%	117567	0.8057858854	117567	0.8057858854

平均值：

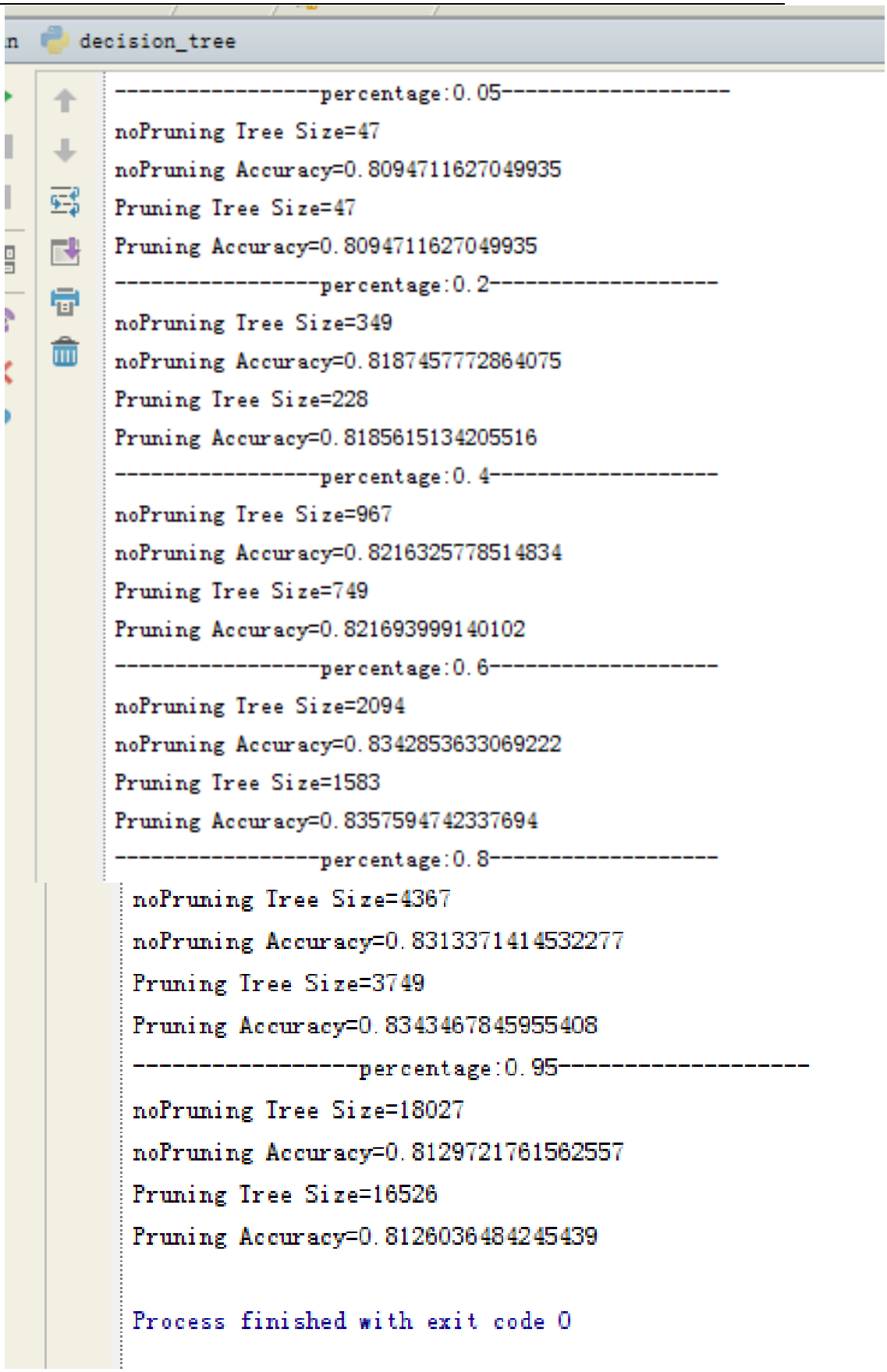
训练集	未剪枝-树点	未剪枝-准确率	后剪枝-树点	后剪枝-准确率
5%	48	0.807772	31	0.808181
50%	1458	0.828532	1256.333	0.829904
100%	117567	0.805847	117567	0.805847

三. 实验分析

我们可以看出随着训练集的增大，准确率先变大后变小，原因在于训练集过大，会导致过渡匹配的现象发生，所以要选择合适比例的训练集和测试集，适当剪枝。

虽然采用增加训练数据和后剪枝的方法，可以适当提高准确率，但也增加了程序运行的时间和运行时所占的内存，造成一定的资源浪费，对电脑的性能也会有一定的要求。

因为训练数据划分训练集和验证集的百分比间隔过大，当训练集 100%时，没有剪枝，所以以上剪枝的效果不是很明显，所以我跑了训练集是 0.05，0.2，0.4，0.6，0.8，0.95 时的剪枝前和剪枝后结果：


decision_tree
----------percentage:0.05----------
noPruning Tree Size=47
noPruning Accuracy=0.8094711627049935
Pruning Tree Size=47
Pruning Accuracy=0.8094711627049935
----------percentage:0.2----------
noPruning Tree Size=349
noPruning Accuracy=0.8187457772864075
Pruning Tree Size=228
Pruning Accuracy=0.8185615134205516
----------percentage:0.4----------
noPruning Tree Size=967
noPruning Accuracy=0.8216325778514834
Pruning Tree Size=749
Pruning Accuracy=0.821693999140102
----------percentage:0.6----------
noPruning Tree Size=2094
noPruning Accuracy=0.8342853633069222
Pruning Tree Size=1583
Pruning Accuracy=0.8357594742337694
----------percentage:0.8----------
noPruning Tree Size=4367
noPruning Accuracy=0.8313371414532277
Pruning Tree Size=3749
Pruning Accuracy=0.8343467845955408
----------percentage:0.95----------
noPruning Tree Size=18027
noPruning Accuracy=0.8129721761562557
Pruning Tree Size=16526
Pruning Accuracy=0.8126036484245439
Process finished with exit code 0

训练集	未剪枝-树点	未剪枝-准确率	后剪枝-树点	后剪枝-准确率
5%	47	0.8094711627	47	0.8094711627
20%	349	0.8187457773	228	0.8185615134
40%	967	0.8216325779	749	0.8216939991
60%	2094	0.8342853633	1583	0.8357594742
80%	4367	0.8313371415	3749	0.8343467846
95%	18027	0.8129721762	16526	0.8126036484

虽然按照前面 5%，50%的训练集和验证集划分，剪枝后准确率上升了，但是从上表可以看出，当按照 20%的训练集和验证集划分时，剪枝的准确率下降，所以由此可以看出训练集和验证集的比例划分也会影响最终测试集的结果。那么，本算法还有提升空间，那就是多次循环测试不同划分比例的训练姐和验证集准确率上升值和下降值之差，**找到剪枝之后准确率提高最多的训练集和验证集划分比例**，在同一比例下也可以**交叉验证**。

四. 参考网址

1. http://blog.csdn.net/sealyao/article/details/6530876
2. http://www.onlamp.com/pub/a/python/2006/02/09/ai_decision_trees.html?page=1
3. https://pypi.python.org/pypi/DecisionTree [内有源代码，看了之后帮助很大]

五. 感谢篇

非常感谢老师的辛勤教导，在这门课上我除了收获了很多的知识，也第一次明白了原则的重要性，从意识到自己要独立完成作业，到上课不能迟到，不然就没有额外奖励加分等好事，我突然发现大学的我貌似有些想法必须开始学着改变。再次感谢老师和助教对我的帮助和启迪~

最后，祝愿老师身体健康，工作顺利~

助教学业顺利，一切都好^_^