北京航空航天大學
BEIHANG UNIVERSITY

机器学习实验报告

实验三、决策树学习

院（系）名称	自动化科学与电气工程学院
专业名称	自动化
学生学号	15031117
学生姓名	柳天宇
指导教师	秦曾昌

2018 年 6 月 19 日

决策树是分类应用中采用最广泛的模型之一。它是一种逼近离散值目标函数的方法,决策树通过把实例从根节点排列到某个叶子结点来分类实例, 叶子结点即为实例所属的分类。 树上的每一个结点指定了对实例的某个属性的测试, 并且该结点的每一个后续分支对应于该属性的一个可能值。在决策树模型的构建过程中，计算信息熵与信息增益是关键部分。

1.知识准备

1.信息熵：在信息论中称为平均信息量，是对被传送的信息进行度量所采用的一种平均值。数学式子来表示就是:一组事件 X_1,…　,X_r，以既定概率 p(X_1),…,p(X_r)出现，其平均值 H(X)就是信息熵，它的值等于每个事件的(自)信息量 I(X)的数学期望，即:

$$H(X)=-\sum_{r=1}^{r}p(Xi)I(Xi)=-\sum_{r=1}^{r}p(X_i)\log p(X_i)$$

2.信息增益：当选择某个特征对数据集进行分类时，分类后的数据集信息熵会比分类前的小，其差值表示为信息增益。信息增益可以衡量某个特征对分类结果的影响大小。

2.决策树生成的一般性步骤：

1.从属性表中选择属性 A_i 作为分类属性；若属性 A_i 的取值有 K_i 个，则将 T 划分为 K_i 个子集 T_1,…, T_{K_i},其中 T_{ij} ={<x,C>|<x,c>}∈T,且 X 的属性取值 A 为第 K_i 个值；

2.从属性表中删除属性 A_i；

3.对于每一个 T_{ij}（1⩽j⩽ K_1），令 $T=T_{ij}$；

4.如果属性表非空，返回（1），否则输出。

3.实验内容及结果分析

3.1 决策树模型的构建

3.1.1 编写代码计算数据集中每个属性的信息熵

信息熵的计算：$$H(X)=-\sum_{r=1}^{r}p(Xi)I(Xi)=-\sum_{r=1}^{r}p(X_i)\log p(X_i)$$

```
def entropy(dataset):                          #计算 entropy
    labelcount={}                              #创建字典，记录每一类的标签，和该类别样本的数目
    numEnt=len(dataset)                        #样本总数
    for feat in dataset:
        culabels=feat[-1]                      #获取类别标签
        if culabels not in labelcount.keys():
            labelcount[culabels]=0
        labelcount[culabels]+=1                #统计每个类别样本的数目
    Ent=0.0
    for key in labelcount:
        prob=float(labelcount[key]/numEnt)     #每个类别样本占总样本的比重
        Ent-=prob*log(prob,2)                  #依据概率计算信息熵
    return Ent
```

3.1.2 选择某一数据集，计算各个属性的信息增益，选出信息量最大的属性

选择鸢尾花数据集，该数据集共采集了鸢尾花的 4 个属性：sepal length（花萼长度）、sepal width（花萼宽度）、petal length（花瓣长度）和 petal width（花瓣宽度）（单位是 cm）。

每个类别样本选取 30 个作为训练集，各属性信息增益计算结果如下：

```
sepallength 特征 的信息增益= 0.9476486514290312
sepalwidth 特征 的信息增益= 0.6128705601877552
petallength 特征 的信息增益= 1.4643541343625635
petalwidth 特征 的信息增益= 1.418078611736113
petallength 是最佳特征
>>>
```

可以看出 petallength 的信息增益最大，是对于该数据集分类的最佳特征。

3.1.3 选择合适的数据结构，从根节点到叶节点构建决策树

决策树的生成是一个递归的过程

```
ID3(Examples, DecisionAttribute, Attributes)
Examples 是训练样本，DecisionAttribute 是分类属性，Attributes 是用于分类的属性
返回可以正确分类 Examples 的决策树

创建树的根节点 Root
若所有 Examples 均为正例，则返回 Label=+的单节点树 Root
若所有 Examples 均为反例，则返回 Label=-的单节点树 Root
若 Attributes 为空，则返回单节点树 Root，Label=Examples 中最普遍的
DecisionAttribute 值
否则
    A←Attribute 中分类 Examples 能力最好(信息增益最大)的属性
    Root 的决策属性←A
    对于 A 的每个可能值 Vi
        在 Root 下加一个新的分支对应 A=Vi
        Examples←Examples 中满足 A 的属性值为 Vi 的子集
        若 Examples 为空
            为该分支添加叶子节点，节点的 Label 为 Label=Examples 中最普遍的
            DecisionAttribute 值
        否则
            在该分支下添加一棵 ID3 决策树子树(Examples, DecisionAttribute,
            Attributes-{A})
返回 Root
```

决策树构建的结果如下：


```
{'petallength': {1.5: 'setosa', 1.0: 'setosa', 3.5: 'versicolor', 4.5: {'sepallength': {6.0: 'versicolor'
, 5.7: 'versicolor', 5.6: 'versicolor', 4.9: 'virginica', 5.4: 'versicolor', 6.4: 'versicolor', 6.2: 'ver
sicolor'}}, 4.0: 'versicolor', 5.0: {'sepallength': {6.7: 'versicolor', 6.3: 'virginica', 6.0: 'virginica
', 5.7: 'virginica'}}, 3.0: 'versicolor', 6.0: 'virginica', 5.5: 'virginica', 1.2: 'setosa', 1.7: 'setosa
', 4.2: 'versicolor', 4.7: 'versicolor', 3.7: 'versicolor', 6.7: 'virginica', 5.7: 'virginica', 5.2: 'vir
ginica', 1.4: 'setosa', 1.9: 'setosa', 4.4: 'versicolor', 3.9: 'versicolor', 4.9: {'sepalwidth': {2.5: 'v
ersicolor', 3.0: 'virginica', 2.8: 'virginica', 3.1: 'versicolor', 2.7: 'virginica'}}, 5.9: 'virginica',
6.9: 'virginica', 6.4: 'virginica', 5.4: 'virginica', 1.1: 'setosa', 1.6: 'setosa', 3.6: 'versicolor', 4.
6: 'versicolor', 4.1: 'versicolor', 5.1: {'sepallength': {6.0: 'versicolor', 6.5: 'virginica', 6.3: 'virg
inica', 5.8: 'virginica', 5.9: 'virginica', 6.9: 'virginica'}}, 5.6: 'virginica', 6.6: 'virginica', 6.1:
'virginica', 1.3: 'setosa', 3.3: 'versicolor', 4.3: 'versicolor', 4.8: {'sepallength': {5.9: 'versicolor'
, 6.8: 'versicolor', 6.0: 'virginica', 6.2: 'virginica'}}, 3.8: 'versicolor', 5.8: 'virginica', 6.3: 'vir
ginica', 5.3: 'virginica'}}
```


可以看出各个特征的取值情况较多，特征的取值近似于连续，如果直接使用该模型对测试集进行分类，无法取得很好效果。

3.1.4 针对特征连续的情况，将特征离散化，完成决策树构建

属性值连续的情况：由于连续属性的可取值数目不再有限，因此，不能直接根据连续属性的可取值来对节点进行划分。此时，我们可以使用连续属性离散化的技术将连续属性转换为离散的。最简单的策略是使用二分法对连续属性进行处

理，即选取最优的划分点对样本集合进行划分。

（1）二分法对特征离散化

假设属性 a 是连续属性，将属性 a 下的值从小到大排序，有{a1,a2,a3,……an}，在 a(i)和 a(i+1)之间取平均值，作为一个划分结点，一共有 n-1 个划分结点，因此我们以这些结点把数据集分为两个子集，分别计算在该结点下属性 a 的信息增益，计算结果有 n-1 个值，在 n-1 个值之中选取信息增益最大的值，以这个值的划分点作为把连续属性 a 变成 2 个类离散化的节点。

此时信息增益的计算如下：

$$\text{Gain}(\text{D},\text{a}) = \max_{t\epsilon T_a} Gain(D,a,t) = \max_{t\epsilon T_a} Ent(D) - \sum_{\lambda\epsilon\{-,+\}} \frac{|D_t^\lambda|}{|D|} Ent(D_t^\lambda)$$

例如在鸢尾花数据集分类时，由于 petallength 这个特征在训练集中的取值情况较多，可视作连续的变化的特征。因此考虑在根节点处采用二分法，在每个取值下的信息增益计算结果如下：

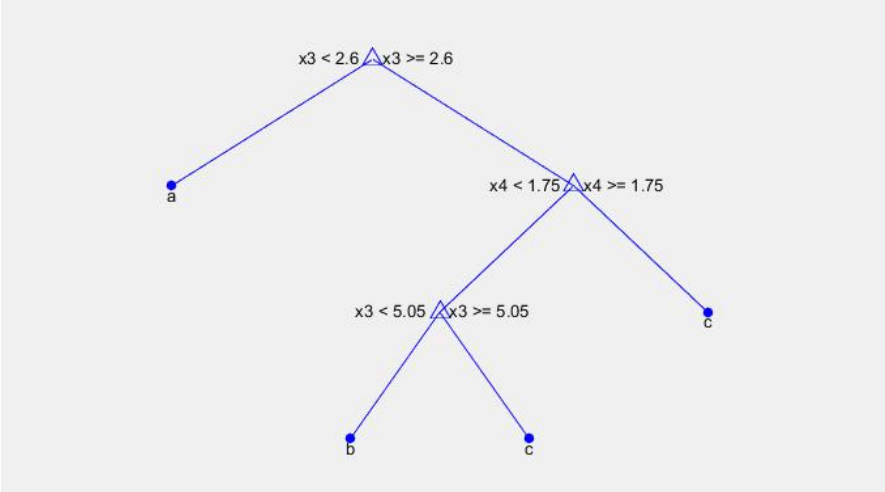
```
petallength 是取连续特征
当前划分值为：1.05，此时的信息增益为：-3.9179275211332967
当前划分值为：1.15，此时的信息增益为：-3.8725169134988375
当前划分值为：1.25，此时的信息增益为：-3.8261301623309603
当前划分值为：1.35，此时的信息增益为：-3.7302330303007065
当前划分值为：1.45，此时的信息增益为：-3.2956183221039033
当前划分值为：1.55，此时的信息增益为：-2.7373729657833823
当前划分值为：1.65，此时的信息增益为：-2.373078656552823
当前划分值为：1.8，此时的信息增益为：-1.871153940411804
当前划分值为：2.6，此时的信息增益为：-1.6666666666666667
当前划分值为：3.4，此时的信息增益为：-1.8155872483091962
当前划分值为：3.55，此时的信息增益为：-1.9830403041425284
当前划分值为：3.75，此时的信息增益为：-2.0433539952654964
当前划分值为：3.95，此时的信息增益为：-2.1366365114056878
当前划分值为：4.05，此时的信息增益为：-2.2289708985646683
当前划分值为：4.15，此时的信息增益为：-2.2499626305860763
当前划分值为：4.25，此时的信息增益为：-2.266739118167698
当前划分值为：4.35，此时的信息增益为：-2.2795526672276893
当前划分值为：4.45，此时的信息增益为：-2.293956570926286
当前划分值为：4.55，此时的信息增益为：-2.266739118167698
当前划分值为：4.65，此时的信息增益为：-2.2289708985646683
当前划分值为：4.75，此时的信息增益为：-2.0938946911559464
当前划分值为：4.85，此时的信息增益为：-2.3251811052193383
当前划分值为：4.95，此时的信息增益为：-2.5115894646085644
当前划分值为：5.05，此时的信息增益为：-2.4724111426894413
当前划分值为：5.15，此时的信息增益为：-2.8171415139100815
当前划分值为：5.3，此时的信息增益为：-2.9667115099633747
当前划分值为：5.45，此时的信息增益为：-3.105087344756242
当前划分值为：5.55，此时的信息增益为：-3.170683531015338
当前划分值为：5.65，此时的信息增益为：-3.4695069432425174
当前划分值为：5.75，此时的信息增益为：-3.6298031132214543
当前划分值为：5.85，此时的信息增益为：-3.680614659988805
当前划分值为：5.95，此时的信息增益为：-3.7302330303007065
当前划分值为：6.05，此时的信息增益为：-3.778719458227803
当前划分值为：6.25，此时的信息增益为：-3.8725169134988375
当前划分值为：6.55，此时的信息增益为：-3.9179275211332967
-1.6666666666666667
```

以 2.6 为划分点可以取得最大的信息增益，故可以将 petallength 特征划分为大于 2.6 和小于 2.6 两类。此外，若二分法对于分类准确率影响较大，可以考虑

将特征的取值划分为若干个小区间。

（2）使用二分法构建决策树

特征均按照二分法作离散化后结果如下图所示：



a:setosa
b:versicolor
c:virginica

X1:sapallength;

X2:sapalwidth;

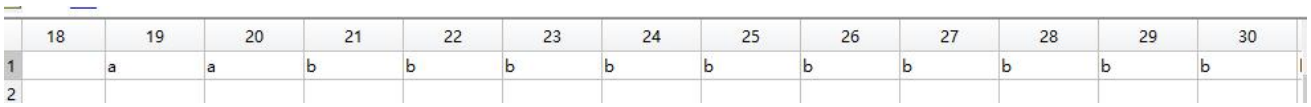
X3:petallength;

X4;petalwidth;

（3）测试集表现

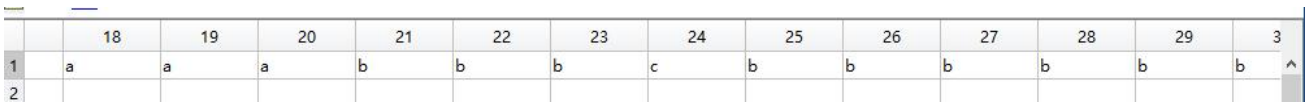
使用该决策树对于测试集中 60 组样本的类别进行预测，与真实类别进行比对得到如下结果：

预测类别：



	18	19	20	21	22	23	24	25	26	27	28	29	30
1		a	a	b	b	b	b	b	b	b	b	b	b
2													

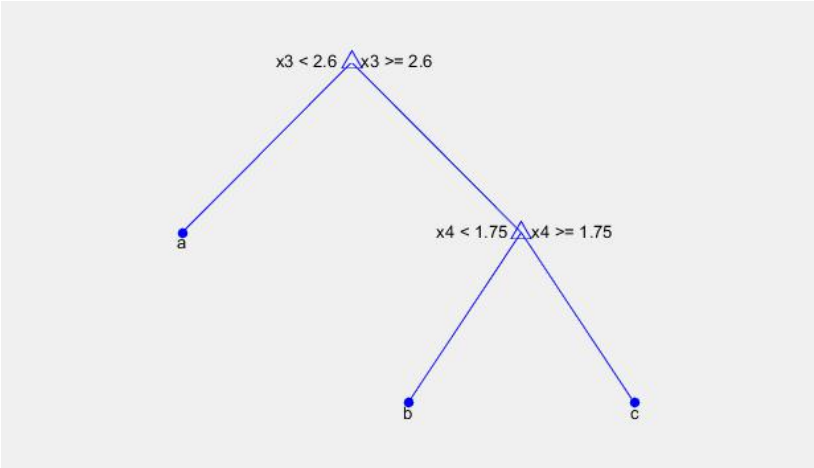
真实类别：



	18	19	20	21	22	23	24	25	26	27	28	29	3
1	a	a	a	b	b	b	c	b	b	b	b	b	b
2													

60 个测试样本中，57 个可正确分类

对决策树进行剪枝处理，去除最后一层：



测试集上表现如下：

预测类别：

20	21	22	23	24	25	26	27	28	29	30	31	32
a	b	b	b	b	b	b	b	b	b	b	b	b

测试类别：

20	21	22	23	24	25	26	27	28	29	30	31	32
a	b	b	b	b	b	b	b	b	b	b	b	b

60 个测试样本中，58 个可正确分类。可见剪枝处理后，分类器性能有所提高。

3.2.决策树模型在实际使用时的一些问题

3.2.1 决策树的分枝数目与模型的精度

增加决策树的分支数目与层数，即使决策树的规模更加庞大可以提高模型在训练集上的表现，使得模型对于训练集样本的拟合效果更好，但在学习的过程中为了尽可能的正确的分类训练样本，不停地对结点进行划分，因此这会导致整棵树的分支过多，也就导致了过拟合。训练集不能够完全等同于所有真实数据，二者之间在特征上可能存在差异性，如果过分地追求模型在训练集上的表现，就相当于把训练集自身的一些独有的特点当成所有数据共同的特性，尤其是当训练集中个别的数据含有噪声时，过拟合对于数据泛化后的性能影响是非常不利的，树的每次分裂都减少了数据集，决策树当数据集很小使预测结果的准确性也会下降。这时我们应该考虑主动去掉一些分支来降低过拟合的程度，提高决策树的数据泛化性能。

模型本身的精确度和模型的稳定性分别对应着偏差和方差。实际中，很难做

到同时提高模型的精度和稳定性。降低偏差的同时就一定程度上提高了方差。但选择合适的分支数目和层数能够使偏差和方差综合造成的不利影响尽可能小。如果想要同时提高稳定性和精度，在选择合适模型的同时，还需要提高训练集数据的真实性，即训练数据与真实数据的接近程度，就是说分类器的分类效果也会受到训练集数据本身的限制。

综合考虑稳定性和精度，在保证训练集上分类准确率的前提下，决策树的分支数和层数不应过多，即当增加分支数与层数准确率的提升已经不明显时应该停止增加模型复杂度。

3.2.2 使用多特征改善决策边界形状

如果把样本的每个属性视为坐标空间的一个坐标轴，单变量决策树所形成的分类边界均是与坐标轴平行的，即是由若干个与坐标轴平行的线段组成的。如果针对于复杂问题进行分类，真实的分类边界往往也比较复杂，而单变量决策树边界的形状较为单一，这样形成的决策树将会很复杂。如果考虑使用多种特征的组合作为分类依据，形成的决策边界在特征空间上就具有了更多的形状，而不仅限于平行于坐标轴的之间，这样使得决策边界形状更加平滑，拟合真实边界的能力更强。

4. 问题讨论：决策树与朴素贝叶斯的对比

（1）贝叶斯分类器的设计需要计算先验概率，而先验概率往往需要一定专业知识，先验概率的准确值并不容易获取。相比朴素贝叶斯分类，决策树的优势在于构造过程不需要任何领域知识或参数设置，因此在实际应用中，对于探测式的知识发现，决策树更加适用。

（2）与贝叶斯方法相比，决策树无须花费大量的时间和进行上千次的迭代来训练模型，适用于大规模数据集，除了训练数据中的信息外不再需要其他额外信息，表现了很好的分类精确度。

（3）当数据真实边界很光滑时，单变量的决策树不容易拟合出决策边界。当特征取值不连续时，决策树性能更好。

（4）如果有很多不相关的变量，决策树表现的不好。决策树的工作是通过找到变量间的相互作用。而朴素贝叶斯是基于特征相互独立的假设，在此种情形下非

常适用。

（5）决策树易于理解和实现，能够直接体现数据的特点，只要通过解释后都有能力去理解决策树所表达的意义。

5.总结

本次实验使用决策树的方法完成了鸢尾花数据集的分类，深化了对于决策树这种机器学习经典算法的理解。相比于其他算法，决策树的结果具有决策树在处理连续变化特征时，必须要对特征进行离散化处理；同时，决策树对于平滑的决策边界拟合性能不佳；此外，在决策树的构建过程中要注意枝叶的修剪否则容易出现过拟合的情况。这说明不同的机器学习方法都有其适合与不适合的场景。我们在面对不同的数据集时，应注意地合理选择学习方法，对于具体问题往往还需对于经典算法进行优化。

Github 网址：https://github.com/ECHOliuty/PR-report