中 国 地 质 大 学

本科生课程论文

课 程 名 称 模式识别 A

教 师 姓 名 蒋良孝

学 生 姓 名 常文瀚

学 生 学 号 20181001095

学 生 班 级 191181

所 在 学 院 计算机学院

完 成 日 期 2021 年 6 月 20 日

目录

一、题目描述

1、主题

复现 ID3 和 C4.5 经典决策树算法，并研究剪枝对决策树分类精确度的影响。

2、开发环境

开发语言：Java

开发平台：Weka、IDEA

二、决策树算法

决策树是一种常用的数据挖掘方法，是一个类似流程图的树型结构。决策树包含三个元素：根结点、内部结点和叶子结点。若要对未知的数据对象进行分类，可以按照决策树的数据结构对数据集中的属性(取值)进行测试，从决策树的根结点到叶结点的一条路径就代表了对相应数据对象的类别预测。决策树是一种分而治之(divide-and-conquer)的决策过程，形成决策树的决策规则有许多，如信息增益，信息增益比，基尼系数等。

1、ID3 算法

ID3 算法的基本思想是：首先在决策树的各级结点上，选择信息增益最大的属性作为分类结点，根据该属性的不同取值分裂出各个子结点，随后采用递归的方法建立决策树的分支，直到样本集中只含有一种类别时停止，得到最终的决策树。

2、信息增益

要想学习理解 ID3 以及一些决策树训练算法首先就要引入“熵”这个概念，熵的度量可以分为等概率均匀分布、不等概率一般分布。

对于信息熵，我认为它是熵的一个具体化或者实例化，或许也可以把它理解成子类继承父类，信息熵的性质和“熵”的概念是相似的，信息熵公式如下。

$$Entropy(D) = -\sum_{k=1}^{|y|} P_k \, log_2 \, P_k$$

信息熵是代表随机变量的复杂度（不确定度），条件熵代表在某一个条件下，随机变量的复杂度（不确定度）。而我们的信息增益恰好是：信息熵-条件熵。换

句话说，信息增益代表了在一个条件下，信息复杂度（不确定性）减少的程度。那么我们现在也很好理解了，在决策树算法中，我们的关键就是每次选择一个特征，特征有多个，那么到底按照什么标准来选择哪一个特征。这个问题就可以用信息增益来度量。如果选择一个特征后，信息增益最大（信息不确定性减少的程度最大），那么我们就选取这个特征，信息增益计算公式如下。

$$Gain(D,a) = Entropy(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|} Entropy(D^v)$$

3、C4.5 算法

C4.5 算法是对 ID3 算法的改进算法，具有 ID3 算法所有的优点，如分类规则易于理解、算法复杂度较低等。C4.5 算法用信息增益比来选择特征，通过递归地对变量进行特征选择，然后用最优特征分割数据集，这个过程持续到所有实例中的子集都落在同一个类中。

4、信息增益率

C4.5 算法使用信息增益率进行特征选择，对 ID3 算法进行优化，计算信息增益率同样需要计算信息增益，但此外还需要计算属性分类信息度量，用分类信息度量来考虑某种属性进行分裂时分支的数量信息和尺寸信息，我们把这些信息称为属性的内在信息。信息增益率用信息增益/内在信息，会导致属性的重要性随着内在信息量的增大而减小（也就是说，如果这个属性本身不确定性就很大，那我就越不倾向于选取它），这样算是对单纯用信息增益有所补偿，分类信息度量公式如下：

$$H(a) = -\sum_{v=1}^{V} \frac{|D^v|}{|D|} log_2 \frac{|D^v|}{|D|}$$

结合信息增益，信息增益率计算公式如下：

$$GainRatio(D,a) = \frac{Gain(D,a)}{H(a)}$$

三、剪枝—防止过拟合的优化方法

在决策树生成的过程中，有些节点分类是意义不大的，而这些节点的存在不仅增加复杂度，而且减少了分类器的正确率。 剪枝算法就是减除决策树中意义不大的节点来从而减小决策树的大小。剪枝算法可以有效缓解过拟合问题，从而提高了预测准确性。

1、预剪枝

预剪枝就是在构造决策树的过程中，先对每个结点在划分前进行估计，若果当前结点的划分不能带来决策树模型泛华性能的提升，则不对当前结点进行划分并且将当前结点标记为叶结点。

2、后剪枝

后剪枝就是先把整颗决策树构造完毕，然后自底向上的对非叶结点进行考察，若将该结点对应的子树换为叶结点能够带来泛华性能的提升，则把该子树替换为叶结点。

本文中实现的是后剪枝中的 Reduced-Error Pruning (REP 错误率降低剪枝)，这个思路很直接，完全的决策树过度拟合，我再使用一个测试数据集来纠正它。对于完全决策树中的每一个非叶子节点的子树，我们尝试着把它替换成一个叶子节点，该叶子节点的类别我们用子树所覆盖训练样本中存在最多的那个类来代替，这样就产生了一个简化决策树，然后比较这两个决策树在测试数据集中的表现，如果简化决策树在测试数据集中的错误比较少，那么该子树就可以替换成叶子节点。该算法以 bottom-up 的方式遍历所有的子树，直至没有任何子树可以替换使得测试数据集的表现得以改进时，算法就可以终止。

四、经典决策树算法的复现

1、ID3 算法的复现

在对 ID3 的复现中调用了 Weka 接口，生成了 Weka ID3 对象，与自己复现的 ID3 决策树进行对比，实验结果表明两个决策树性能一致，算法实现正确（数据集为：weather.nominal.arff）。

```
===== Id3 Result =====
Correctly Classified Instances          12               85.7143 %
Incorrectly Classified Instances         2               14.2857 %
Kappa statistic                          0.6889
Mean absolute error                      0.1429
Root mean squared error                  0.378
Relative absolute error                 30      %
Root relative squared error             76.6097 %
Total Number of Instances               14
```

图（1） Weka ID3 算法分类结果

```
===== MyId3 Result =====
Correctly Classified Instances          12               85.7143 %
Incorrectly Classified Instances         2               14.2857 %
Kappa statistic                          0.6889
Mean absolute error                      0.1429
Root mean squared error                  0.378
Relative absolute error                 30      %
Root relative squared error             76.6097 %
Total Number of Instances               14
```

图（2） 复现 ID3 算法分类结果

```
outlook = sunny
|  humidity = high: no
|  humidity = normal: yes
outlook = overcast: yes
outlook = rainy
|  windy = TRUE: no
|  windy = FALSE: yes
```

图（3） ID3 算法生成的决策树

2、C4.5 算法的复现

```
===== J48 Result =====
Correctly Classified Instances           7               50      %
Incorrectly Classified Instances         7               50      %
Kappa statistic                         -0.0426
Mean absolute error                      0.4167
Root mean squared error                  0.5984
Relative absolute error                 87.5    %
Root relative squared error            121.2987 %
Total Number of Instances               14
```

图（4） Weka C4.5 算法分类结果

```
===== My J48 Result =====
Correctly Classified Instances           8               57.1429 %
Incorrectly Classified Instances         6               42.8571 %
Kappa statistic                         -0.1351
Mean absolute error                      0.4286
Root mean squared error                  0.6547
Relative absolute error                 90      %
Root relative squared error            132.6919 %
Total Number of Instances               14
```

图（5） 复现 C4.5 算法分类结果

五、剪枝的实现

1、算法说明

在实现剪枝的过程中，我参考了很多资料最终选择了使用保险公司基准 (COIL 2000) 数据集，并选择实现了后剪枝策略。我首先在没有任何停止策略的情况下生成了一棵决策树，该决策树在训练集上的精确度达到 100%，出现过拟合现象，在这之后进行剪枝。

2、实验结果

```
Attribute used for splitting at root = 0
Height of the tree       = 7
Accuracy on test set     = 87.86810302734375
Accuracy on training set = 100.0
```

图（6） 生成的过拟合决策树

```
----------------------Pruning starts-------------------

Prediction accuracy on validation set = 88.13770294189453
Prediction accuracy on validation set = 88.44877624511719
Prediction accuracy on validation set = 88.84280395507812
Prediction accuracy on validation set = 89.2990493774414
Prediction accuracy on validation set = 89.31978607177734
Prediction accuracy on validation set = 90.06636047363281
Prediction accuracy on validation set = 90.48112487792969
Prediction accuracy on validation set = 90.85441589355469
Prediction accuracy on validation set = 90.93737030029297
Prediction accuracy on validation set = 91.35213470458984
Prediction accuracy on validation set = 91.6632080078125
Prediction accuracy on validation set = 92.0364990234375
Prediction accuracy on validation set = 92.32683563232422
Prediction accuracy on validation set = 92.45126342773438
Prediction accuracy on validation set = 92.5549545288086
Prediction accuracy on validation set = 93.19783782958984
Prediction accuracy on validation set = 93.3637466430664
Prediction accuracy on validation set = 93.75778198242188
Prediction accuracy on validation set = 93.79924774169922
Prediction accuracy on validation set = 94.08959197998047

Accuracy after pruning on test set = 94.08959197998047
Accuracy after pruning on training set = 93.69999694824219
```

图（7） 剪枝过程与剪枝结果

可以通过实验结果看到，在经过 REP 错误率降低剪枝后，决策树在测试集上的精确度不断升高，最终由 88.13%升至 94.08%，而其在训练集上的精确度也从 100%降到了 93.69%，与其在测试集上的精确度基本持平，我们可以认为剪枝使该决策树的泛化性能得到了提升。

六、总结与体会

经过模式识别这一课程的学习，我初步的了解了机器学习的历史以及相关的十大经典算法，每一个算法都能引起我非常大的兴趣，在对课堂上收获的知识进行总结后，我也认识到单单从课堂上获取知识是不够的，必须辅以实践，或者更多资料的阅读，才能更加深刻的理解知识。正如这次结课作业让我对于如何生成一棵决策树，并消除其过拟合现象有了更加清晰的理解，希望在今后的学习中，我能够带着更大的热情去探究课堂上学习到的内容。

七、引用与参考

[1]邵旻晖.决策树典型算法研究综述[J].电脑知识与技术,2018,14(08):175-177.

[2]章晓. 决策树 ID3 分类算法研究[D].浙江工业大学,2014.

[3]张小轩. ID3 算法的研究及优化[D].山东科技大学,2017.