读书报告

09118119 黄一凡

# 自己提出的问题

#### 1.在计算entropy时为什么需要用log，以及log的底数是否需要和最后类的数量相同？

这里使用log是在entropy公式中定义的：

不需要，我认为这里log的底数对结果无实际影响，是log即可。对于不同的底数，总是在相同条件下达到最大值：

因此，log的底数对的计算没有影响。

#### 2.为什么一个attribute的值越多，其增益就越大，感觉好像是这样的，是否有数学证明？

在《Web Data Mining》中对这个问题的描述为：The gain criterion tends to favor attributes with many possible values. 这里我的理解是，这只是一个趋势，并不是充要条件。可以举反例来证明，例如我们并不能说任意有三个值的attribute的增益 一定比有两个值的大。

而这一趋势在解决问题时也不是完全不用考虑，比如一个attribute中的每个值互不相同，那么这就需要用到gainRatio来解决。

# 别人提出的问题

#### 1. 为避免rule pruning后一个例子符合多条rule而对rule进行排序，此时如何排序？按照什么准则？

这里的顺序的作用是避免某一个data同时符合几条rule而不知道该归为哪一类的情况。因此，只需随机确定一个固定的顺序即可。

在讨论过程中，有同学提出了是否可以按照rule的support和confidence值来进行排序，考虑到经过rule pruninng后，很大一部分 rule的support和confidence值是未知的，需要重新遍历整个数据集来进行计算，增加了算法的复杂度，不知道是否有必要这样来 排序，需要后续进一步的研究。

#### 2.以cross-validation为例，按理来说k次交叉验证后应得到的k个模型分别有不同的accuracy， 那取这k个accuracy的平均值有何意义？它并不能代表k个模型中的任意一个。

Cross-validation的目的是评价某一种分类算法在此数据上的表现如何，由此来决定使用哪一种算法。尽管k次交叉验证后生成了 k个不同的模型，但是这k个模型都是基于决策树算法生成的，只是其中某些参数不同。

因此，平均值代表的是由某个算法生成的k个模型的准确度，评价的对象是这个算法，而不是每一个模型。

#### 3.比起单纯看模型的accuracy来说，计算recall和precision可以提供哪些额外的信息来进一步评估模型？

Recall和precision可以更加全面、准确的来对模型进行评估。比如在一个包含10000个数据的数据集，有9900个为正，100个为负， 那么一个算法只需简单地将所有数据判断为正便可以得到很高的准确率，而这很明显是我们不想看到的。如果采用recall和precison 就可以很好得避免这个问题，因为它们可以只考虑某一类数据的准确度。

# 读书计划

#### 本周所读：

3.1-3.3

#### 下周计划：

4.1-4.3

# 读书摘要

下面是我读书时做的一些笔记整理：

# 3 Supervised Learning

In this chapter, we focus on one particular type of supervised learning, that is, learning a target function that can be used to predict the values of a discrete class attribute.

This chapter introduce a number of such supervised learning techniques.

## 3.1 Basic Concepts

Step 1: a learning algorithm uses the training data to generate a classification model

Step 2: the learned model is tested using the test set to obtain the classification accuracy.

- If the accuracy of the learned model on the test data is satisfactory, the model can be used in real-world tasks to predict classes of new cases

- If the accuracy is not satisfactory, we need to go back and choose a different learning algorithm and/or do some further processing of the data

A practical learning task typically involves many iterations of these steps before a satisfactory model is built.

From the next section onward, we will study several supervised learning algorithms.

## 3.2 Decision Tree Induction

There are many possible trees that can be learned from one single data.

In parctice, a small and accurate tree is prefered.

Each leaf node of the decision tree gives a class value, which is the output. below each class means that x out of y training examples that reach this leaf node have the class of the leaf.

For most real-life data sets, :

Indeed, a decision tree can be converted to a set of if-then rules.

- Each path from the root to a leaf forms a rule

- All the decision nodes along the path form the conditions of the rule

- The leaf node or the class forms the consequent

- In most classification systems, support and confidence values are not provided

- A decision tree only finds a subset of rules that exist in data, which is sufficient for classification

The problem becomes building the best tree that is small and accurate. However, finding the best tree is a NP-complete problem.

### 3.2.1 Learning Algorithm

The learning of a tree is typically done using the divide-and-conquer strategy that recursively partitions the data to produce the tree.

To partition the data, we have to select the best attribute by which the maximum purity is obtained.

A function called impurity function can help finding the best attribute, which is the key in decision tree learning.

This is a greedy algorithm with no backtracking. Once a node is created, it will not be revised or revisited no matter what happens subsequently.

### 3.2.2 Impurity Function

The most popular impurity functions used for decision tree learning are information gain and information gain ratio. Let us first discuss information gain, which can be extended slightly to produce information gain ratio.

The information gain measure is based on the entropy function from information theory:

is the probability of class in dataset , which is the number of examples of class in divided by the total number of examples in .

In the entropy computation, we define

When the data becomes purer and purer, the entropy value becomes smaller and smaller. The entropy measures the amount of impurity or disorder in the data.

#### Information Gain

1. Given a data set , we first compute the impurity value of , , by impurityEval-1 function.
2. To find out which attribute can reduce the impruity most, each attribute is evaluated. Let the number of possible values of the attribute be . The entropy after the partition by is, by impurityEval-2 function:
3. The information gain of attribute is computed with:
4. The attribute with largest gain is then selected to partition the data.

#### Information Gain Ratio

The gain criterion tends to favor attributes with many possible values, sometimes this will lead to some problems.

This method works because if Ai has too many values the denominator will be large.

### 3.2.3 Handling of Continuous Attributes

To apply the decision tree building method, we can divide the value range of attribute Ai into intervals at a particular tree node. Each interval can then be considered a discrete value. Based on the intervals, gain or gainRatio is evaluated in the same way as in the discrete case.

Clearly, we can divide Ai into any number of intervals at a tree node. However, two intervals are usually sufficient. We need to find a threshold value for the division.

We should choose the threshold that maximizes the gain(or gainRatio)

- This is not a problem because although for a continuous attribute Ai the number of possible values that it can take is infinite, the number of actual values that appear in the data is always finite

- The threshold value that maximizes the gain(gainRatio) value is selected

- For a continuous attribute, we do not remove attribute A because an interval can be further split recursively in subsequent tree extensions

- Handling of continuous (numeric) attributes has an impact on the efficiency of the decision tree algorithm

### 3.2.4 Some Other Issues

#### Tree Pruning and Overfitting

Overfitting is usually caused by noise in the data. It may also be due to the complexity and randomness of the application domain.

To reduce overfitting in the context of decision tree learning, we perform pruning of the tree, i.e., to delete some branches or sub-trees and replace them with leaves of majority classes.

There are two main methods to do this, stopping early in tree building (which is also called pre-pruning) and pruning the tree after it is built (which is called post-pruning).

Postpruning has been shown more effective.

There are three ways to perform The general idea of post-pruning is to estimate the error of each tree node. If the estimated error for a node is less than the estimated error of its extended sub-tree, then the sub-tree is pruned.