**Experiment 3 Decision Tree Learning for Classification**

**3.1 Introduction**

Decision tree induction is one of the simplest and yet most successful learning

algorithms. A decision tree (DT) consists of internal and external nodes and the interconnections between nodes are called branches of the tree. An internal node is a

decision-making unit to decide which child nodes to visit next depending on different

possible values of associated variables. In contrast, an external node also known as a

leaf node, is the terminated node of a branch. It has no child nodes and is associated

with a class label that describes the given data. A decision tree is a set of rules in a tree

structure, each branch of which can be interpreted as a decision rule associated with nodes visited along this branch.

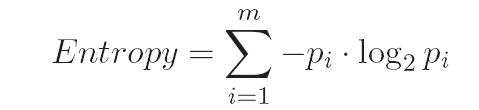
**3.2 Principle and Theory**

Decision trees classify instances by sorting them down the tree from root to leaf nodes.This tree-structured classifier partitions the input space of the data set recursively into mutually exclusive spaces. Following this structure, each training data is identified as

belonging to a certain subspace, which is assigned a label, a value, or an action to characterize its data points. The decision tree mechanism has good transparency in that we can follow a tree structure easily in order to explain how a decision is made. Thus interpretability is enhanced when we clarify the conditional rules characterizing the tree.

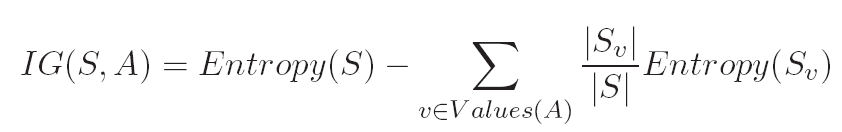
Entropy of a random variable is the average amount of information generated by observing its value. Consider the random experiment of tossing a coin with probability of heads equal to 0.9, so that P(Head) = 0.9 and P(Tail) = 0.1. This provides more information than the case where P(Head) = 0.5 and P(Tail) = 0.5. Entropy is used to evaluate randomness in physics, where a large entropy value indicates that the process is very random. The decision tree is guided heuristically according to the information content of each attribute. Entropy is used to evaluate the information of each attribute; as a means of classification. Suppose we have *m* classes, for a particular attribute, we denoted it by pi by the proportion of data which belongs to classCiwhere i = 1, 2, … m.

The entropy of this attribute is then:



We can also say that entropy is a measurement of the impurity in a collection of training examples: larger the entropy, the more impure the data is. Based on entropy, Information Gain (IG) is used to measure the effectiveness of an attribute as a means

of discriminating between classes.



where all examples *S* is divided into several groups (i.e. *Sv* for *v* ∈ *Values(A)*) according to the value of *A*. It is simply the expected reduction of entropy caused by partitioning the examples according to this attribute.

**3.3 Objective**

The goals of the experiment are as follows:

(1) To understand why we use entropy-based measure for constructing a decision

tree.

(2) To understand how Information Gain is used to select attributes in the process

of building a decision tree.

(3) To understand the equivalence of a decsion tree to a set of rules.

(4) To understand why we need to prune the tree sometimes and how can we

prune? Based on what mesure we prune a decsion tree.

(5) To understand the concept of Soft Decsion Treees and why they are imporant

extensions to classical decision trees.

**3.4 Contents and Procedure**

**Stage 1:**

(1) According to the above principle and theory in section 3.2, implement the

code to calculate the information entropy of each attribute.

(2) Select the most informative attribute from a given classification problem (e.g.,

we will be given the Iris Dataset from the UCI Machine Learning Repository)

(3) Find an appropriate data structure to represent a decion tree. Building the tree

from the root to leaves based on the principal discussed in section 3.2 by

using Information Gain guided heuritics.

**Stage 2:**

(1) Now consider the case of with continuous attributes or mixed attributes (with

both continuous and discrete attributes), how can we deal with the decision

trees? Can you propose some approaches to do discretization?

(2) Is there a tradeoff between the size of the tree and the model accuracy? Is

there existing an optimal tree in both compactness and performance?

(3) For one data element, the classical decsion tree gives a hard bounday to

decide which branch to follow, can you propose a “soft approach” to increase

the robustness of the decision tree?

(4) Compare to the Naïve Bayes, what are the advantages and disvantages of the

decision tree learning?

**Stage 3**：

Explore the questions in the previous section and design experiments to answer

these questions. Complete and submit an experiment report about all experiment results with comparative analysis and a summary of experiences about this experiment study.

**3.5 Results and Report**

**Stage 1:**

I chose the Iris Dataset from the UCI Machine Learning Repository as my train dataset to complete this experiment. At First, I code some functions to compute the Entropy and IG(Information Gain). Then, because all attributes is continuous, I chose binary tree to generate my decision tree, in order to simplify the process of generating. I will mention it later.

As you can see, for the train dataset, the accuracy of prediction equals 100 percent. But the tree has so many nodes (19 nodes), and some nodes contains just few cases in the dataset. The size of the tree should have been smaller. So I use pruning to reduce the size of the decision tree.

**Stage 2:**

1. For discrete attributes, it’s easy to compute the IG of them, and select one of which IG is the biggest.

For continuous attributes,if the dataset is not too big, we can find the gap between the values of the same attributes. For example, in the Iris Dataset, when we create the root node, we need to select the best attribute and the best “pivot” value in order to create 2 child node. At First, we can select the first attribute “sepal\_length”。 We can **sort** the values of all 150 case and then compute 149 gaps.



So we can find IG of every gap, and find the best gap as the “pivot” value. Then, we need to find the best gaps of 3 other attributes. Compare the 4 values of IG，finally we can find the best attribute and the best “pivot” value.

If the dataset is too big to compute like this, we can use another way yo find the “gaps”. We still need to sort all cases in dataset by the value of one attribute, and then we can pay attention to the classes between them. For example, we have 6 cases in dataset. They belong to 2 class(A, B). The sorted dataset can be expressed like [A,C,C,C,A,A]. We can get 2 gaps:



1. Yes. Usually, the bigger the size of decision tree, the accuracy of the model. By the Pruning Algorithm, we can find a better tree (rather than an optimal tree), in both compactness and performance.There are many methods in the Pruning Algorithm. I will show you one of them that has been used in my experiment.

At First, we can defined a variable ɑ(ɑ∈(0,1]) . In one non-leaf node, before we create 2 child nodes for it, we can find the class that has the most number of cases in this node. If the proportion of them is bigger than ɑ, we set this node as a leaf node and let its class be the class we find. If not, we create two child node for it. In this way, we can reduce the size of the tree a lot, but only reduce the accuracy a little.

In this experiment, when I let ɑ = 0.92, I realized the size of decision tree become smaller a lot, but the accuracy become smaller a little.

**Before Pruning:**



**After Pruning:**



1. Classic Decision Tree use branch to decide which class one case belongs to.We can use possibility in every node. For example, there are one case in one node and this node has two child node. According to classic decision tree, this case belongs to the left child node.Now we compute the possibilities that it belongs to the two child node. We can let the possibility that it belongs to the left child node be the accuracy of the left child tree, and let another possibility be (1- the accuracy of the right child tree). In this way, one case will travel every branch in the tree. We can find a leaf node which has the most biggest possibility as final result.
2. Advantages: When there are many attributes, decision Tree can forecast faster. The more cases in the dataset, the bigger accuracy the decision tree has. On the other hand, Naïve Bayes are not concern about the amount of cases in dataset, and the accuracy is stable.

disadvantage: It will be over-fitting, Especially when the amount of cases is not big enough. It will take more time to generate the tree and do pruning.