Calculating certainty in decision trees: Certainty vs Entropy.

Alejandro Penate-Diaz

In dedication to my parents, whose unconditional love made this research possible.

Abstract

This research introduces a certainty function, to replace the entropy function used to calculate information gain when building decision trees, in algorithms like ID3[1], C4.5[2] and C5.0. The new function is simpler than the entropy function and calculates certainty more accurately, potentially consuming less time in the process.

I will also introduce another formula that help deal with causal analysis. It is the certainty-raising inequality. This formula can be derived from the original certainty function, and provide insights into the cause-effect relations in the data.

Keywords: ID3, C4.5, C5.0, entropy, certainty, uncertainty, decision tree, splitting criteria, purity, impurity.

Calculating certainty in decision trees: Certainty vs Entropy.

The creation of decision trees and respective inductive rules for a given dataset, is based on the calculation of information gain of its non-class attributes, with respect to the class attribute, and the selection of the ones that provide more information gain, in order to split the dataset into smaller sets containing the same value of the attribute used to split it. There are several algorithms based on this technique like the ID3 [1], C4.5 [2] and C5.0. Usually the calculation of the entropy of attributes is involved, and subsequently the calculation of information gain, which is the difference in entropy before and after the dataset is split.

The conditional entropy of attribute value ***j*** is described by the function:

$$H\left(C|a_j\right) = -\sum_{i=1}^{N} p\left(c_i|a_j\right)*log_2 p\left(c_i|a_j\right)$$

Where N is the number of values of the class attribute, C is the class attribute and A is the current attribute.

And the total conditional entropy of the attribute is calculated by:

$$H\left(C|A\right) = \sum_{j=1}^{M} p\left(a_j\right)*H\left(C|a_j\right)$$

This method has proved effective so far in that it creates small trees and overall good predictive accuracy.

In this paper, I will describe the use of another function that can be used as a splitting criterion. It is, unlike entropy, a purity function, and will call it certainty function for the sake of clarity. The conditional certainty function could be described as:

$$I\left(C|a_j\right) = \sum_{i=1}^{N} \left| p\left(c_i \cap a_j\right) - \frac{p\left(a_j\right)}{N} \right|$$

Which reads as the summation of the absolute value of the joint probability of c_i and a_j, minus the average probability of a_j.

And it is used to calculate the certainty of attribute value *j* of attribute *A* respect to class attribute *i* value.

The idea behind the formula is that above and below the average is considered certainty and added, so being $p\left(a_j\right)$ the total probability for the attribute value *j*, which is distributed for the *N* values of class attribute *C*, it follows that $p\left(a_j\right)/N$ is the average probability.

And the total attribute certainty is the sum of the certainty of its values:

$$I(C|A) = \sum_{j=1}^{M} I\left(C|a_j\right)$$

Where M is the number of values of the attribute A.

The key insights that led me to this function was the realization that instead of minimizing uncertainty, we should maximize certainty. And also, that conditional probability was not required, the joint probability would suffice, although later I have found that this is the equivalent to doing the following:

$$I\left(C|a_j\right) = \sum_{i=1}^{N} \left| p\left(c_i|a_j\right) - \frac{1}{N} \right|$$

$$I\left(C|A\right) = \sum_{j=1}^{M} p\left(a_j\right) * I\left(C|a_j\right)$$

So, this could be called “conditional certainty”.

It can be shown that this function reaches a minimum of zero when the classes in all the examples are equally distributed, and conversely reaches a maximum of 1 when there are two classes and all the examples belong to the same class. When there are more than two classes the maximum can be greater than 1, behaving in this way similarly to the Entropy criteria but the other way around. Thus, this is a purity function instead of the more common impurity functions, such as Entropy and Gini. It is easy to see that when building decision trees what we need to find is purity instead of impurity because we need to determine the most important attributes, not the least important ones. It seems that Quinlan realized this and turned the Entropy into Information Gain and subsequently Gain Ratio, which are measures that we want to maximize instead of minimizing, while keeping the entropy around. The certainty function is meant to eliminate the need for impurity functions.

This way we can select the attribute with greater conditional certainty as the root of our tree. And the root attribute can be considered the main cause of the class attribute. For example, the attribute “pressure" would be the main cause of class attribute “storm”. More about causality analysis in the next section.

The formal definition of the function is as follows:

Let *X* be a discrete random variable on a finite set $X = \{x_1, \ldots, x_n\}$, with probability distribution function *p(x) = Pr(X=x)*. The certainty *I(X) of X* is defined as:

$$I(X) = \sum_{x \in X}^{N} \left| p(x) - \frac{1}{N} \right|$$

The main idea is that above and below the average is certainty. Being 1 the sum of all probabilities, then 1/N is the average. For binary classification N = 2. Then it can be shown that:

$$I(X) = \sum_{x \in X}^{N} \left| p(x) - \frac{1}{N} \right| = \left| p(x) - \frac{1}{2} \right| + \left| 1 - p(x) - \frac{1}{2} \right| = \left| p(x) - 0.5 \right| + \left| 0.5 - p(x) \right|$$

and then plotting the function would result in the following graph:

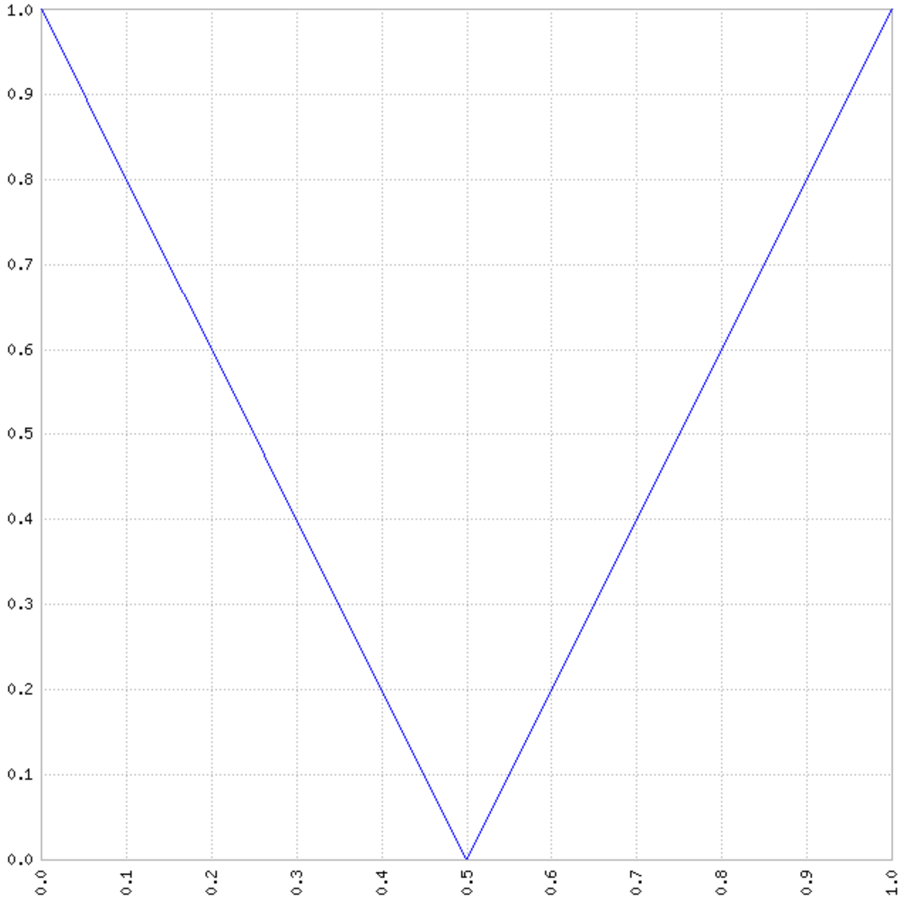

Fig. 1: Certainty function plot for two-class probabilistic system.

Certainty and causality.

In the topic of causality and probability theory, there is an inequality that stands out from the very beginning. It is the probability-raising inequality. The probability-raising inequality states that for an event to be a cause of another, the conditional probability of the effect, given that the cause occurs, must be greater that the probability of the effect alone.

The general form of the original inequality is the following:

$$P\left(c|a\right) > P(c)$$

I will introduce what can constitute a better approach, although it may look very similar. Instead of comparing the probabilities we will compare certainties. So the new inequality now takes the form of:

$$I\left(C|A\right) > I(C)$$

The main difference is that in the original inequality, we are comparing events *a* and *c*, while in the new one, we are comparing whole attributes or variables *A* and *C* .

The idea is that if an attribute can increase the certainty of a probabilistic system, then this attribute is of great importance in reference to this system. This way we can consider attribute *A* a cause of attribute *C*. This could be called the certainty-raising inequality.

If we want to know whether an event *a* causes event *c* then we have to calculate the "causal certainty":

$$\frac{p\left(c \mid a\right)}{I\left(C \middle| A\right)} > \frac{p(c)}{I(C)}$$

This results are important to be able to better analyze and assess the data and the different attributes. For example, a medical doctor would be able to better identify the cause of an illness in a patient.

Results

In order to compare results, the algorithm ID3 was run over several datasets from the UCI website, using C5.0 and the certainty function. The first 80% of the examples in the datasets was used to train, the rest as test. Many datasets were tested but only those which its correct guesses were at least three times the number of incorrect guesses were selected, as a way to filter out the datasets with insufficient examples to learn a good model.

The certainty function almost perfectly fits the training data: this was confirmed with the full **KDDCUP 1999** dataset with 5 million examples, the training error for the certainty function was just 12, while the C5.0 train errors were 338! At the same time the test error was also lower: 87 vs 123! This was also confirmed with the **Skin Segmentation** dataset with 245,057 examples; the train and test error for the certainty function were 9 and 36 respectively, while C5.0 were 189 and 66. Yet another example is the **Statlog (Shuttle)** dataset, for which the certainty function produced a perfect fit of 0 training errors and just 3 test errors, while the C5.0 produced 36 training errors and 15 test errors. In the classic **Monk's Problems** dataset, the certainty function produces less training errors and less test errors than C5.0 in the **Monks-1** and **Monks-2** datasets. Add to that that it also generates less training and test errors in the **Optical Recognition of Handwritten Digits** and the **Pen-Based Recognition of Handwritten Digits** datasets. Also, I ran the **Crime (San Francisco)** dataset with the certainty function and C5.0 (no pruning), the certainty created a tree of 16,283 nodes, and generated 109 train errors and 316 test errors, while the C5.0 created a tree of 12,406 nodes, and generated 562 train errors and 392 test errors. In the

Breast Cancer Wisconsin (Diagnostic) dataset, the certainty function produced 0 train errors and 6 test errors, while C5.0 (no pruning) produced 3 train errors and 8 test errors. With pruning activated, C5.0 produced 7 train errors and 8 test errors. In the **Arcene** dataset, the certainty function produced 0 train errors and 25 test errors, while C5.0 produced 4 train errors and 39 test errors!

It can be seen that this is a trend for many datasets and not just a coincidence. The certainty function almost perfectly fits the training data while maintaining lower error on the test set than a state-of-the-art algorithm such as C5.0.

Conclusion

I have introduced a new certainty function that can be used when building decision trees, instead of the more common impurity functions such as Entropy and Gini. The results are promising in several datasets, opening the door for more research based on this new certainty function.

I have also introduced the certainty-raising inequality, that helps when performing causal analysis on the data.

Compliance with Ethical Standards

Funding: This study was funded by its author.
Conflict of Interest: The authors declare that they have no conflict of interest.

References

1. Quinlan, J.R. (1986). Induction of Decision Trees. Mach. Learn. 1, 1 (Mar. 1986), 81-106
2. Quinlan, J.R. C4.5: Programs for Machine Learning. Morgan Kaufmann Publishers, 1993.