Occam’s Razor and a Non-Syntactic Measure of Decision Tree Complexity
Goutam Paul, Department of Computer Science, State University of New York, Albany
goutam@cs.albany.edu

Motivation

- Occam’s Razor (1285-1349)
  “Puraüitas non est ponenda sine necessitate”: plurality should not be assumed without necessity
- Machine Learning Interpretation
  If two models have the same performance on the training set, choose the simpler
- Justifications
  For parsimonious reasons and predictive accuracy

The Geometry

- Each training or test instance is a point in an m-dimensional instance space
- Each test of an attribute places a hyper-plane in the instance space, perpendicular to that attribute axis
- The hyper-planes partition the instance space
- A tree T with n leaf nodes partitions the instances into n subsets: B_1, B_2, ..., B_n

Decision Trees

- Information Gain - preference to shorter trees
- Shorter trees have better generalization accuracy (e.g. pruning)
- Some evidence (Webb, JAIR 1996) apparently against Occam’s Razor

C4.5 vs C4.5X

- C4.5 has more nodes and branches, and therefore is considered more complex
- Both C4.5 and C4.5X have the same training set accuracy
- C4.5X has better generalization accuracy for some data sets, contrary to Occam’s Razor

Our Approach

- A decision tree partitions the instance space into a set of regions
- The C4.5X tree may be more complex in representation, but the description of its partitions may be less complex
- Occam’s Razor holds in our non-syntactic measure of complexity of the partitions

Partitioning of the Instance Space by Decision Tree

Traditional Measure

- Complexity of a tree T with n nodes and height h is traditionally measured as a function of n and h, and the simplest and the most popular such measure is
  \[ C(T) = n \]

Our Measure using Kolmogorov Complexity

- Any object x can be represented by a binary string
- Definition of Kolmogorov Complexity of a binary string x:
  \[ K(x) = \min\{ l(p): p is a program for x \} \]
  where \( l(p) \) denotes the length of the program p
- Important features of this complexity:
  - It is representation Independent (Non-syntactic)
  - It measures the complexity of the object, not of the source, unlike information theoretic measures
  - A substring of a string may have more Kolmogorov Complexity than the whole string, and hence a C4.5X tree may have less Kolmogorov Complexity than its C4.5 subtree

- If f is a recursive injective mapping from pairs of binary strings to binary strings, the Kolmogorov Complexity of a pair of strings x and y is defined as follows:
  \[ f(x, y) = x <_{xy} y \]
  \[ K(x, y) = K(<_{xy} x, y) \]

Complexity of Decision Tree Partitioning

- Extending the above definition to n such strings
  \[ x_1, x_2, \ldots, x_n \]
  \[ K(x_1, x_2, x_3, \ldots, x_n) = K(<_{xy} x_1, x_2, x_3, \ldots, x_n) \]

- For a decision tree T, instead of measuring its complexity by the description of its splits, we measure its complexity by the description of the partitions of the instance space as induced by T. If the instances are partitioned into n subsets (blocks) B_1, B_2, ..., B_n, we define the K-complexity of T as
  \[ C(T) = K(B_1, B_2, ..., B_n) \]

Traditional Measure

- Ours is hereafter referred as Occam’s Razor
- The idea behind Occam’s Razor is the hope that the simpler model will give better predictive accuracy
- In our experiments, we define Occam’s Razor as holding, if and only if for two classifiers T_i and T_j, either
  \[ C(T_i) \leq C(T_j) \text{ and } GA(T_i) \geq GA(T_j) \]
  or
  \[ C(T_j) \leq C(T_i) \text{ and } GA(T_j) \geq GA(T_i) \]
  otherwise we say that Occam’s razor fails (GA means generalization accuracy)
  - Out of 100 experiments, more than 70% support Occam’s razor in all the datasets

Training Set / Test Set Issues

- We have measured the complexity of partitioning on the test set, rather than on the training set, because C4.5 and C4.5X produce the same partitioning on the training set and both have the same training set accuracies

The idea behind Occam’s Razor is that the simpler model will give better predictive accuracy.

Discussions

- Our complexity measure is generic, and hence we can also measure the complexity based on partitioning on the training set, when, for example, we are considering two different decision tree generating algorithms having different training set partitioning

Conclusions and Future Work

Summary

- Occam’s Razor says to prefer less complex models
- How to measure the complexity is an important issue
- Ideally, complexity measure should be non-syntactic
- Occam’s Razor is typically not violated in our non-syntactic measure of decision tree complexity

Two Further Applications

- Choosing between two or more decision trees based on the minimum complexity criterion
- Prediction of a new test instance x as follows: predict the class of x as the most occurring class label amongst the instances in block B_i, where
  \[ j = \arg\min_{1 \leq i \leq n} \{ K(B_i, U(x)) = K(B_i) \} \]

In other words, we can set the class label of x to be the most occurring class label amongst the instances in the block which has the least increase in complexity, when we add the instance x to that block.

- For prediction, instead of using difference in complexity measure, use of other similarity metrics are also possible, such as the ratio of complexities, or the relative increase in complexity etc.

Future Plans

- Explore the above applications
- Compare the Minimum Message Length / Minimum Description Length approaches with our work