UW Math 480 Final Project (Draft)

Ayla Lampard, Jason Uanon

May 21, 2013

1 Introduction

1.1 Motivation

Machine learning is a branch is artificial intelligence that deals with systems (computer programs) that learn from data. It is focused on using data to make certain predictions or generalizations of that data, often using statistics or information theory to assist in making the best possible "guess."

Decision trees are any systems that use a tree-like graph to model decisions in machine learning. At each node of the tree, a decision can be made that narrows down the possibilities, eventually reaching a leaf node that contains that tree's prediction.

Figure 1 shows an example of a decision tree.

Decision trees provide a simple model for us to answer general questions of the form: given training data that contains attributes $X = \{x_1, \ldots, x_n\}$ and satisfying a trait y, can we predict whether future instances of this data also satisfy y?

1.2 Data

We will be using the Adult Data Set from the UCI Machine Learning Repository [1]. This data is freely-available online and comes from the U.S. Census Bureau from 1994. It contains over 32,000 training instances and 16,000 test instances (although it does contain missing values, denoted with "?". The number of instances with missing values, however, is low enough that we could simply not consider them).

Using this data, we will create a decision tree that will predict whether a person's income exceeds $50,000 per year. The data itself contains 14 attributes, which are (listed with their data categories):

- age: continuous.
- workclass: Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
- fnlwgt: continuous.

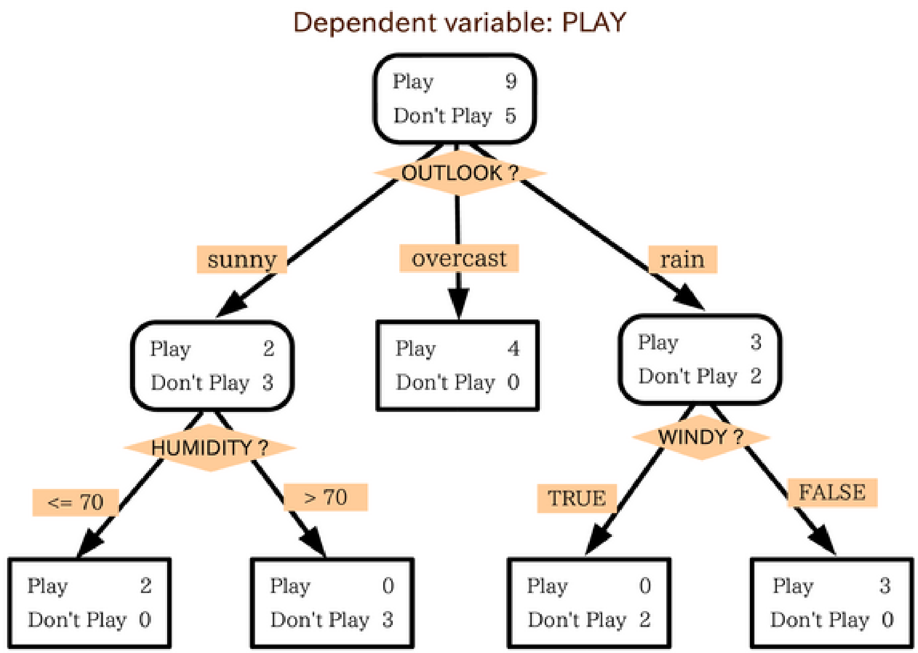

Figure 1: Example of a decision tree

- education: Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
- education-num: continuous.
- marital-status: Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.
- occupation: Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
- relationship: Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
- race: White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
- sex: Female, Male.
- capital-gain: continuous.
- capital-loss: continuous.
- hours-per-week: continuous.

- native-country: United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.

The initial tree will be created using the training data; the tree will then be tested on the test data, giving us feedback on its accuracy.

2 Problem Setup

2.1 Definitions

We will first define some terms that will be used. Note that the following formulas use the logarithm with base 2. The natural logarithm (with base e) is often used, but in this case, information theory deals with bits of information: 0 or 1. Hence, the formulas describe the amount of "disorder" that can be expected among the bits used.

Entropy measures the amount of disorder in a random variable [2]. Let X be a random variable with $p(x)$ the probability that $X = x$. Mathematically, it can be expressed as

$$\begin{aligned} H(X) &= \sum_{i=1}^{n} p(x_i) \log_2 \left(\frac{1}{p(x_i)} \right) \\ &= -\sum_{i=1}^{n} p(x_i) \log_2 \left(p(x_i) \right) \end{aligned}$$

The **conditional entropy** of a random variable X (with events x_i) conditioned on a random variable Y (with events y_j) is

$$H(X|Y) = -\sum_{i=1}^{n} p(x_i) \sum_{j=1}^{m} p(y_j|x_i) \log_2 \left(p(y_j|x_i) \right)$$

The **information gain** of a random variable X conditioned on a random variable Y is

$$IG(X) = H(Y) - H(Y|X)$$

Information gain will be our primary criterium that will determine what attribute to next split on; it is used in the ID3 decision tree algorithm to find the best attribute (e.g., the one that probabilistically reveals the most information regarding our target classification) [3].

Another way to interpret information gain is the expected reduction in entropy (disorder) caused by partitioning the examples according to this attribute.

2.2 The Algorithm

We will be using the ID3 Decision Tree algorithm using information gain as the splitting criteria. It can be summarized as follows:

- Start from the empty decision tree
- Select the next best attribute i that maximizes information gain (i.e., maximizing $IG(X_i)$)
- Recursively build the children of the root node

It is essentially a greedy algorithm that grows the tree top-down, continuing until the tree either classifies all of the training examples, or until all attributes have been used (for our purposes, with thousands of training examples, the latter case will occur).

2.3 Overfitting & Pruning

The algorithm previously described grows each branch of the tree just enough to perfectly classify the training examples. When the number of training examples is too small to produce a representative sample of the true target funtion (i.e., how the data behaves asymptotically), this can lead to suboptimal behavior on the test data [3]. There may also be "noise" in the data that misrepresents the target function. In these cases, we may produce trees that overfit the data. More precisely, given a hypothesis space H, a hypothesis $h \in H$ is said to **overfit** the training data if there exists some alternative hypothesis $h' \in H$ such that h has a smaller error rate than h' over the training data, but h' has a smaller error rate than h over the entire distribution of instances [3].

One heuristic to reduce overfitting is to **prune** the tree. That is, we seek to reduce the size of the decision tree while minimizing the effect on the training set. We could cross check with the test data to check if pruning actually increases accuracy on the overall data set.

2.4 Implementation

We will be using Python 2.7.3 for the implementation of the decision tree algorithms. As part of the project, we will also create a Cython version using Cython 0.15.1 to enhance its performance, and compare the relative speeds of the two implementations.

References

[1] Bache, K. and Lichman, M. (2013). *UCI Machine Learning Repository: Adult Data Set.* [http://archive.ics.uci.edu/ml/datasets/Adult]. Irvine, CA: University of California, School of Information and Computer Science.

[2] Segaran, Toby. *Programming Collective Intelligence*. O'Reilly, California, 2007.

[3] Mitchell, Tom. *Machine Learning*. McGaw-Hill, 1997.