CS 217: Artificial Intelligence and Machine Learning **Jan-Apr 2024**

Lecture 10: Decision Trees

Lecturer: Swaprava Nath *Scribe(s): Tejas Sharma*

10.1 The Objective

We are to create a classifier on whether a given student will pass or not, depending on the data. Like, we have a list of students with backgrounds, mock tests taken, and online classes taken (variables with a fixed no. of outcomes).

We look at the features above and frame a decision tree. Now, given the next student, we label him as pass or fail by traversing this decision tree.

Let us record it like this by mock tests taken:

- **Yes**: 5 Pass, 4 Fail
- **No**: 3 Pass, 4 Fail

A better scheme is by background first since two of them directly determine the result.

- **CSE**: 4 Pass, 0 Fail (certain)
- **Math**: 4 Pass, 4 Fail (uncertain – split by **mock test**):
  - **Yes**: 2 Pass, 2 Fail (uncertain – split by **online courses**):
    - **Yes**: 1 Pass, 1 Fail
    - **No**: 1 Pass, 1 Fail.
  - **No** (mock test): 3 Pass, 0 Fail
- **Others**: 0 Pass, 4 Fail (certain)

10.1.1 Example: Iris Dataset

We plot a graph of Petal width vs Petal width (cm).

We form a decision tree regarding petal width and then petal length. Petals in a certain region form a class. We want to classify them.

10.1.2 How to Build the Tree?

We look at the variable in the dataset (field) with the least entropy or the most skewed distribution.

10.1.2.1 Entropy of a Random Variable

Let $\mathbf{X}$ be a categorical random variable, with $p(x) = P(X = x) \forall x$. We define

$$H(\mathbf{X}) = -\sum_{x \in \mathbf{X}} p(x) \cdot log_{\|\mathbf{X}\|} p(x), \quad \mathbf{X} = 0, 1 \text{ is a binary variable, H measured in bits and } |X| = 2.$$

$H(x)$ is always nonnegative and is always less than one.

Think of $H(x)$ as an expectation of the logarithm of the probability. We define $H[\mathbf{X}]$ as the entropy.

10.1.2.2 Conditional Entropy

We know the price distribution of one of the ingredients but not the price of the dish itself, for example. We look at $P(x|y) = P(\mathbf{X} = x|\mathbf{Y} = y)$.

Correspondingly, we define conditional entropy as

$$H[\mathbf{X}|\mathbf{Y}] = -\sum_{y \in \mathbf{Y}} \sum_{x \in \mathbf{X}} p(x, y) \cdot log_{\|\mathbf{X}\|} p(x|y) = -\sum_{y \in \mathbf{Y}} p(y) \left(\sum_{x \in \mathbf{X}} p(x|y) \cdot log_{\|\mathbf{X}\|} p(x|y) \right)$$

Clearly, the inner bracket evaluates to $H(X|Y = y)$. We further define $I(X, Y) = H(X) - H(Y)$.

Rather, we can define conditional entropy based on each value of y as $H(X|Y) = \sum_{y \in \mathbf{Y}} p(y) H(X|y)$

10.1.2.3 The main Algorithm: How to go about it

```
while (stopping criteria not met) do
    find the features that yield maximum net conditional entropy or information.
```

Other metric used: **Gini index.**

10.1.2.4 Where to Stop Building the Tree?

[Context: Avoid overfitting]
If the information gained on taking the next parameter is the same for all variables, we stop there. Of course, if we reach an atomic distribution, we must stop anyway.

If we have a shallow tree – we do not have enough power to distinguish or classify the data and identify the output variable given test elements.

If we have too deep a tree, our model gets too specific to the exact training points given and is unlikely to be accurate given the next point or the test elements. For such a tree, we can pre-prune or early stop the creation of the tree until the test error goes up. (We split the given data into training and test data).

Yet another method is the Ensemble method: we use averages of different models to create the decision tree. Like average of this method and Gini index tree.