Data Mining, CSCI 347, Fall 2019
Recursive Algorithms - Decision Trees, Sept. 16

Trees

- "Divide-and-conquer" approach
- Tree built by splitting on attributes, where the number of children is usually equal to the number of values for the attributes
- The leaves tell the classification
- Attributes typically only get tested once

Constructing a decision tree can be expressed recursively:

1. Select an attribute to place as the root node
2. Make one branch for each possible value, splitting the example set into subsets, one for every value of the attribute.
3. Repeat the process for each branch (recursion)
4. Base case - stop if all instances have the same class, there are no more attributes to split on, or have reached a pre-defined depth.

Selecting which attribute to split on:

- Want to get smallest tree
- Heuristic (guideline): choose the attribute that produces the "purest" nodes
- In the case that no attribute produces "pure" nodes, split on the attribute that gives the greatest information gain (from Information Theory)

Information Theory

- Information is a reduction in uncertainty
- Measure "purity" in "bits"
    - 0 bits – no uncertainly
    - Higher the value, the more uncertainty
- Entropy:

    $\text{entropy}(p_1, p_2, \ldots, p_n) = -p_1 \log p_1 - p_2 \log p_2 \ldots - p_n \log p_n$ bits
- Decisions can be made in several stages:

    info for one 2-way split:

    $\text{info}([a,b]) = \text{entropy}(a/(a+b), b/(a+b))$

    info for one 3-way split:

    $\text{info}([a,b,c]) = \text{entropy}(a/(a+b+c), b/(a+b+c), c/(a+b+c))$

    info for two splits, a three way and a two way:

    $\text{info}([a,b,c],[d,e]) = (a+b+c)/(a+b+c+d+e)*\text{info}([a,b,c]) + (d+e)/(a+b+c+d+e)*\text{info}([d,e])$