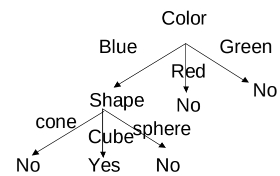
Decision Trees

CPS 270

Ron Parr

Decision Trees

- Decision trees try to construct small, consistent hypothesis
- Suppose our concept is “blue cube”



Facts About Decision Trees

- If the concept has d conjuncts, there will be a decision tree for this concept with depth d
- Decision trees are very bad for some functions:
  - Parity function
  - Majority function
- For errorless data, you can always construct a decision tree that correctly labels every element of the training set, but it may be exponential in size

Decision Tree Algorithms

- Aim for:
  - Small decision trees
  - Robustness to misclassification
- Constructing the shortest decision tree is intractable
- Standard approaches are greedy
- Classical approach is to split tree using an information-theoretic criterion

Growing Decision Trees

```
Repeat until no good leaves
  Pick leaf
  Split = choose_variable(variabes – all_parents(leaf))
  For val in domain(split)
   new_leaf = new_leaf(split=val)
   new_leaf.instances=leaf.stances s.t. split=val

For leaf in tree
  classification(leaf)=majority_classification(leaf)
```

Information Theory

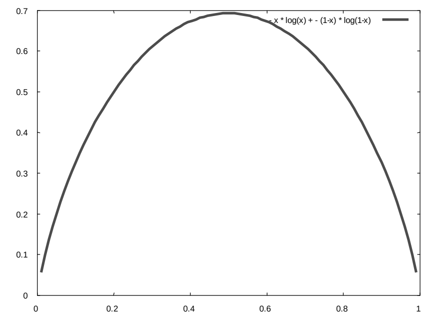
- Roughly speaking, information theory measures the expected number of bits needed to communicate information from one person to another
- Suppose person1 is flipping a coin with bias p
- Person1 wants to tell person2 the sequence of results
- What is the expected number of bits person 1 will send to person 2?
- Note relation to compression

Information Content

$$I(p_1,\ldots,p_n) = E(\#\text{bits}) = \sum_{i=1}^{n} -p_i \log_2(p_i)$$

For an unbiased coin, the information content is 1.
For a totally biased coin, the information content is 0.

Information Content



Information Content of a Leaf

$$I\left(\frac{p}{p+n},\frac{n}{p+n}\right) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}$$

Information gain of a split:

$$I\left(\frac{p}{p+n},\frac{n}{p+n}\right) - \sum_{i=1}^{v}\frac{p_i+n_i}{p+n} I\left(\frac{p_i}{p_i+n_i},\frac{n_i}{p_i+n_i}\right)$$

Gain Example

- Suppose we have seen:
  - Red tetrahedron(f), Blue sphere(t), Blue cone(t), green cone(f)
- Is it better to split on shape or color?
- Information of original set is: 1
- Information gain of splitting on cone:
- Information gain of splitting on blue:

Favoring Small Examples

- Information gain (and other splitting criteria)
  - Are greedy
  - Favor small trees
- This makes representation an issue yet again
- Suppose you want to learn “parity(+) and blue”
- Hard to learn with decision trees, but
  - If we treat parity like a state variable, then it’s easy
  - Call these derived variables features or attribrutes

Decision Tree Conclusion

- Simple method
- Works surprisingly well in many cases
- Issues:
  - Continuous variables
  - Missing values
  - Expressive power