Decision Trees

Decision Trees - play golf?

Finding a good tree

Decision tree algorithm

Decision trees

• Popular C4.5 excellent “off the shelf” alg.
• Efficient hypothesis space
  – Variable sized: bigger trees for more complicated hypotheses
  – Can mix discrete and numeric attributes
• Small trees are easily understandable
• Decision tree successes: (many)
  – flying (Sammuti et.al. 92), better than simulator pilots
  – BP GasOil (Mitchie ’86) for separation at off-shore platforms 2500 years (10 man-years by hand, done in 100 man days), saved many millions

Finding a good tree

• Want accuracy on training data but don’t want to memorize it
• Small tree good for generalization
• Finding smallest tree consistent with data is NP hard
• Use greedy search to build trees top down -- can miss XOR

Decision tree algorithm

• If data pure (all one class) then return a leaf predicting the class
• Otherwise:
  – Pick a test to split on (discrete attribute or attribute+threshold)
  – Insert node with test into tree
  – split training data based on test, and recursively construct each branch from proper portion of data

How to chose test?

• Use a splitting criteria like number of mistakes or information gain
• Often viewed as “impurity” measure, goal: minimize the impurity
• Try all possible splits and use the one that maximizes the criteria
• Many algorithms use measures related to information theory
Impurity measures

• If have n total examples with n_+ positive examples, let p=n_+/n
• Examples of impurity functions (of p)
  – Gini index: 2p(1-p)
  – Entropy: -p lg p - (1-p) lg (1-p)
  – Error rate: 1-max[p, 1-p]
  – Generalized entropy for multiple classes

Impurity of split

• If have set S of n examples split into S_1 and S_2. Let n_1 be number examples in S_1 and p_1 be fraction of S_1 labeled positive (n_2, p_2 similar)
• Badness of split is defined avg impurity:
  \[ \frac{n_1 \text{ impurity}(p_1) + n_2 \text{ impurity}(p_2)}{n} \]
• Pick split with least badness
• Generalizes to multi-way splits (need to penalize them too)

Decision Trees can overfit

• Overfitting - modeling the particulars of the data set rather than the underlying pattern
• Def: Hypothesis h \in H overfits the data if there is an h' \in H such that h better on training data but h' generalizes better.
• Complexity of decision trees lets them fit the noise

ID3 overfits (Mitchell)

Avoiding overfitting:

• Early stopping (PrePruning)
• PostPruning (more popular)
  – Based on validation set
  – Rule post-pruning:
    • convert to rules, generalize if accuracy improves, and apply rules in order of accuracy
Rule Extraction from Trees

C4.5 Rules
(Quinlan, 1993)

Decision tree comments:

- Several packages e.g. ID3, C4.5
  - Weka has J48 and BFTree
- Often heavily engineered to handle
  missing data, overfitting, numeric vs.
  nominal attributes, etc.

- Multivariate trees
- Regression trees

Model Selection in Trees:

Random Forests

- Pick small random subset of features to try at
each node rather than exhaustive search
- Build many trees and predict with most
  frequent prediction
- Subset saves time, robust against missing
  data
- Ensemble reduces variance - don’t need
  pruning
- Ho ’95, Brieman ’01

Exercise

1. Make up an XOR.arff data file with 12
   examples. Each example \( x \) labeled
   with \( x_1 \) XOR \( x_2 \), and add 10 irrelevant
   and random binary features.
2. Try Weka’s ID3 on the XOR data.
3. Try Weka’s J48 on the soybean data -
   is the tree understandable?

Decision Tree Summary:

- Model: trees (flexible size)
- Data: numeric and nominal
- Interpretable: if small tree
- Noise/outliers: OK with pruning
- Irrelevant features: fair
- Missing attributes: some tricks
- Computation time: OK