Extension I: Continuous Attributes

• Discretize into ranges
  – big, medium, small
  – Use care and prior understanding and luck to make it work

• Split by introducing thresholds
  – Form of test: $A < c$
  – Partition into $A < c$ and $A \geq c$.
  – Calculate information
  – How to find the split point with the highest gain?
    • For each continuous feature $A$:
      • Sort examples by $A$
      • Evaluate each mid-point as a possible threshold
      • Real parameter but finite interesting distinctions
Extension I: Continuous Attributes

- Choose c w/ highest information gain
- Real parameter; but based on data, only finite number of interesting distinctions
- Thought question: why use the mid-point?
- Generalizes (with some computational effort) to more intervals
Extension II: Missing Attributes

• E.g., Medical tests not yet run

• Training:
  – Information Gain splitting on attribute ‘a’
  – In some of the examples ‘a’ is not given

• Testing:
  – Classify an example x
  – But x is missing the value of ‘a’
Missing Attributes

Outlook = ???, Temp = Hot, Humidity = Normal, Wind = Strong, label = ??

Estimate Outlook:
Blend by labels
1/3 Yes + 1/3 Yes +1/3 No = Yes
Blend by counts & labels
label weighted by est. prob.
Recall Final Decision Tree

Thought Questions

Are these actual world examples or did someone make them up?

How could we tell?

Instead of 14 training examples, suppose we had 14,000, same pattern, same tree

Is it possible that we build the wrong tree? (what would that mean?)
Overfitting
(very important & general phenomenon!)
(it is precisely the well-known statistical notion of significance)

• Concept performs
  – well on training data (drawn from X according to $D$)
  – poorly on additional examples from same distribution

• Excess flexibility in hypothesis space $H$
  – Finds training set pattern not in population
  – Concept selection from too little (insignificant) training data
  – NB: in statistics, “significant” $\neq$ “large”

• Confidence in selecting $c \in H$
  – Diversity of $H$ reflects our ignorance
  – Training set $Z$ provides information
  – $\text{Information}(Z) \geq \text{Information need}(H)$  [>> for high confidence]

• Often Learning algorithms cannot tell

• Low confidence: expected behavior of “best” concept (on $Z$) is poor on underlying population

• Extreme: Rote learning & No generalization
Overfitting

Insignificnace: conclusions drawn from too little evidence
Overfitting in Decision Trees

• A rich set of tests
  – Split to (nearly) homogeneity
  – Few examples at leaves
  – Accurate on training set
  – Poor *expected* performance on new data
  – Noise: either real or sampling

• With limited training examples, deep decision trees are bad
  – Low confidence in lower split choices (why?)
  – Hypothesis space H become too expressive
How to Reduce Overfitting in Decision Trees

- Restrict the expressiveness of $H$
- Limit the set of tests available
- Limit the depth
  - No deeper than $N$ (say 3 or 12 or 86)
  - But how to choose?
- Limit the minimum number of examples used to select a split
  - Need at least $M$ (is 10 enough? 20?)
  - How to choose? (Adjust for number of tests? Their outcomes?)
  - Want significance; Statistical hypothesis testing can help
- BEST: Learn an overfit tree and prune
  - Partition the labeled examples
  - A: training data – grow the tree
  - B: pruning data – prune the tree from leaves
  - What would be the pruning algorithm?
Bias / Variance Tradeoff

• Learner chooses $c \in H$ based on $Z$ (training set)

• Tradeoff (with limited data):
  – **Bias**: $c$’s choice only partially due to $Z$
  – **Variance**: $c$’s change from a re-sampled $Z’$

• How likely would we build a similar concept
  – From a different training sample?
  – From this training set used differently?

• What makes concepts similar?
Variance Reduction: Bagging

- General technique
- Easy, Effective, Useful
- Average over a set of quasi-independent concepts
- Quasi-independent???
- Partition training set $Z$, different ways
- With each, grow & prune a decision tree
- Classify new examples by vote of concepts
- “Decision Forest”
Cross Validation  
(Popular & Useful)

• Estimating Parameters  
  – Classification Accuracy  
  – Blending or other Learning Parameters
• Partition training data into subsets for  
  – Training, Testing
• Each partitioning yields quasi-independent learning
• How to partition?  
  – Often into N sets (folds)  
    • N=5, 10,...  
    • N=|Z| (Leave One Out)  
  – Train on N-1 sets, Test on the remaining one  
  – All N ways
• Agreement is evidence for confidence  
  Can combine results (e.g., averaging)
Cross Validation in DTs

• Partition training set N ways
• Train on N-1, Prune on 1 to best tree
• Repeat N times
• Yield quasi-independent estimates of best tree
• Estimate best depth (poor)
• Estimate best number of leaves (better)
• Estimate a model of good decisions (best)
• Decide on a split termination (using above)
• Use all the data, stop at split termination