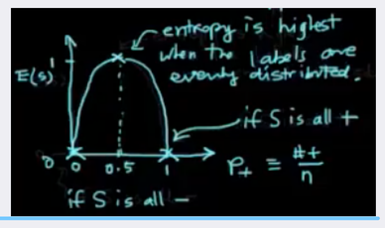
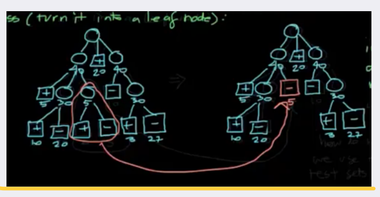
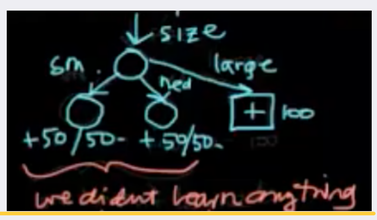
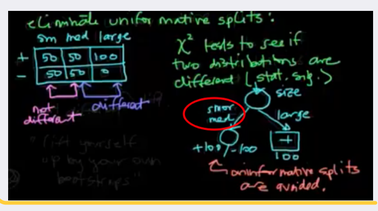
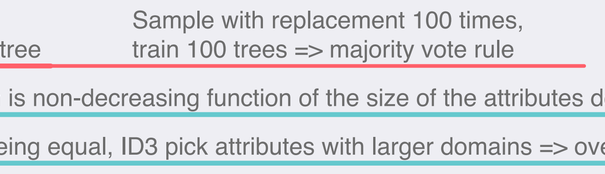
- W08
  - Introduction to Decision Trees (8A)
    - rules
      - Irrelevant feature
      - Best order to examine the feature values
      - Partition
  - Entropy & Information Gain (8B)
    - entropy
      - $E(S) = -\sum_i p_i log_2(p_i)$
      - 
        
      - A good attribute lower entropy
    - Information gain
      - $G(S, A) = E(S) - \sum_{v \in V_A} \frac{|S_v|}{|S|} E(S_v)$
      - Weight each subgroup by its relative frequency
  - ID3 Algorithm (8C)

**Decision Tree Learning (ID3)**

```
def id3( data, attributes, default)
  if data is empty, return default
  if data is homogeneous, return class label.
  if attributes is empty, return majority-label( data)
  best_attr = pick_best_attribute( data, attributes)
  node = new Node( best_attribute)
  default_label = majority-label( data)
  for value in the domain of best_attr
    subset = examples in data where best_attr == value
    child = id3( subset, attributes - best_attr, default_label)
    add child to node
  end
  return node
end
```

  - Overfitting in Decision Trees (8D)
    - Better generalize
      - Rare occurrences or outliers
    - 1. Prune
      - C4.5
        - Not record the majority class for EVERY node
        - Randomly replace an internal node to leaf (with its majority class), check if improve accuracy
        - 
          
        - Essentially it is hill climbing
        - 10 fold cross validation to do the test
          - 8: Train
          - 1: Prune
          - 1: Test
    - 2. CHAID
      - More records, more reliable
      - With information gain, only one of two attribute values need to be really good for the split.
        - 
          
      - X^2, Chi-square: eliminate uniformative splits
        - To see if two distributions are statistically different
        - To do: lump the values that do not create statistically significant partitions together
        - 
          
  - Random Forests (8E)
    - 3. Random Forests
      - Statistic issue: Not enough data
        - Repeated sampling: Expensive
        - Standard error
        - bootstrap
          - Treat sample as population. Sample with replacement, calculate the average, use the variance of all those averages
          - Confidence bound: STD
          - Computational expensive: Ok now a days
      - Apply to decision tree
        - Sample with replacement 100 times, train 100 trees => majority vote rule
  - Regression Trees and Other Improvements to ID3 (8F)
    - Domain sizes
      - Information gain is non-decreasing function of the size of the attributes domain
      - If other things being equal, ID3 pick attributes with larger domains => overfitting
      - Solution: Normalized information or the gain ratio
      - Split:
        - $Split(S, A) = -\sum_{v \in V_A} \frac{|S_v|}{|S|} log_2(\frac{|S_v|}{|S|})$
        - Split is the normalizer for normalized information gain or gain ratio.
    - Numerical values
      - bucketize
      - Mid-points as threshold, binary splits
        - After one split, we can use the rest number (because number come with order)
        - 
          
    - Regression trees
      - Why?
        - $y = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 \cdots + \theta_n x_n$
        - Interaction terms Theta_23 X_2 X_3 …
        - 
          
        - averages and the sum of squared errors (SSE)
        - 
          
      - 1. Start with average y for all data
      - 2. Calculate SSE
      - 3. Pick attribute that lower SSE the most
      - 4. Recurse on partitions