CS / Philo 372

Lecture 8
More on Learning

Decision Trees

- Algorithm
- Start: place all examples in a "leaf"
- Loop:
  - Find a leaf that is not all the same category.
    - it will become an internal node
  - try all possible features to create new leaves
  - select the feature that maximizes the some criterion
    - When using an Occam bias, information gain is common
    - The formula below balances homogeneity of the leaves against the number of leaves added.

$$r(A)=\sum_{i=1}^{v}\left(\frac{(p_i+n_i)}{(p+n)}\right)*\left(\left(\frac{-p_i}{(p_i+n_i)}\right)\log_2\left(\frac{p_i}{(p_i+n_i)}\right)+\left(\frac{-n_i}{(p_i+n_i)}\right)\log_2\left(\frac{n_i}{(p_i+n_i)}\right)\right)$$

$$gain(A)=\left(\left(\frac{-p}{(p+n)}\right)\log_2\left(\frac{p}{(p+n)}\right)+\left(\frac{-n}{(p+n)}\right)\log_2\left(\frac{n}{(p+n)}\right)\right)-r(A)$$

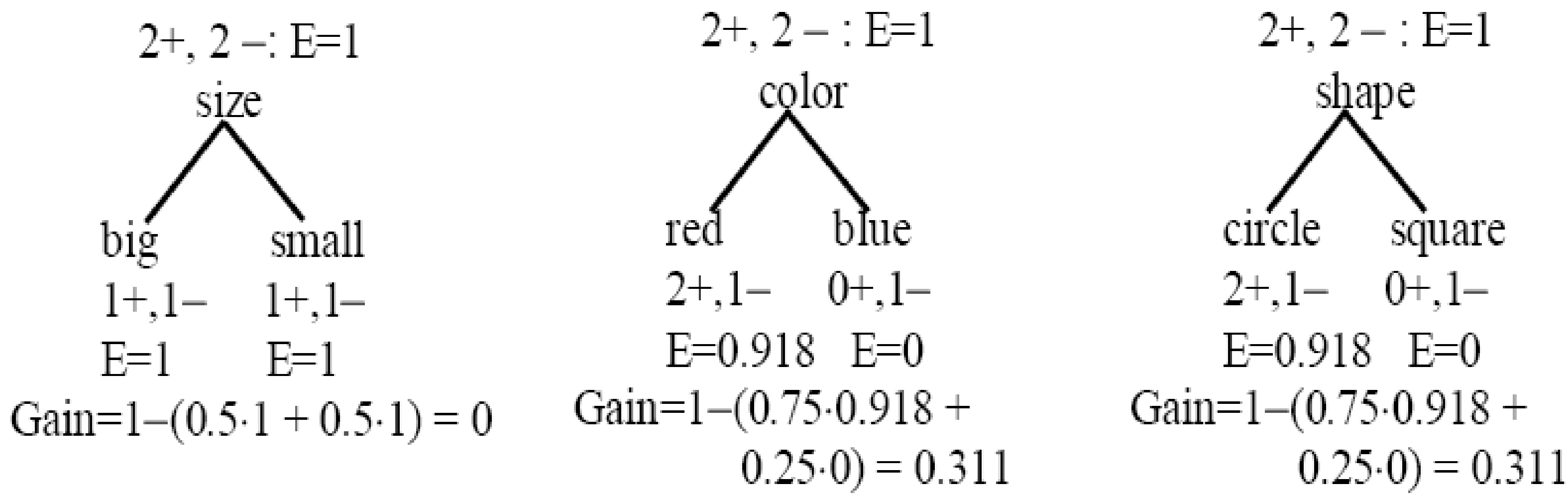
Information Gain

- The information gain of a feature F is the expected reduction in entropy resulting from splitting on this feature.

$$Gain(S,F) = Entropy(S) - \sum_{v \in Values(F)} \frac{|S_v|}{|S|} Entropy(S_v)$$

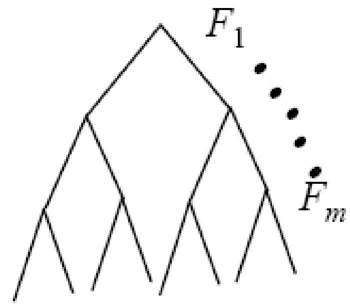
where S_v is the subset of S having value v for feature F.

- Entropy of each resulting subset weighted by its relative size.
- Example:
  - <big, red, circle>: + <small, red, circle>: +
  - <small, red, square>: – <big, blue, circle>: –



Complexity of building Trees

- Worst case builds a complete tree where every path test every feature. Assume n examples and m features.



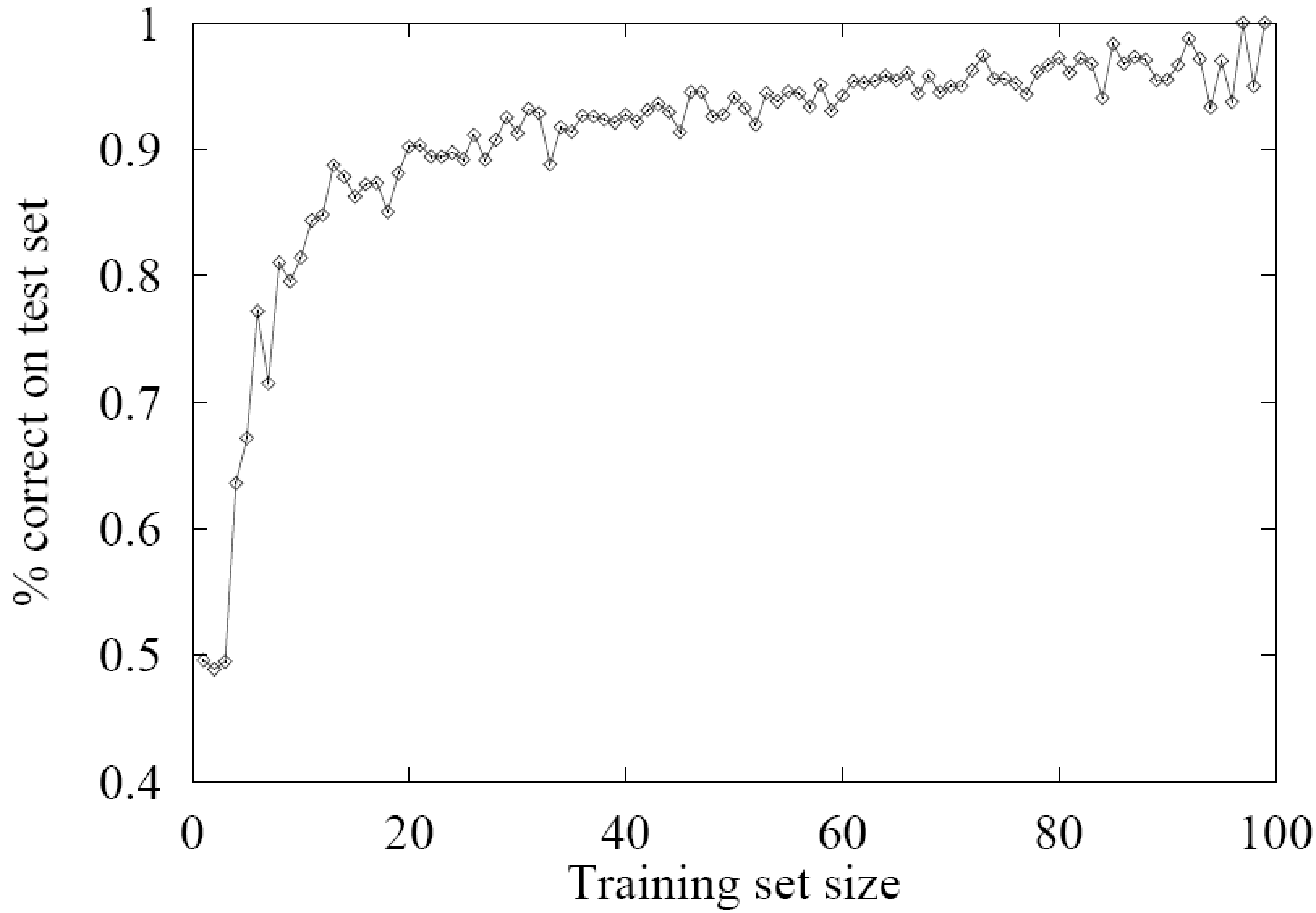
Maximum of n examples spread across all nodes at each of the m levels

- At each level, i, in the tree, must examine the remaining $m-i$ features for each instance at the level to calculate info gains.

$$\sum_{i=1}^{m} i \cdot n = O(nm^2)$$

- However, learned tree is rarely complete (number of leaves is $\leq n$). In practice, complexity is linear in both number of features (m) and number of training examples (n).

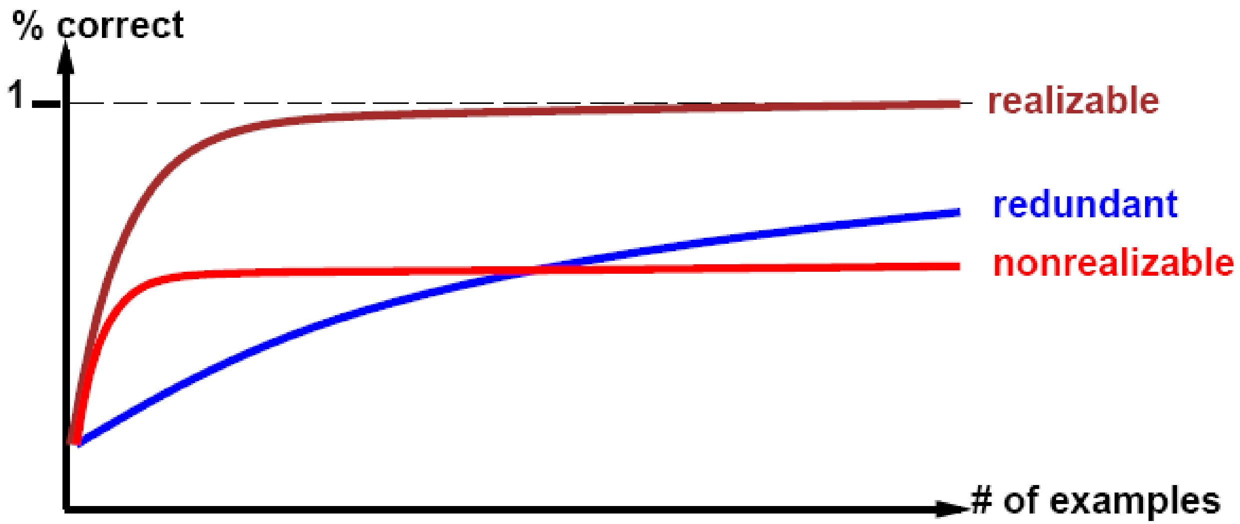
Learning Curves



Limitations on Learning

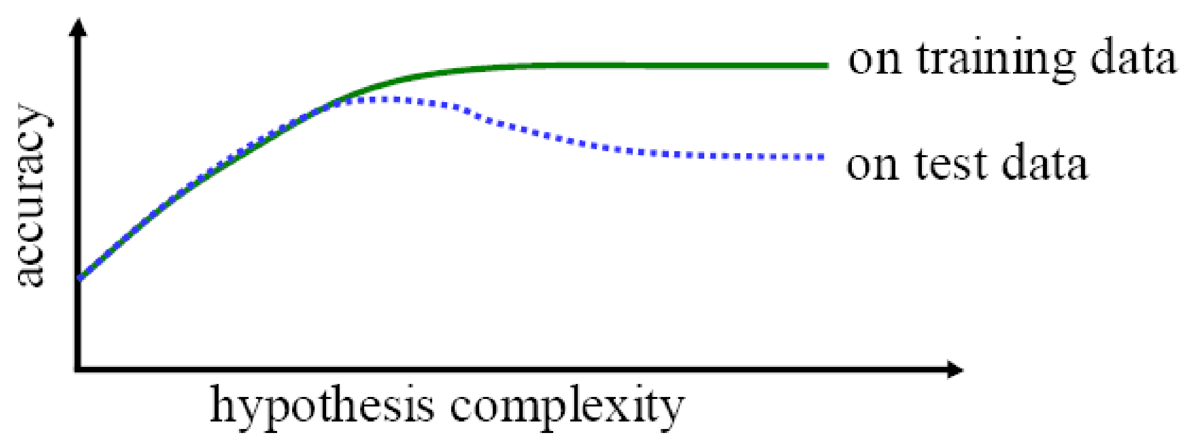
Learning curve depends on

- realizable (can express target function) vs. non-realizable
  Non-realizability can be due to
  - missing attributes, or
  - restricted hypothesis class (e.g., thresholded linear function)
- redundant expressiveness (e.g., loads of irrelevant attributes)



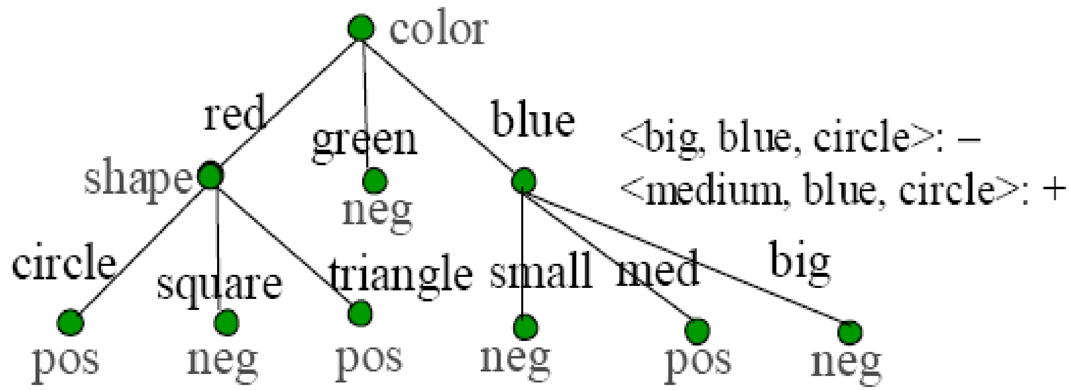
Overfitting

- Learning a tree that classifies the training data perfectly may not lead to the tree with the best generalization to unseen data.
  - There may be noise in the training data that the tree is erroneously fitting.
  - The algorithm may be making poor decisions towards the leaves of the tree that are based on very little data and may not reflect reliable trends.
- A hypothesis, *h*, is said to overfit the training data is there exists another hypothesis which, *h*′, such that *h* has less error than *h*′ on the training data but greater error on independent test data.



Noise

- Category or feature noise can easily cause overfitting.
  - Add noisy instance <medium, blue, circle>: pos (but really neg)



- Noise can also cause different instances of the same feature vector to have different classes. Impossible to fit this data and must label leaf with the majority class.
  - <big, red, circle>: neg (but really pos)
- Conflicting examples can also arise if the features are incomplete and inadequate to determine the class or if the target concept is non-deterministic.

Limitations of Learning

- "PAC" learning
  - "Probably Approximately Correct"
  - Idea
    - How much data to I need to with **some amount of confidence**, say that my result will be correct at least a **certain percentage of the time**
- Positive Result
  - The number of examples required for learn a concept is $N \geq \frac{1}{e} * (\ln(\frac{1}{delta}) + \ln(H))$
  - where
    - e = maximum acceptable error rate
    - delta => (1-delta) is probability that the hypothesis will be e acceptable
    - H size of hypothesis space

More PAC

- Negative Result
  - for boolean functions
    - hypothesis space size = 2^(2^n)
    - example space size: 2^n
    - PAC estimates therefore are larger than size of example space
    - So an alg learning in space of all boolean functions can be no better than a lookup table
    - For any unclassified example, there are as many correct as incorrect hypotheses
      - NFL

Limitations on Learning
Mistake Bounded Learning

- Like O() notation for algorithm analysis,
  - how many examples that require – in the worst case – to get correct answer
  - Assuming learning bias is correct
- Idea – the antagonistic teacher
  - You guess the answer
    - if correct teacher says "yes"
    - if incorrect teacher gives an example your concept gets wrong
    - trick: teacher does not have a fixed concept
    - How many examples to id an rectangle in 2d?

Ensemble Methods -- Committees

- Observation
  - different learning methods have different biases
  - results in making different mistakes
  - This leads to hope that using some sort of "committee" might improve overall performance
- Result
  - Banko & Brill observe that for "small" example sets committee votes are helpful, but as examples increase committee is little better than single best
  - Others report committee is often worse than best individual
    - Why?

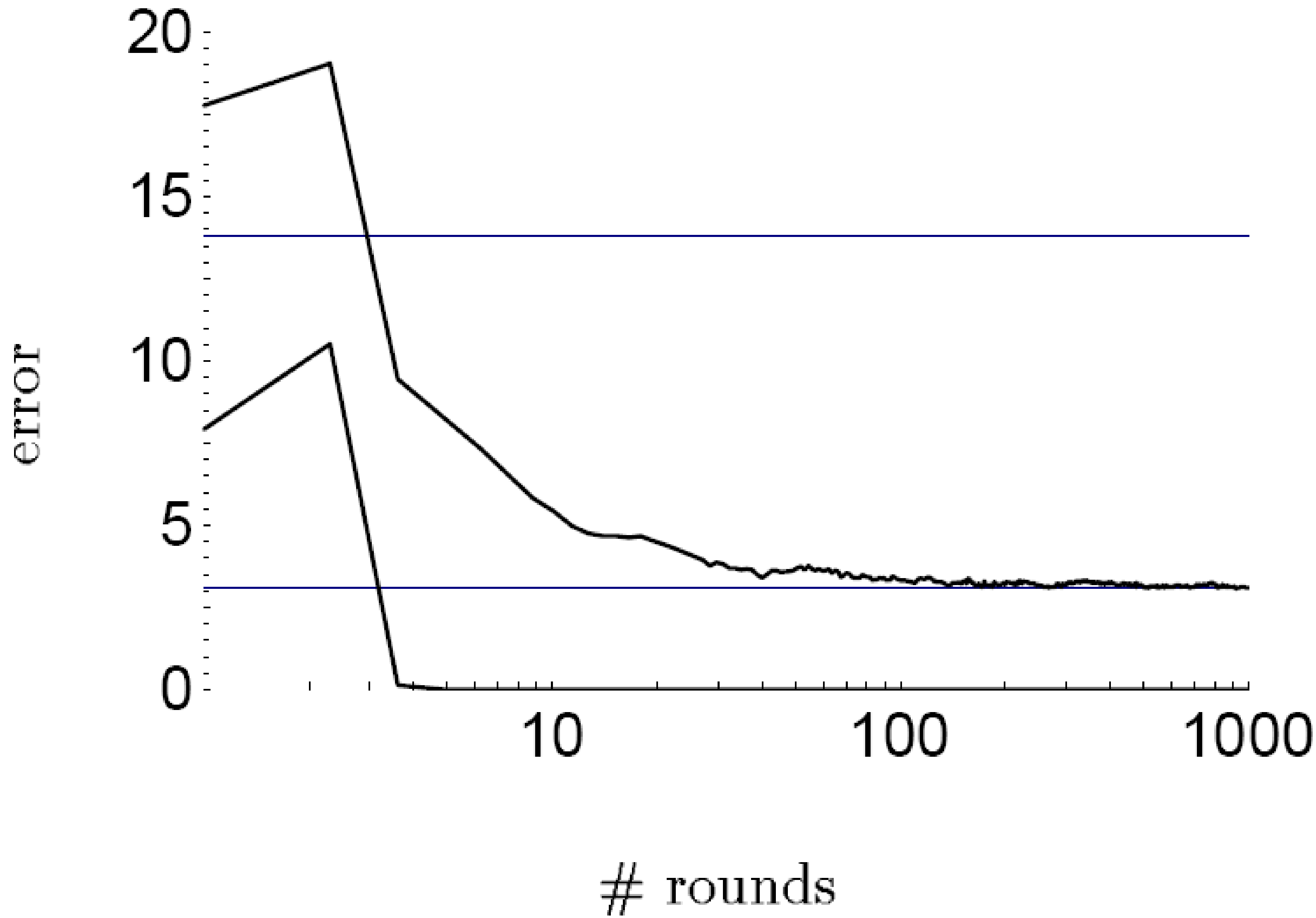
Ensembles
Stacked Generalization

- Idea – different training sets cause bias
  - so train N classifiers each using a different N-1 examples (called level 0 data)
  - Then create a new data set in which the output of the N classifiers is the input and the correct answer is the output (called level 1 data)
- Ting (1999) showed that this approach can be very effective
- As with committees it can be hard to beat the best
- relies on example bias so large datasets ...

Ensembles -- boosting

- Central idea – weighted examples
  - tell learner to care about some examples more than others.
    - weighting is often useful in real world
      - cost/risk of testing in medical diagnosis
    - Apply to decision trees?
  - Concept
    - Learning phase
      - begin with all examples equally weighted and build a classifier
      - increase weight of examples misclassified and decrease weight of correct
      - Repeat M times recording training set correctness
    - Classification
      - on new example class is correctness weighted sum of M classifiers

Boosting -- adaboost



boosting III

- Note that test set accuracy continues to improve even when training set accuracy is 100%!!!!
- This is consistently observed
- **EXPLAIN**
- On many datasets test set accuracy will eventually go down.
  - Why?