



Ensembles of Classifiers

Larry Holder
CptS 570 – Machine Learning
School of Electrical Engineering and Computer Science
Washington State University



References

- T. Dietterich (2000). Ensemble Methods in Machine Learning. In J. Kittler and F. Roli (Ed.) First International Workshop on Multiple Classifier Systems, *Lecture Notes in Computer Science* (pp. 1-15). New York: Springer Verlag.



Learning Task

- Given a set S of training examples $\{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_m, y_m)\}$
 - Sampled from unknown function $y = f(\mathbf{x})$
 - Each x_i is a feature vector $\langle x_{i,1}, \dots, x_{i,n} \rangle$ of n discrete or real-valued features
 - Class $y \in \{1, \dots, K\}$
 - Example may contain noise
- Find hypothesis h approximating f



Ensemble of Classifiers

- Goal
 - Improve accuracy of supervised learning task
- Approach
 - Use an ensemble of classifiers, rather than just one
- Challenges
 - How to construct ensemble
 - How to use individual hypotheses of ensemble to produce a classification



Ensembles of Classifiers

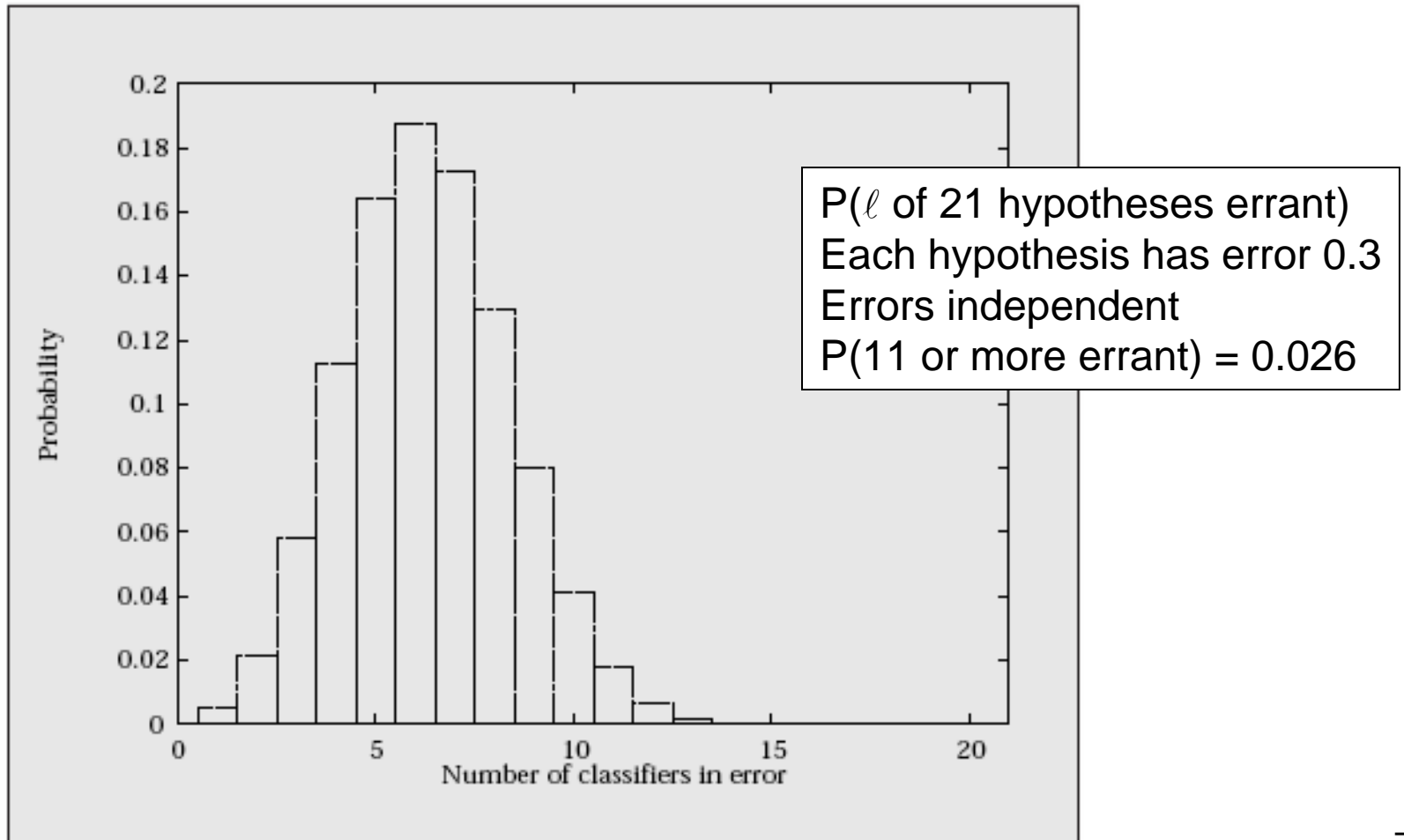
- Given ensemble of L classifiers h_1, \dots, h_L
- Decisions based on combination of individual h_ℓ
 - E.g., weighted or unweighted voting
- How to construct ensemble whose accuracy is better than any individual classifier?



Ensembles of Classifiers

- Ensemble requirements
 - Individual classifiers disagree
 - Each classifier's error < 0.5
 - Classifiers' errors uncorrelated
- THEN, ensemble will outperform any h_ℓ

Ensembles of Classifiers (Fig. 1)





Constructing Ensembles

- Sub-sampling the training examples
 - One learning algorithm run on different sub-samples of training to produce different classifiers
 - Works well for *unstable* learners, i.e., output classifier undergoes major changes given only small changes in training data
- Unstable learners
 - Decision tree, neural network, rule learners
- Stable learners
 - Linear regression, nearest-neighbor, linear-threshold (perceptron)



Sub-sampling the Training Set

- Methods
 - Cross-validated committees
 - k -fold cross-validation to generate k different training sets
 - Learn k classifiers
 - Bagging
 - Boosting



Bagging

- Given m training examples
- Construct L random samples of size m with replacement
 - Each sample called a *bootstrap replicate*
 - On average, each replicate contains 63.2% of training data
- Learn a classifier h_ℓ for each of the L samples



Boosting

- Each of the m training examples weighted according to classification difficulty $p_\ell(\mathbf{x})$
 - Initially uniform: $1/m$
- Training sample of size m for iteration ℓ drawn with replacement according to distribution $p_\ell(\mathbf{x})$
- Learner biased toward higher-weight training examples – if learner can use $p_\ell(\mathbf{x})$
- Error ε_ℓ of classifier h_ℓ used to bias $p_{\ell+1}(\mathbf{x})$
- Learn L classifiers
 - Each used to modify weights for next learned classifier
- Final classifier a weighted vote of individual classifiers



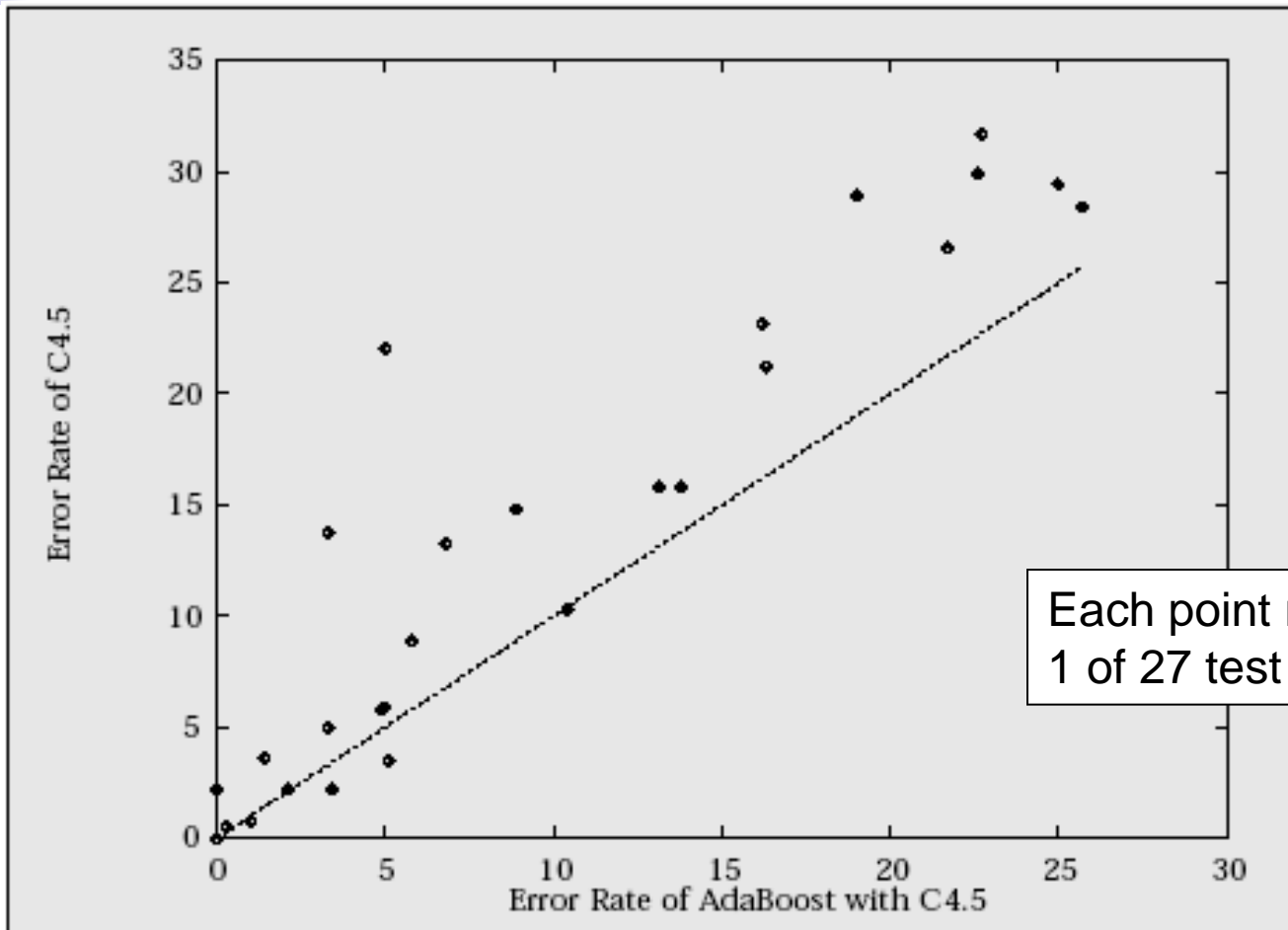
AdaBoost

Input: a set S , of m labeled examples: $S = \{(\mathbf{x}_i, y_i), i = 1, 2, \dots, m\}$,
labels $y_i \in Y = \{1, \dots, K\}$
LEARN (a learning algorithm)
a constant L .

```
[1] initialize for all  $i$ :  $w_1(i) := 1/m$  initialize the weights
[2] for  $\ell = 1$  to  $L$  do
[3]   for all  $i$ :  $p_\ell(i) := w_\ell(i) / (\sum_i w_\ell(i))$  compute normalized weights
[4]    $h_\ell := \text{LEARN}(p_\ell)$  call LEARN with normalized weights.
[5]    $\epsilon_\ell := \sum_i p_\ell(i) [h_\ell(\mathbf{x}_i) \neq y_i]$  calculate the error of  $h_\ell$ 
[7]   if  $\epsilon_\ell > 1/2$  then
[8]      $L := \ell - 1$ 
[9]     goto 13
[10]   $\beta_\ell := \epsilon_\ell / (1 - \epsilon_\ell)$ 
[11]  for all  $i$ :  $w_{\ell+1}(i) := w_\ell(i) \beta_\ell^{1 - [h_\ell(\mathbf{x}_i) \neq y_i]}$  compute new weights
[12] end for
```

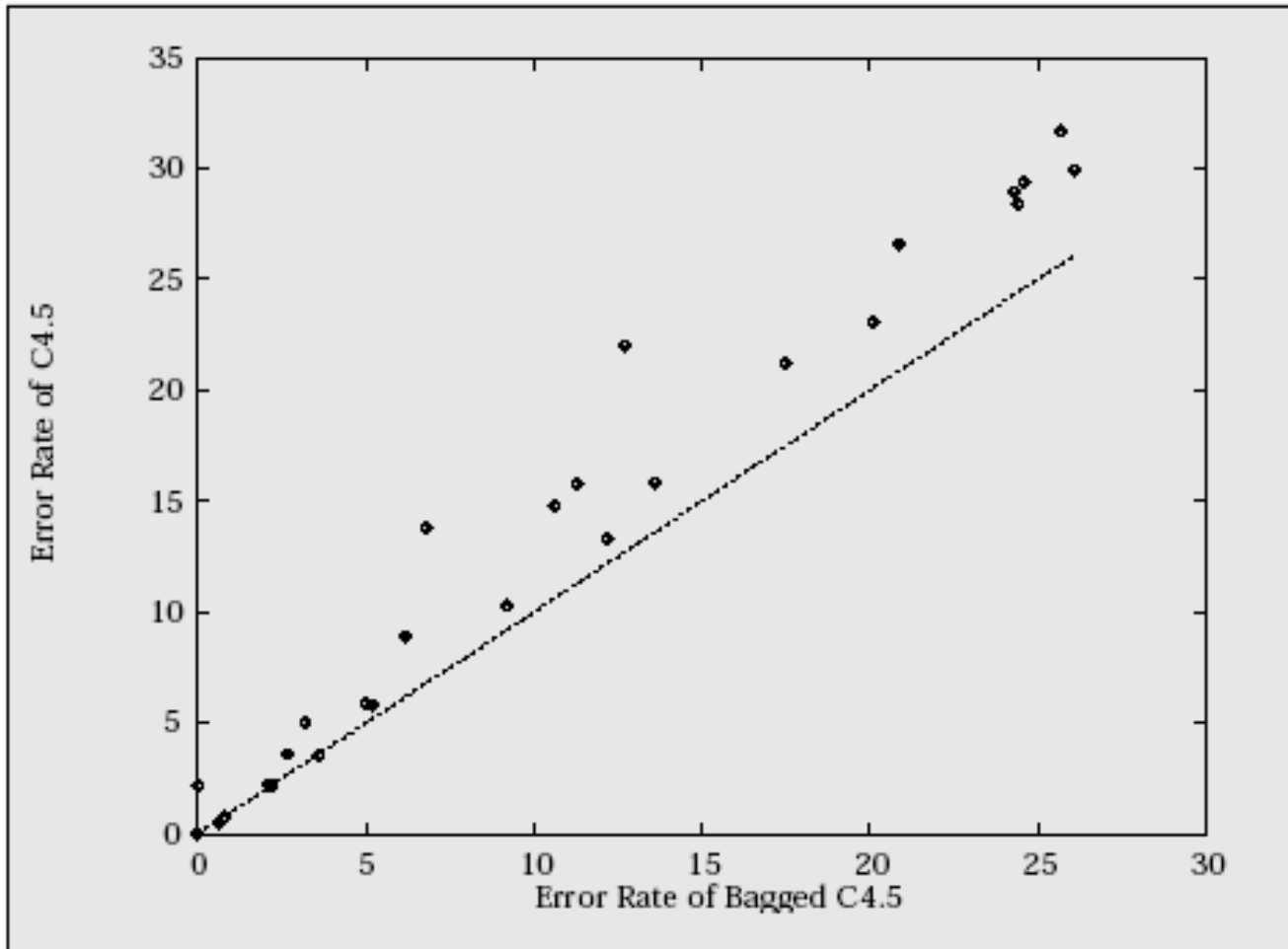
[13] **Output:** $h_f(\mathbf{x}) = \operatorname{argmax}_{y \in Y} \sum_{\ell=1}^L \left(\log \frac{1}{\beta_\ell} \right) [h_\ell(\mathbf{x}) = y]$

C4.5 with/without Boosting

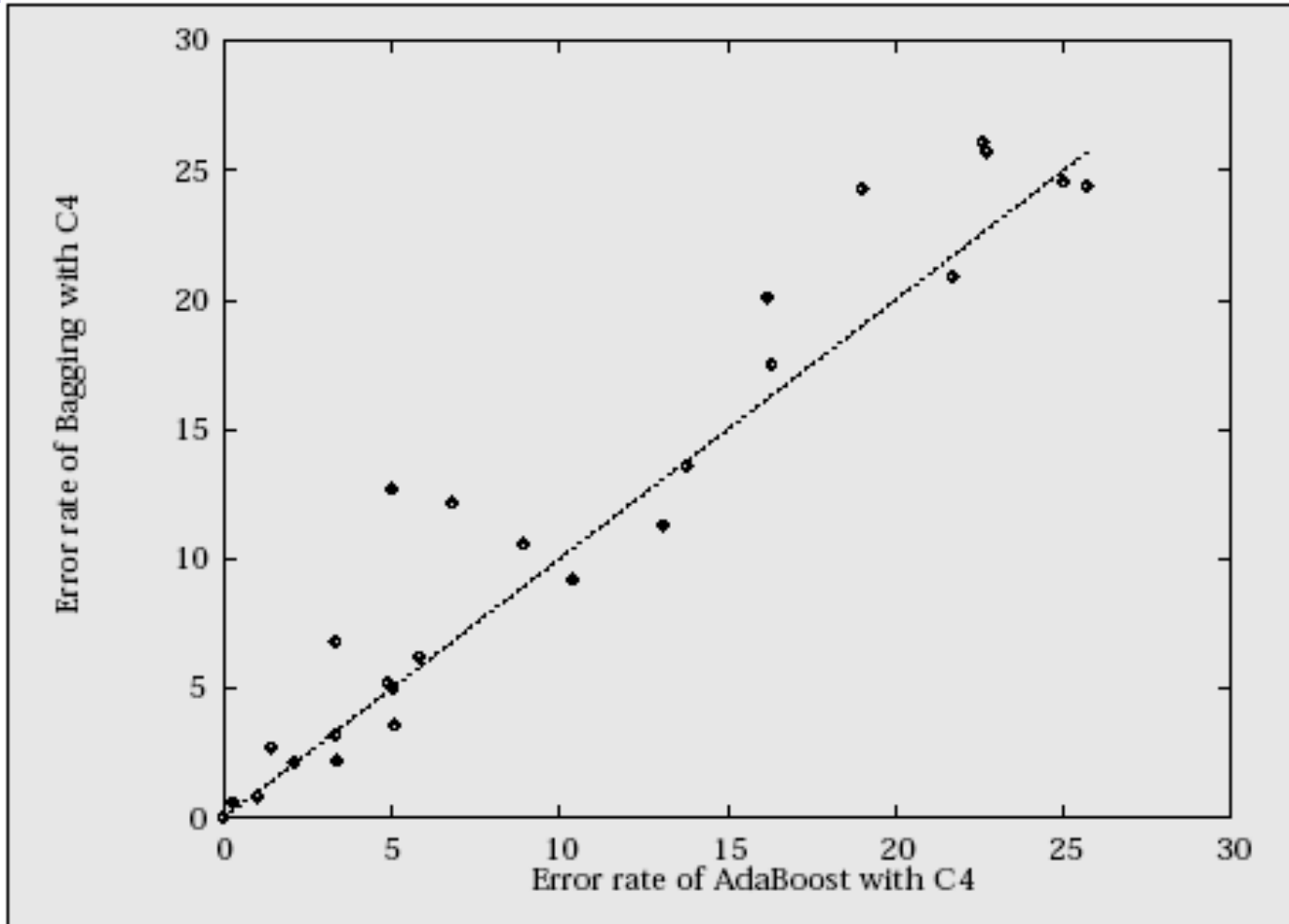


Each point represents
1 of 27 test domains.

C4.5 with/without Bagging



Boosting vs. Bagging





Constructing Ensembles

- Manipulating input features
 - Classifiers constructed using different subsets of features
 - Works only when some redundancy in features



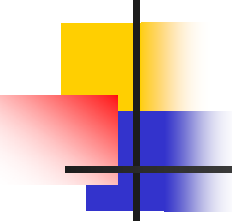
Constructing Ensembles

- Manipulating Output Targets
 - When large number K of classes
 - Generate L binary partitions of K classes
 - Generate L classifiers for these 2-class problems
 - Classify according to class whose partitions received most votes
 - Similar to error-correcting codes
 - Generally improves performance



Constructing Ensembles

- Injecting Randomness
 - Multiple neural nets with different random initial weights
 - Randomly-selected split attribute among top 20 in C4.5
 - Randomly-selected condition among top 20% in FOIL (Prolog rule learner)
 - Adding Gaussian noise to input features
 - Make random modifications to current h and use these classifiers weighted by their posterior probability (accuracy on training set)



Constructing Ensembles using Neural Networks

- Train multiple neural networks minimizing error and correlation with other networks' predictions
- Use a genetic algorithm to generate multiple, diverse networks
- Have networks also predict various sub-tasks (e.g., one of the input features)



Constructing Ensembles

- Use several different types of learning algorithms
 - E.g., decision tree, neural network, nearest neighbor
 - Some learners' error rates may be bad (i.e., > 0.5)
 - Some learners' predictions may be correlated
 - Need to check using, e.g., cross-validation



Combining Classifiers

- Unweighted vote
 - Majority vote
 - If h_ℓ produce class probability distributions $P(f(x)=k | h_\ell)$

$$P(f(x) = k) = \frac{1}{L} \sum_{l=1}^L P(f(x) = k | h_l)$$

- Weighted vote
 - Classifier weights proportional to accuracy on training data
- Learning combination
 - Gating function (learn classifier weights)
 - Stacking (learn how to vote)



Why Ensembles Work

- Uncorrelated errors made by individual classifiers can be overcome by voting
- How difficult is it to find a set of uncorrelated classifiers?
- Why can't we find a single classifier that does as well?



Finding Good Ensembles

- Typical hypothesis spaces H are large
- Need a large number (ideally $\lg(|H|)$) of training examples to narrow the search through H
- Typically, sample S of size $m \ll \lg(|H|)$
- The subset of hypotheses H consistent with S forms a good ensemble

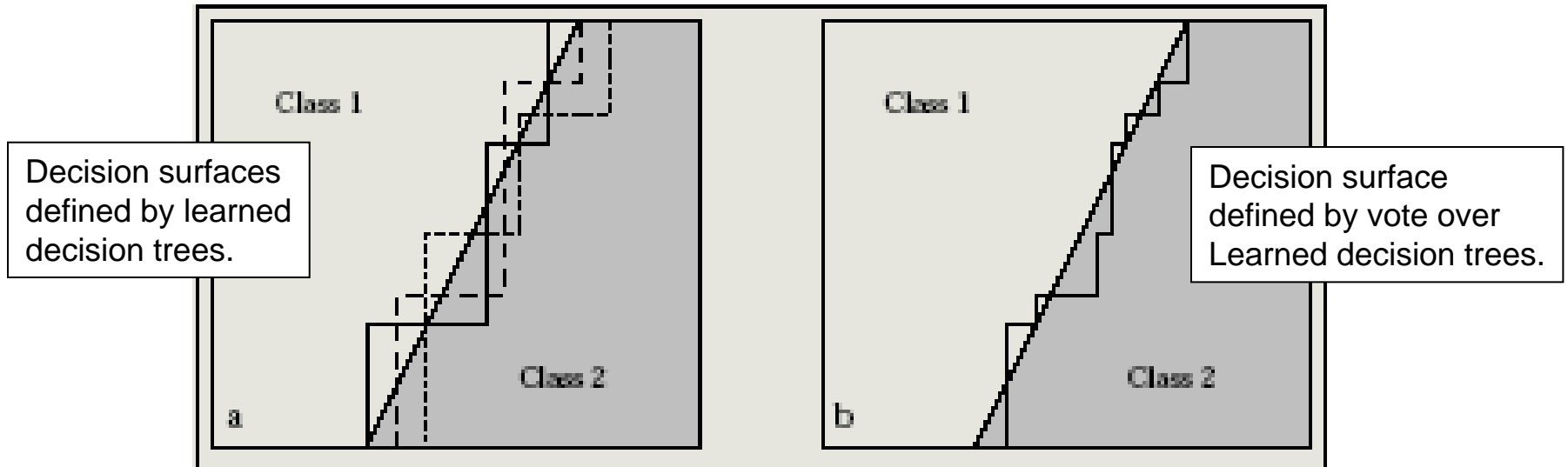


Finding Good Ensembles

- Typical learning algorithms L employ greedy search
- Not guaranteed to find optimal hypothesis (minimal size and/or minimal error)
- Generating hypotheses using different perturbations of L produces good ensembles

Finding Good Ensembles

- Typically, the hypothesis space H does not contain the target function f
- Weighted combinations of several approximations may represent classifiers outside of H





Summary

- Advantages

- Ensemble of classifiers typically outperforms any one classifier

- Disadvantages

- Difficult to measure correlation between classifiers from different types of learners
- Learning time and memory constraints
- Learned concept difficult to understand