Ensembles

Computer Science 760

Patricia J Riddle

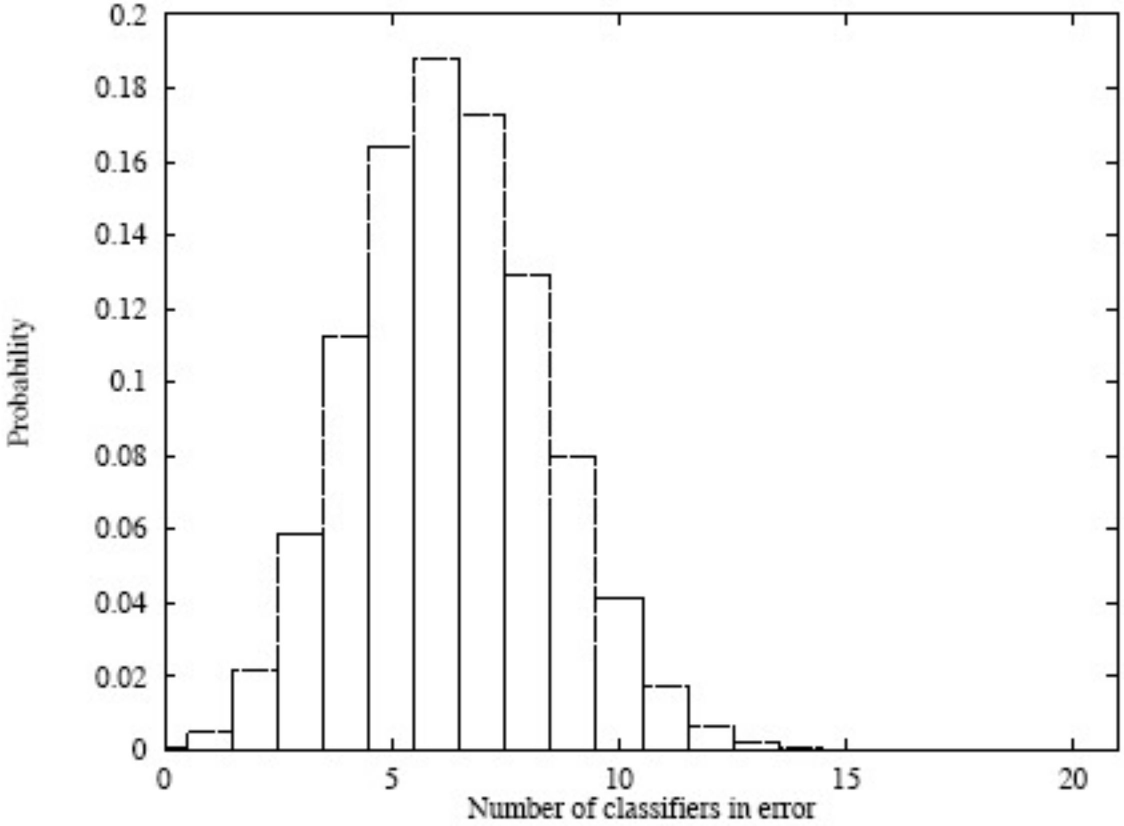
Ensembles of Classifiers

- An ensemble of classifiers is a set of classifiers whose individual decisions are combined in some way (typically weighted or unweighted voting) to classify new examples
- Ensembles are often much more accurate than the individual classifiers that make them up

Key to Ensembles

- An ensemble can only be more accurate than its component classifiers if the individual classifiers disagree with one another
- If individual hypotheses make uncorrelated errors at rates exceeding 0.5, then the error rate of the voted ensemble increases.
- Key: individual classifiers with error rates below 0.5 whose errors are at least somewhat uncorrelated

Probability that Majority Vote is Wrong



Constructing Ensembles

- Subsampling works especially well for unstable learning algorithms
- Bagging - bootstrap replicate - 63.2 percent
- Cross-validated committees
- Adaboost - adjusts probability distribution over training instances

Manipulating the Input Features - feature selection

- Volcanoes on Venus - 8 subsets of 119 input features and 4 network sizes

- Failure on sonar data - only works when input features are highly redundant

Manipulating the Output Target

- Error Correcting Output Coding - randomly partition K classes into two subsets A and B, learn a classifier, repeat the process L times
- Each member of each class receives a vote and the class with the most votes is the prediction of the ensemble
- Methods for designing good error-correcting codes can be applied
- Has been combined with Adaboost
- ECOC has also been combined with feature-selection

Injecting Randomness

- Different initial weights in ANN - didn’t perform as well as bagging and cross-validated committees
- Decision tree split criteria which chooses randomly among the best 20 tests at each node
- Others used weighted random choice
- In ANN bootstrap sampling of training data and adding Gaussian noise to the input features
- Markov chain Monte Carlo method - injecting randomness with vote proportional to posterior probability

Algorithm Specific Methods

- Backpropagation - train several networks simultaneously and use a correlation penalty in the error function
- Genetic operators to generate new network topologies - multiplicative term that incorporates the diversity of the classifiers - prune to N best networks

Algorithm Specific Methods II

- Training on auxiliary task as well as the main task - diverse classifiers can be learned with one primary task but with different auxiliary tasks such as predicting one of its input features
- Network whose secondary prediction is best is the winner - encourage different networks to become experts at predicting the auxiliary task in different local regions - causes the errors in the primary output to become decorrelated
- Decision Trees - option trees - equivalent and more understandable than bagging

Combining Different Algorithms

- Some perform much worse than others

- No guarantee of diversity

- Weighted combination

Combining Classifiers

1. Unweighted vote
   - bagging, ECOC - shown to be robust
   - probability estimate if classifier can produce class probability estimates
2. Many weighted voting methods:
   - Regression - weight should be inversely proportional to the variance of the estimates of h
   - Classification - weights proportional to accuracies
3. Learn good weights - gating function or gating network - overfitting problem
4. Stacking - use outputs of L classifiers as attributes for target in leave one out - good results in combining different forms of linear regression

Why Ensembles Work

- Why should it be possible to find ensembles of classifiers that make uncorrelated errors?
- Why shouldn’t we be able to find a single classifier that performs as well as an ensemble?

1. Statistical - Training data might not be sufficient - in 2 class problem need O(log(H)) examples minimum - many equally good hypothesis on the amount of data we have seen

Why Ensembles Work Visual

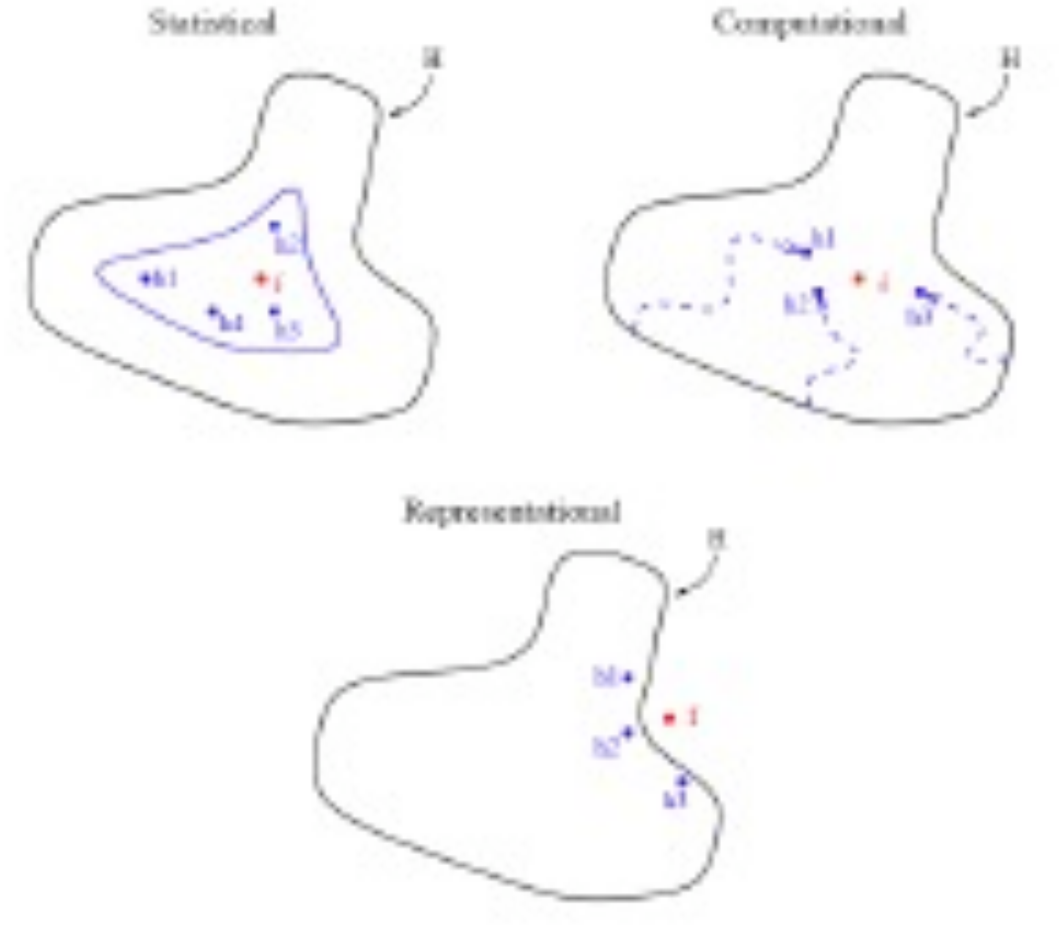


Fig. 2. Three fundamental reasons why an ensemble may work better than a single classifier

OR….

2. Computational - Difficult search problems - smallest decision tree consistent with the data, finding the weights for the smallest possible Neural Network consistent with the training data - NP-hard
   - Use search heuristics - so even if there is a unique best hypothesis we might not find it - so find suboptimal approximations
   - So ensembles combine different suboptimal approximations

OR…

3. Representational - Hypothesis space may not combine the true function - weighted combinations of approximations might be able to represent classifiers outside of H
    - Just complex decision trees but way too large for the available data

Decision Boundary

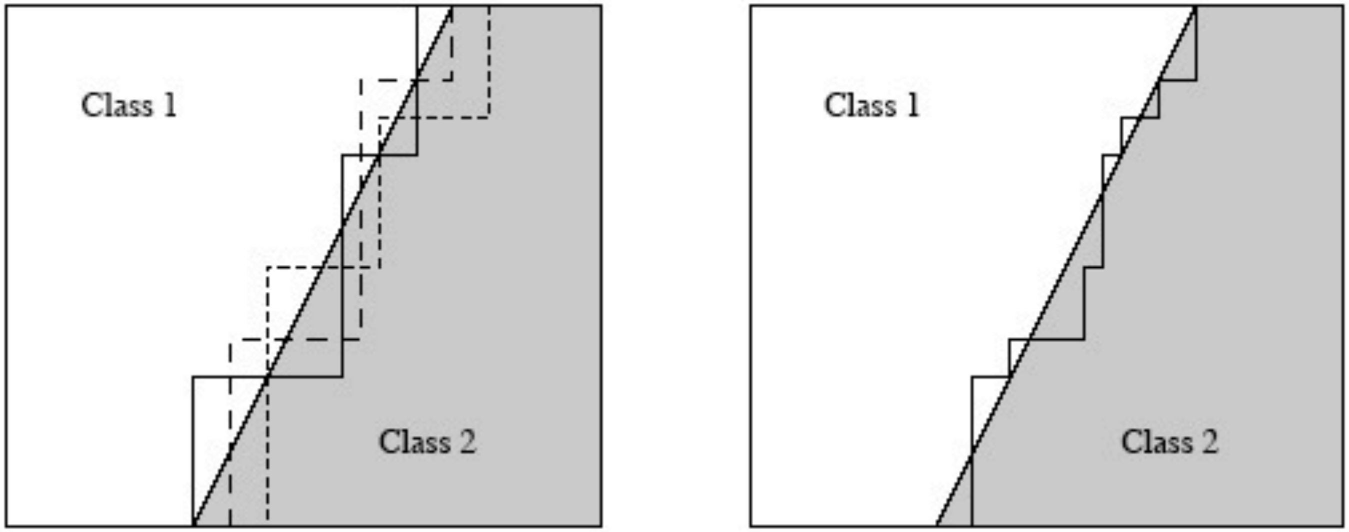


Fig. 4. The left figure shows the true diagonal decision boundary and three staircase approximations to it (of the kind that are created by decision tree algorithms). The right figure shows the voted decision boundary, which is a much better approximation to the diagonal boundary.

Michael Goebel’s PhD Thesis

- Comparison of cross-validated communities and bagging
- One was better sometimes and the other sometimes
- Why?
  - Size of the dataset with respect to the hypothesis space
- 1/2 bag and double bag
- Why is bagging always better?

Open Problems

1. When to use which ensemble methods - Adaboost best except when there is noisy data - so NEVER
2. Bagging and ECOC combined perform better than either separately - other combinations should be explored
3. Few systematic studies of ensembles on ANN and rule-learning systems
4. Ensembles take a lot of memory - can they be converted to less redundant representations?
5. Ensembles provide little insight - can we obtain explanations from ensembles?