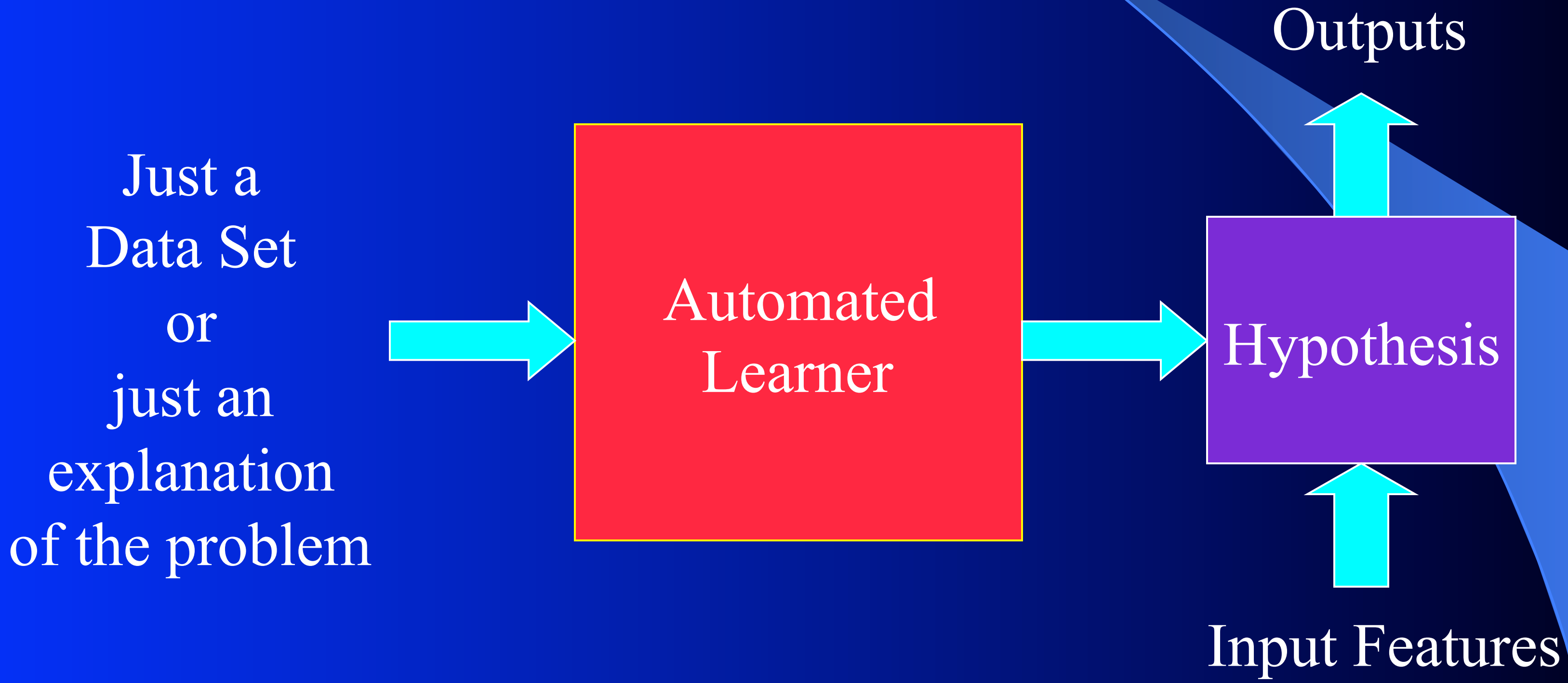
Ensembles

A “Holy Grail” of Machine Learning



Just a
Data Set
or
just an
explanation
of the problem

Automated Learner

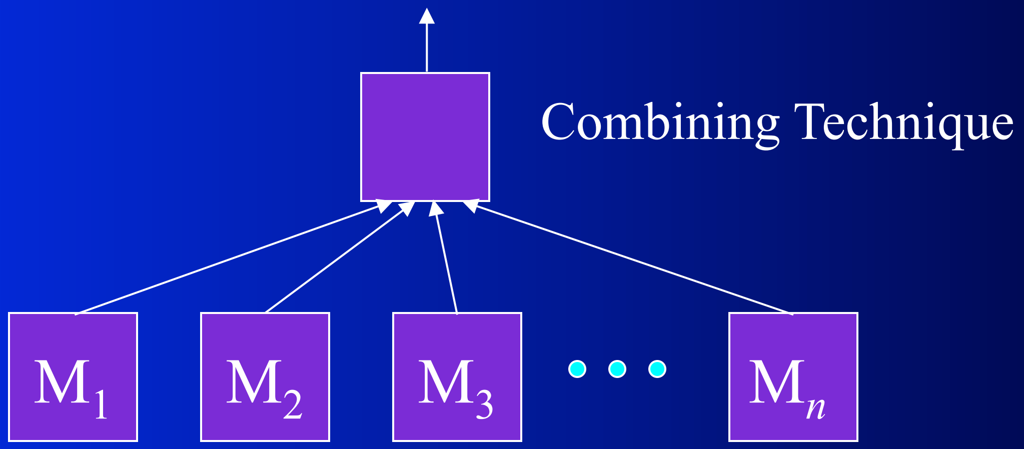
Hypothesis

Outputs

Input Features

Ensembles

- Multiple diverse models (Inductive Biases) are trained on the same problem and then their outputs are combined to come up with a final output
- The specific overfit of each learning model can be averaged out
- If models are diverse (uncorrelated errors) then even if the individual models are weak generalizers, the ensemble can be very accurate
- Many different Ensemble approaches
  - Stacking, Gating/Mixture of Experts, Bagging, Boosting, Wagging, Mimicking, Heuristic Weighted Voting, Combinations



Ensembles are Scriptural

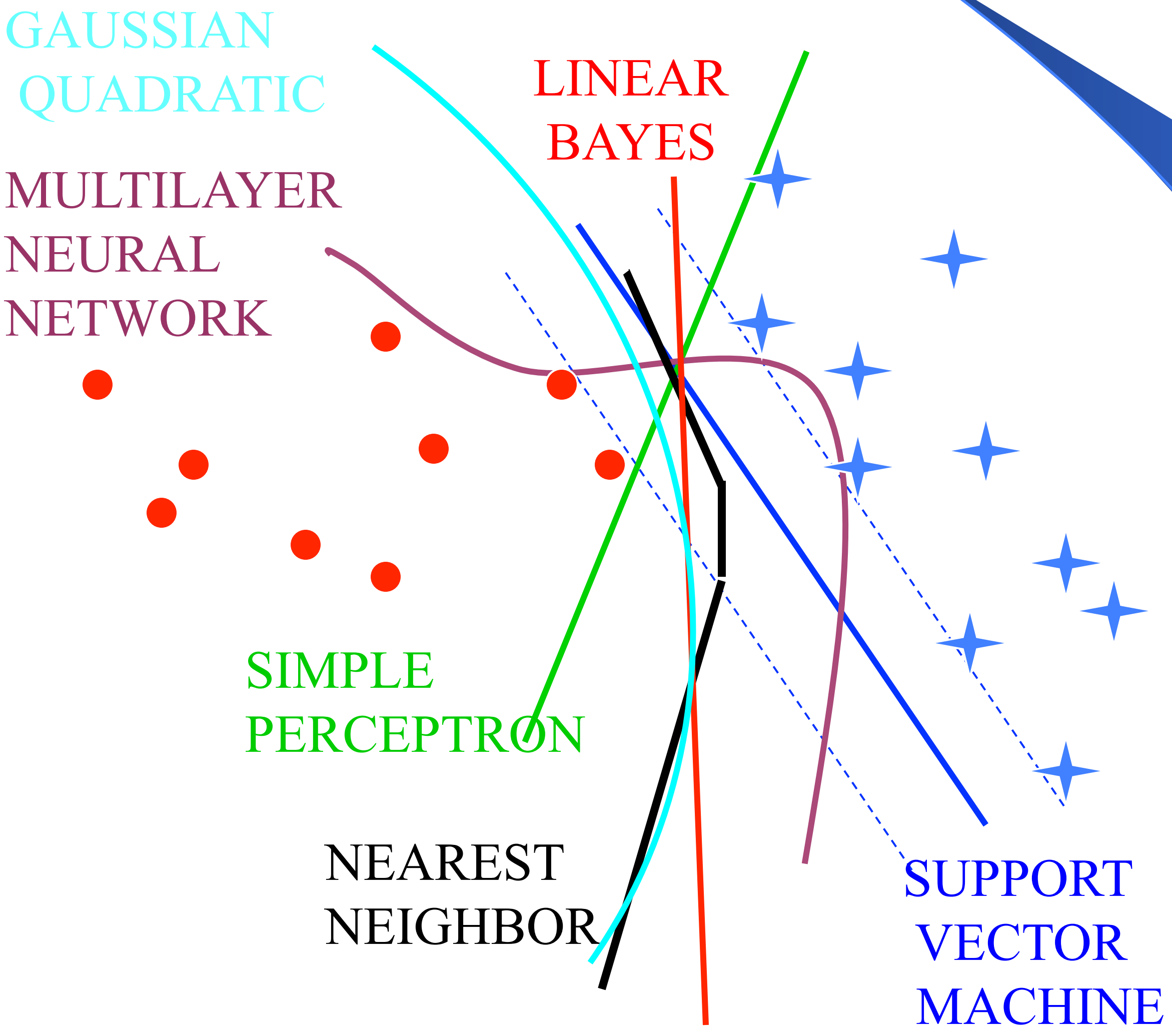
Mosiah 29:26, 27 Now it is not common that the voice of the people desireth anything contrary to that which is right; but it is common for the lesser part of the people to desire that which is not right; therefore this shall ye observe and make it your law--to do your business by the voice of the people.

And if the time comes that the voice of the people doth choose iniquity, then is the time that the judgments of God will come upon you; yea, then is the time he will visit you with great destruction even as he has hitherto visited this land.

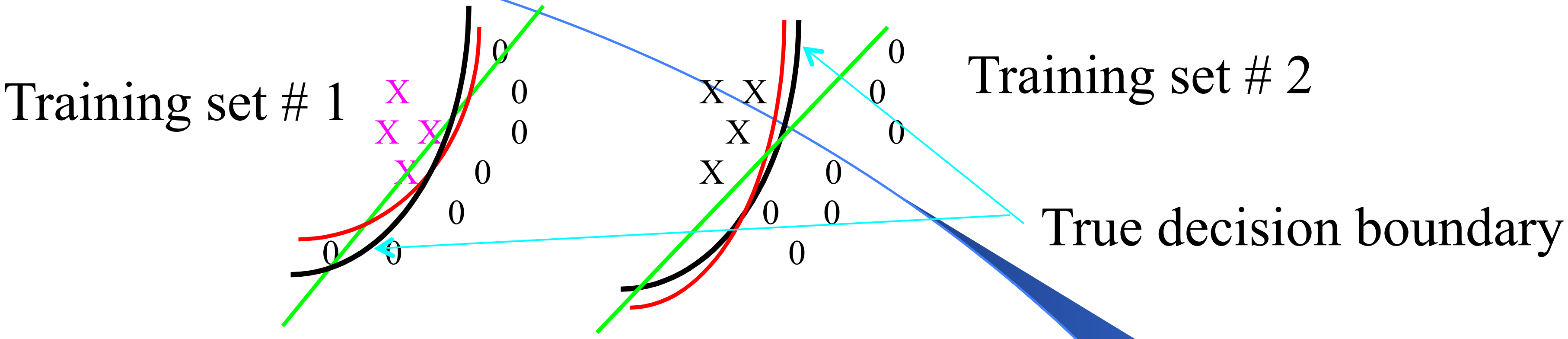
Bias vs. Variance

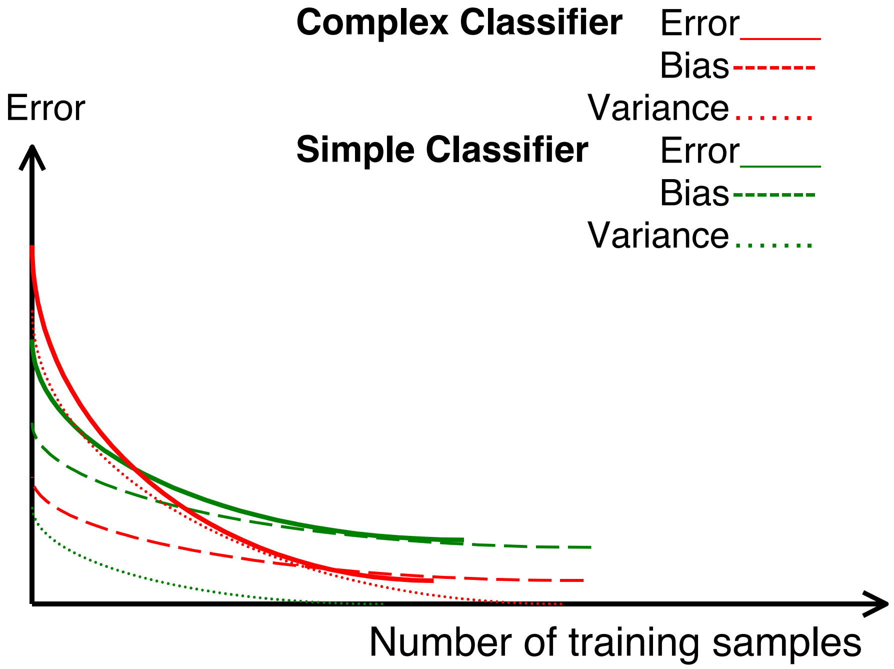
- Learning models can have error based on two basic issues: Bias and Variance
  - "Bias" measures the basic capacity of a learning approach to fit the task
  - "Variance" measures the extent to which different hypotheses trained using a learning approach will vary based on initial conditions, training set, etc.
- MLPs trained with backprop have lower bias error because they can fit the task well, but have relatively high variance error because each model might fall into odd nuances (overfit) based on training set choice, initial weights, and other parameters – Typical with the more complex models we want
- Naïve Bayes has high bias error (doesn't fit that well), but has no variance error.
- We would like low bias error and low variance error
- Ensembles using multiple trained models with high-variance and low-bias error can average out the variance, leaving just the bias
  - Less worry about overfit with the base models (stopping criteria, etc.)

Some classifiers



CLASSIFIER BIAS AND VARIANCE





CLASSIFIER BIAS AND VARIANCE DON'T ADD!

Any classifier can be shown to be better than any other.

Amplifying Weak Learners

- Combining weak learners
  - Assume *n* induced models which are independent of each other with each having accuracy of about 60% on a two class problem. While one model is not dependable, if a good majority of a group of these lean in one direction, then we can have high confidence.
  - If all *n* give the same class output then you can be confident it is correct with probability 1-(1-.6)^n. For *n*=10, confidence would be 99.4%.
  - Normally not independent (e.g. similar training sets). If all *n* were the same model, then no advantage could be gained.
  - Also, unlikely that all *n* would give the same output, but if a majority did, then still get an overall accuracy better than the base accuracy of the models
  - If *m* models say class 1 and *w* models say class 2, then *P*(majority_class) = 1 – Binomial(*n*, min(*m*,*w*), .6)

$$P(r) = \frac{n!}{r!(n-r)!} p^r (1-p)^{n-r}$$

Bagging

- Bootstrap aggregating (Bagging)
- Induce *m* learners starting with the same initial parameters with each training set chosen uniformly at random with replacement from the original data set, training sets might be 2/3^{rds} of the data set – still need to save some separate data for testing
- All *m* hypotheses have an equal vote for classifying novel instances
- Great way to improve overall accuracy by decreasing variance. Consistent significant empirical improvement.
- Does not overfit (whereas boosting may), but may be more conservative overall on accuracy improvements
- Bigger *m* the better (diminishing), but need to consider efficiency trade-off
- Often used with the same learning algorithm and thus best for those which tend to give more diverse hypotheses based on initial random conditions
- Could use other schemes to improve the diversity between learners
  - Different initial parameters, sampling approaches, etc.
  - Different learning algorithms
  - The more diversity the better - (yet most often used with the same learning algorithm and just different training sets)

Boosting

- Boosting by resampling - Each TS_t is chosen randomly with distribution D_t with replacement from the original training data. D_1 has all instances equally likely to be chosen. Typically each TS_t is the same size as the original data set.
  - Induce first model. Change D_{t+1} so that instances which are mis-classified by the current model on its current TS have a higher probability of being chosen for future training sets.
  - Keep training new models until stopping criteria met
    - *M* models induced
    - Overall Accuracy levels out
    - Most recent model has accuracy less than 50% on its TS
- All models vote, but each model's vote is scaled by its accuracy on the training set it was trained on
- Boosting is more aggressive than bagging on accuracy but in some cases can overfit and do worse – can theoretically converge to training set
  - On average better than bagging, but worse for some tasks
  - In rare cases can be worse than the non-ensemble approach

Boosting

- Another approach to boosting is to have each base model train on the entire training set but have the ML algorithm take each current instance weighting into account during learning.
- How might you do that for
  - MLPs
  - Decision Trees
  - *k*-NN
- Then still have each model vote weighted by its overall accuracy

Boosting

- Another approach to boosting is to have each base model train on the entire training set but have the ML algorithm take each current instance weighting into account during learning.
- How might you do that for
  - MLPs – Scale learning rate by weight
  - Decision Trees – instance membership is scaled by weight
  - *k*-NN – node vote is scaled by weight
- Then still have each model vote weighted by its overall accuracy

Ensemble Creation Approaches

- A good goal is to get less correlated errors between models
- Injecting randomness – initial weights, different learning parameters, etc.
- Different Training sets – Bagging, Boosting, different features, etc.
- Forcing differences – different objective functions, auxiliary tasks
- Different machine learning models
  - Obvious, but surprisingly it is less used
- One aspect of *COD* (Classifier Output Distance) research - which algorithms are most different and thus most appropriate to ensemble

Ensemble Combining Approaches

- Unweighted Voting (e.g. Bagging)
- Weighted voting – based on accuracy, etc. (e.g. Boosting)
- Stacking - Learn the combination function
  - Higher order possibilities
  - Which algorithm should be used for the stacker
  - Must match the input/output data types between models
  - Stacking the stack, etc.
- Gating function/Mixture of Experts – The gating function uses the input features to decide which expert or combination (weights) of experts to use in the vote with experts being strong in different part of the input space
- Heuristic Weighted Voting – differs for each instance

Ensemble Summary

- Other Models – Random Forests, Boosted stumps, Cascading, Arbitration, Delegation, PDDAGS (Parallel Decision DAGs), Bayesian Model Averaging and Combination, Clustering Ensemble, etc.
- Efficiency Issues
  - Wagging (Weight Averaging) - Multi-layer?
  - Mimicking - Oracle Learning, semi-supervised
- Great way to decrease variance/overfit
- Almost always gain accuracy improvements with Ensembles