

# CMSC 726

## Lecture 6: Ensemble Methods

Lise Getoor  
October 26, 2010

**ACKNOWLEDGEMENTS:** The material in this course is a synthesis of materials from many sources, including: Hal Daume III, Mark Drezde, Carlos Guestrin, Andrew Ng, Ben Taskar, Eric Xing, and others. I am very grateful for their generous sharing of insights and materials.

# Today's Topics

- ▶ Ensemble Methods

- Boosting
- Reading, Bishop section 14.3
- Optional, Boosting Survey by Shapire on elms (focus on first portion)

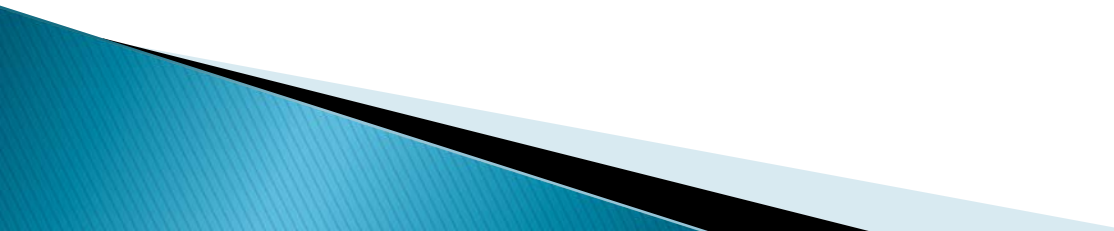
# Combining Classifiers

- ▶ Building on the outputs of supervised models
  - How can be combine outputs?
  - Approach: build complex models from simpler models
- ▶ Can I learn many different classifier using a supervised learning algorithm and combine them to get a better classifier?
  - Yes
- ▶ Ensemble learning
  - Combine an ensemble of classifiers

# Weak Learning

- ▶ Let's say I have a weak learner
  - A learning algorithm that is marginally better than random
- ▶ Can I take this weak learner and make it better?
  - Can I combine weak learners to make a strong learner?

# Weak Learning Example

- ▶ Consider an auto mechanic
    - You want to learn how to fix your car
    - Approach 1: Ask the mechanic to teach you how to diagnose car problems?
      - Problem: very complicated, unlikely to get a good answer
    - Approach 2: Show the mechanic a car with a problem
      - Ask how to fix this problem
      - Problem: you'll get a good answer, but only for that problem
      - Solution: repeat until you get enough answers to cover all scenarios
- 

# Weak Learning Example

- ▶ Weak learner: the 1 rule the mechanic teaches you for each car problem
- ▶ Strong learner: the full knowledge of how to fix cars
- ▶ Can we use the weak learner to build a strong learner?
  - If we ask the mechanic enough questions, will we become an expert?

# Boosting

- ▶ Let's say I have a weak learner, can I take this weak learner and make it better?
- ▶ Question first proposed in 1988:
  - “Does weak learnability imply strong learnability?”
    - Asked about PAC learning, Kearns and Valiant, 1988
- ▶ Answer came in 1989 by Schapire
  - Yes! “boost” a weak learner to get a strong learner!

# Some Boosting History

- ▶ Schapire 1989
  - First boosting algorithm, cited 1565 times on Google Scholar
  - Show slight improvements in theory
- ▶ Freund 1990
  - An optimal algorithm that boosts by majority
- ▶ Drucker, Schapire and Simard 1992
  - First experiments using boosting
  - Limited by practical considerations
- ▶ Freund and Schapire 1996
  - AdaBoost– the first practical boosting algorithm
    - Cited 3034 times on Google Scholar
  - The rest is history



# General Boosting Approach

- ▶ Look at subset of data
  - Make simple rule to classify data (weak)
  - Repeat (make many rules)
- ▶ Boosting:
  - Boost weak rules into strong predictors

# A Formal View of Boosting

- ▶ Given training set  $\{x_i, y_i\}_{i=1}^m$  and weak learner  $f$
- ▶ Binary labels  $y_i \in \{-1, +1\}$
- ▶ For each boosting iteration
  - Construct distribution  $D_t$  on  $m$  examples
  - Learn weak hypothesis  $h_t$  using  $f$  with error:
$$\varepsilon_t = P_{D_t}[h_t(x_i) \neq y_i]$$
- ▶ Output final hypothesis

# Questions

- ▶ How do we choose subsets of the data?
- ▶ How do we combine all the rules into predictor?
- ▶ The answer: AdaBoost

# AdaBoost

- ▶ Given:  $\{x_i, y_i\}_{i=1}^N$  where:  $y_i \in \{-1, +1\}$ 
  - Initialize  $D_1(i) = 1/m$

# AdaBoost

- ▶ For each iteration  $t=1$  to  $T$ :
  - Train weak learner using distribution  $D_t$
  - Get weak hypothesis  $h_t$  with error  $\varepsilon_t = P_{D_t}[h_t(x_i) \neq y_i]$
  - Choose  $\alpha_t = \frac{1}{2} \log\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$
  - Update

$$\begin{aligned} D_{t+1}(i) &= \frac{D_t(i)}{Z_t} \times \begin{cases} e^{-\alpha} & \text{if } h_t(x_i) = y_i \\ e^{\alpha} & \text{if } h_t(x_i) \neq y_i \end{cases} \\ &= \frac{D_t(i) \exp(-\alpha y_i h_t(x_i))}{Z_t} \end{aligned}$$

- $Z_t$  is a normalization constant

# AdaBoost

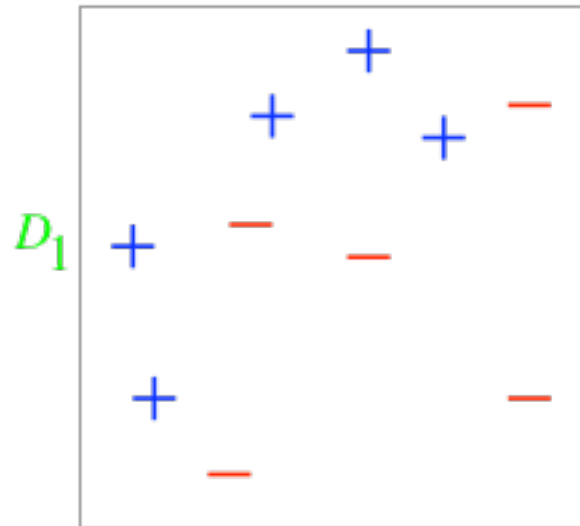
- ▶ Output the final hypothesis:

$$H(x) = \text{sign}\left(\sum_{t=1}^T \alpha_t h_t(x)\right)$$

# Notes

- ▶  $\alpha_t$  measures importance assigned to  $h_t$
- ▶ As  $\alpha$  gets larger, error gets smaller
- ▶ Weight tends to concentrate on hard examples
  - Sound familiar?
    - SVMs!
    - Weight on examples close to the margin
    - We'll come back to this point

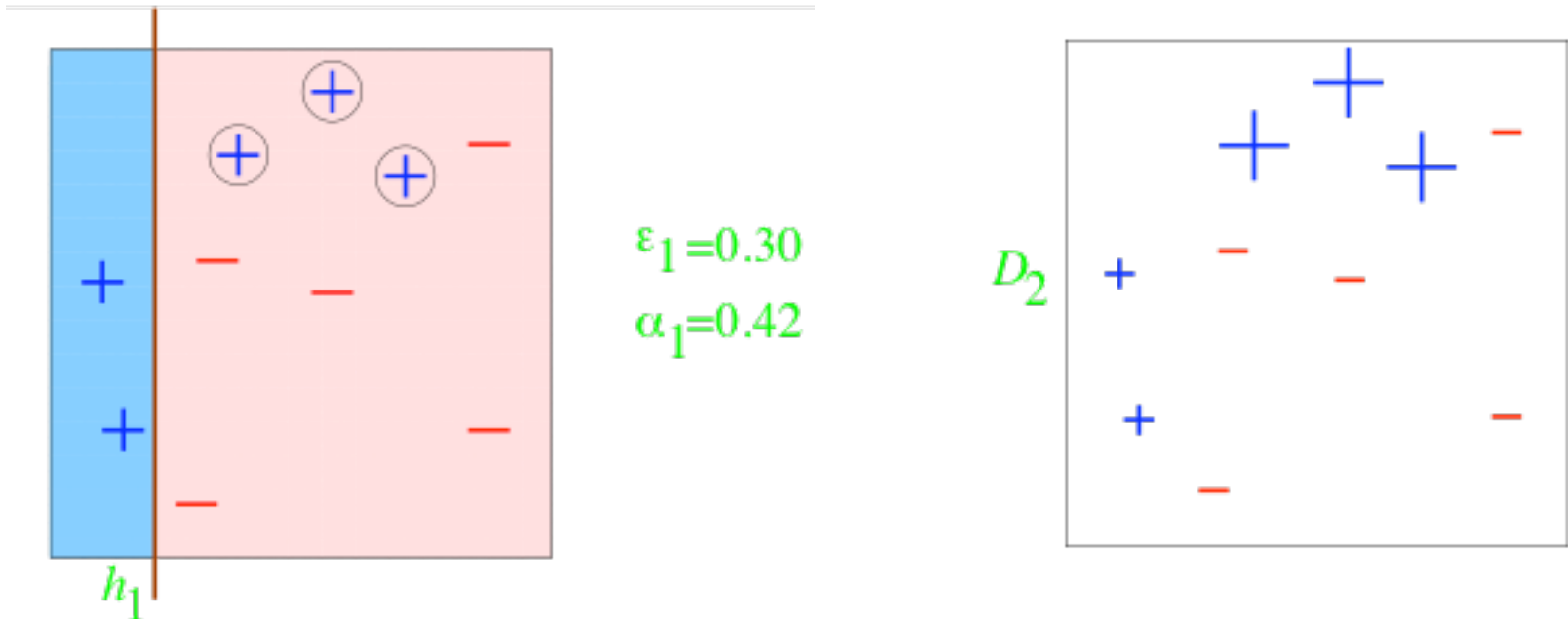
# Example



- ▶ A set of labeled points with a uniform distribution

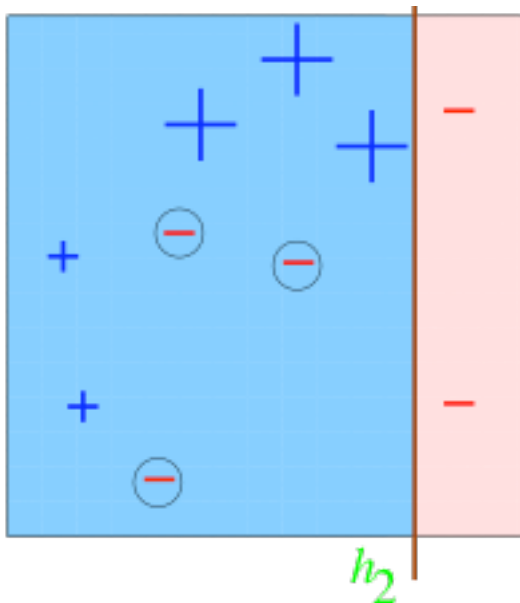


# Round 1

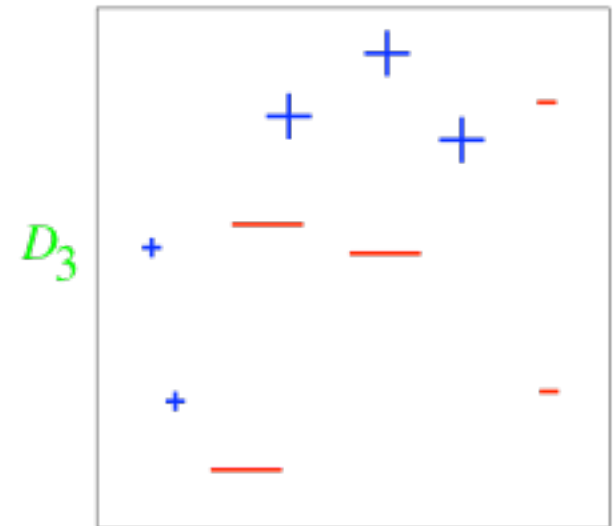


- ▶ Learn hypothesis, measure error, set  $\alpha$
- ▶ Recompute distribution placing more weight on incorrect examples

# Round 2

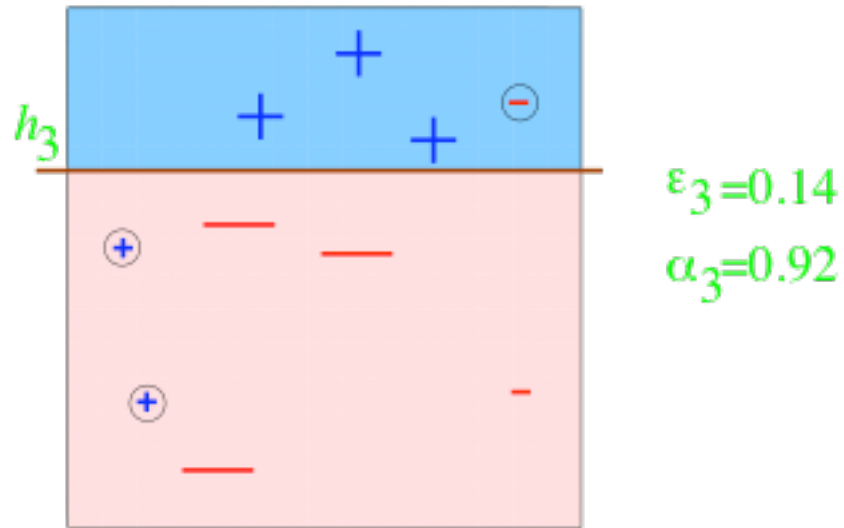


$$\epsilon_2 = 0.21$$
$$\alpha_2 = 0.65$$



- ▶ Learn hypothesis, measure error, set  $\alpha$
- ▶ Recompute distribution placing more weight on incorrect examples

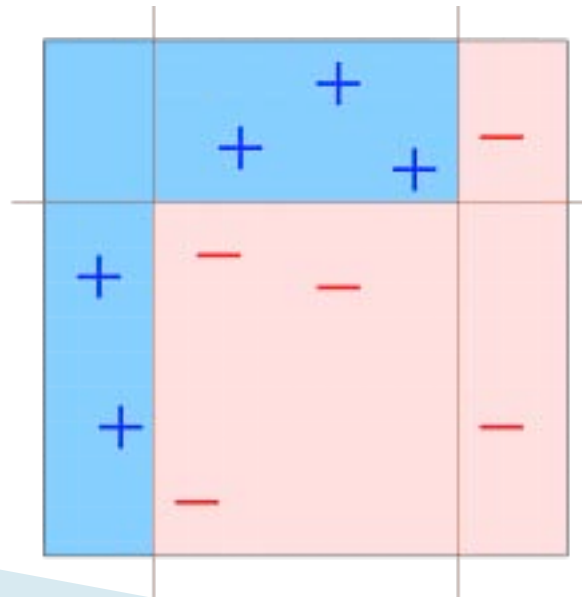
# Round 3



# Final Model

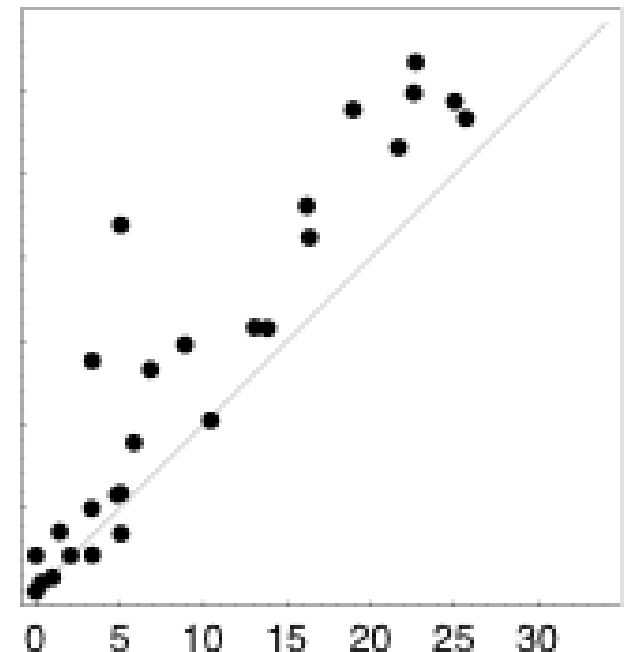
$H_{\text{final}}$

$$= \text{sign} \left( 0.42 \begin{array}{|c|} \hline \text{blue} \\ \hline \text{red} \end{array} + 0.65 \begin{array}{|c|} \hline \text{blue} \\ \hline \text{red} \end{array} + 0.92 \begin{array}{|c|} \hline \text{blue} \\ \hline \text{red} \end{array} \right)$$



# Why is Boosting Good?

- ▶ Boosting achieves good empirical results
- ▶ Why?
- ▶ Many answers
  - Statistical View of Boosting
  - Boosting and Max Margin
  - PAC Learning (learning theory)
  - Game theory



Boosting Stumps error (X axis) vs C4.5 Decision Tree error (Y axis)  
Points above the line are better for boosting

# Statistical View of Boosting

- ▶ What is boosting doing?
  - We normally think of classifiers in terms of loss or likelihood
  - What is the objective function for boosting?

# Generalization Bound

- ▶ Schapire and Singer showed a generalization bound for AdaBoost

$$\frac{1}{m} |i : H(x_i) \neq y_i| \leq \frac{1}{m} \sum_i \exp(-y_i f(x_i)) \quad \begin{aligned} f(x) &= \sum \alpha_t h_t(x) \\ H(x) &= \text{sign}(f(x)) \end{aligned}$$

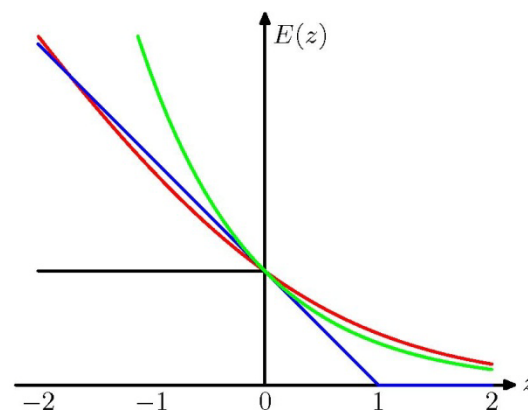
- ▶ Therefore, AdaBoost is trying to find a linear combination  $f$  of base classifiers which minimize

$$\sum_i \exp(-y_i f(x_i)) = \sum_i \exp\left(-y_i \sum_t \alpha_t h_t(x_i)\right)$$

- ▶ On each round, AdaBoost sets  $h$  and chooses  $\alpha$  to add another term to this summation so as to maximally reduce the sum of the exponents
- ▶ Conclusion: AdaBoost is optimizing this objective

# Exponential Loss

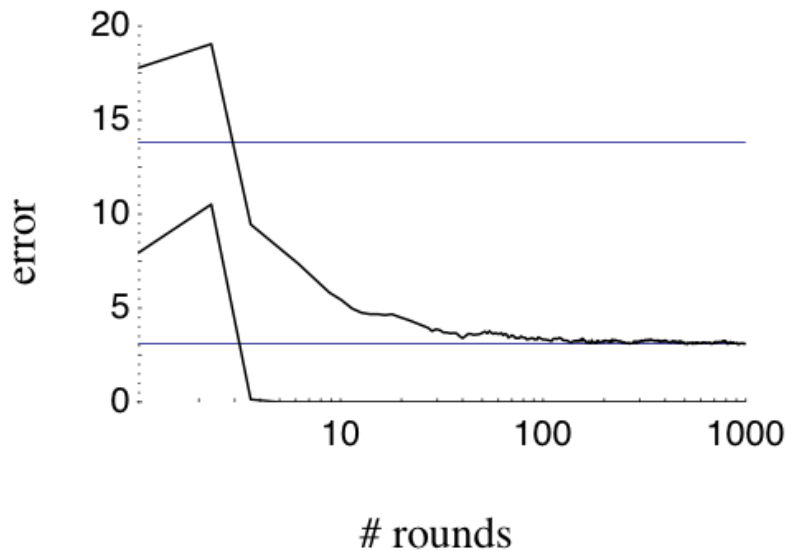
- ▶ What is  $\sum \exp(-y_i f(x_i))$ ?
  - Exponential loss – in green



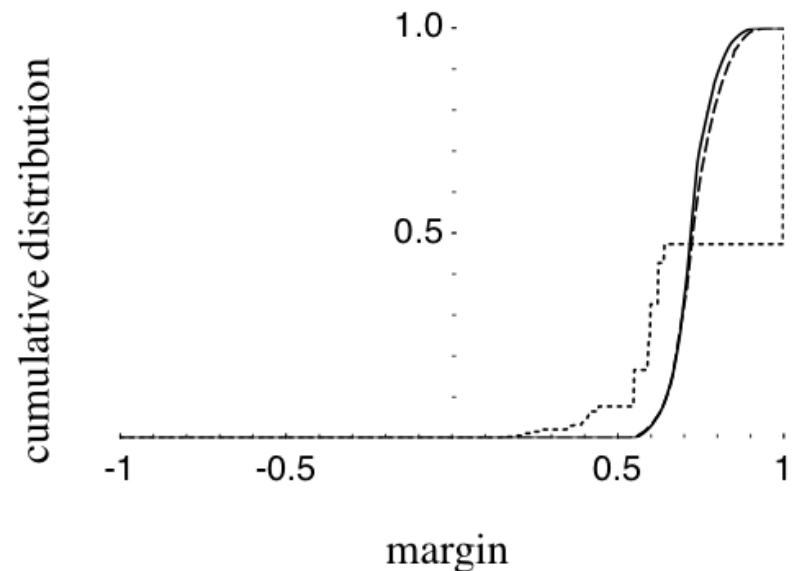
- Can be shown that this upper bounds the negative log likelihood of the weighted average of logistic regression models
  - Hastie, Tibshirani and Friedman, 2000



# Connections to Margins



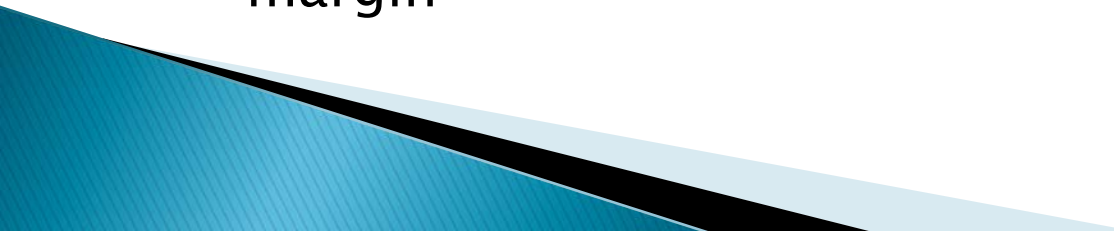
Learning curve for training (lower) and test (upper)



Cumulative distribution of margins of the training examples  
5,100,1k iterations (dashed to solid)

Boosting increases the margin even after training error goes to 0

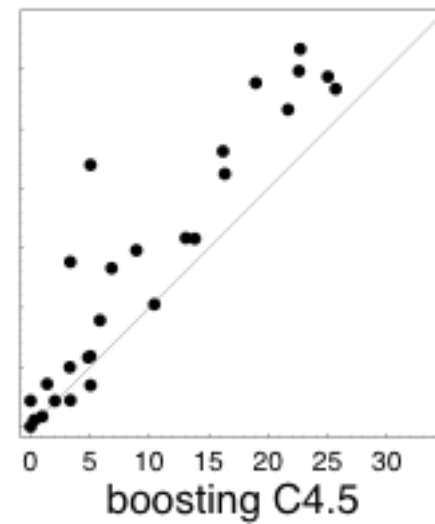
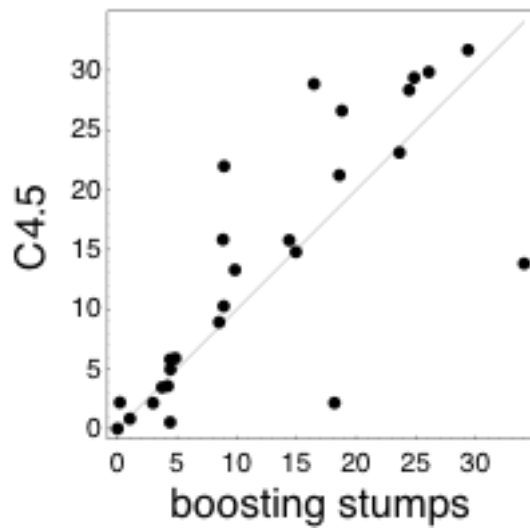
# Differences

- ▶ The norms are different
    - In high dimensional space, the effective enforced margins may be very different
  - ▶ SVM requires quadratic programming
    - Boosting requires linear programming
  - ▶ Finding high dimensional separators
    - SVM uses kernels (inner products of examples)
    - Boosting uses weak learners to handle this
    - There is usually a big difference between the learning spaces of the weak learners and the kernels
  - ▶ This isn't the whole story
    - Only part of the bound corresponds to maximizing the margin
- 

# What Should We Boost?

- ▶ If boosting improves a base classifier...
  - Boost the best classifier we can!
    - Boosted SVMs, KNN, Perceptrons, Logistic Regression, etc.
  - Problem: you have to train many of these, may not be efficient
- ▶ But boosting can boost a *weak* learner
  - For simplicity, let's use a weak learner
  - Decision stumps: very weak, can only look at 1 feature at a time
    - Decision trees with depth of 1

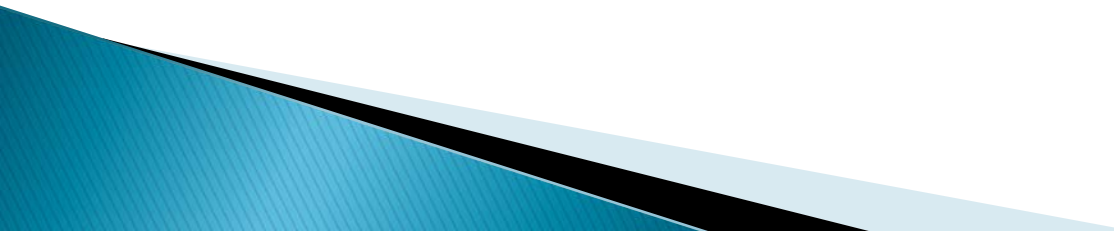
# Boosting Results



# Practical Advantages

- ▶ Easy to implement
- ▶ Very fast
- ▶ No parameters to tune
- ▶ Not specific to any weak learner
- ▶ Well motivated by learning theory
- ▶ Can identify outliers
- ▶ Extensions to:
  - Multi-class, Ranking, Regression

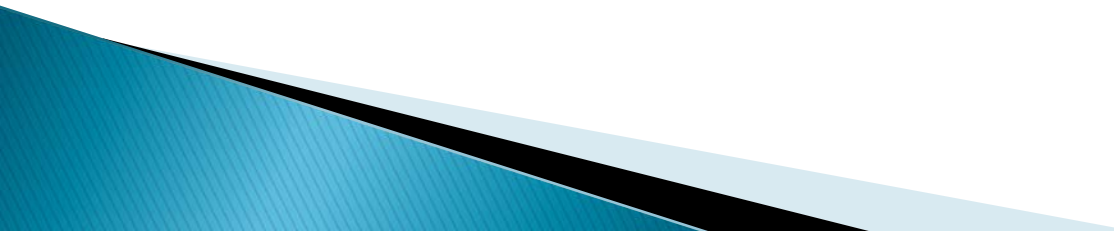
# Other approaches in brief

- ▶ We've talked about several ways to combine
  - ▶ What other are combinations good?
  - ▶ An example: you want to get advice about which stocks to invest in
    - What should you do?
    - Call the same stock broker 100 times and average?
    - Call 100 different stock brokers and average?
  - ▶ Mixture of Experts
- 

# Diversity in Experts

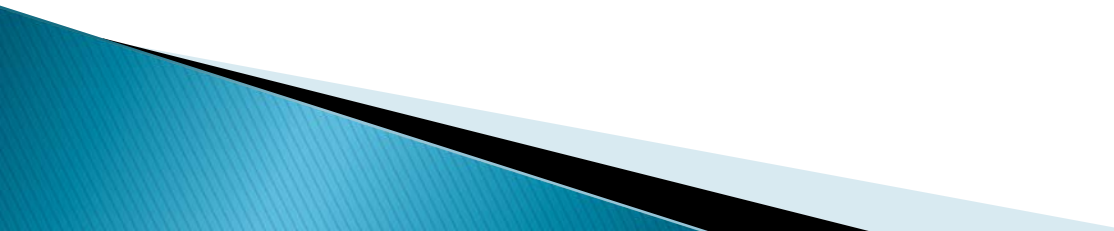
- ▶ We can improve by combining multiple classifiers since they have a diversity of opinions
  - They won't all make the same mistakes
  - If they are all very good, then we can vote them to get even better
    - Reality: they do make some of the same mistakes
  - This is the idea behind boosting
    - If you can do a bit better than random (weak), then you can boost that to good performance (strong)

# Creating Diversity

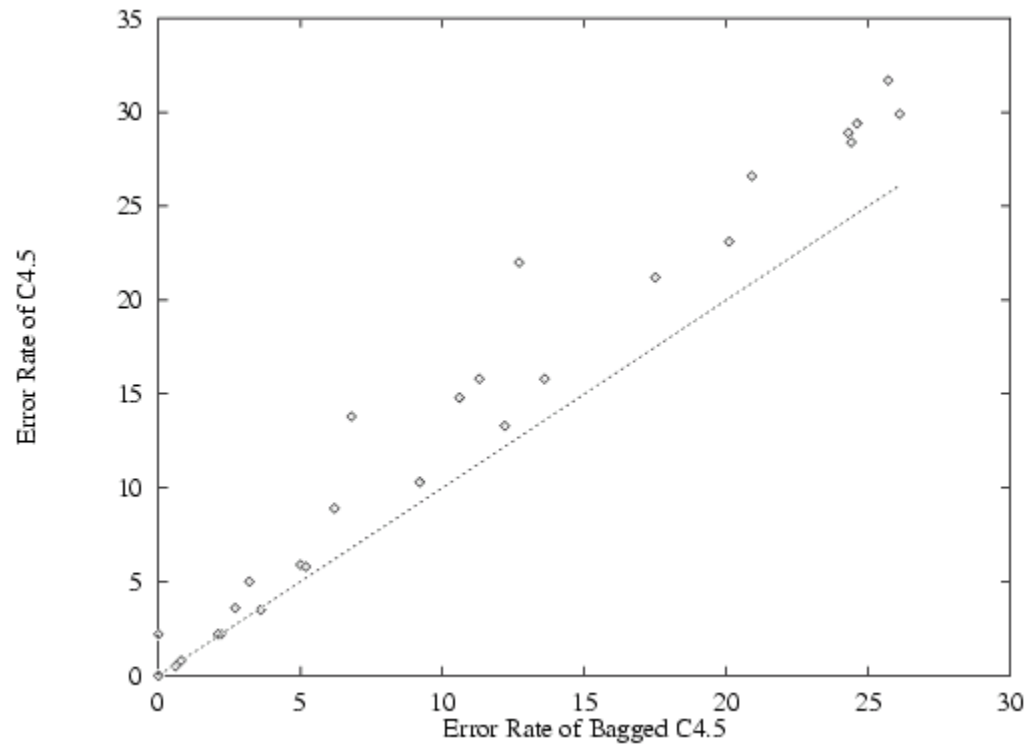
- ▶ We can create diversity by using  $K$  different classifiers
  - ▶ We can also create diversity by creating  $K$  different *datasets*
  - ▶ Bagging: create many different datasets by hiding some of the data.
    - Instance bagging
    - Feature bagging
- 



# Instance Bagging

- ▶ Given  $m$  examples for training
    - Create  $K$  datasets
      - Select  $m$  examples with *replacement* from the training set
      - Train a classifier on the dataset
  - ▶ Final output: voting of the  $K$  classifiers
    - Or: weighted majority of the  $K$  classifiers
  - ▶ Called bootstrap aggregation in statistics
- 

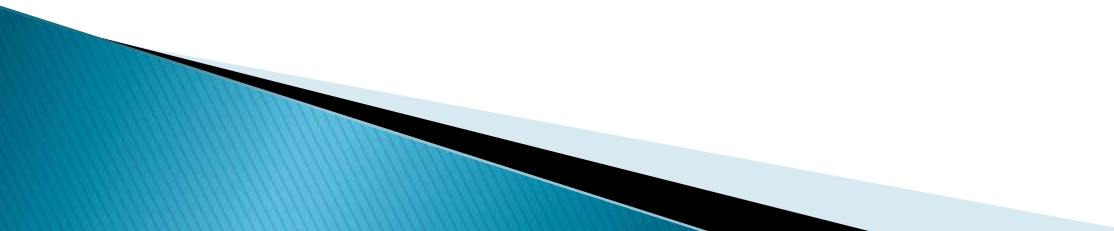
# Bagging Decision Trees (Freund & Schapire)



# Feature Bagging

- ▶ Given  $N$  examples for training
  - Create  $K$  datasets
    - Select  $(K-1)/K$  of the features to use
      - Ignore the rest
    - Train a classifier on the dataset
- ▶ Final output: voting of the  $K$  classifiers
  - Or: weighted majority of the  $K$  classifiers

# Bias / Variance Heuristics

- ▶ Models that fit the data poorly have high bias: “inflexible models” such as linear regression, regression stumps
  - ▶ Models that can fit the data very well have low bias but high variance: “flexible” models such as nearest neighbor regression, regression trees
  - ▶ This suggests that bagging of a flexible model can reduce the variance while benefiting from the low bias
- 

# Summary

- ▶ We can do a lot by combining a little
  - Boosting: turns weak learners into strong learners
- ▶ Mixtures of experts
  - Weighted majority uses the predictions of the best experts
- ▶ Diversity helps
  - Create artificial diversity
    - Instance bagging
    - Feature bagging
  - Diversity in representations for semi-supervised learning
    - Co-training