

Errors - Review

- Error of the hypothesis vs error of the algorithm?
- Know the training and test set error, good estimate of the classifier's performance?
- Classifier Error = noise + bias² + variance
- How we calculate bias and variance for a classifier*
 - $T_{1...n}$: Training sets drawn randomly from population
- Bias is the expected (mean) error over all training sets
- Variance is the variability of the error.
- Why would a decision tree be biased? Have a high variance?

Errors

The **true error** of hypothesis h with respect to target function f and distribution \mathcal{D} is the probability that h will misclassify an instance drawn at random according to \mathcal{D} .

$$error_{\mathcal{D}}(h) \equiv \Pr_{x \in \mathcal{D}}[f(x) \neq h(x)]$$

The **sample error** of h with respect to target function f and data sample S is the proportion of examples h misclassifies

$$error_S(h) \equiv \frac{1}{n} \sum_{x \in S} \delta(f(x) \neq h(x))$$

Where $\delta(f(x) \neq h(x))$ is 1 if $f(x) \neq h(x)$, and 0 otherwise.

How well does $error_S(h)$ estimate $error_{\mathcal{D}}(h)$?

Bias and Variance

1. *Bias*: If S is training set, $error_S(h)$ is optimistically biased

$$bias \equiv E[error_S(h)] - error_{\mathcal{D}}(h)$$

For unbiased estimate, h and S must be chosen independently

2. *Variance*: Even with unbiased S , $error_S(h)$ may still vary from $error_{\mathcal{D}}(h)$

??? What else ???

Model Uncertainty

- What's wrong with making predictions from one model?
 - May have two or more equally accurate models that give different predictions.
 - May have two models that are quite fundamentally different

Ensemble of Models Techniques

- Bayesian Modeling Averaging
 - $\Pr(c, x \mid D, H) = \sum_{h \in H} \Pr(c, x \mid h) \cdot \Pr(h \mid D)$
 - Weight each model's prediction by how good the model is.
 - Can this approach be applied to C4.5 Dtrees?
- Bagging (Bootstrap Aggregation), 1996.
 - Improves accuracy
 - Seminal paper says on 19 of 26 data sets improves accuracy by 4%.

The Bagging Algorithm

- Building the Models

For $i = 1$ to k // k is the number of bags

$T_i = \text{BootStrap}(D)$ // D is the training set

Build Model M_i from T_i (ie. Induce the tree)

End

- Applying the Models To Make a Prediction

For a test set example, x

For $i = 1$ to k // k is the number of bags

$C_i = M_i(x)$

End

Prediction is the class with the most vote.

Take A Bootstrap Sample

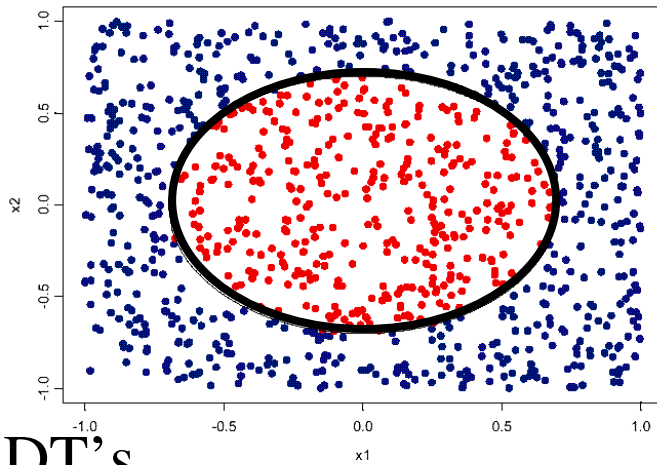
Sample with replacement

Bootstrapping and model building can be easily parallelized

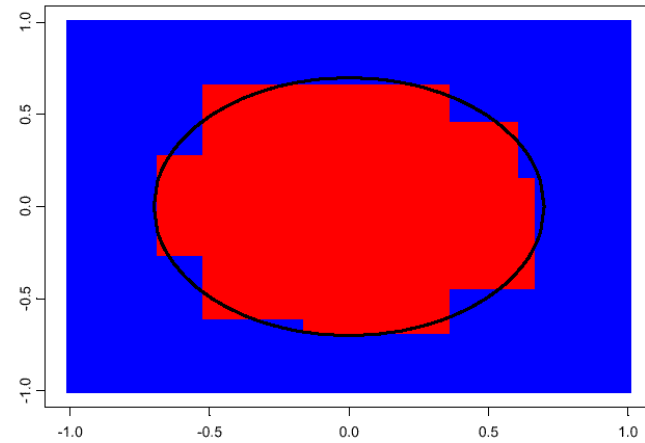
Original	1	2	3	4	5	6	7	8
Training set 1	2	7	8	3	7	6	3	1
Training set 2	7	8	5	6	4	2	7	1
Training set 3	3	6	2	7	5	6	2	2
Training set 4	4	5	1	4	6	4	3	8

Example of Bagging

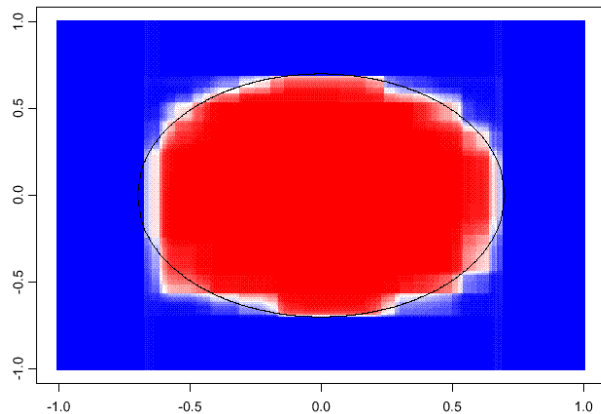
Problem



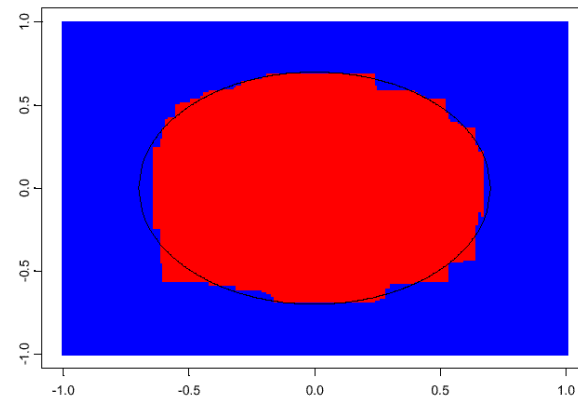
Single DT Solution



100 DT's



Bagging Solution



Boosting – The Idea

- Take weak learners (marginally better than random guessing) make them stronger.
- Freund and Schapire, 95 – AdaBoost
- AdaBoost premise
 - Each training instances has equal weight
 - Build first Model from training instances
 - Training instances that are classified incorrectly given more weight
 - Build another model with re-weighted instances and so on and so on.

Boosting Pseudo Code

- Initialize distribution over the training set $D_1(i) = 1/m$
- For $t = 1, \dots, T$:
 1. Train *Weak Learner* using distribution D_t .
 2. Choose a weight (or confidence value) $\alpha_t \in \mathbf{R}$.
 3. Update the distribution over the training set:

$$D_{t+1}(i) = \frac{D_t(i)e^{-\alpha_t y_i h_t(x_i)}}{Z_t} \quad (2)$$

Where Z_t is a normalization factor chosen so that D_{t+1} will be a distribution

- Final vote $H(x)$ is a weighted sum:

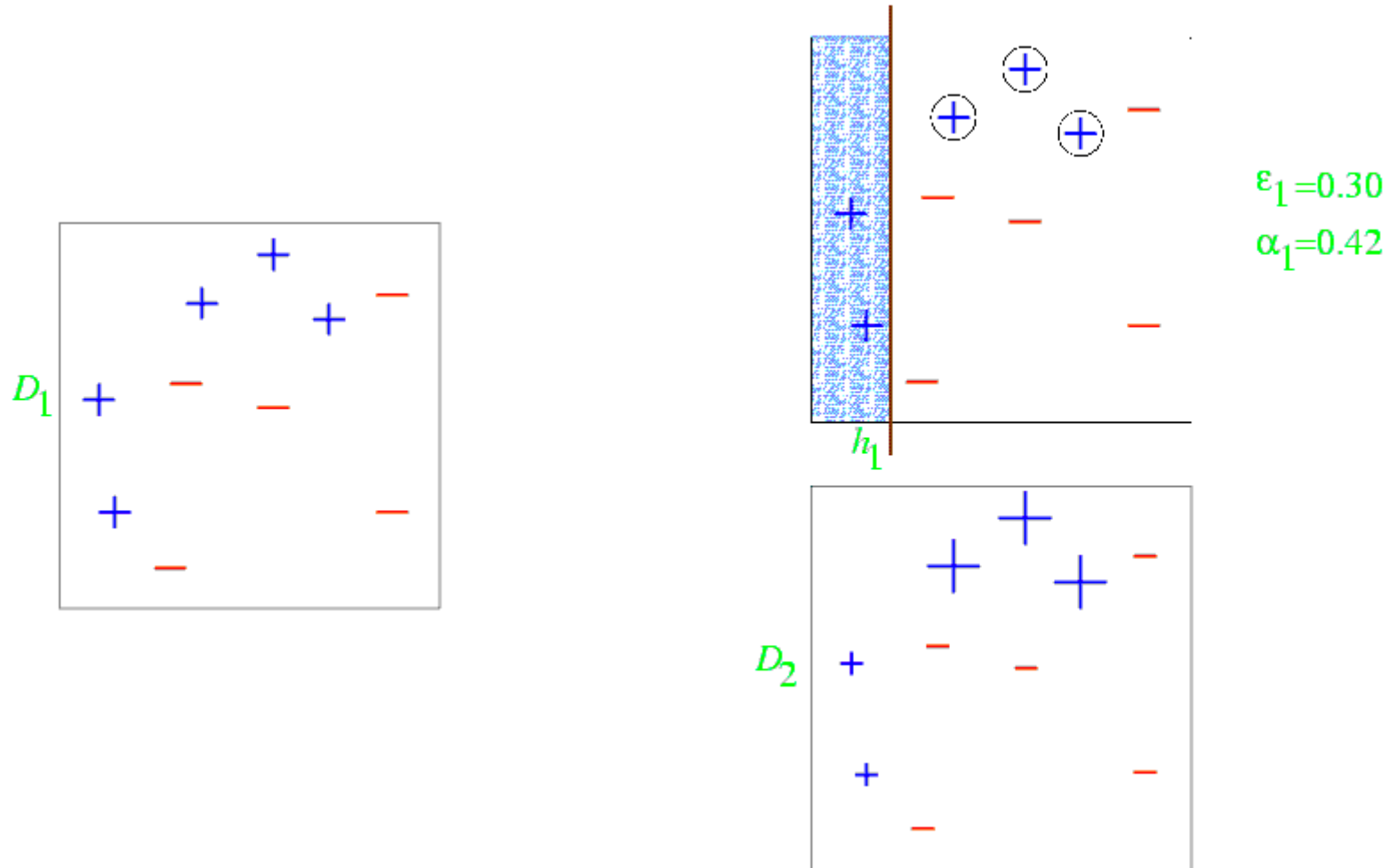
$$H(x) = \text{sign}(f(x)) = \text{sign} \left(\sum_{t=1}^T \alpha_t h_t(x) \right) \quad (3)$$

Some Implementation Comments

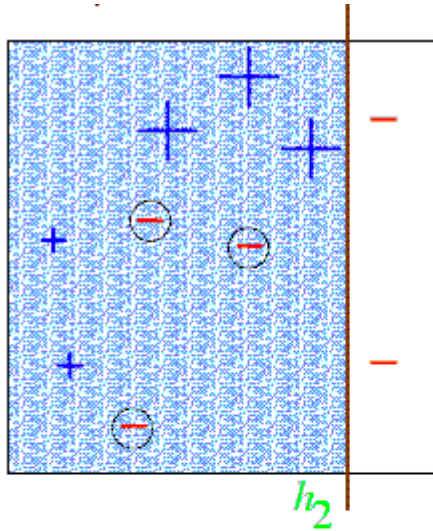
- Difficult to parallelize
- Factoring instance weights into decision tree induction.
- Tree vote is weighted inversely to error.
- Adaptive Boosting (AdaBoosting) according to the tree error
- Free scaled down version of C5.0 incorporates boosting available at <http://www.rulequest.com/download.html>

Toy Example (Freund COLT 99)

Round 1

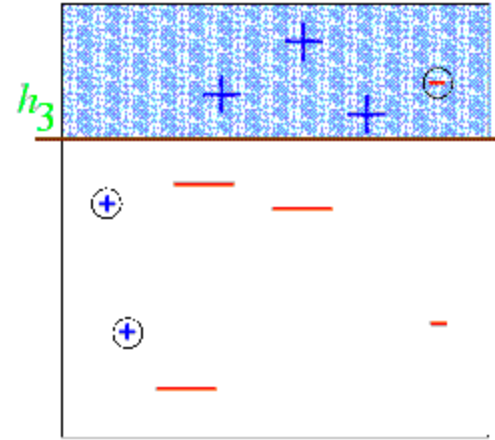


Round 2 + 3



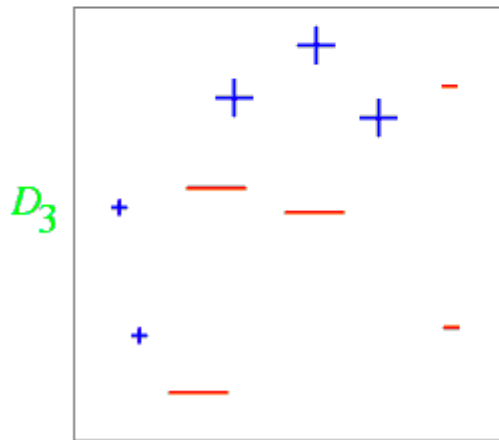
$$\epsilon_2 = 0.21$$

$$\alpha_2 = 0.65$$

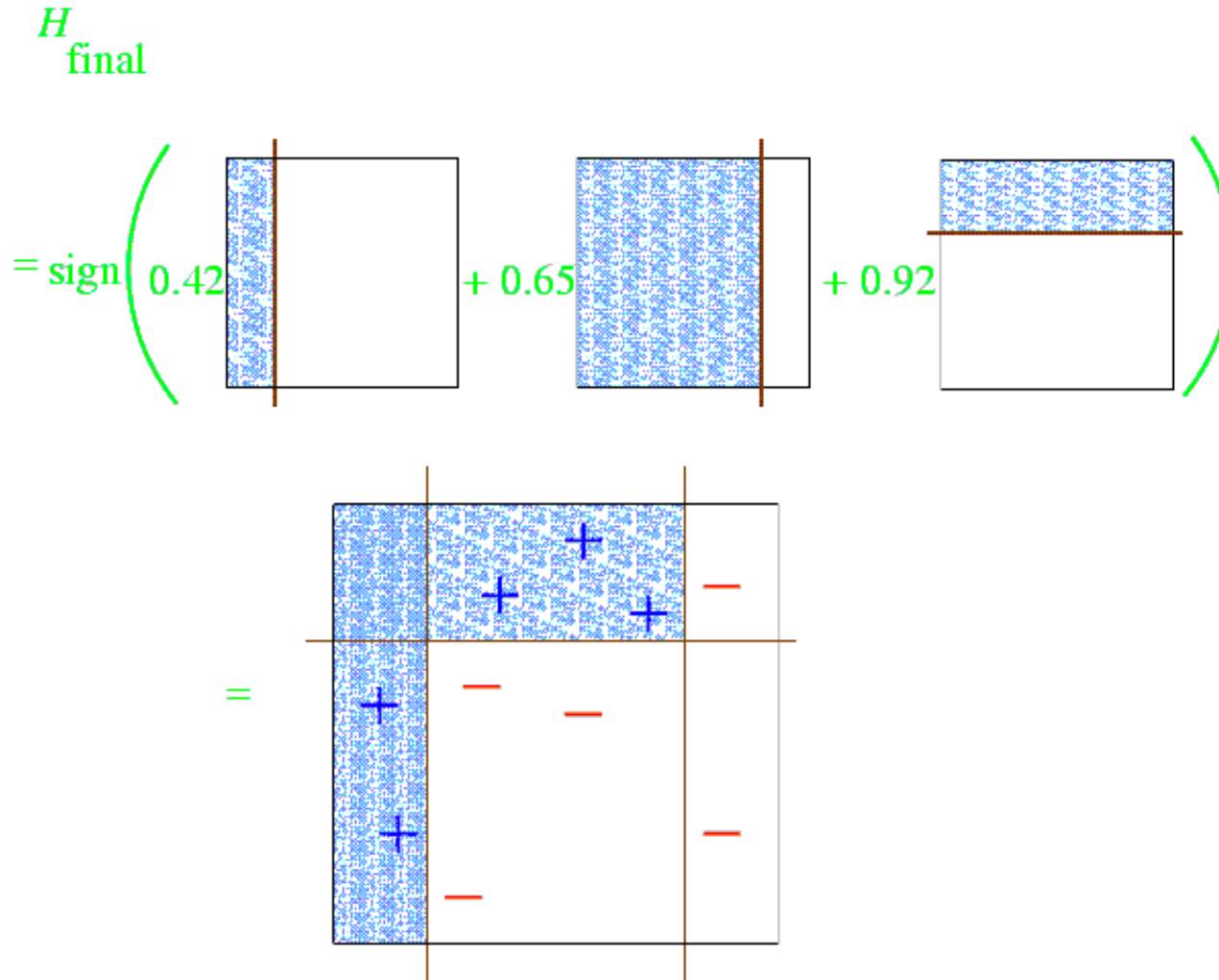


$$\epsilon_3 = 0.14$$

$$\alpha_3 = 0.92$$



Final Hypothesis



Demo at <http://www.cs.huji.ac.il/~yoavf/adaboost/index.html>

Some Insights into Boosting

- Final aggregate model will have no training error (given some conditions).
- Seems to over-fit but reduces test set error
- Larger margins on training set correspond to better generalization error
 - $\text{Margin}(x) = y \sum \alpha_j h_j(x) / \sum \alpha_j$

The Performance of Models and Learners

- Error of the hypothesis vs error of the learning algorithm?
- Know the training and test set error, good estimate of the learner's performance?
- Learners Error = noise + bias² + variance
- How we calculate bias and variance for a learner*
 - $T_{1\dots n}$: Training sets drawn randomly from population
- Bias is the difference in error over all training sets – true error.
- Variance is the variability of the error.
- Why would a decision tree be biased? Have a high variance?

Ensemble Techniques Reduce Error

- Decision trees are known to have a high variance, particularly when overfitted.
- BMA
 - Expected cost of Bayesian prediction is the noise.
 - Why?
- Bagging
 - Reduces variance but not bias
- Boosting
 - Reduces what?

Ensemble Technique Scorecard

	BMA	Bagging	Boosting
<i>Reduces Variance Or bias</i>	Both	Variance	Bias*
<i>Voting Scheme</i>	Degree of Belief in Model	Equal	Depends on Model Error
<i>Requirement of Learners</i>	Bayesian	Unstable	Weak (consistently better than random guessing)
?	?	?	?

Retrospective on Decision Trees

- Representation and search
- Does Bagging and Boosting change model representation space?
- Do they change search preference?
- Order of data presented does not count.