BOOSTING (ADABOOST ALGORITHM)

Eric Emer

Consider Horse-Racing Gambler

- Rules of Thumb for determining Win/Loss:
  - Most favored odds
  - Fastest recorded lap time
  - Most wins recently, say, in the past 1 month

- Hard to determine how he combines analysis of feature set into a single bet.

Consider MIT Admissions

Table 1: MIT Admissions Training Data

ID	Name	**Admit/Deny**	Region	Gender	GoodAtMath	Athlete	SAT
1	Andrew	Admit	East	M	Y	N	2280
2	Burt	Deny	East	M	N	N	2180
3	Charlie	Deny	East	M	N	Y	2400
4	Derek	Admit	West	M	Y	N	2260
5	Erica	Admit	Deep South	F	N	N	2360
6	Faye	Admit	Midwest	F	Y	N	2350
7	Greg	Admit	West	M	N	Y	2290
8	Helga	Deny	Midwest	F	N	Y	2380
9	Ivana	Admit	International	F	Y	N	2310
10	Jan	Deny	International	M	N	Y	2150

- 2-class system (Admit/Deny)
- Both Quantitative Data and Qualitative Data
  - We consider (Y/N) answers to be Quantitative (-1,+1)
  - Region, for instance, is qualitative.

Rules of Thumb, Weak Classifiers

Table 1: MIT Admissions Training Data

ID	Name	**Admit/Deny**	Region	Gender	GoodAtMath	Athlete	SAT
1	Andrew	Admit	East	M	Y	N	2280
2	Burt	Deny	East	M	N	N	2180
3	Charlie	Deny	East	M	N	Y	2400
4	Derek	Admit	West	M	Y	N	2260
5	Erica	Admit	Deep South	F	N	N	2360
6	Faye	Admit	Midwest	F	Y	N	2350
7	Greg	Admit	West	M	N	Y	2290
8	Helga	Deny	Midwest	F	N	Y	2380
9	Ivana	Admit	International	F	Y	N	2310
10	Jan	Deny	International	M	N	Y	2150

- Easy to come up with rules of thumb that correctly classify the training data at better than chance.
  - E.g. IF “GoodAtMath”==Y THEN predict “Admit”.
- Difficult to find a single, highly accurate prediction rule. This is where our Weak Learning Algorithm, AdaBoost, helps us.

What is a Weak Learner?

- For any distribution, with high probability, given polynomially many examples and polynomial time we can find a classifier with generalization error better than random guessing.

$$\epsilon < \frac{1}{2}, \text{ also denoted } \gamma > 0 \text{ for generalization error } (\frac{1}{2} - \gamma)$$

Weak Learning Assumption

- We assume that our Weak Learning Algorithm (Weak Learner) can consistently find weak classifiers (rules of thumb which classify the data correctly at better than 50%)
- Given this assumption, we can use boosting to generate a single weighted classifier which correctly classifies our training data at 99%-100%.

AdaBoost Specifics

- How does AdaBoost weight training examples optimally?
  - Focus on difficult data points. The data points that have been misclassified most by the previous weak classifier.

- How does AdaBoost combine these weak classifiers into a comprehensive prediction?
  - Use an optimally weighted majority vote of weak classifier.

AdaBoost Technical Description

Given training data $(x_1, y_1), ..., (x_m, y_m)$
$y_i \in \{-1, +1\}$, $x_i \in X$ is the object or instance, y_i is the classification.

for $t = 1, \ldots, T$
  create distribution D_t on $\{1, \ldots, m\}$
  select weak classifier with smallest error ϵ_t on D_t
    $\epsilon_t = \Pr_{D_t}[h_t(x_i) \neq y_i]$
    $h_t : X \to \{-1, +1\}$
output single classifier $H_{\text{final}}(x)$

Missing details: How to generate distribution? How to get single classifier?

Constructing Dt

$$D_1(i) = \frac{1}{m}$$

and given D_t and h_t:

$$D_{t+1} = \frac{D_t(i)}{Z_t} c(x)$$

$$c(x) = \begin{cases} e^{-\alpha_t} & : y_i = h_t(x_i) \\ e^{\alpha_t} & : y_i \neq h_t(x_i) \end{cases}$$

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

where Z_t = normalization constant

$$\alpha_t = \frac{1}{2} \ln \frac{1-\epsilon_t}{\epsilon_t} > 0$$

Getting a Single Classifier

$$H_{final}(x) = \mathrm{sign}(\sum_t \alpha_t h_t(x))$$

Mini-Problem

Table 2: MIT Admissions Training Data

ID	Name	**Admit/Deny**	# of High School Detentions	SAT
1	Andrew	Deny	3	2050
2	Burt	Admit	1	2200
3	Charlie	Admit	2	2090
4	Derek	Deny	4	2230
5	Erica	Admit	5	2330
6	Faye	Deny	6	2220
7	Greg	Admit	6	2390
8	Helga	Admit	7	2320
9	Ivana	Deny	8	2330
10	Jan	Deny	8	2090

Training Error Analysis

$$\text{Thm: training error}(H_{final}) \leq e^{-2\gamma^2 T}$$

Claim: $\text{training error}(H_{final}) \leq \prod_t 2\sqrt{\epsilon_t(1-\epsilon_t)}$

$\text{suppose } \epsilon_t = 1/2 - \gamma_t$ then,

$$\text{training error}(H_{final}) \leq \prod_t \sqrt{1-4\gamma_t^2}$$

$$\text{training error}(H_{final}) \leq \exp\left(-2\sum_t \gamma_t^2\right)$$

$$\text{if for all } t\text{: } \gamma_t \geq \gamma > 0 \text{ then } \boxed{\text{training error}(H_{final}) \leq e^{-2\gamma^2 T}}.$$

Proof

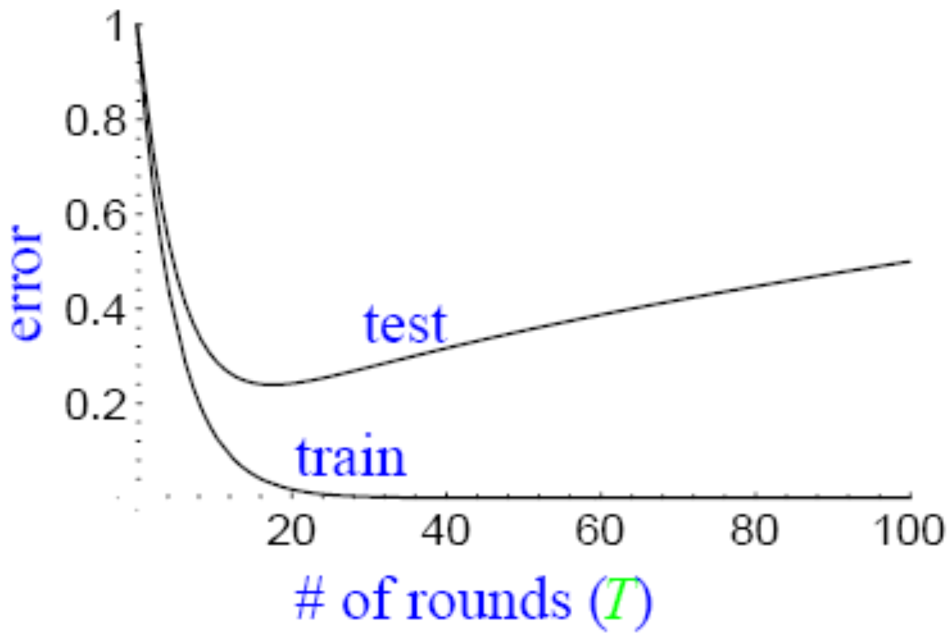
Thm: training error$(H_{final}) \leq e^{-2\gamma^2 T}$

- Step 1: unwrapping the recurrence
- Step 2: Show $\text{training error}(H_{final}) \leq \prod_t Z_t$
- Step 3: Show $Z_t = 2\sqrt{\epsilon_t(1-\epsilon_t)}$

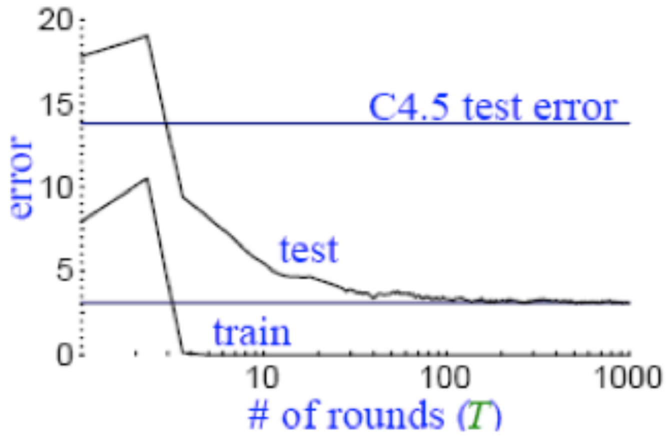
How might test error react to AdaBoost?

We expect to encounter:

- Occam's Razor
- Overfitting



Empirical results of test error



(boosting C4.5 on "letter" dataset)

•Test error does not increase even after 1000 rounds.
•Test error continues to drop after training error reaches zero.

	# rounds		
	5	100	1000
train error	0.0	0.0	0.0
test error	8.4	3.3	3.1

Difference from Expectation: The Margins Explanation

- Our training error only measures correctness of classifications, neglects *confidence* of classifications. How can we measure confidence of classifications?

$$H_{final}(x) = \mathrm{sign}(f(x))$$

$$f(x) = \frac{\sum_t \alpha_t h_t}{\sum_t \alpha_t} \in [-1, 1]$$

$$\mathrm{margin}(x, y) = y f(x)$$

- Margin(x,y) close to +1 is high confidence, correct.
- Margin(x,y) close to -1 is high confidence, incorrect.
- Margin(x,y) close to 0 is low confidence.

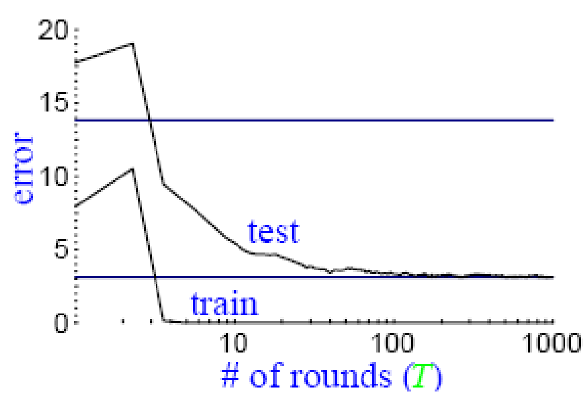
Empirical Evidence Supporting Margins Explanation

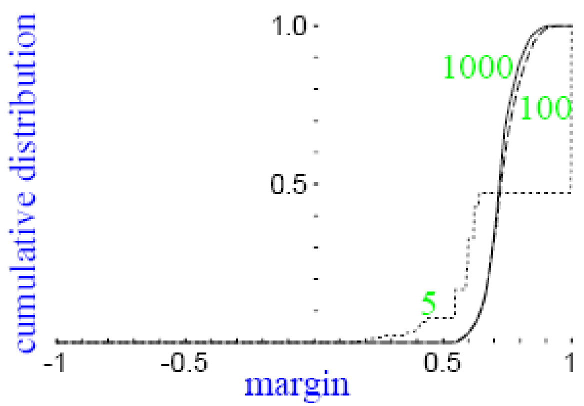
$$H_{final}(x) = \text{sign}(f(x))$$

$$f(x) = \frac{\sum_t \alpha_t h_t}{\sum_t \alpha_t} \in [-1, 1]$$

$$\text{margin}(x, y) = y f(x)$$

	# rounds		
	5	100	1000
train error	0.0	0.0	0.0
test error	8.4	3.3	3.1
% margins ≤ 0.5	7.7	0.0	0.0
minimum margin	0.14	0.52	0.55





Cumulative distribution of margins on training examples

Pros/Cons of AdaBoost

Pros

- Fast
- Simple and easy to program
- No parameters to tune (except T)
- No prior knowledge needed about weak learner
- Provably effective given Weak Learning Assumption
- versatile

Cons

- Weak classifiers too complex leads to overfitting.
- Weak classifiers too weak can lead to low margins, and can also lead to overfitting.
- From empirical evidence, AdaBoost is particularly vulnerable to uniform noise.

Predicting College Football Results

Training Data: 2009 NCAAF Season
Test Data: 2010 NCAAF Season

Passes attempted	Interceptions	Points gained
Passes completed	Interception yardage	Points allowed
Passes intercepted	Passing touchdowns allowed	Red zone scoring percentage
Pass completion percentage	Passing yards allowed	Red zone field goal percentage
Pass rating	Rushing touchdowns allowed	Red zone touchdowns allowed
Passing touchdowns	Rushing yards allowed	Red zone field goals allowed
Passing yardage	Sacks	Third down conversion percentage
Rushes attempted	Sack yardage	Third down conversion percentage allowed
Rushing average	Tackles for loss	Quarterback hurries
Rushing touchdowns	Tackles for loss yardage	Passes broken up
Rushing yardage	Forced fumbles	Kicks or punts blocked

Figure 1: A subset of the fundamental statistics used as features

	Train Error	**Test Error**
Always Home	-	43.25%
Always Away	-	56.75%
Random	-	52.25%
Logistic	14.11%	38.52%
SVM (Linear)	7.77%	34.43%
SVM (Poly)	0%	35.66%
SVM (RBF)	0%	47.54%
GentleBoost	0%	27.46%
ModestBoost	9.41%	30.74%

(a) Results for straight bets