Boosting Algorithms
Weak and Strong Learning

PAC Learning model

- Assume a fixed distribution D over domain X
- Examples: $\langle x, c^*(x) \rangle$
  - c^* is the target function
- Goal:
  - With high probability $(1-\delta)$
  - find h in H such that
  - $error(h,c) < \varepsilon$
- δ and ε are arbitrarily small.

Weak Learning - Confidence

- Assume we are given a learning algorithm
  - with confidence $\delta = \frac{1}{2}$
  - but arbitrary accuracy $\varepsilon > 0$

- Can we boost the confidence?
  - How?!

BoostConfidence Algorithm

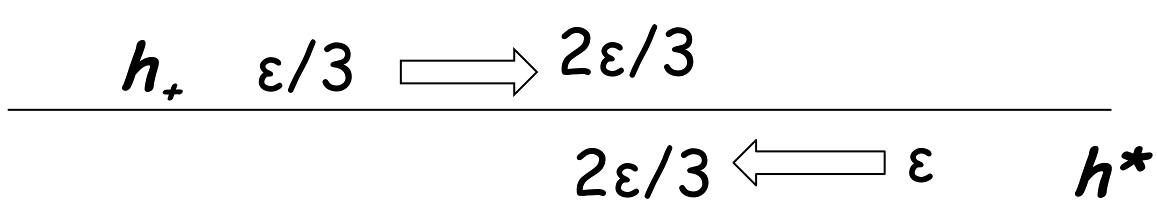
- **Input**: Algorithm **A** and parameter **δ**
- Create $k=log(2/\delta)$ independent problems
  - – Sample S_i for i-th copy
  - – Run Algorithm **A** on S_i with accuracy $\varepsilon/3$
  - – Let h_i be the hypothesis that **A** outputs on S_i
- Take a new sample S of size $m=\varepsilon^{-2} log(2k/\delta)$
  - – Return the best hypothesis h_i on S
    - call it h^*

BoostConfidence Analysis

- Fix copy i:
  - The probability that $error\ (h_i) < \varepsilon/3$ at least ½
- The probability that
  - some h_i has $error(h_i) < \varepsilon/3$
  - at least $1\text{-}2^{-k}=1\text{-}\ \delta/2$
    - Holds for $k=log(2/\delta)$
- Assume this holds!
  - Namely, some h_i has $error(h_i) < \varepsilon/3$
    - denote it by $\boldsymbol{h}_+$

BoostConfidence Analysis

- With probability at least $1 - \delta/2$
  - for every h_i: $|\boldsymbol{error(h_i)\text{-}obst\text{-}error(h_i)}| \leq \boldsymbol{\varepsilon/3}$
    - Chernoff bound using $m = \varepsilon^{-2} \log(2k/\delta)$
- Assume this holds!
- Together, with probability at least $1\text{-}\delta$
  - $error(h^*) \leq \varepsilon$



Weak Learning - Accuracy

- Assume: ***error(h,c)* < ½-γ**
- The parameter ***γ>0*** is small
  - constant, 1/poly
- Intuitively: a much easier task
- Question:
  - Assume **C** is weak learnable,
  - Is **C** PAC (strong) learnable?

Weak Learning - Definition

Weak Learning

- Algo A weak learns C using H if
- exists **γ>0**
- for all c in C
- **for any distribution D**
- for all δ> ½
- Outputs h in H such that
  - with prob 1- δ
  - ***error(c,h) < ½ - γ***

Strong Learning

- Algo A strong learns C using H if
- for all **ε>0**
- for all c in C
- ~~for any distribution D~~
- for all δ> ½
- Outputs h in H such that
  - with prob 1- δ
  - ***error(c,h) < ε***

Weak learning – definition

- Why do we need **ANY** distribution
- Example:
  - Consider the following distribution over bits
  - if $x_1=x_2=0$ then $c^*(x)=$ some hard function
  - otherwise $c^*(x)=0$
- Uniform Distribution
  - Predicting 0 will be correct 87.5%
  - Getting above than is impossible.
- Hard distribution
  - only $x_1=x_2=0$

Three Weak Learners

- One weak learner
  - only one thing to do !
- Two weak learners
  - what to do if they disagree?
- Three weak learners
  - Can we improve accuracy?

- Example
  - weak learners=literals

x	- y	x	- y
11110	- 1	10011	- 0
10010	- 1	10111	- 0
10111	- 1	10011	- 0
10001	- 1	00100	- 0
10101	-1	00000	- 0

Three weak learners

- First weak learner
  - use the distribution D
  - get h_1
- Second weak learner
  - How can we force new h?
  - Set D_2 s.t.
  - h_1 has error ½
  - Get h_2
    - why $h_2 \neq h_1$?
- Third weak learner
  - What are "interesting inputs"?
  - $h_2(x) \neq h_1(x)$
    - let D_3 be such inputs
  - get h_3
- How will we predict?
  - If $h_2(x) \neq h_1(x)$ using $h_3(x)$
  - Else $h_2(x)$ (or $h_1(x)$)
    - **majority**

3 Weak Learners - Performance

- Define D_2 and D_3
- Dist D_2:
  - Select random b={0,1}
  - If b=0
    - Sample x from D_1 until $h_1(x)=c^*(x)$
  - Else (b=1)
    - Sample x from D_1 until $h_1(x)\neq c^*(x)$
- Dist D_3
  - if $h_1(x)\neq h_2(x)$ sample

- Formally

$$D_2(x)=\begin{cases}\dfrac{0.5}{1-p}D_1(x) & \text{if } \mathrm{h}_1(x)=c^*(x)\\ \dfrac{0.5}{p}D_1(x) & \text{if } \mathrm{h}_1(x)\neq c^*(x)\end{cases}$$

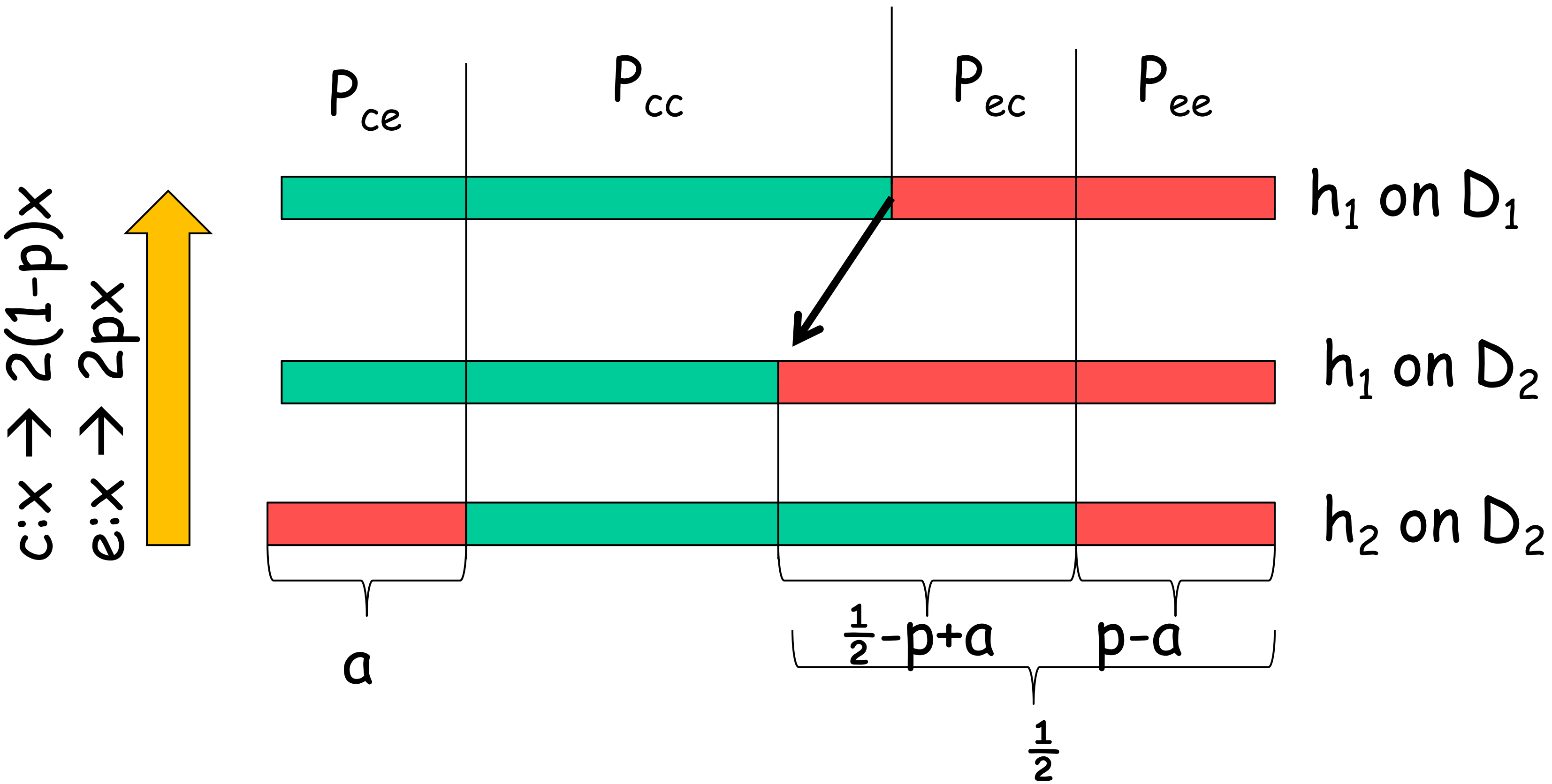
$$p=\Pr[\mathrm{h}_1(x)\neq c^*(x)]$$

$$D_3(x)=\begin{cases}\dfrac{D_1(x)}{Z} & \text{if } \mathrm{h}_1(x)\neq h_2(x)\\ 0 & \text{if } \mathrm{h}_1(x)=h_2(x)\end{cases}$$

$$Z=\Pr[\mathrm{h}_1(x)\neq h_2(x)]$$

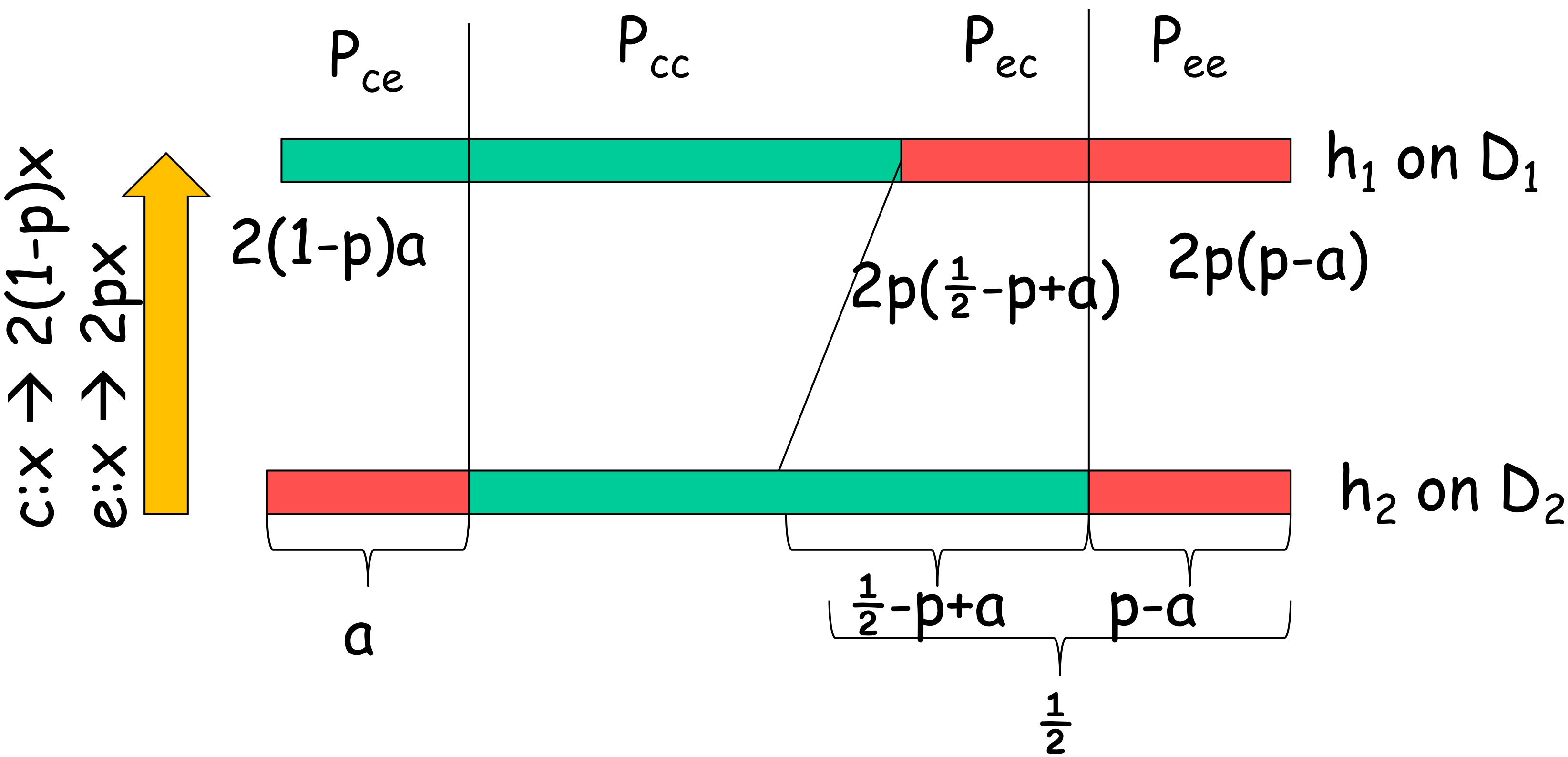
3 Weak Learners - Analysis

- Assume all WL are $\boldsymbol{p=½ -\gamma}$



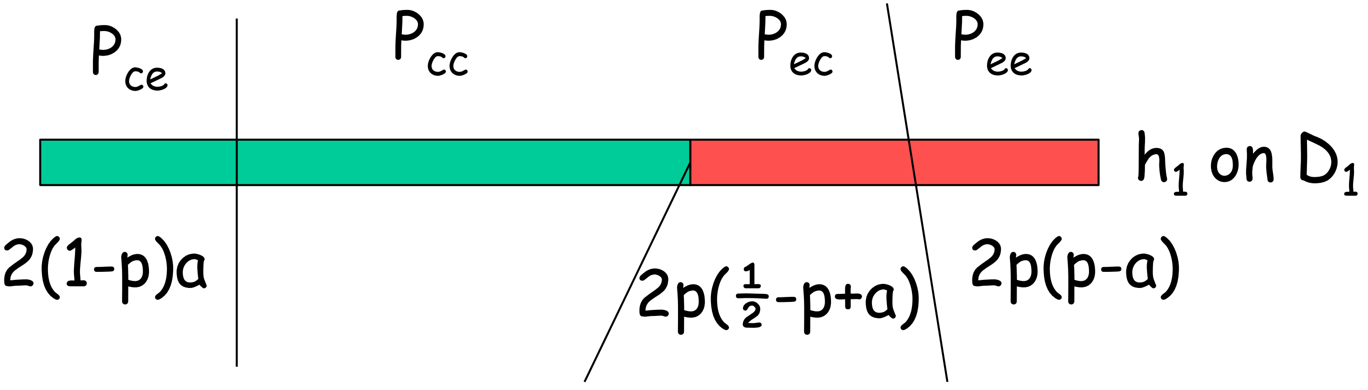
3 Weak Learners - Analysis

- Assume all WL are p=½ -γ



3 Weak Learners - Analysis

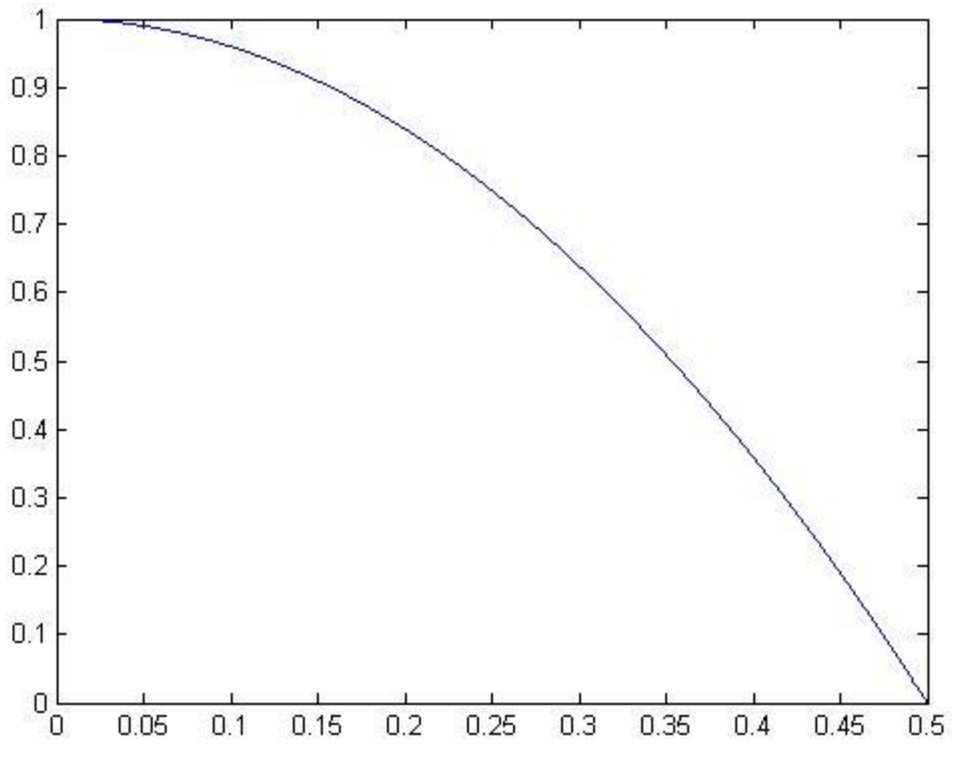
- Assume all WL are p=½ -γ



Error = $P_{ee}+p(P_{ec}+P_{ce})=3p^2-2p^3$

3 Weak Learners - Analysis

New Error =Old Error*$(1-4\gamma^2)$
small γ; medium γ; large γ near 1/2



What about more hypothesis

- The CS way
  - Do it recursively
  - Can push down the error arbitrarily
- We show a more constructive way

ADABOOST:
ADAPTIVE BOOSTING

AdaBoost: Overview

- Build a linear classifier
  - basic elements, weak learners
  - Ensemble method
- An “online” approach
  - each time add one more classifier
- Fit the sample **S**

- Each time step t
  - have a hypothesis F_t
  - Select a distribution D_t
    - on **S**
  - Find a weak learner h_t
    - w.r.t D_t
  - Add h_t to the hypothesis
    - decide on weight a_t
    - $F_{t+1}(x)=F_t(x)+a_t h_t(x)$
    - predict $\mathbf{sign}(F_{t+1}(x))$

AdaBoost: Algorithm

- Weak Learners ***H***

  h: $\boldsymbol{X} \rightarrow \{+1,-1\}$
- Initialization:
  - Sample $\boldsymbol{S}=\{x_1,\ldots,x_m\}$
  - $D_1(x_i) = 1/m$
- Prediction

$F(x)=sign(\Sigma\alpha_t h_t(x))$

- Step $t = 1, \ldots, T$
  - Receive h_t
    - WL w.r.t. D_t
  - Define
    - $\varepsilon_t = Pr[h_t(x) \neq c^*(x)]$
    - $\alpha_t = ½ \log (1- \varepsilon_t) / \varepsilon_t$
  - Define D_{t+1},

$$D_{t+1}(x_i) = \frac{D_t(x_i)}{Z_t} \cdot \begin{cases} e^{-\alpha_t} & y_i = h_t(x_i) \\ e^{\alpha_t} & y_i \neq h_t(x_i) \end{cases}$$

$$\propto D_t(x_i) e^{-\alpha_t y_i h_t(x_i)}$$

AdaBoost: Intiution

- How do we change the distribution?
  - Error → weight increases
  - Correct → weight decreases
  - Focus on the “hard” examples
- What are the parameters?
  - The weak learning class $\boldsymbol{H}$
  - The number of iterations $\boldsymbol{T}$
    - Assume to be inputs

AdaBoost: Analysis

- Theorem:
  - Given $\varepsilon_1, \ldots, \varepsilon_T$
  - the error $\boldsymbol{\varepsilon}$ of $\boldsymbol{F}$ is bounded by

$$\varepsilon \leq 2^T \prod_{t=1}^{T} \sqrt{\varepsilon_t (1 - \varepsilon_t)}$$

- Proof based on three claims

AdaBoost: Analysis

Claim 1:

$$D_{T+1}(x_i) = \frac{D_1(x_i)e^{-y_i f(x_i)}}{\prod_t Z_t}$$

where $f(x)=\Sigma\alpha_t h_t(x)$

Corollary 1:

$$e^{-y_i f(x_i)} = mD_{T+1}(x_i)\prod_t Z_t$$

Proof

- For $t+1$ we have

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

- unravel recurrence

$$D_{T+1}(x_i) = D_1(x_i)\prod_{t=1}^{T} \frac{e^{-y_i \alpha_t h_t(x_i)}}{Z_t}$$

$$= D_1(x_i)\frac{e^{-y_i \sum_{t=1}^{T} \alpha_t h_t(x_i)}}{\prod_t Z_t}$$

$$= D_1(x_i)\frac{e^{-y_i f(x_i)}}{\prod_t Z_t}$$

AdaBoost: Analysis

Claim 2:

$$error(F,S) \le \prod_{t=1}^{T} Z_t$$

Proof

$$\begin{aligned}
error(F,S) &= \frac{1}{m}\sum_{i=1}^{m} I(y_i \ne F(x_i)) \\
&= \frac{1}{m}\sum_{i=1}^{m} I(y_i f(x_i) \le 0) \\
&\le \frac{1}{m}\sum_{i=1}^{m} e^{-y_i f(x_i)} \\
&= \frac{1}{m}\sum_{t=1}^{T} m D_{T+1}(x_i) \prod_t Z_t \\
&= \left(\prod_t Z_t\right) \sum_{i=1}^{m} D_{T+1}(x_i) \\
&= \prod_t Z_t
\end{aligned}$$

z<0 → e^-z>1

Corollary 1

AdaBoost: Analysis

Claim 3:

$$Z_t = 2\sqrt{\varepsilon_t(1-\varepsilon_t)}$$

Proof

$$\begin{aligned} Z_t &= \sum_{i=1}^{m} D_t(x_i) e^{-y_i \alpha_t h_t(x_i)} \\ &= \sum_{i:y_i = h_t(x_i)} D_t(x_i) e^{-\alpha_t} + \sum_{i:y_i \neq h_t(x_i)} D_t(x_i) e^{\alpha_t} \\ &= (1-\varepsilon_t) e^{-\alpha_t} + \varepsilon_t e^{\alpha_t} \end{aligned}$$

Recall

$$\sum_{i:y_i \neq h_t(x_i)} D_t(x_i) = \varepsilon_t$$

AdaBoost: Analysis

- Completing the proof
  - Z_t valid for any α_t
  - Minimize Z_t

$$\frac{dZ_t}{d\alpha_t} = -(1-\varepsilon_t)e^{-\alpha_t} + \varepsilon_t e^{\alpha_t}$$

$$\alpha_t = \frac{1}{2}\ln\left(\frac{1-\varepsilon_t}{\varepsilon_t}\right)$$

$$\prod_t Z_t = \prod_t 2\sqrt{\varepsilon_t(1-\varepsilon_t)}$$

AdaBoost: Analysis

- Theorem:
  - Given ε_1, ... , ε_T
  - the error $\boldsymbol{\varepsilon}$ of $\boldsymbol{F}$ is bounded by

$$\varepsilon \leq 2^T \prod_{t=1}^{T} \sqrt{\varepsilon_t (1 - \varepsilon_t)}$$

- Proof based on three claims

AdaBoost: Fixed bias

- Assume $\varepsilon_t = \frac{1}{2} - \gamma$
- We bound:

$$error(F,S) \le \prod_t Z_t$$

$$= (1 - 4\gamma^2)^{T/2}$$

$$\le e^{-2\gamma^2 T}$$

Additional Issues

- Boosting the margin
- Overfitting
- When to stop
- Advantages
- Weaknesses
- Implementation