CS4442/9542b
Artificial Intelligence II
prof. Olga Veksler

Lecture 5

Machine Learning

Boosting

Boosting: Motivation

- Hard to design accurate classifier which generalizes well
- Easy to find many rule of thumb or weak classifiers
  - a classifier is weak if it is slightly better than random guessing
  - example: if an email has word "money" classify it as spam, otherwise classify it as not spam
    - likely to be better than random guessing
- Can we combine several weak classifiers to produce an accurate classifier?
  - Question people have been working on since 1980's
  - Ada-Boost (1996) was the first practical boosting algorithm

Ada Boost: General form

- Assume 2-class problem, with labels +1 and -1
  - $\mathbf{y}^i$ in {-1,1}
- Ada boost produces a discriminant function:

$$\mathbf{g}(\mathbf{x}) = \sum_{\mathbf{t}=1}^{\mathbf{T}} \alpha_{\mathbf{t}} \mathbf{h}_{\mathbf{t}}(\mathbf{x}) = \alpha_1 \mathbf{h}_1(\mathbf{x}) + \alpha_2 \mathbf{h}_2(\mathbf{x}) + \ldots \alpha_{\mathbf{T}} \mathbf{h}_{\mathbf{T}}(\mathbf{x})$$

- Where $\mathbf{h}_t(\boldsymbol{x})$ is a weak classifier, for example:

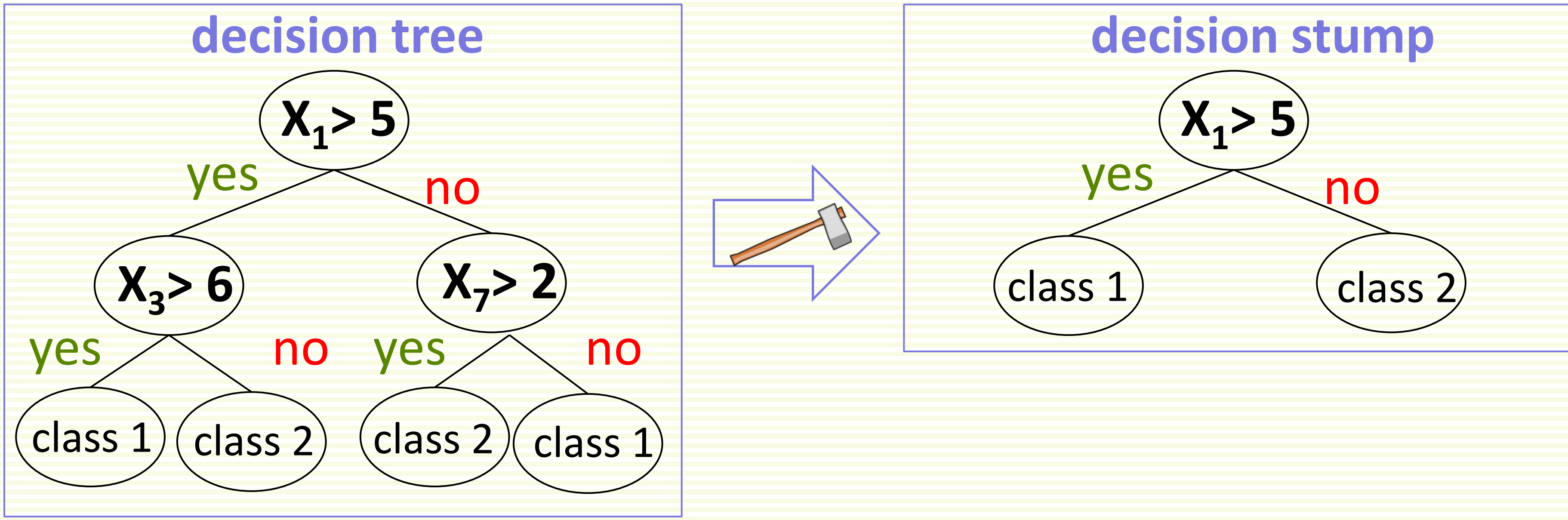
$$\mathbf{h}_{\mathbf{t}}(\mathbf{x}) = \begin{cases} -1 & \text{if email has word “money”} \\ 1 & \text{if email does not have word “money”} \end{cases}$$

- The final classifier is the sign of the discriminant function

$$\mathbf{f}_{\text{final}}(\mathbf{x}) = \text{sign}[\mathbf{g}(\mathbf{x})]$$

Ada Boost: Weak Classifiers

- Degenerate decision trees (decision stumps) are frequently used as weak classifiers



- Based on thresholding just one feature

$$\mathbf{h}_t(\mathbf{x}) = \begin{cases} -1 & \text{if } \mathbf{x}_3 > 10 \\ 1 & \text{if } \mathbf{x}_3 \leq 10 \end{cases} \qquad \mathbf{h}_t(\mathbf{x}) = \begin{cases} -1 & \text{if } \mathbf{x}_7 > 60 \\ 1 & \text{if } \mathbf{x}_7 \leq 60 \end{cases}$$

Ada Boost: Weak Classifiers

- Based on thresholding one feature

$$\mathbf{h}_t(\mathbf{x}) = \begin{cases} -1 & \text{if } \mathbf{x}_3 > 10 \\ 1 & \text{if } \mathbf{x}_3 \le 10 \end{cases} \qquad \mathbf{h}_t(\mathbf{x}) = \begin{cases} -1 & \text{if } \mathbf{x}_7 > 60 \\ 1 & \text{if } \mathbf{x}_7 \le 60 \end{cases}$$

- Reverse **polarity**:

$$\mathbf{h}_t(\mathbf{x}) = \begin{cases} -1 & \text{if } \mathbf{x}_3 \le 10 \\ 1 & \text{if } \mathbf{x}_3 > 10 \end{cases} \qquad \mathbf{h}_t(\mathbf{x}) = \begin{cases} -1 & \text{if } \mathbf{x}_7 \le 60 \\ 1 & \text{if } \mathbf{x}_7 > 60 \end{cases}$$

- There are approximately 2*n*d distinct decision stump classifiers, where
  - **n** is number of training samples, **d** is dimension of samples
  - We will see why later
- Small decision trees are also popular weak classifiers

Idea Behind Ada Boost

- Algorithm is iterative
- Maintains distribution of weights over the training examples
- Initially weights are equal
- Main Idea: at successive iterations, the weight of misclassified examples is increased
- This forces the algorithm to concentrate on examples that have not been classified correctly so far

Weighted Examples

- Training examples are weighted with distribution **D(x)**
- Many classifiers can handle weighted examples
- But if classifier does not handle weighted examples we can sample **k > n** examples according to distribution **D(x)**:

original data:

D(x): 1/16 1/4 1/16 1/16 1/4 1/16 1/4

data resampled according to **D(x)**:

- Apply classifier to the resampled data

Idea Behind Ada Boost

- misclassified examples get more weight
- more attention to examples of high weight
- Face/nonface classification problem:

Round 1

	1/7	1/7	1/7	1/7	1/7	1/7	1/7
best weak classifier:	✓	✗	✓	✓	✗	✓	✗
change weights:	1/16	1/4	1/16	1/16	1/4	1/16	1/4

Round 2

best weak classifier:	✓	✓	✓	✗	✗	✗	✓	✓	✓	✓
change weights:		1/8	1/32	11/32		1/2		1/8	1/32	1/32

Idea Behind Ada Boost

Round 3

- out of all available weak classifiers, we choose the one that works best on the data we have at round 3
- we assume there is always a weak classifier better than random (better than 50% error)
-  image is 50% of our data
- chosen weak classifier **has to** classify this image correctly

More Comments on Ada Boost

- Ada boost is very simple to implement, provided you have an implementation of a "weak learner"
- Will work as long as the "basic" classifier $\mathbf{h}_t(\mathbf{x})$ is at least slightly better than random
  - will work if the error rate of $\boldsymbol{h}_t(\boldsymbol{x})$ is less than 0.5
  - 0.5 is the error rate of a random guessing for a 2-class problem
- Can be applied to boost any classifier, not necessarily weak
  - but there may be no benefits in boosting a "strong" classifier

Ada Boost for 2 Classes

Initialization step: for each example $\mathbf{x}$, set

$$D(x) = \frac{1}{N}$$, where **N** is the number of examples

Iteration step (for **t** = 1…T):

1. Find best weak classifier $h_t(x)$ using weights $D(x)$
2. Compute the error rate ε_t as

$$\varepsilon_t = \sum_{i=1}^{N} D(x^i) \cdot I[y^i \neq h_t(x^i)]$$

$$I[y^i \neq h_t(x^i)] = \begin{cases} 1 & \text{if } y^i \neq h_t(x^i) \\ 0 & \text{otherwise} \end{cases}$$

3. compute weight α_t of classifier h_t

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

4. For each x^i, $D(x^i) = D(x^i) \cdot \exp(-\alpha_t \cdot y^i \cdot h_t(x^i))$
5. Normalize $D(x^i)$ so that $\sum_{i=1}^{N} D(x^i) = 1$

$$f_{final}(x) = \text{sign} [\sum \alpha_t h_t(x)]$$

Ada Boost: Step 1

1. Find best weak classifier $\mathbf{h}_t(\mathbf{x})$ using weights $\mathbf{D}(\mathbf{x})$
   - use resampled data if classifier does not handle weights
   - decision stump weak classifier handles weights

D(x):	**1/16**	**1/4**	**1/16**	**1/16**	**1/4**	**1/16**	**1/4**
$\mathbf{X}_3$:	1	8	7	6	4	9	9
	✗	✓	✓	✓	✓	✗	✗

- weak classifier: $$\mathbf{h}_t(\mathbf{x}) = \begin{cases} 1 \ (\textbf{face}) & \text{if } \mathbf{x_3} > 5 \\ -1 \ (\textbf{not face}) & \text{if } \mathbf{x_3} \le 5 \end{cases}$$
- error rate: 1/16 + 1/16 + 1/4 = 3/8

Ada Boost: Step 1

1. Find best weak classifier $\mathbf{h}_t(\mathbf{x})$ using weights $\mathbf{D}(\boldsymbol{x})$

- Give to the classifier the re-sampled examples:

- To find the best weak classifier, go through **all** weak classifiers, and find the one that gives the smallest error on the re-sampled examples

weak classifiers	$\mathbf{h}_1(\mathbf{x})$	$\mathbf{h}_2(\mathbf{x})$	$\mathbf{h}_3(\mathbf{x})$	…………	$\mathbf{h}_m(\mathbf{x})$
errors:	0.46	0.36	0.16		0.43

the best classifier $h_t(x)$
to choose at iteration t

Ada Boost: Step 2

2. Compute $\boldsymbol{\varepsilon}_t$ the error rate as

$$\varepsilon_t = \sum_{i=1}^{N} D(x^i) \cdot I[y^i \neq h_t(x^i)]$$

$$I[y^i \neq h_t(x^i)] = \begin{cases} 1 & \text{if } \; y^i \neq h_t(x^i) \\ 0 & \text{otherwise} \end{cases}$$


1/16 ✓


1/4 ✓


1/16 ✓


1/16 ✗


1/4 ✗


1/16 ✓


1/4 ✓

$$\varepsilon_t = \frac{1}{4} + \frac{1}{16} = \frac{5}{16}$$

- $\boldsymbol{\varepsilon}_t$ is the weight of all misclassified examples added
  - the error rate is computed over original examples, not the re-sampled examples
- If a weak classifier is better than random, then $\boldsymbol{\varepsilon}_t < ½$

Ada Boost: Step 3

3. compute weight α_t of classifier $\mathbf{h}_t$

$$\alpha_t = \frac{1}{2} \log \left((1 - \boldsymbol{\varepsilon}_t) / \boldsymbol{\varepsilon}_t \right)$$

In example from previous slide:

$$\boldsymbol{\varepsilon}_t = \frac{5}{16} \Rightarrow \alpha_t = \frac{1}{2} \log \frac{1 - \frac{5}{16}}{\frac{5}{16}} = \frac{1}{2} \log \frac{11}{5} \approx 0.4$$

- Recall that $\boldsymbol{\varepsilon}_t < \frac{1}{2}$
- Thus $(1 - \boldsymbol{\varepsilon}_t) / \boldsymbol{\varepsilon}_t > 1 \Rightarrow \alpha_t > 0$
- The smaller is $\boldsymbol{\varepsilon}_t$, the larger is α_t, and thus the more importance (weight) classifier $\mathbf{h}_t(x)$

$$\text{final}(\mathbf{x}) = \text{sign} \left[\sum \alpha_t \mathbf{h}_t(\mathbf{x}) \right]$$

Ada Boost: Step 4

4. For each x^i, $D(x^i) = D(x^i) \cdot \exp(-\alpha_t \cdot y^i \cdot h_t(x^i))$])

from previous slide $\alpha_t = 0.4$

1/16	1/4	1/16	1/16	1/4	1/16	1/4
✓	✓	✓	✗	✗	✓	✓
⇓	⇓	⇓	⇓	⇓	⇓	⇓
(1/16) exp(**-0.4**)	(1/4) exp(**-0.4**)	(1/16) exp(**-0.4**)	(1/16) exp (**0.4**)	(1/4) exp (**0.4**)	(1/16) exp(**-0.4**)	(1/4) exp(**-0.4**)

- weight of misclassified examples is increased
- weight of correctly classified examples is decreased

Ada Boost: Step 5

5. Normalize **D**(**x**^{i}) so that $\sum$**D**(**x**^{i}) = 1

from previous slide:

0.04 0.17 0.04 0.14 0.56 0.04 0.17

- after normalization

0.03 0.15 0.03 0.12 0.48 0.03 0.15

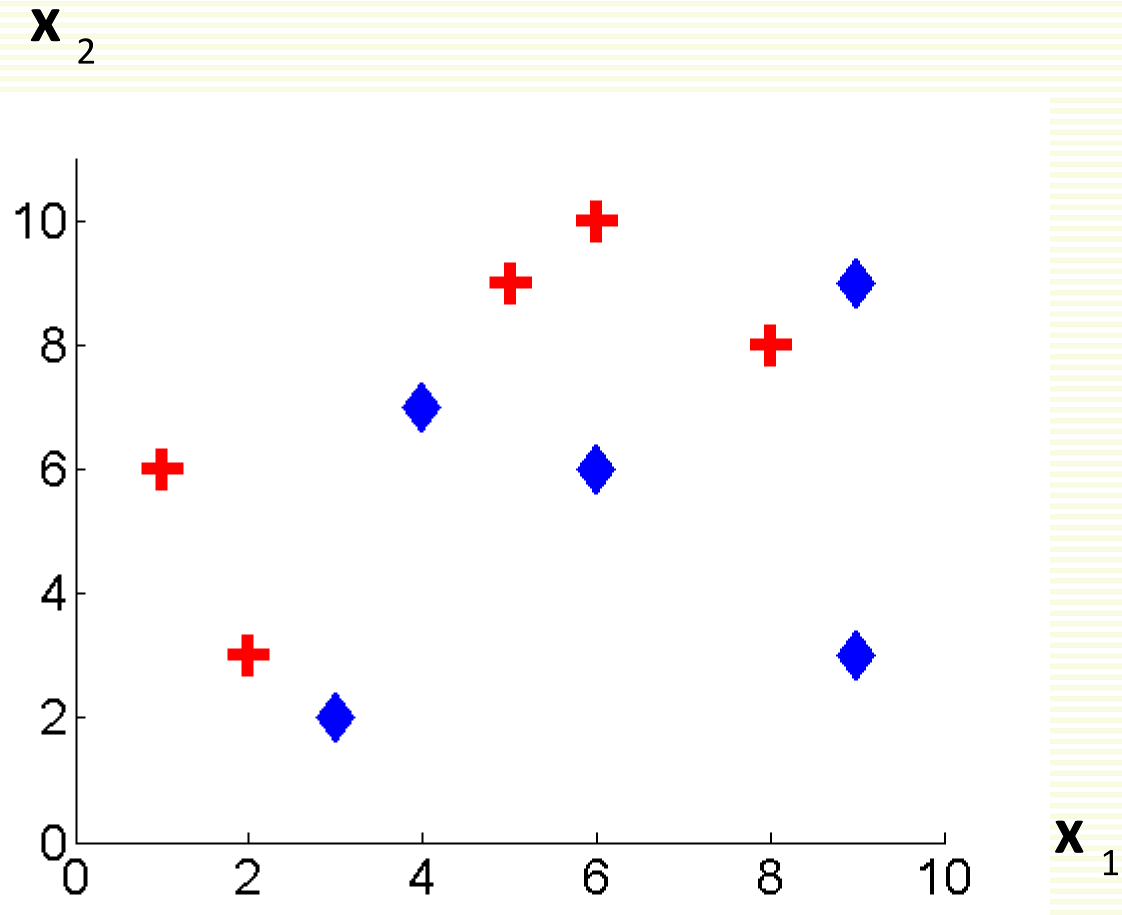
- In Matlab, if **D** is weights vector, normalize with

D = **D**./sum(**D**)

AdaBoost Example

$$C1 = \begin{bmatrix} 2 & 3 \\ 1 & 6 \\ 5 & 9 \\ 6 & 10 \\ 8 & 8 \end{bmatrix}$$

$$\mathbf{C2} = \begin{bmatrix} 3 & 2 \\ 4 & 7 \\ 6 & 6 \\ 9 & 3 \\ 9 & 9 \end{bmatrix}$$

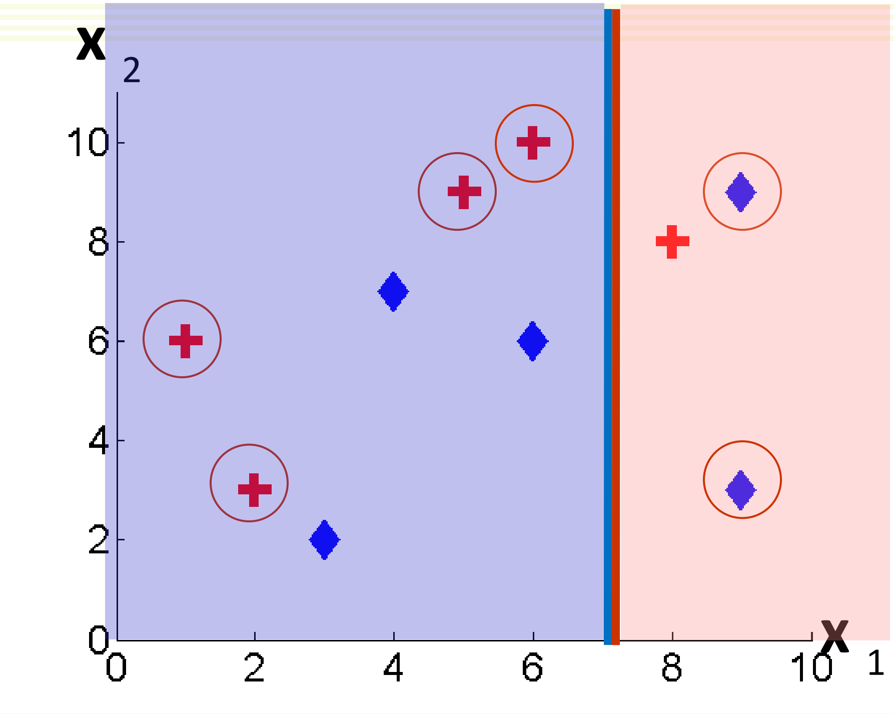


AdaBoost Example

- Decision stump weak classifiers

$$h_t(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x_1} > 7 \\ -1 & \text{if } \mathbf{x_1} \leq 7 \end{cases}$$

- 6 samples misclassified, error 0.6



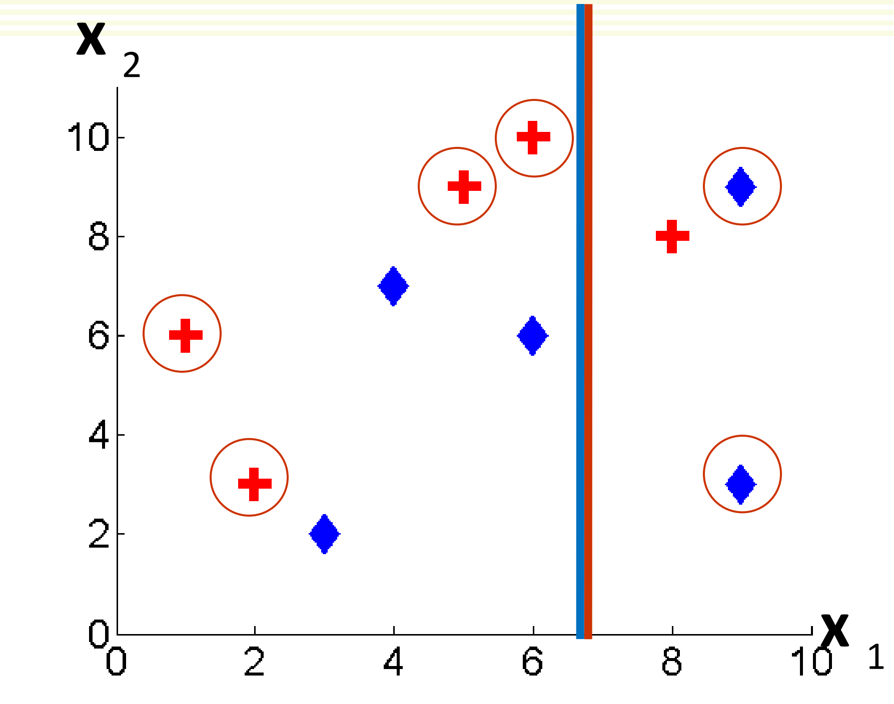
$$C1 = \begin{bmatrix} 2 & 3 \\ 1 & 6 \\ 5 & 9 \\ 6 & 10 \\ 8 & 8 \end{bmatrix} \quad C2 = \begin{bmatrix} 3 & 2 \\ 4 & 7 \\ 6 & 6 \\ 9 & 3 \\ 9 & 9 \end{bmatrix}$$

AdaBoost Example

- How many distinct classifiers based on thresholding feature 1 are there ?

$$h_t(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x_1} > 7 \\ -1 & \text{if } \mathbf{x_1} \leq 7 \end{cases}$$

- 6 samples misclassified



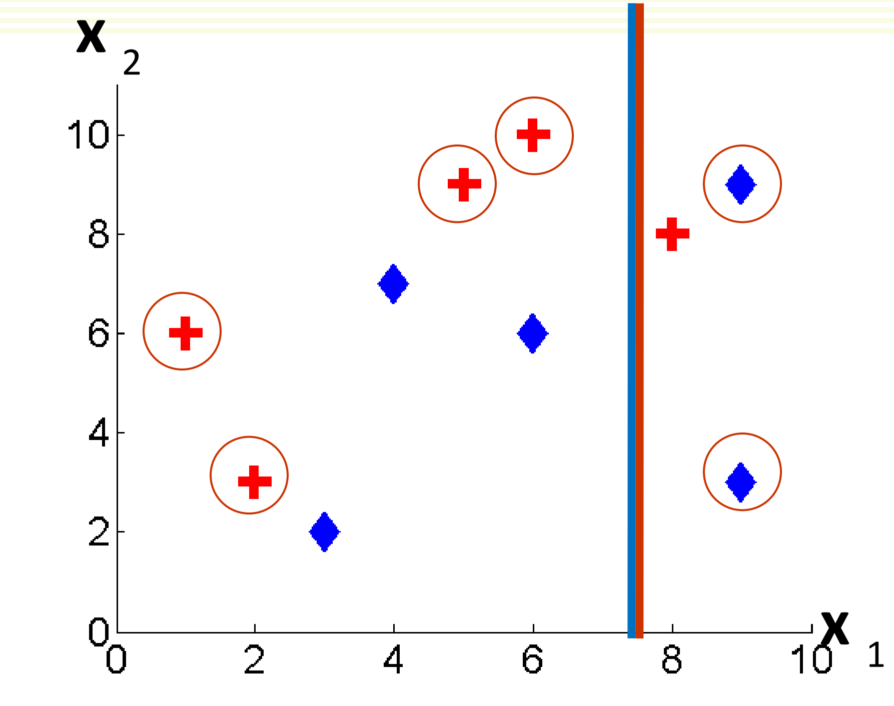
$$C1 = \begin{bmatrix} 2 & 3 \\ 1 & 6 \\ 5 & 9 \\ 6 & 10 \\ 8 & 8 \end{bmatrix} \quad \mathbf{C2} = \begin{bmatrix} 3 & 2 \\ 4 & 7 \\ 6 & 6 \\ 9 & 3 \\ 9 & 9 \end{bmatrix}$$

AdaBoost Example

- How many distinct classifiers based on thresholding feature 1 are there ?

$$\mathbf{h}_t(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x_1} > 7.5 \\ -1 & \text{if } \mathbf{x_1} \leq 7.5 \end{cases}$$

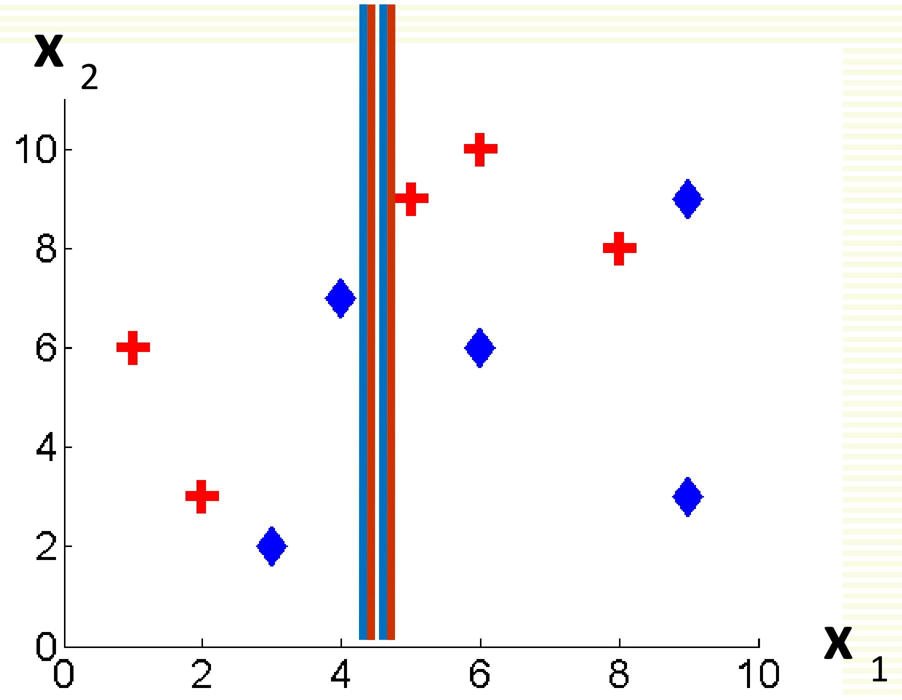
- 6 samples misclassified, same classifier as with threshold 7



$$C1 = \begin{bmatrix} 2 & 3 \\ 1 & 6 \\ 5 & 9 \\ 6 & 10 \\ 8 & 8 \end{bmatrix} \qquad \mathbf{C2} = \begin{bmatrix} 3 & 2 \\ 4 & 7 \\ 6 & 6 \\ 9 & 3 \\ 9 & 9 \end{bmatrix}$$

AdaBoost Example

$$C1 = \begin{bmatrix} 2 & 3 \\ 1 & 6 \\ 5 & 9 \\ 6 & 10 \\ 8 & 8 \end{bmatrix} \qquad C2 = \begin{bmatrix} 3 & 2 \\ 4 & 7 \\ 6 & 6 \\ 9 & 3 \\ 9 & 9 \end{bmatrix}$$



- Values of feature **1** in C1 and C2:

  1, 2, 3, 4, 5, 6, 8, 9

- Thresholds between any two consecutive values give same classifier
  - take two thresholds between 4 and 5, for example:

$$h_t(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x_1} > 4.2 \\ -1 & \text{if } \mathbf{x_1} \leq 4.2 \end{cases} \qquad h_t(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x_1} > 4.8 \\ -1 & \text{if } \mathbf{x_1} \leq 4.8 \end{cases}$$

  - get the same classifier with error 0.3

AdaBoost Example

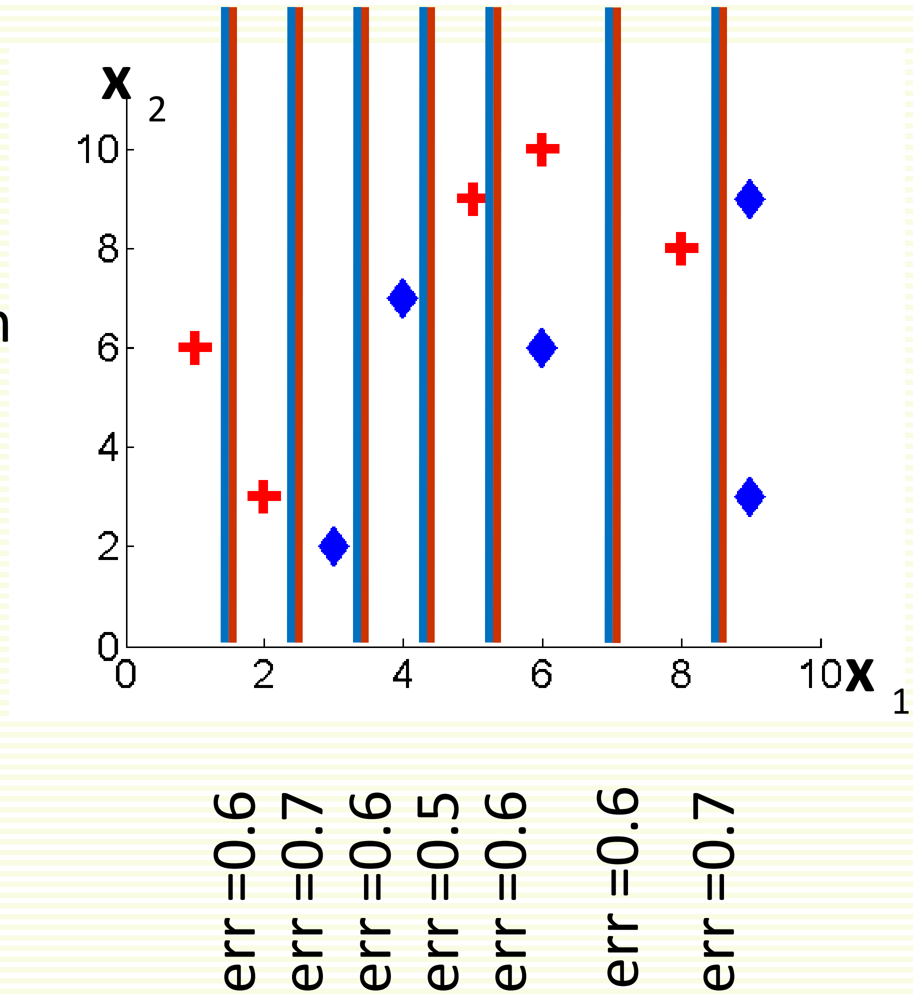
- Values of feature **1** in C1 and C2:

  1, 2, 3, 4, 5, 6, 8, 9

- Take one threshold between each pair of feature values:

  $a \in \{1.5, 2.5, 3.5, 4.5, 5.5, 7, 8.5\}$

$$\mathbf{h}_t(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x_1} > a \\ -1 & \text{if } \mathbf{x_1} \leq a \end{cases}$$



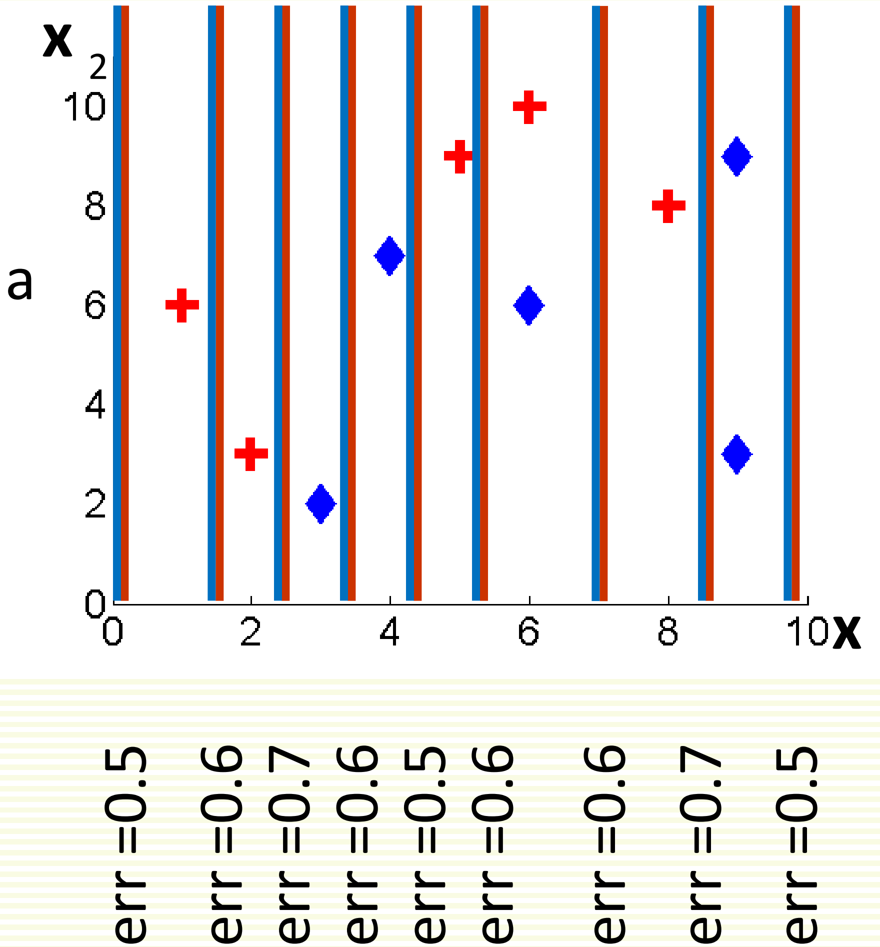
AdaBoost Example

- Values of feature **1** in C1 and C2:

  1, 2, 3, 4, 5, 6, 8, 9

- Two more distinct classifiers using a value smaller and larger than any value for feature 1, but these classifiers are largely useless:

  $a \in \{0,10\}$



$$h_t(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x}_1 > a \\ -1 & \text{if } \mathbf{x}_1 \leq a \end{cases}$$

- Thresholds leading to distinct classifiers

$a \in \{0, 1.5, 2.5, 3.5, 4.5, 5.5, 7, 8.5, 10\}$

AdaBoost Example

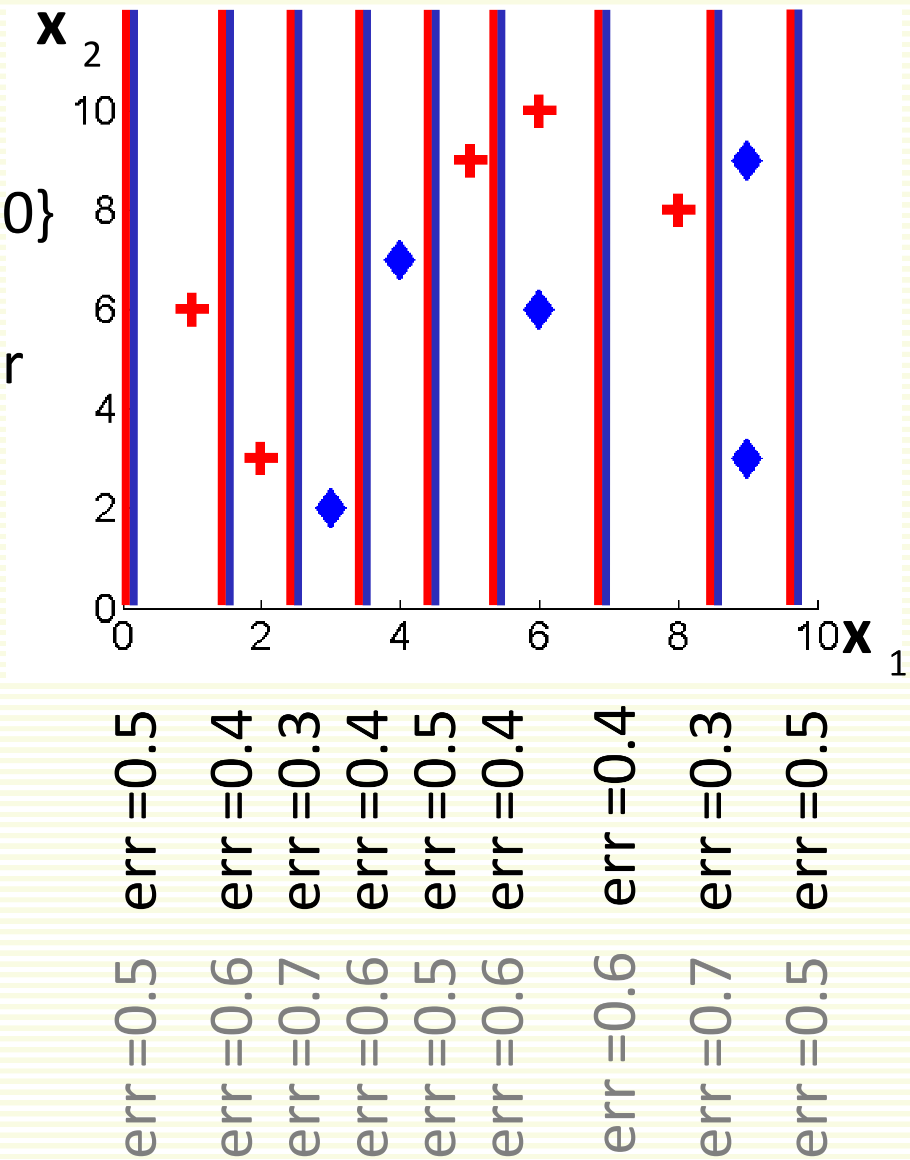
- Thresholds leading to distinct classifiers

$a \in \{0, 1.5, 2.5, 3.5, 4.5, 5.5, 7, 8.5, 10\}$

- Reverse polarity to double number of classifiers:

$$h_t(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x_1} \le a \\ -1 & \text{if } \mathbf{x_1} > a \end{cases}$$

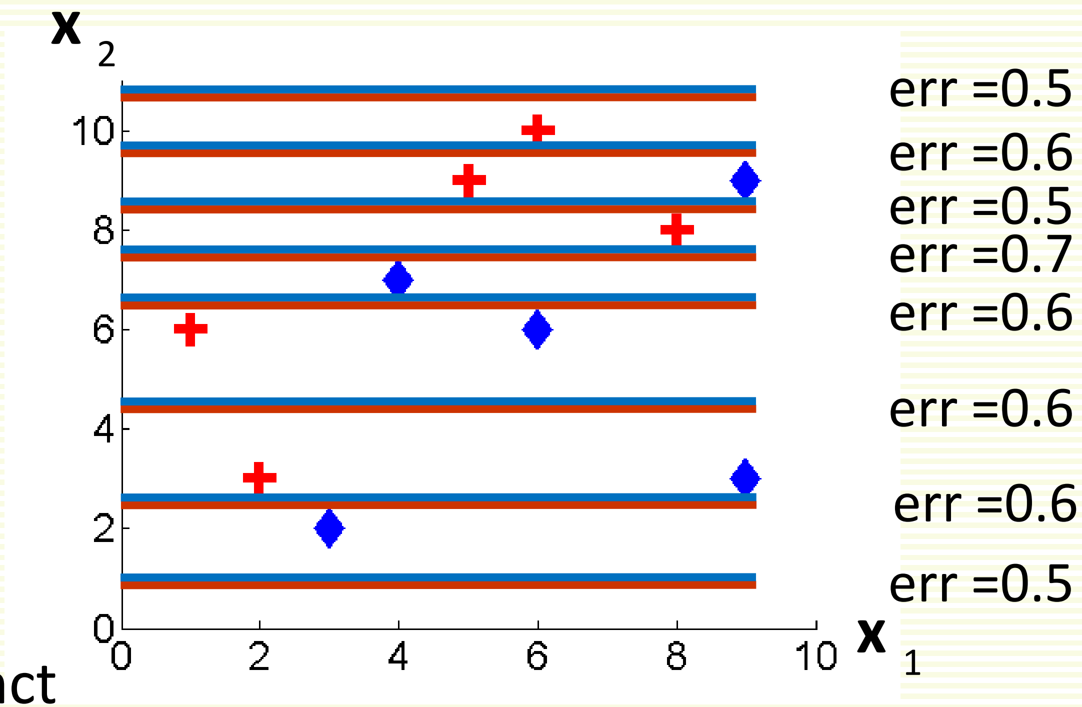
- Note error rates are reversed, compared to the same threshold but different polarity



AdaBoost Example

- Similar for feature 2

$$C1 = \begin{bmatrix} 2 & 3 \\ 1 & 6 \\ 5 & 9 \\ 6 & 10 \\ 8 & 8 \end{bmatrix} \qquad C2 = \begin{bmatrix} 3 & 2 \\ 4 & 7 \\ 6 & 6 \\ 9 & 3 \\ 9 & 9 \end{bmatrix}$$



- Distinct values of feature 2:

{2,3,6,7,8,9,10}

- Thresholds leading to distinct classifiers

$a \in \{1, 2.5, 4.5, 6.5, 7.5, 8.5, 9.5, 11\}$

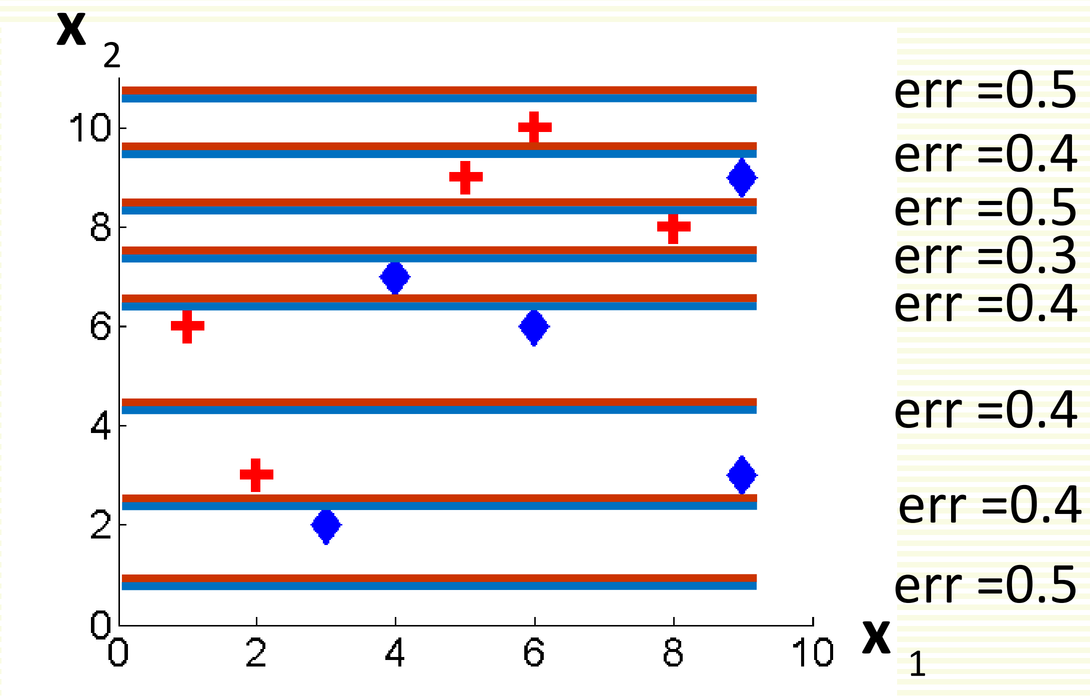
$$\mathbf{h}_t(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x}_2 \leq a \\ -1 & \text{if } \mathbf{x}_2 > a \end{cases}$$

AdaBoost Example

- Reverse polarity
- Thresholds leading to distinct classifiers

$$h_t(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x_2} > a \\ -1 & \text{if } \mathbf{x_2} \le a \end{cases}$$

$a \in \{1, 2.5, 4.5, 6.5, 7.5, 8.5, 9.5, 11\}$



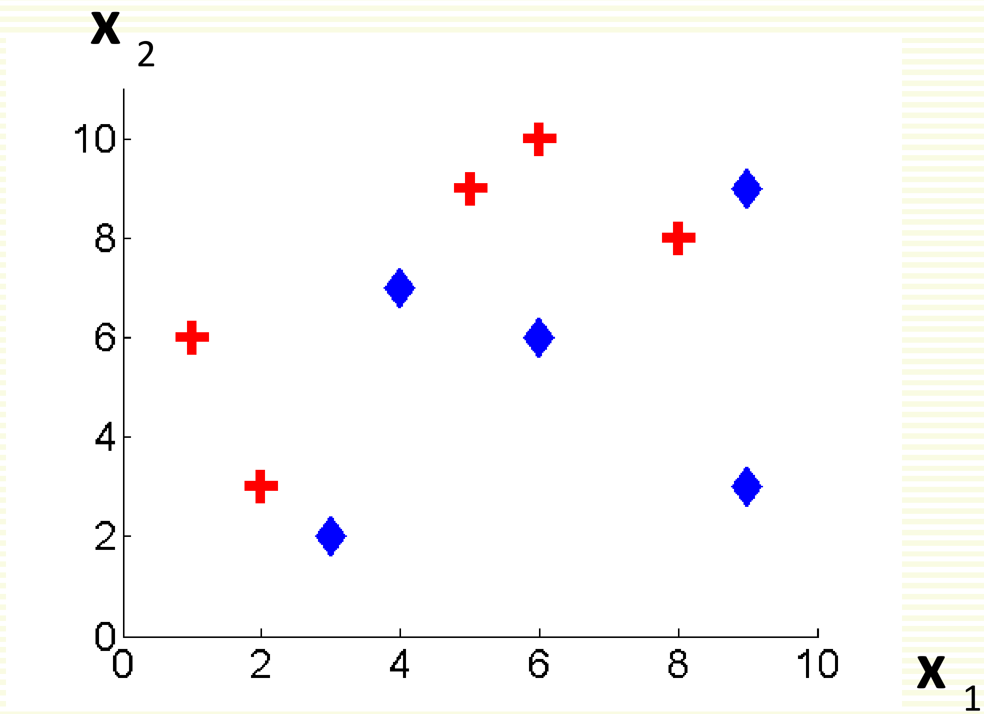
AdaBoost Example

- Thus total number of decision-stump weak classifiers is, approximately, **2 · n · d**
  - **d** is number of features
  - **n** is times number of samples
  - **2** comes from polarity
- Small (shallow) decision trees are also popular as weak classifiers
  - gives more weak classifiers

AdaBoost Example

$$C1 = \begin{bmatrix} 2 & 3 \\ 1 & 6 \\ 5 & 9 \\ 6 & 10 \\ 8 & 8 \end{bmatrix}$$

$$C2 = \begin{bmatrix} 3 & 2 \\ 4 & 7 \\ 6 & 6 \\ 9 & 3 \\ 9 & 9 \end{bmatrix}$$



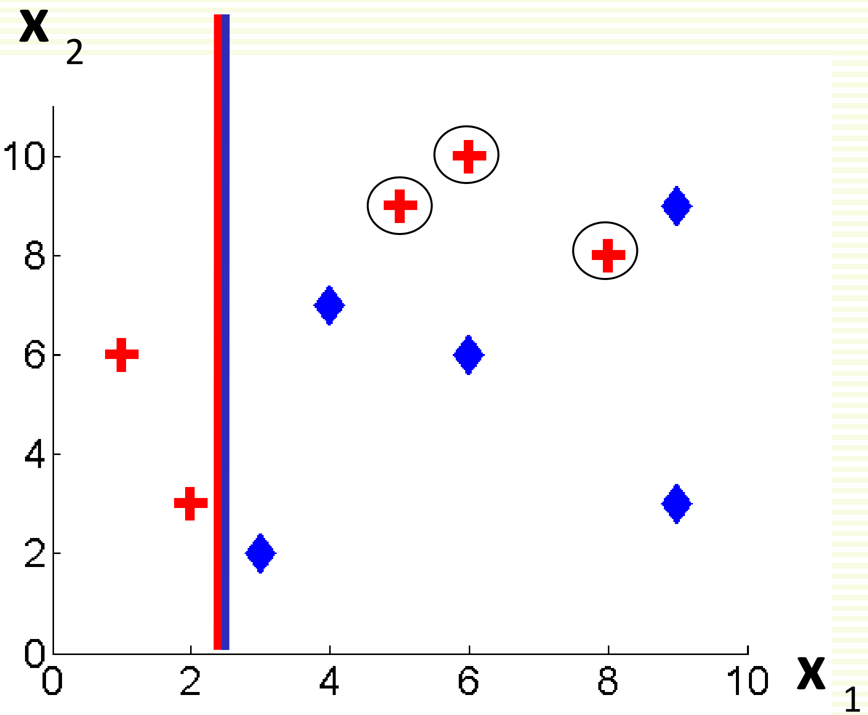
- Initialization: all examples have equal weights

D_1 = [0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1, 0.1]^T

AdaBoost Example: Round 1

- Classifier chosen:

$$\mathbf{h}_1(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x_1} \le 2.5 \\ -1 & \text{if } \mathbf{x_1} > 2.5 \end{cases}$$



- $\varepsilon_1 = 0.3, \alpha_1 = 0.42$

$$C1 = \begin{bmatrix} 2 & 3 \\ 1 & 6 \\ 5 & 9 \\ 6 & 10 \\ 8 & 8 \end{bmatrix}$$

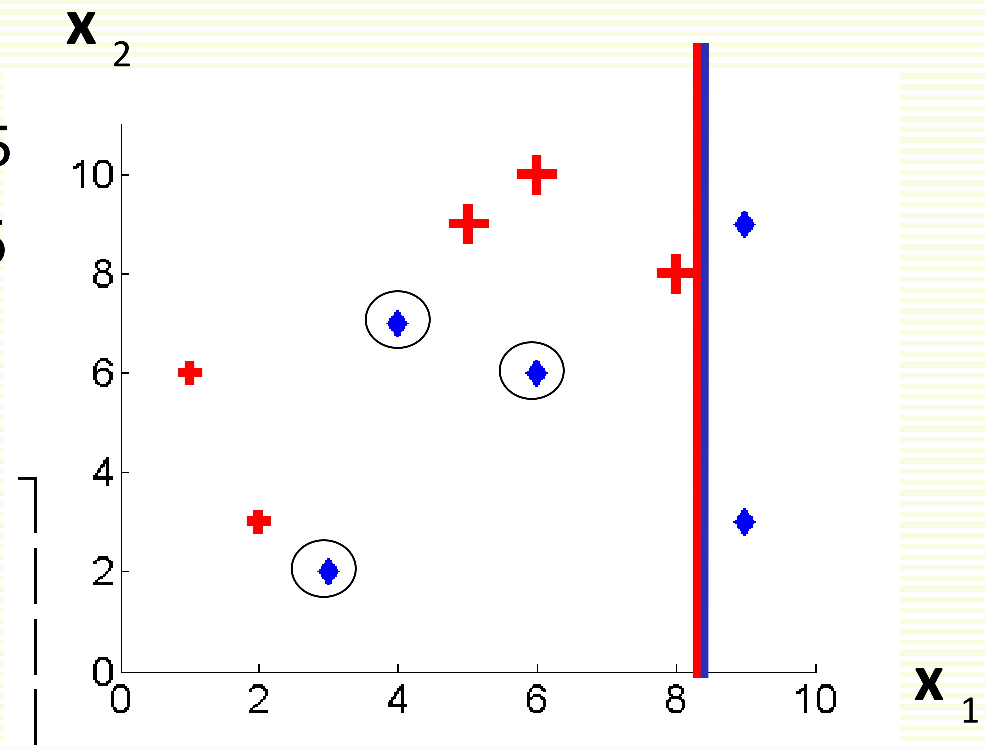
$$\mathbf{C}2 = \begin{bmatrix} 3 & 2 \\ 4 & 7 \\ 6 & 6 \\ 9 & 3 \\ 9 & 9 \end{bmatrix}$$

$$\mathbf{D}_2 = \begin{bmatrix} 0.07 \\ 0.07 \\ 0.17 \\ 0.17 \\ 0.17 \\ 0.07 \\ 0.07 \\ 0.07 \\ 0.07 \\ 0.07 \end{bmatrix}$$

AdaBoost Example: Round 2

- Classifier chosen:

$$h_2(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x_1} \le 8.5 \\ -1 & \text{if } \mathbf{x_1} > 8.5 \end{cases}$$



- $\varepsilon_2 = 0.21,\ \alpha_2 = 0.66$

$$C1 = \begin{bmatrix} 2 & 3 \\ 1 & 6 \\ 5 & 9 \\ 6 & 10 \\ 8 & 8 \end{bmatrix}$$

$$C2 = \begin{bmatrix} 3 & 2 \\ 4 & 7 \\ 6 & 6 \\ 9 & 3 \\ 9 & 9 \end{bmatrix}$$

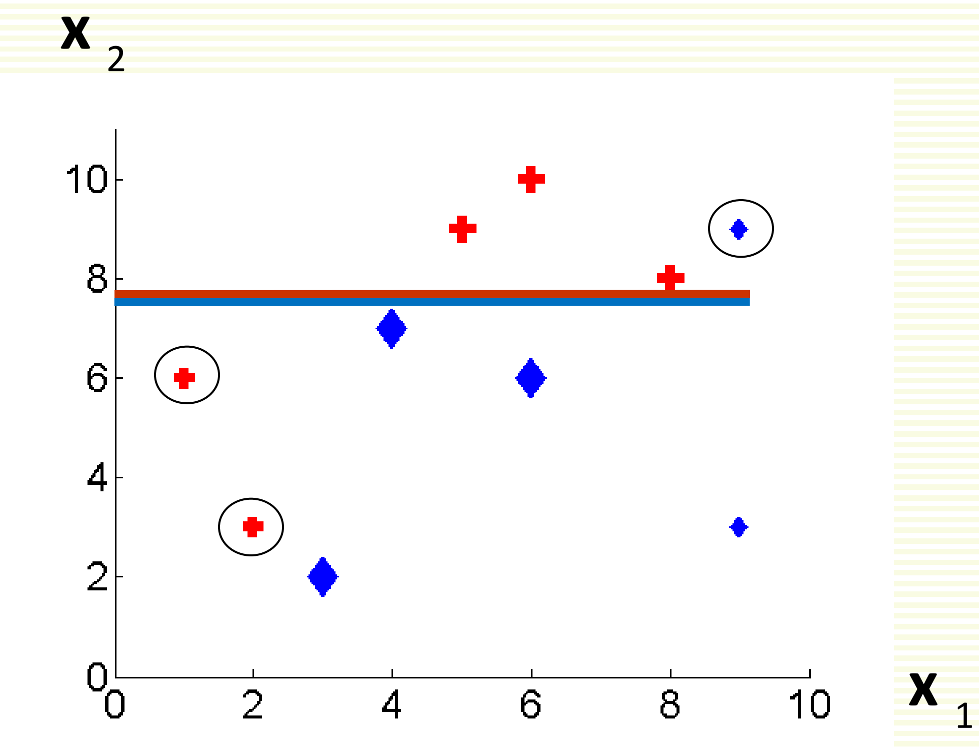
$$D_3 = \begin{bmatrix} 0.04 \\ 0.04 \\ 0.10 \\ 0.10 \\ 0.10 \\ 0.17 \\ 0.17 \\ 0.17 \\ 0.04 \\ 0.04 \end{bmatrix}$$

AdaBoost Example: Round 3

- Classifier chosen:

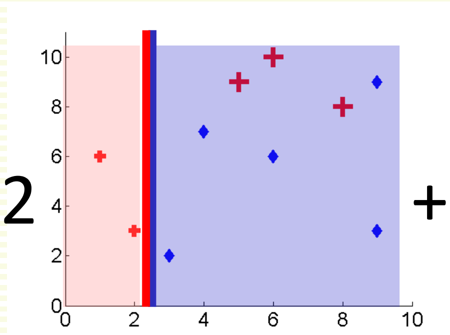
$$\mathbf{h}_3(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x_2} > 7.5 \\ -1 & \text{if } \mathbf{x_2} \leq 7.5 \end{cases}$$

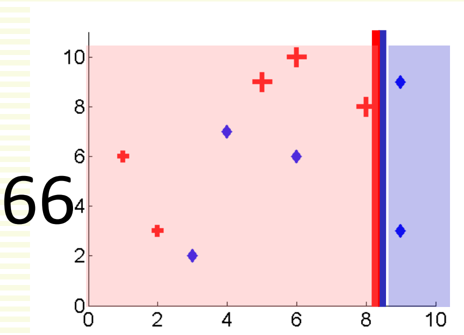
- $\varepsilon_3 = 0.12$, $\alpha_3 = 1.0$

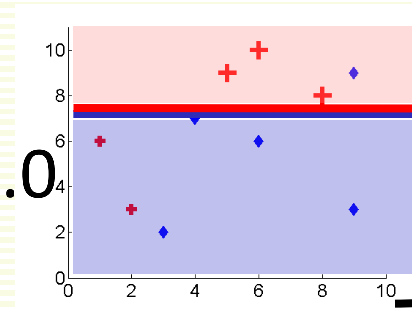


AdaBoost Final Classifier

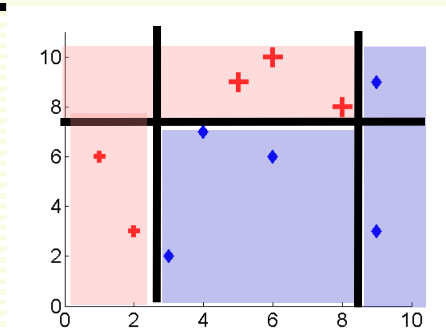
$\mathbf{f}_{\text{final}}(\mathbf{x}) =$

sign[0.42 

+0.66 

+1.0 

]

= 

AdaBoost Comments

- Can show that training error drops exponentially fast

$$Err_{train} \leq \exp\left(-2\sum_t \gamma_t^2\right)$$

- Here $\gamma_t = \varepsilon_t - 1/2$, where ε_t is classification error at round t
- Example: let errors for the first four rounds be, 0.3, 0.14, 0.06, 0.03, 0.01 respectively. Then

$$Err_{train} \leq \exp\left[-2\left(0.2^2 + 0.36^2 + 0.44^2 + 0.47^2 + 0.49^2\right)\right]$$

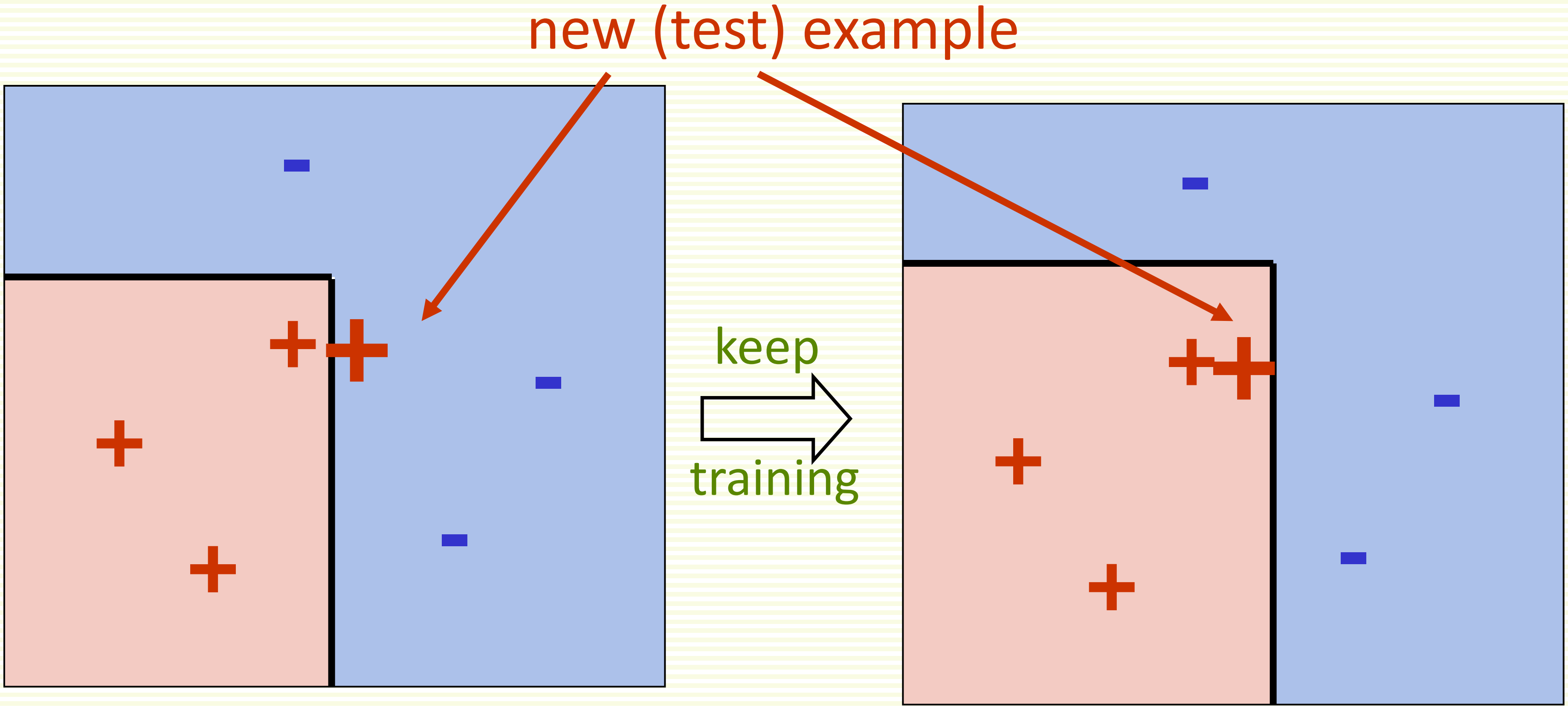
$$\approx 0.19$$

- Thus log (**n**) rounds of boosting are sufficient to get zero training error
  - provided weak learners are better than random

AdaBoost Comments

- We are really interested in the generalization properties of $\mathbf{f}_{FINAL}(\mathbf{x})$, not the training error
- AdaBoost was shown to have excellent generalization properties in practice
  - the more rounds, the more complex is the final classifier, so overfitting is expected as the training proceeds
  - but in the beginning researchers observed no overfitting of the data
  - It turns out it does overfit data eventually, if you run it really long
- It can be shown that boosting increases the margins of training examples, as iterations proceed
  - larger margins help better generalization
  - margins continue to increase even when training error reaches zero
  - helps to explain empirically observed phenomena: test error continues to drop even after training error reaches zero

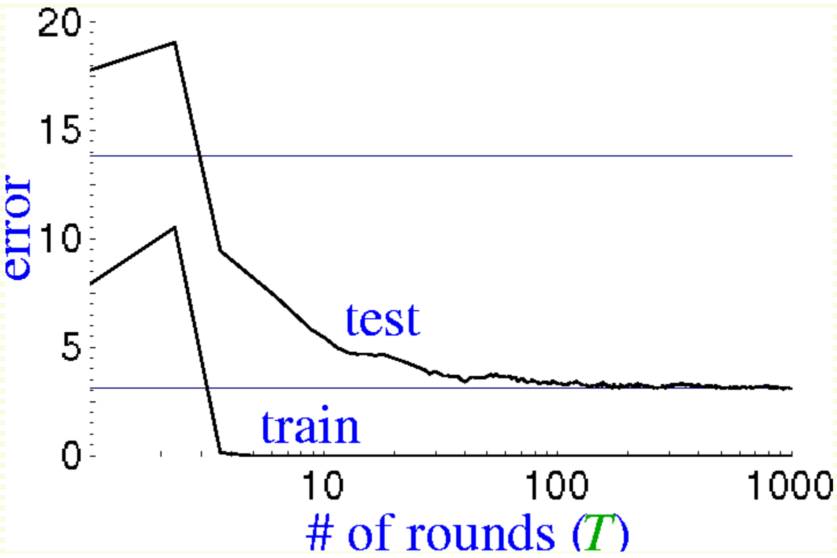
AdaBoost Example

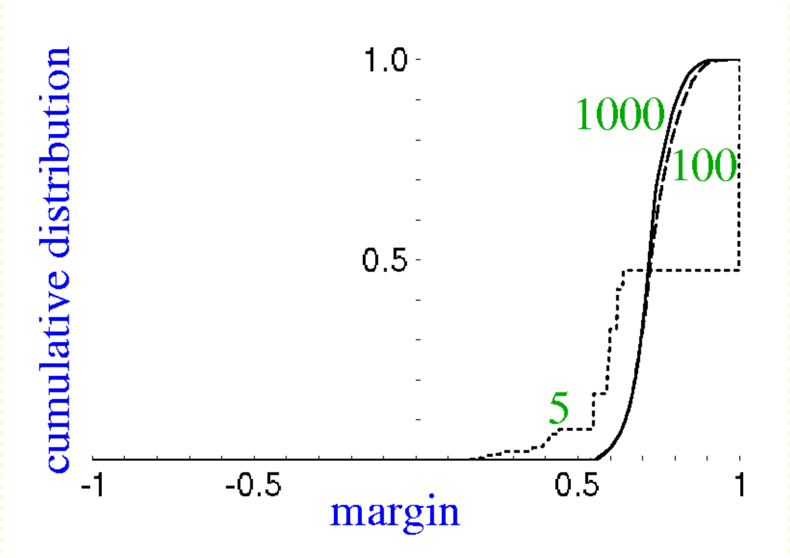


- zero training error

- zero training error
- larger margins helps better genarlization

Margin Distribution





Iteration number	5	100	1000
training 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

Practical Advantages of AdaBoost

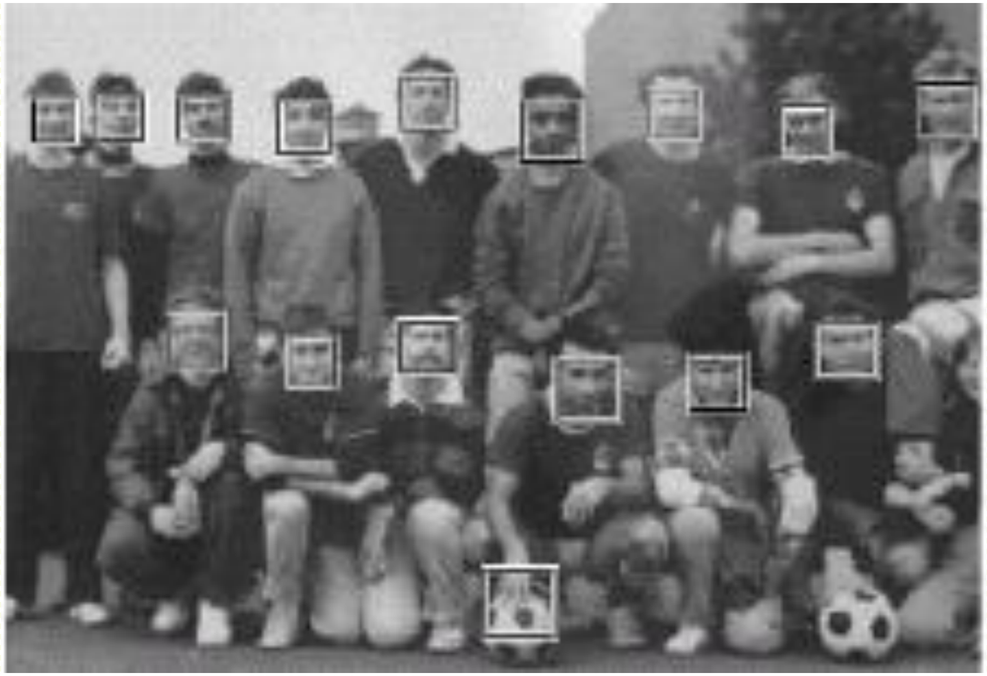
- Can construct arbitrarily complex decision regions
- Fast
- Simple
- Has only one parameter to tune, **T**
- Flexible: can be combined with any classifier
- provably effective (assuming weak learner)
  - shift in mind set: goal now is merely to find hypotheses that are better than random guessing

Caveats

- AdaBoost can <u>fail</u> if
  - weak hypothesis too complex (overfitting)
  - weak hypothesis too weak ($\gamma_t \to 0$ too quickly),
    - underfitting
- empirically, AdaBoost seems especially susceptible to noise
  - noise is the data with wrong labels

Applications

- Face Detection



- Object Detection

http://www.youtube.com/watch?v=2_0SmxvDbKs