6.034 FALL 2020

RECITATION 10: ADABOOST

- ☐ ADABOOST + SUPERVISED ML
- ☐ HOW TO ADABOOST
- ☐ EXAMPLE (2012 Q4)
- ☐ ADABOOST FACTS

ADABOOST + SUPERVISED ML

Supervised ML

[Bayes Nets, KNNs, ID Trees, NNs, SUMs] ← weak classifiers (h_i)

$$H(\vec{x}) = \text{sign}\left(\sum \alpha_i \, h_i(\vec{x})\right)$$

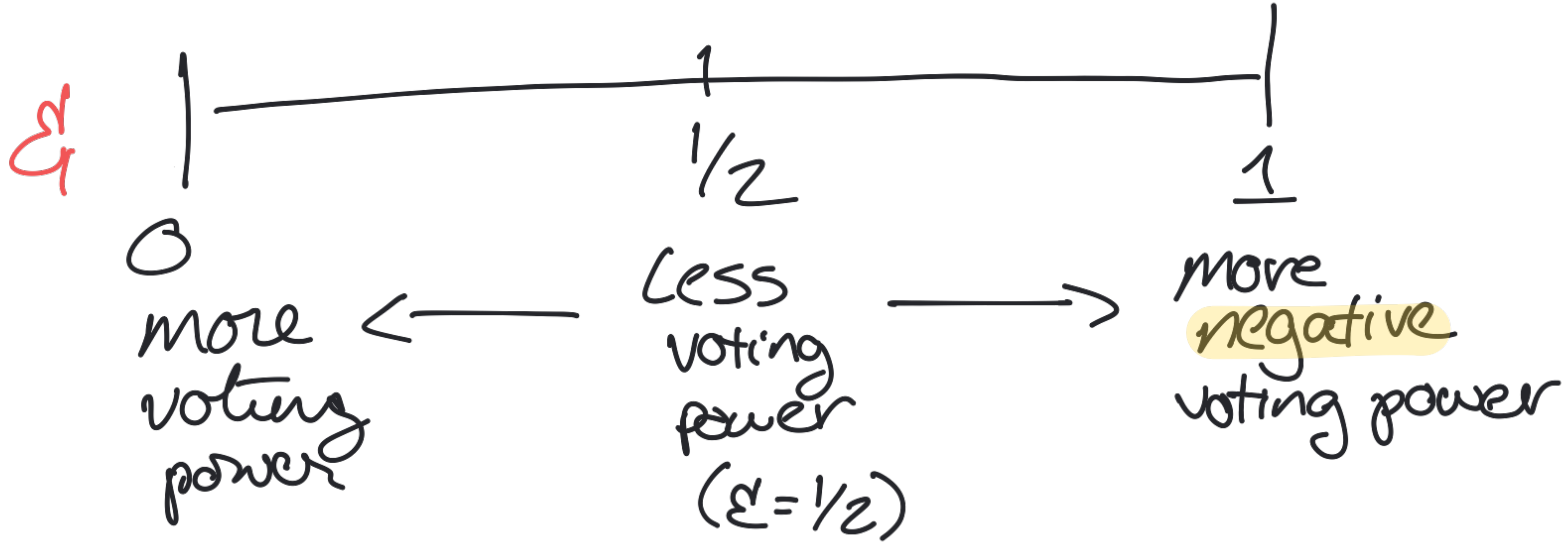
α_i: voting power

$h_i(\vec{x})$: weak classifier

ADABOOST + SUPERVISED ML

Each classifier has an error rate ε

$$\varepsilon \in [0, 1]$$



HOW TO ADABOOST

$H(\vec{x})$ assembled in a series of rounds

Each round:

1) Pick the best weak classifier ($h^*(\vec{x})$)
2) Assign voting power α^*
3) Append $h^*(\vec{x})$ to $H(\vec{x})$

HOW TO ADABOOST

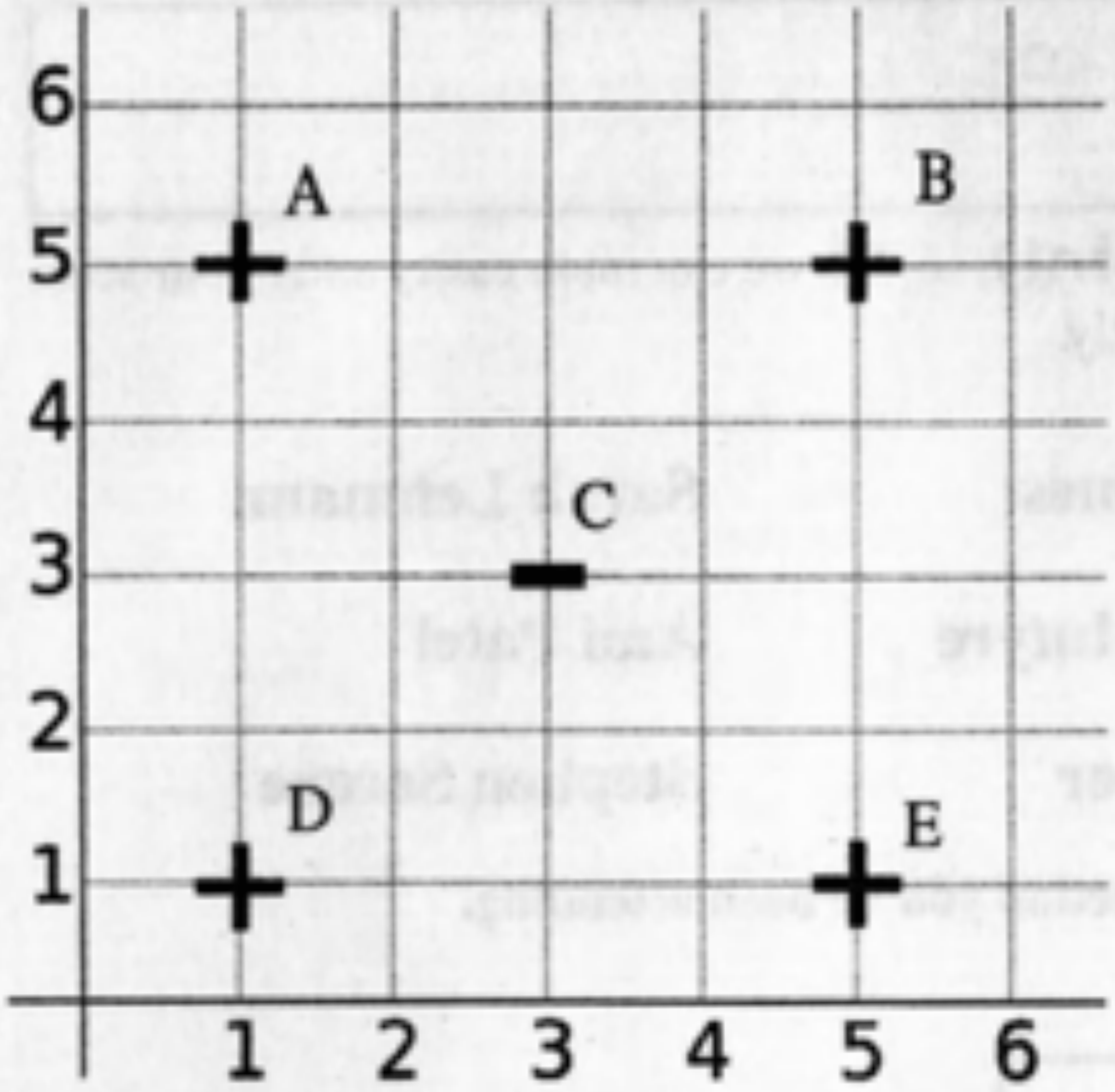
What is the best classifier ($h^*(\vec{x})$)?

- ☐ Makes fewest (or most) mistakes
- ☐ Adjusted to emphasize points that were previously misclassified

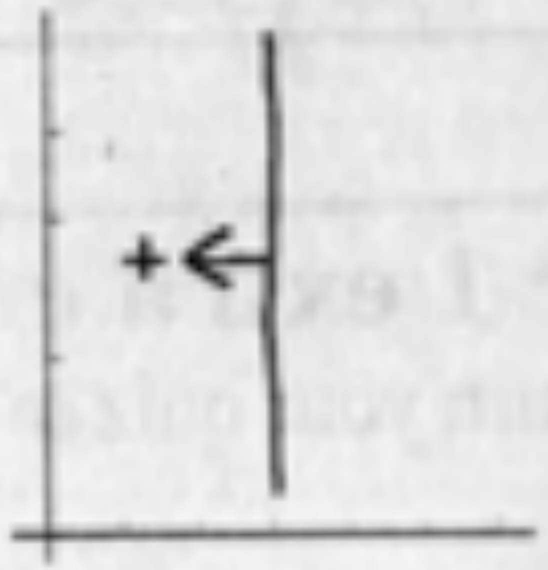
★ Mistakes are sometimes more informative than correct predictions

EXAMPLE (2012 Q4)

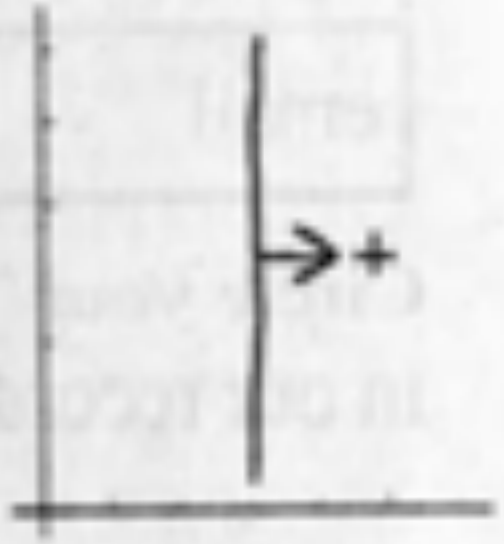
In this problem, you'll use boosting to construct a classifier for the following training dataset:



For your collection of weak classifiers, you'll use vertical line tests with either of the following forms



X ≤ T



X ≥ T

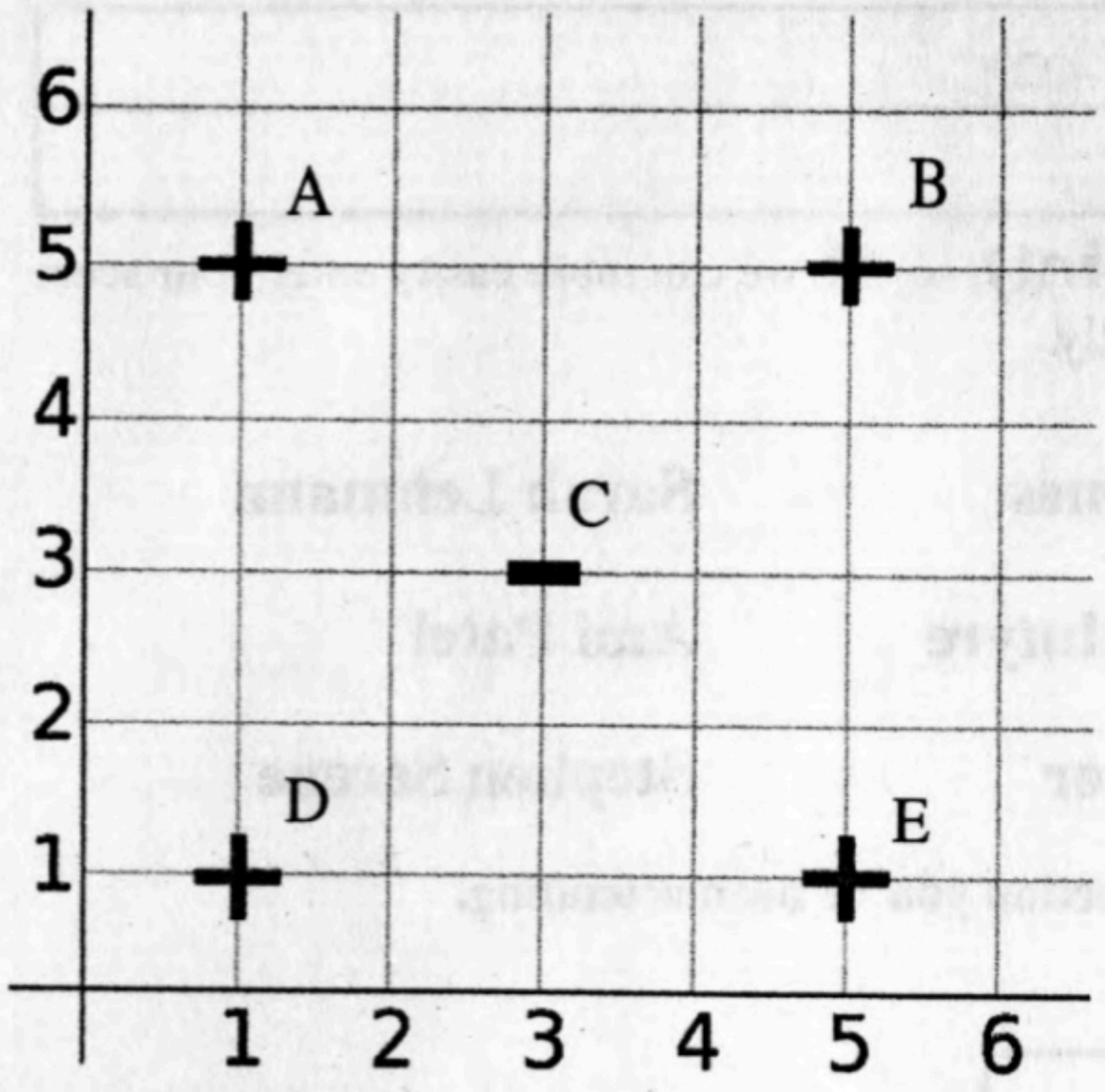
$$h(x, y) = \begin{cases} +1 & \text{if } x \leq T \\ -1 & \text{if } x > T \end{cases} \qquad h(x, y) = \begin{cases} +1 & \text{if } x \geq T \\ -1 & \text{if } x < T \end{cases}$$

The first kind of test classifies a point as positive if it's to the LEFT of a certain vertical line.

The second kind of test classifies a point as positive if it's to the RIGHT of a certain vertical line.

EXAMPLE (2012 Q4)

In this problem, you'll use boosting to construct a classifier for the following training dataset:

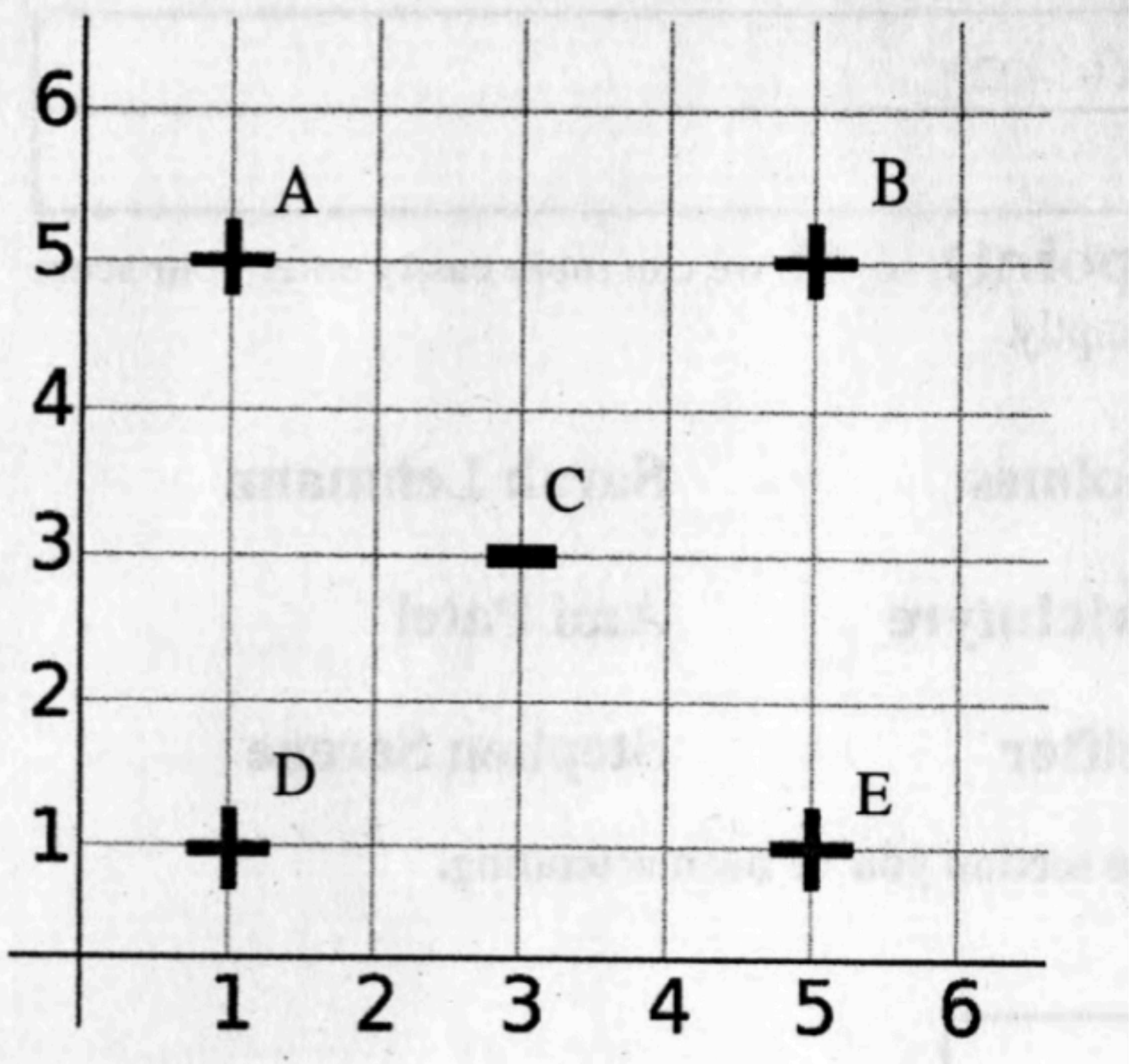


Weak classifiers $h_i(x)$

$h_1 \rightarrow$	$x \leq 2$	$x \geq 2$	$\leftarrow h_4$
$h_2 \rightarrow$	$x \leq 4$	$x \geq 4$	$\leftarrow h_5$
$h_3 \rightarrow$	$x \leq 6$	$x \geq 6$	$\leftarrow h_6$

EXAMPLE (2012 Q4)

In this problem, you'll use boosting to construct a classifier for the following training dataset:



$h_1: x \leq 2$	B, E
$h_2: x \leq 4$	B, C, E
$h_3: x \leq 6$	C
$h_4: x \geq 2$	A, C, D
$h_5: x \geq 4$	A, D
$h_6: x \geq 6$	A, B, D, E

	Round 1	Round 2	Round 3
w_A			
w_B			
w_C			
w_D			
w_E			
$h_1\ (x<2)$			
$h_2\ (x<4)$			
$h_3\ (x<6)$			
$h_4\ (x>2)$			
$h_5\ (x>4)$			
$h_6\ (x>6)$			

weights of the points

ϵ of each h_i

EXAMPLE (2012 Q4)

Adaboost Algorithm

1) Uniformly initialize all the weights: $\omega = \frac{1}{N}$

	Round 1	Round 2	Round 3
w_A	1/5		
w_B	1/5		
w_C	1/5		
w_D	1/5		
w_E	1/5		
$h_1(x<2)$			
$h_2(x<4)$			
$h_3(x<6)$			
$h_4(x>2)$			
$h_5(x>4)$			
$h_6(x>6)$			

EXAMPLE (2012 Q4)

Adaboost Algorithm

1) Uniformly initialize all the weights: $w = \frac{1}{N}$

2) Calculate error rate ε_i for each $h_i(\vec{x})$

$$\varepsilon_i = \sum_{\substack{\text{wrong}\\ \text{point } j}} w_j$$

	Round 1	Round 2	Round 3
w_A	$1/5$		
w_B	$1/5$		
w_C	$1/5$		
w_D	$1/5$		
w_E	$1/5$		
$h_1 (x<2)$	$2/5$		
$h_2 (x<4)$	$3/5$		
$h_3 (x<6)$	$1/5$		
$h_4 (x>2)$	$3/5$		
$h_5 (x>4)$	$2/5$		
$h_6 (x>6)$	$4/5$		

EXAMPLE (2012 Q4)

Adaboost Algorithm

1) Uniformly initialize all the weights: $\omega = \frac{1}{N}$
2) Calculate error rate ε_i for each $h_i(\vec{x})$
3) Pick best $h_i(\vec{x})$ - lowest ε_i $(h^*(\vec{x}))$

	Round 1	Round 2	Round 3
w_A	1/5		
w_B	1/5		
w_C	1/5		
w_D	1/5		
w_E	1/5		
$h_1(x<2)$	2/5		
$h_2(x<4)$	3/5		
$h_3(x<6)$	1/5		
$h_4(x>2)$	3/5		
$h_5(x>4)$	2/5		
$h_6(x>6)$	4/5		

	Round 1	Round 2	Round 3
$h^*(\vec{x})$	$h_3: x < 6$		
ε_1	$1/5$		
α^*			

$$H(\vec{x}) = \text{sign}(\cdot \cdot \cdot)$$

EXAMPLE (2012 Q4)

Adaboost Algorithm

1) Uniformly initialize all the weights: $\omega = \frac{1}{N}$
2) Calculate error rate ε_i for each $h_i(\vec{x})$
3) Pick best $h_i(\vec{x})$ - lowest ε_i $(h^*(\vec{x}))$
4) Calculate voting power (α^*) for $h^*(\vec{x})$

$$\alpha = \frac{1}{2} \ln\left(\frac{1-\varepsilon}{\varepsilon}\right)$$

	Round 1	Round 2	Round 3
$h^*(\vec{x})$	$h_3 : x < 6$		
ε_1	$1/5$		
α^*	$\frac{1}{2}\ln(4)$		

$$\downarrow$$

$$H(\vec{x}) = \text{sign}\left(\frac{1}{2}\ln(4) \cdot h_3(\vec{x}) \ldots\right)$$

EXAMPLE (2012 Q4)

Adaboost Algorithm

1) Uniformly initialize all the weights: $\omega = \frac{1}{N}$

2) Calculate error rate ε_i for each $h_i(\vec{x})$

3) Pick best $h_i(\vec{x})$ - lowest ε_i $(h^*(\vec{x}))$

4) Calculate voting power (α^*) for $h^*(\vec{x})$

5) FINISHED?

EXAMPLE (2012 Q4)

When are we FINISHED?

- ☐ if $H(\vec{x})$ is good enough
- ☐ sufficient rounds
- ☐ no good classifiers left
  - ☐ $h^*(\vec{x})$ has $\mathcal{E}_t = 1/2$

EXAMPLE (2012 Q4)

Adaboost Algorithm

1) Uniformly initialize all the weights: $\omega = \frac{1}{N}$
2) Calculate error rate ε_i for each $h_i(\vec{x})$
3) Pick best $h_i(\vec{x})$ - lowest ε_i $(h^*(\vec{x}))$
4) Calculate voting power (α^*) for $h^*(\vec{x})$
5) FINISHED? $\longrightarrow$ YES: STOP | NO: step 6
6) Update weights to emphasize points that were misclassified

EXAMPLE (2012 Q4)

Update weights:

$$w_{i,new} = \begin{cases} \frac{1}{2}\left(\frac{1}{1-\varepsilon}\right) w_{i,old}, & \text{if } \vec{x}_i \text{ classified correctly} \\ \frac{1}{2}\left(\frac{1}{\varepsilon}\right) w_{i,old}, & \text{if } \vec{x}_i \text{ classified incorrectly} \end{cases}$$

	Round 1	Round 2	Round 3
w_A	1/5	1/8	
w_B	1/5	1/8	
w_C	1/5	1/2	
w_D	1/5	1/8	
w_E	1/5	1/8	
$h_1\ (x<2)$	2/5		
$h_2\ (x<4)$	3/5		
$h_3\ (x<6)$	1/5		
$h_4\ (x>2)$	3/5		
$h_5\ (x>4)$	2/5		
$h_6\ (x>6)$	4/5		

	Round 1	Round 2	Round 3
w_A	1/5	1/8	
w_B	1/5	1/8	
w_C	1/5	1/2	
w_D	1/5	1/8	
w_E	1/5	1/8	
$h_1 (x<2)$	2/5	2/8	
$h_2 (x<4)$	3/5	6/8	
$h_3 (x<6)$	1/5	4/8	
$h_4 (x>2)$	3/5	6/8	
$h_5 (x>4)$	3/5	2/8	
$h_6 (x>6)$	4/5	4/8	

	Round 1	Round 2	Round 3
$h^*(\vec{x})$	$h_3 : x < 6$	$h_1 : x < 2$	
ε_t	$1/5$	$1/4$	
α^*	$\frac{1}{2}\ln(4)$	$\frac{1}{2}\ln(3)$	

$$H(\vec{x}) = \text{sign}\left(\tfrac{1}{2}\ln(4)\cdot h_3(\vec{x}) + \tfrac{1}{2}\ln(3)\cdot h_1(\vec{x}) \ldots\right)$$

	Round 1	Round 2	Round 3
w_A	1/5	1/8	1/12
w_B	1/5	1/8	3/12
w_C	1/5	1/2	4/12
w_D	1/5	1/8	1/12
w_E	1/5	1/8	3/12
$h_1\ (x<2)$	2/5	2/8	
$h_2\ (x<4)$	3/5	6/8	
$h_3\ (x<6)$	1/5	4/8	
$h_4\ (x>2)$	3/5	6/8	
$h_5\ (x>4)$	2/5	2/8	
$h_6\ (x>6)$	4/5	4/8	

	Round 1	Round 2	Round 3
w_A	1/5	1/8	1/12
w_B	1/5	1/8	3/12
w_C	1/5	1/2	4/12
w_D	1/5	1/8	1/12
w_E	1/5	1/8	3/12
$h_1 (x<2)$	2/5	2/8	1/2
$h_2 (x<4)$	3/5	6/8	10/12
$h_3 (x<6)$	1/5	4/8	4/12
$h_4 (x>2)$	3/5	6/8	1/2
$h_5 (x>4)$	2/5	2/8	2/12
$h_6 (x>6)$	4/5	4/8	8/12

	Round 1	Round 2	Round 3
$h^*(\vec{x})$	$h_3: x < 6$	$h_1: x < 2$	$h_5: x > 4$
ϵ_t	$1/5$	$1/4$	$2/12$
α^*	$\frac{1}{2}\ln(4)$	$\frac{1}{2}\ln(3)$	$\frac{1}{2}\ln(5)$

$$H(\vec{x}) = \text{sign}\left(\frac{1}{2}\ln(4)\cdot h_3(\vec{x}) + \frac{1}{2}\ln(3)\cdot h_1(\vec{x}) + \right.$$

$$\left. 1/2\ln(5)\, h_5(\vec{x})\right)$$

FACTS ABOUT BOOSTING

(except for weird degenerate cases)

Useful trick: weights update proportionally from prior values

1) $\sum_{i}^{N} w_i = 1$

2) $\sum_{\text{right}} w_i = \sum_{\text{wrong}} w_j = 1/2$

3) $h^*(\vec{x})$ from the previous round will have $\varepsilon_t = 1/2$ in the next round

4) After first round: $0 < w_i \leq 1/2$ and $w_i \neq 0$

5) $\varepsilon = 0$ ... $\varepsilon = 1/2$ ... $\varepsilon = 1$

$\alpha = \infty$ ... $\alpha = 0$ ... $\alpha = -\infty$

FACTS ABOUT BOOSTING

(except for weird degenerate cases)

6) The same $h_i(\vec{x})$ will never be chosen twice in a row (see 2 and 3)

7) You cannot classify perfectly with 2 imperfect $h_i(\vec{x})$ because one will have a larger α

8) $W_{outlier}$ (the weight of an outlier point) will continually increase until some chosen $h_i(\vec{x})$ classifies it (the system gets anxious)

9) If 3+ classifiers have disjoint errors, then one can make $H(\vec{x})$ perfect by setting all α equal*.
(But it's not guaranteed AdaBoost will pick them first.)

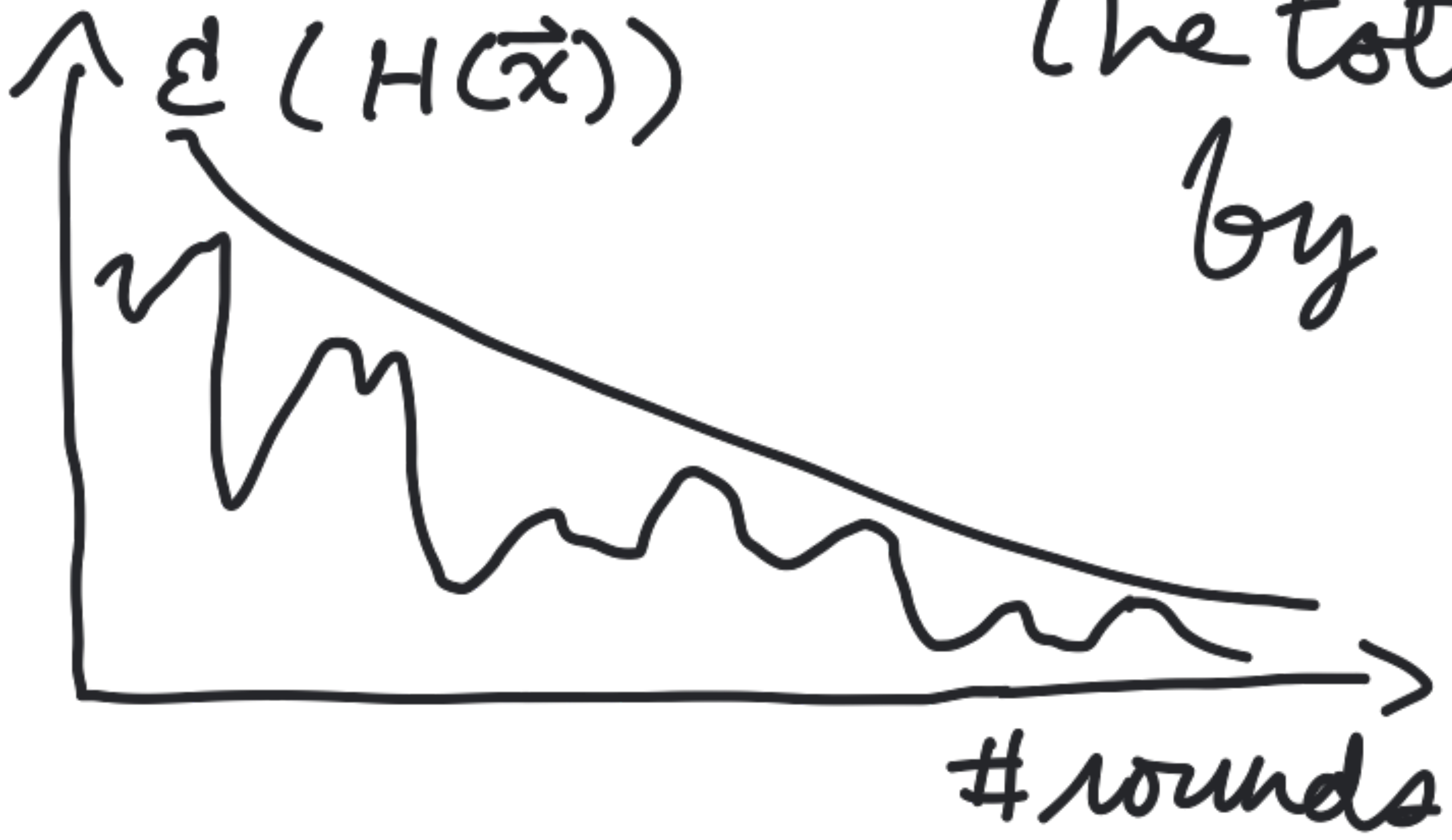
*or if αs satisfy the Δ inequality

FACTS ABOUT BOOSTING

(except for weird degenerate cases)

10) If 3+ classifiers have <u>overlapping errors</u>, it may still be possible to construct a perfect $H(\vec{x})$ from them.

11) AdaBoost tends to <u>not</u> overfit because it gives less weight to $h_i(\vec{x})$s that correctly classify outliers

12) 

The total error is upper-bounded by some exponential decay function.

FACTS ABOUT BOOSTING

(except for weird degenerate cases)

err($h_i(\vec{x})$)
= set of errors
of $h_i(\cdot)$

13) for min(ε) is best:

if $\mathrm{err}(h_1(\vec{x})) \supset \mathrm{err}(h_2(\vec{x}))$

then $\varepsilon(h_1) > \varepsilon(h_2)$

so $h_1(\vec{x})$ would never be picked

for $\max(|\varepsilon - \frac{1}{2}|)$ is best:

if $\mathrm{err}(h_1(\vec{x})) \supset \mathrm{err}(h_2(\vec{x})) \supset \mathrm{err}(h_3(\vec{x}))$,

then $\varepsilon(h_1) > \varepsilon(h_2) > \varepsilon(h_3)$

so $h_2(\vec{x})$ would never be picked