BOOSTING

Ronen Nir & Itay Evron

Outline

- Boosting – General idea
- AdaBoost
  - Visual demo
  - Detailed example
  - From a loss perspective

Boosting – General idea

- Create a strong classifier by combining multiple weak classifiers
- A weak classifier guarantees a slightly better-than-random error rate

Initialize a uniform distribution $D^{(1)}$ over trainset S

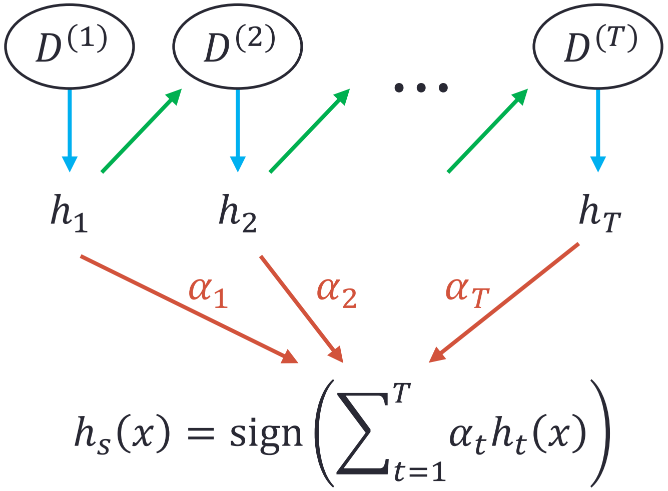
For t=1, …, T:

Learn (weak) model $h_t = \mathcal{A}(S, D^{(t)})$

Compute error

Update the distribution $D^{(t+1)}$

Return final hypothesis

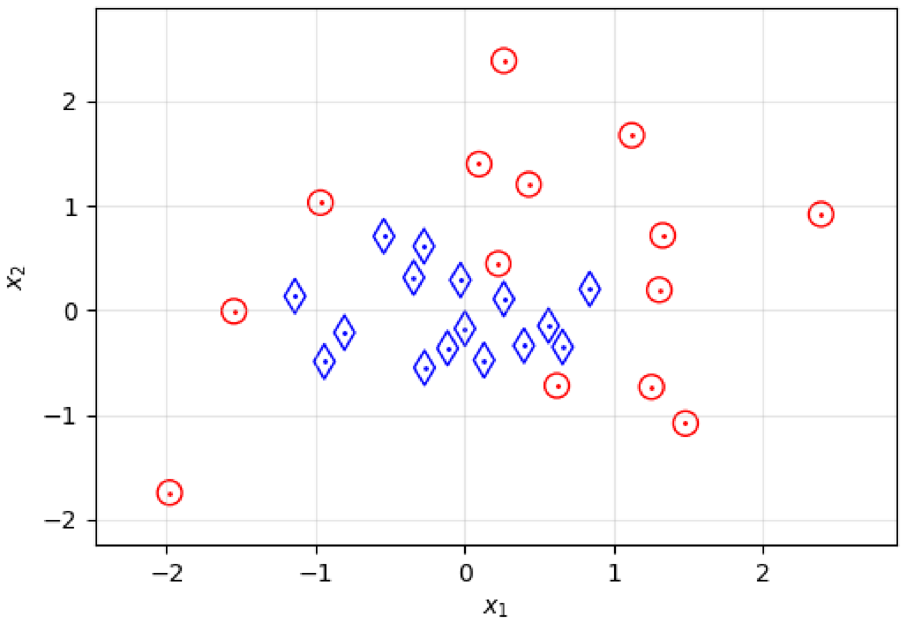


Based on Illustration from
“Pattern Recognition and Machine Learning”, Bishop, 2006

AdaBoost – Visualization

Dataset

- 2 features x_1, x_2
- 2 classes (red/blue)

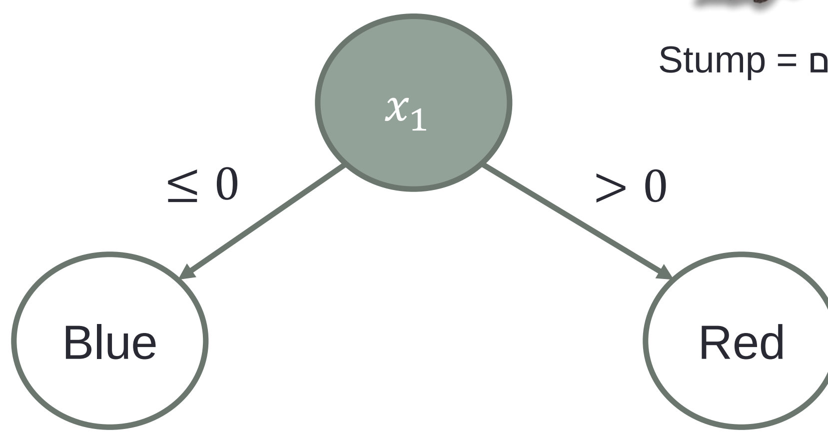


Weak learner choice

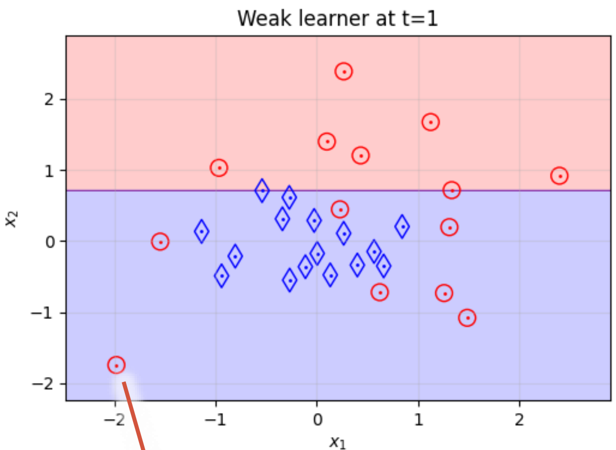
Decision stumps,

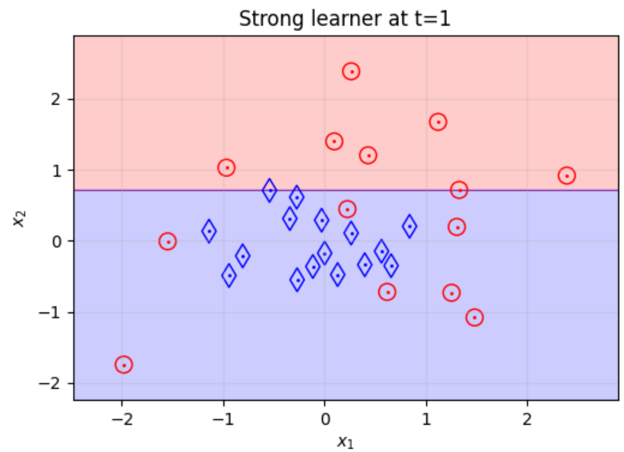
i.e., decision trees with depth 1

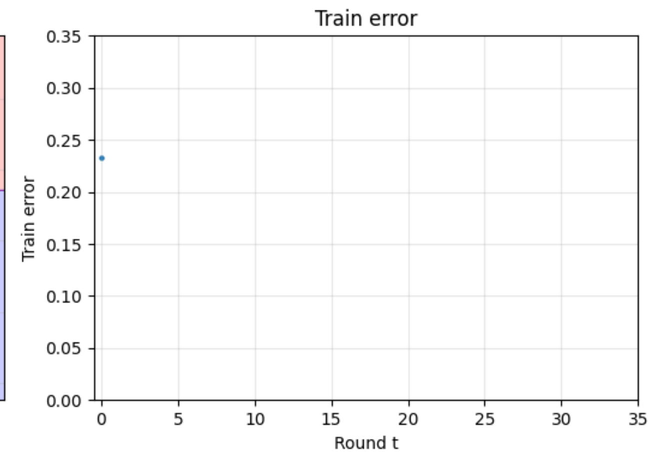
Stump = גֶּדֶם



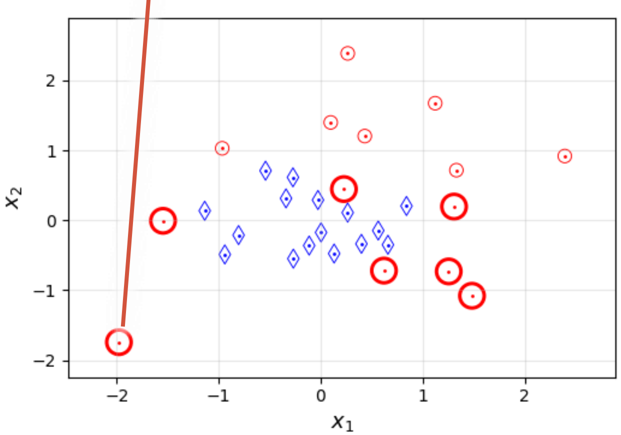
AdaBoost – Visualization





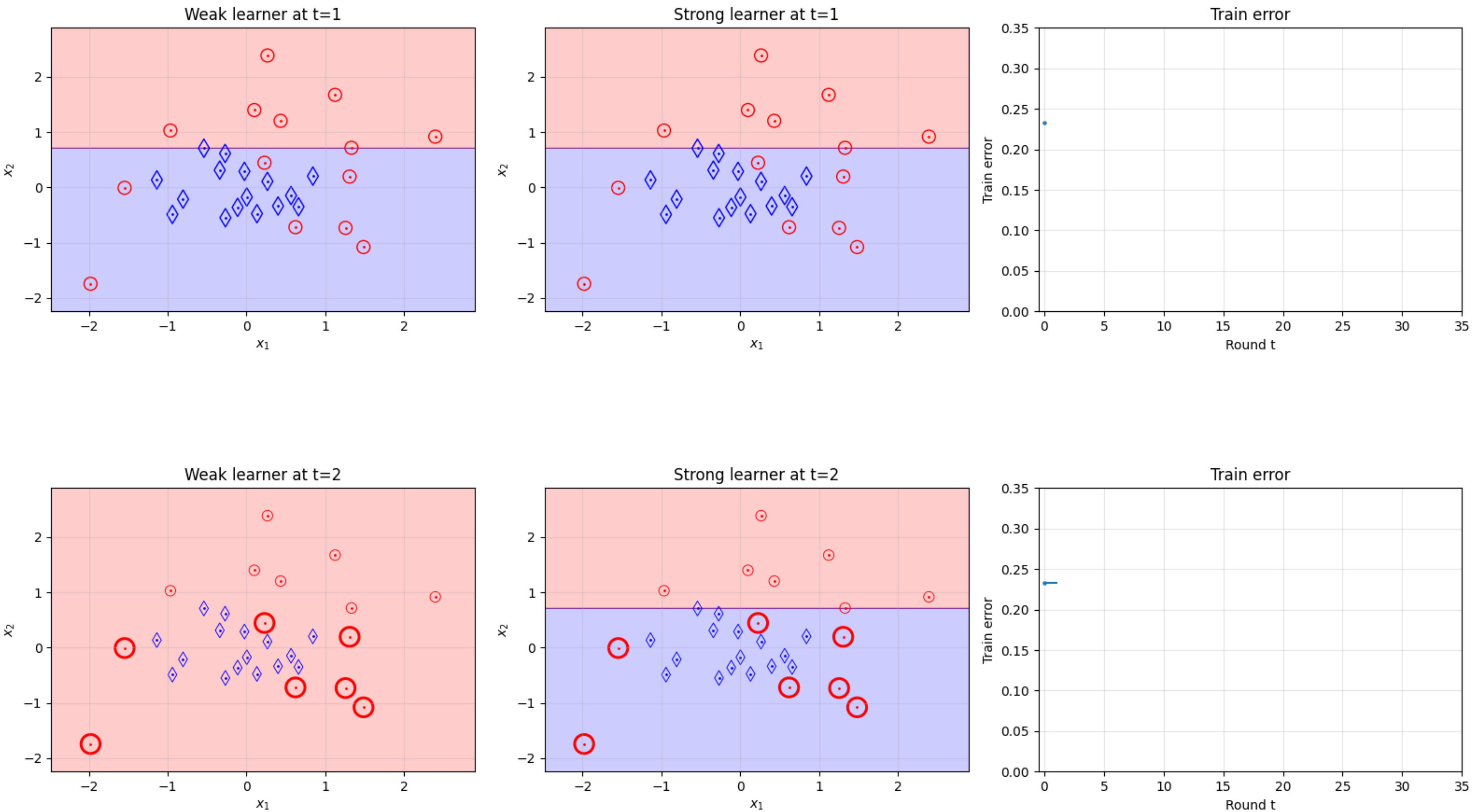


Misclassified points
get higher probability



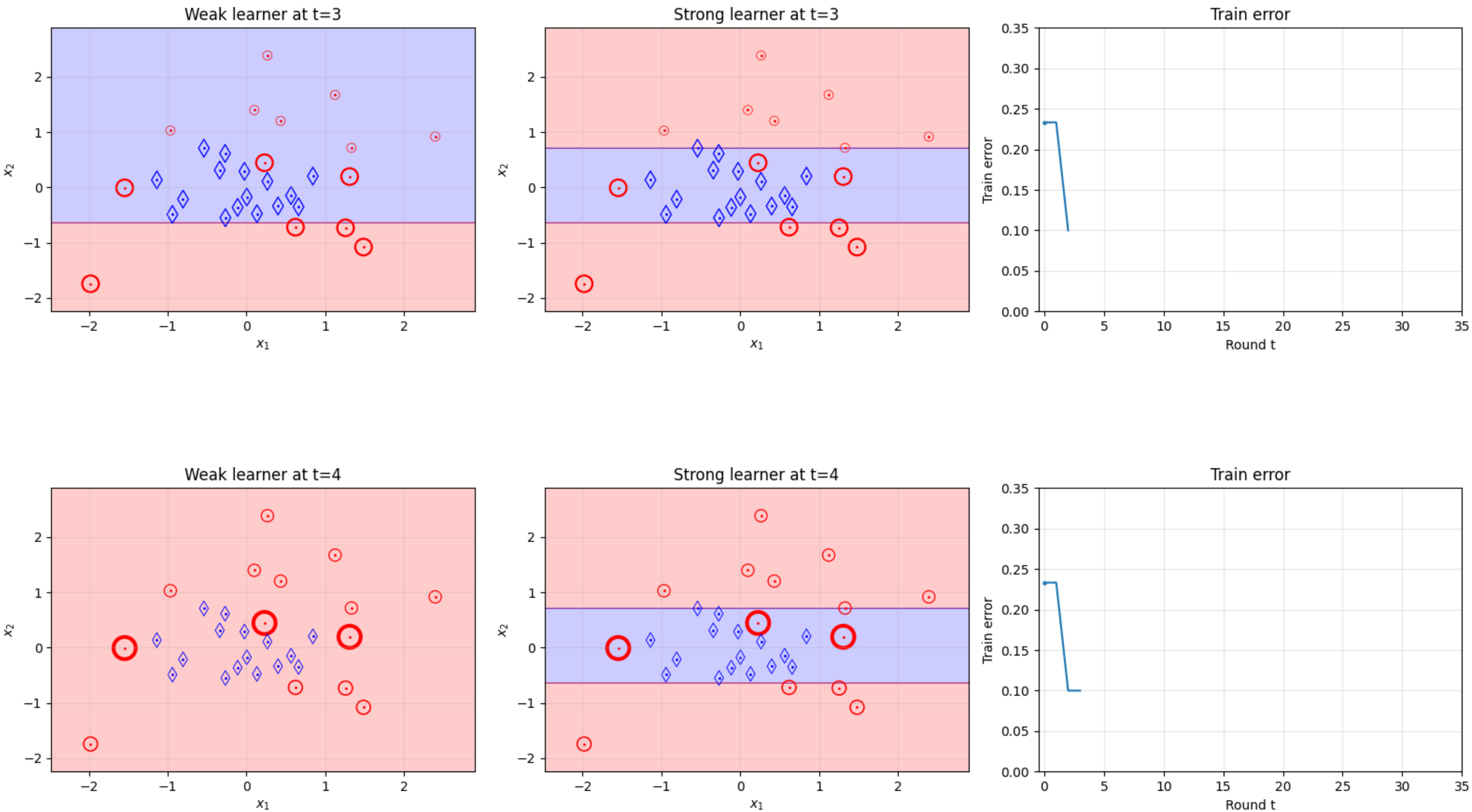
Based on a simulation by Geoff Ruddock

AdaBoost – Visualization



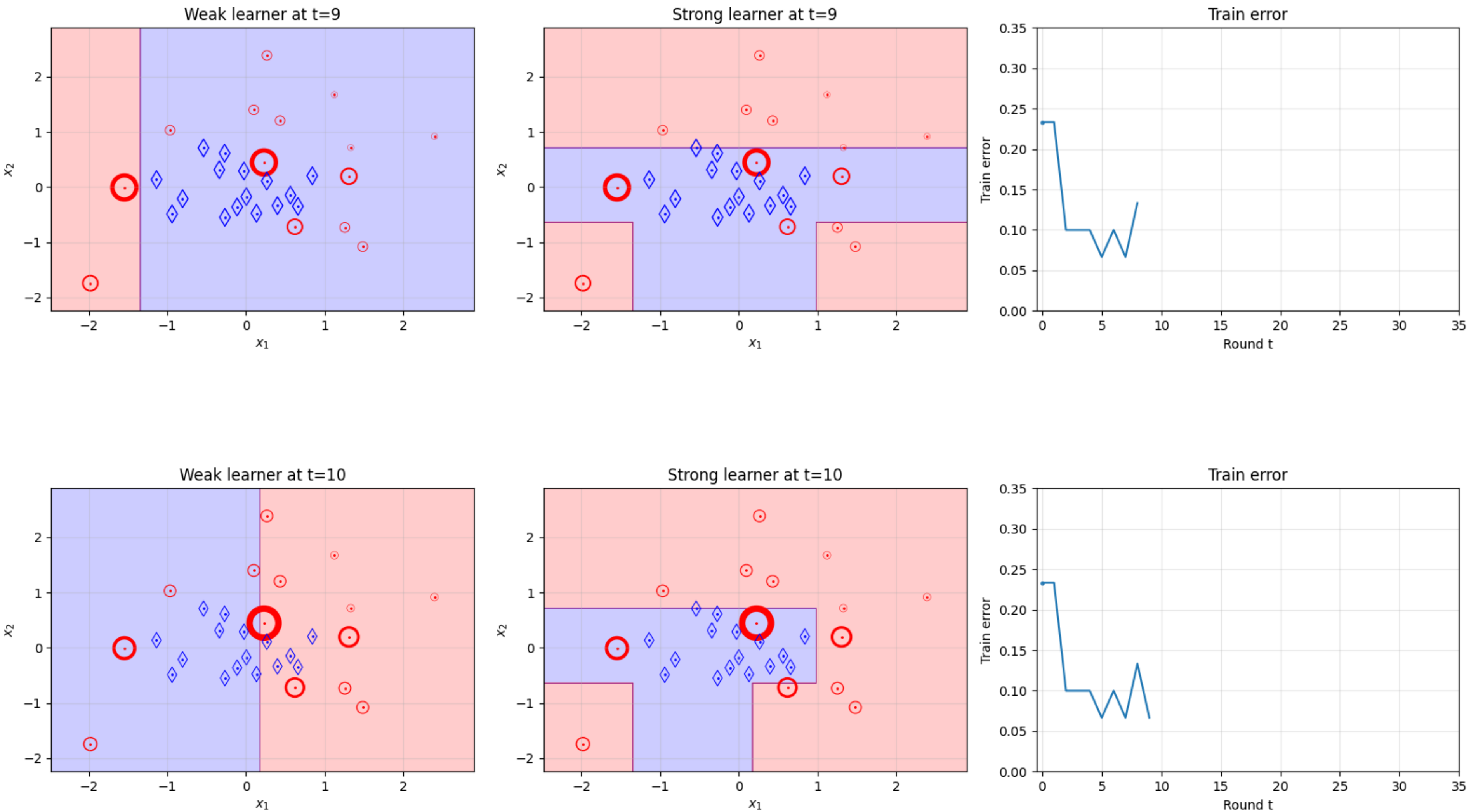
Based on a simulation by Geoff Ruddock

AdaBoost – Visualization



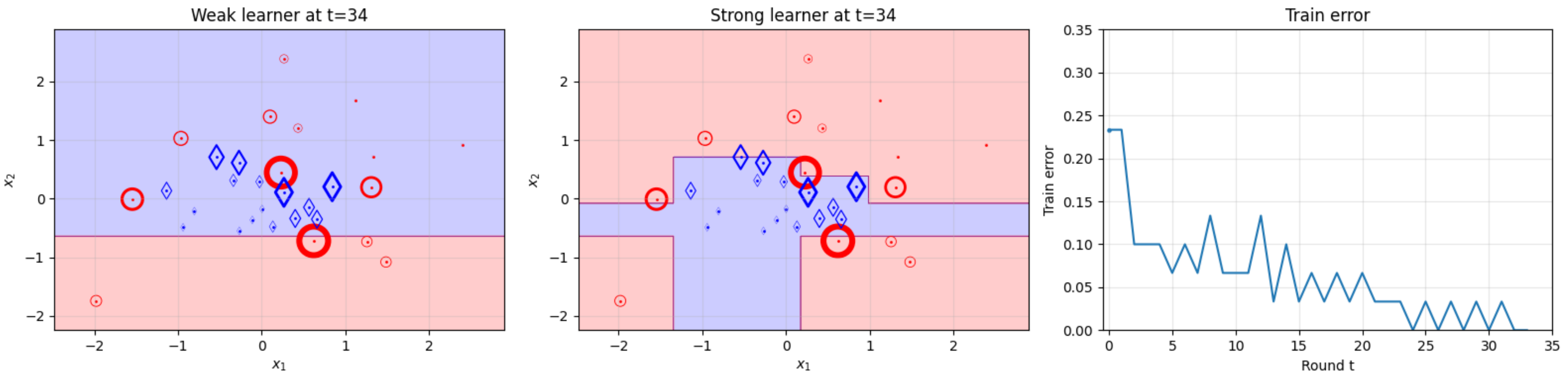
Based on a simulation by Geoff Ruddock

AdaBoost – Visualization

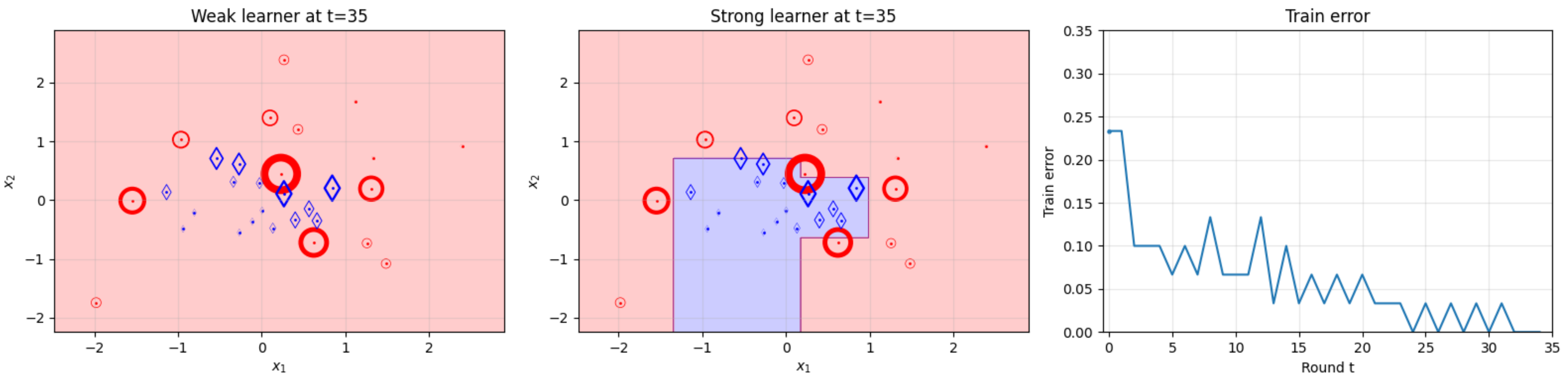


Based on a simulation by Geoff Ruddock

AdaBoost – Visualization



Where do you think we'll get better generalization?



Based on a simulation by Geoff Ruddock

Detailed Example

Some students want to avoid "hard" courses.

They want to classify courses as "easy" or "hard".

- First, they collect (very little) data.

Course ID	Hard?	Final exam?	Theoretical?	Advanced?	HW Number
1	1	-1	1	1	1
2	1	-1	1	-1	3
3	1	-1	1	-1	5
4	1	1	-1	-1	5
5	1	1	-1	1	5
6	-1	-1	1	-1	1
7	-1	-1	-1	-1	3

Model choice

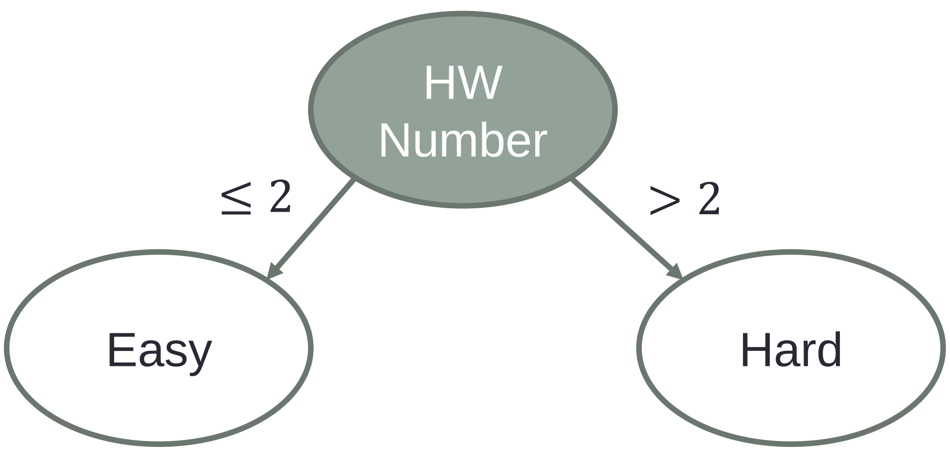
Some students want to avoid “hard” courses.

They want to classify courses as “easy” or “hard”.

- First, they collect (very little) data.

Course ID	Hard?	Final exam?	Theoretical?	Advanced?	HW Number
⋮	⋮	⋮	⋮	⋮	⋮

- Then, they train a model using AdaBoost with decision stumps.



Building weak classifiers

Course ID	Hard?	Final exam?	Theoretical?	Advanced?	HW Number
1	1	-1	1	1	1
2	1	-1	1	-1	3
3	1	-1	1	-1	5
4	1	1	-1	-1	5
5	1	1	-1	1	5
6	-1	-1	1	-1	1
7	-1	-1	-1	-1	3

After some preprocessing, we propose the following weak decision stumps classifiers. >4

Exercise: Which samples do they misclassify?

Classifier	Attribute	Value	Misclassified
A	Constant "Hard"		
B	Theoretical	1	
C	Advanced	1	
D	# HW	> 2	
E	# HW	> 4	

Proposed weak classifiers

Classifier	Attribute	Value	Misclassified
A	Constant “Hard”		6, 7
B	Theoretical	1	4, 5, 6
C	Advanced	1	2, 3, 4
D	# HW	> 2	1, 7
E	# HW	> 4	1, 2

AdaBoost

- We are ready to run AdaBoost.

Initialize $D^{(1)} = \left(\frac{1}{m}, \frac{1}{m}, \ldots, \frac{1}{m}\right)$

For t=1, …, T:

 # Learn (weak) model

$$h_t = \mathcal{A}(S, D^{(t)})$$

 # Compute error on current distribution

$$\epsilon_t = \sum_i D_i^{(t)} \cdot \mathbf{1}_{h_t(x_i) \neq y_i}$$

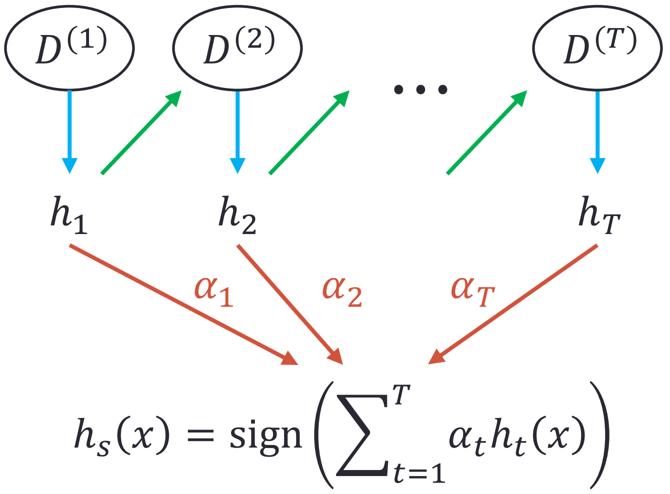
 # Update weights and distribution

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

$$D_i^{(t+1)} = \frac{D_i^{(t)} \exp(-\alpha_t y_i h_t(x_i))}{\sum_j D_j^{(t)} \exp(-\alpha_t y_j h_t(x_j))}$$

 # Return final hypothesis

$$h_s(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$



- Fill in the classifiers, errors, weights, and distributions of the first 3 rounds.

	$t=1$
$D_1^{(t)}$	$^1/_7$
$D_2^{(t)}$	$^1/_7$
$D_3^{(t)}$	$^1/_7$
$D_4^{(t)}$	$^1/_7$
$D_5^{(t)}$	$^1/_7$
$D_6^{(t)}$	$^1/_7$
$D_7^{(t)}$	$^1/_7$
h_t	pick the classifier with the lowest weight
ϵ_t	1\7 + 1\7
α_t	1\2 * ln(5\2)

Initialize $D^{(1)} = \left(\frac{1}{m}, \frac{1}{m}, \ldots, \frac{1}{m}\right)$

For t=1, …, T:

 # Learn (weak) model

$$h_t = \mathcal{A}(S, D^{(t)})$$

 # Compute error on current distribution

$$\epsilon_t = \sum_i D_i^{(t)} \cdot \mathbf{1}_{h_t(x_i) \neq y_i}$$

 # Update weights and distribution

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

$$D_i^{(t+1)} = \frac{D_i^{(t)} \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

Possible classifiers (ERM)

#	Attribute	Value	Misclassified	ϵ
A	Constant “Hard”		6, 7	
B	Theoretical	1	4, 5, 6	
C	Advanced	1	2, 3, 4	
D	# HW	> 2	1, 7	
E	# HW	> 4	1, 2	

	$t=1$	$t=2$
$D_1^{(t)}$	$^1/_7$	$^1/_{10}$
$D_2^{(t)}$	$^1/_7$	$^1/_{10}$
$D_3^{(t)}$	$^1/_7$	$^1/_{10}$
$D_4^{(t)}$	$^1/_7$	$^1/_{10}$
$D_5^{(t)}$	$^1/_7$	$^1/_{10}$
$D_6^{(t)}$	$^1/_7$	$^{2.5}/_{10}$
$D_7^{(t)}$	$^1/_7$	$^{2.5}/_{10}$
h_t	**A**	
ϵ_t	$^2/_7$	
α_t	$0.5 \ln \frac{5}{2}$	

Updating the distribution

- We used the following weak classifier:

#	Attribute	Value	Misclassified
A	Constant "Hard"		6, 7 => we wan't to up their weight

- We wish to compute the new distribution

$$D_i^{(t+1)} = \frac{D_i^{(t)} \exp\left(-\alpha_t \overbrace{y_i h_t(x_i)}^{\in\{1,-1\}}\right)}{Z_t}$$

- First, let us compute the unnormalized distribution

$\frac{1}{Z_t}$ במקום לרשום

$$D_i^{(2)} \propto \begin{cases} D_i^{(1)} \exp(\alpha_1) = \frac{1}{7} e^{\frac{1}{2}\ln\left(\frac{5}{2}\right)} = \frac{1}{7} \cdot \sqrt{\frac{5}{2}} \propto \sqrt{\frac{5}{2}} \propto 5 & i = 6,7 \\ D_i^{(1)} \exp(-\alpha_1) = \frac{1}{7} e^{-\frac{1}{2}\ln\left(\frac{5}{2}\right)} = \frac{1}{7}\sqrt{\frac{2}{5}} \propto \sqrt{\frac{2}{5}} \propto 2 & else \end{cases}$$

$sum = 5 \cdot 2 + 2 \cdot 5 = 20 \Rightarrow D_7^{(2)} = D_6^{(2)} = \frac{5}{20} = \frac{2.5}{10}$

- Now, fill in the normalized distribution

	$t = 1$	$t = 2$
$D_1^{(t)}$	$^1/_7$	$^1/_{10}$
$D_2^{(t)}$	$^1/_7$	$^1/_{10}$
$D_3^{(t)}$	$^1/_7$	$^1/_{10}$
$D_4^{(t)}$	$^1/_7$	$^1/_{10}$
$D_5^{(t)}$	$^1/_7$	$^1/_{10}$
$D_6^{(t)}$	$^1/_7$	$^{2.5}/_{10}$
$D_7^{(t)}$	$^1/_7$	$^{2.5}/_{10}$
h_t	**A**	
ϵ_t	$^2/_7$	
α_t	$0.5 \ln \frac{5}{2}$	

Initialize $D^{(1)} = \left(\frac{1}{m}, \frac{1}{m}, \ldots, \frac{1}{m}\right)$

For t=1, …, T:

 # Learn (weak) model

$$h_t = \mathcal{A}(S, D^{(t)})$$

 # Compute error on current distribution

$$\epsilon_t = \sum_i D_i^{(t)} \cdot \mathbf{1}_{h_t(x_i) \neq y_i}$$

 # Update weights and distribution

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

$$D_i^{(t+1)} = \frac{D_i^{(t)} \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

Possible classifiers (ERM)

#	Attribute	Value	Misclassified	ϵ
A	Constant "Hard"		6, 7	1/2
B	Theoretical	1	4, 5, 6	4.5/10
C	Advanced	1	2, 3, 4	3/10
D	# HW	> 2	1, 7	3.5/10
E	# HW	> 4	1, 2	2/10

	$t=1$	$t=2$
$D_1^{(t)}$	$1/7$	$1/10$
$D_2^{(t)}$	$1/7$	$1/10$
$D_3^{(t)}$	$1/7$	$1/10$
$D_4^{(t)}$	$1/7$	$1/10$
$D_5^{(t)}$	$1/7$	$1/10$
$D_6^{(t)}$	$1/7$	$2.5/10$
$D_7^{(t)}$	$1/7$	$2.5/10$
h_t	**A**	**E**
ϵ_t	$2/7$	$1/5$
α_t	$0.5 \ln \frac{5}{2}$	$\ln 2$

Initialize $D^{(1)} = \left(\frac{1}{m}, \frac{1}{m}, \dots, \frac{1}{m}\right)$

For t=1, …, T:

 # Learn (weak) model

$h_t = \mathcal{A}(S, D^{(t)})$

 # Compute error on current distribution

$\epsilon_t = \sum_i D_i^{(t)} \cdot \mathbf{1}_{h_t(x_i) \neq y_i}$

 # Update weights and distribution

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

$D_i^{(t+1)} = \frac{D_i^{(t)} \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$

$\mathcal{E}_1 = \frac{1}{10} + \frac{1}{10} = \frac{1}{5}$

$D_i^{(3)} = \frac{D_i^{(2)} e^{-\ln(2) \cdot y_i h_i(x_i)}}{Z_t}$

$D_2^{(3)} = D_1^{(3)} \propto D_1^{(2)} e^{\ln(2)} = \frac{2}{10} \propto 4$

$D_3^{(3)} = D_4^{(3)} = D_5^{(3)} \propto D_3^{(2)} e^{-\ln(2)} = \frac{1}{20} \propto 1$

$D_6^{(3)} = D_7^{(3)} \propto D_6^{(2)} e^{-\ln(2)} = \frac{2.5}{20} \propto 2.5$

Sum $= 4 \cdot 2 + 3 \cdot 1 + 2 \cdot 2.5 = 16$

$D_2^{(3)} = D_1^{(3)} = \frac{4}{16} = \frac{8}{32}$

$D_3^{(3)} = D_4^{(3)} = D_5^{(3)} = \frac{1}{16} = \frac{2}{32}$

$D_6^{(3)} = D_7^{(3)} = \frac{2.5}{16} = \frac{5}{32}$

	$t=1$	$t=2$	$t=3$
$D_1^{(t)}$	$^1/_7$	$^1/_{10}$	$^8/_{32}$
$D_2^{(t)}$	$^1/_7$	$^1/_{10}$	$^8/_{32}$
$D_3^{(t)}$	$^1/_7$	$^1/_{10}$	$^2/_{32}$
$D_4^{(t)}$	$^1/_7$	$^1/_{10}$	$^2/_{32}$
$D_5^{(t)}$	$^1/_7$	$^1/_{10}$	$^2/_{32}$
$D_6^{(t)}$	$^1/_7$	$^{2.5}/_{10}$	$^5/_{32}$
$D_7^{(t)}$	$^1/_7$	$^{2.5}/_{10}$	$^5/_{32}$
h_t	**A**	**E**	**B**
ϵ_t	$^2/_7$	$^1/_5$	$^9/_{32}$
α_t	$0.5\ln\frac{5}{2}$	$\ln 2$	$0.5\ln\frac{23}{9}$

Updating the distribution

- We used the following weak classifier:

#	Attribute	Value	Misclassified
E	# HW	> 4	1, 2

- We attach results for a 3rd iteration of AdaBoost without relevant computations.

- Extra: manually run this iteration by yourselves and prove the presented results.

Final hypothesis

Computed weights

	$t = 1$	$t = 2$	$t = 3$
h_t	**A**	**E**	**B**
α_t	$0.5 \ln \frac{5}{2}$	$\ln 2$	$0.5 \ln \frac{23}{9}$

Chosen classifiers

#	Attribute	Value	Misclassified
A	Constant "Hard"		6, 7
E	# HW	> 4	1, 2
B	Theoretical	1	4, 5, 6

- Final hypothesis: $$h_S(x) = sign\left(0.5 \ln \frac{5}{2} A(x) + \ln 2\, E(x) + 0.5 \ln \frac{23}{9} B(x)\right)$$ $$\approx sign(0.46 + 0.69E(x) + 0.47B(x))$$

  where: A, B, E return $+1$ or -1

- How many of the data points are classified correctly?
  - Notice: each two weights $\alpha's$ are larger than the third
  - We get a simple majority vote
- Only one wrong prediction (example 6)

AdaBoost: Guarantees

- AdaBoost constructs a strong hypothesis $h_s(x) = \mathrm{sign}(\sum_{k=1}^{t} \alpha_k h_k(x))$.
- Guarantee: AdaBoost's training error after T iterations

  is bounded by $L_S(h_s) \triangleq \frac{1}{m} \sum_i \mathbf{1}_{h_t(x_i) \neq y_i} \leq \exp\{-\gamma^2 T\}$ for some $\gamma \in (0, {}^1/_2)$.

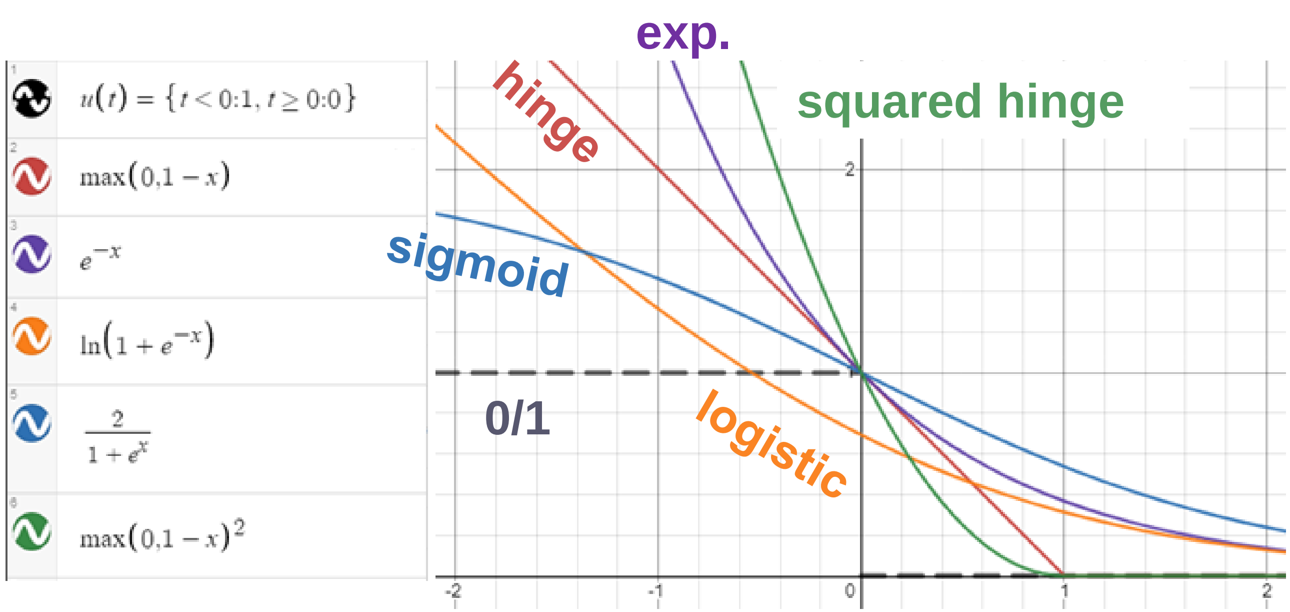
  (under mild conditions; without proof)
- Corollary: AdaBoost reaches zero training error eventually. (under the same conditions)

AdaBoost optimizes the exponential loss

- AdaBoost constructs a strong hypothesis $h_s(x) = \cancel{\mathrm{sign}}\left(\sum_{k=1}^{t} \alpha_k h_k(x)\right)$.
  - Why does it work?
  - We will show that AdaBoost greedily optimizes the exponential loss.

Focus on the “unthresholded” hypothesis

- Recall the exponential loss: $\mathcal{L}_{\exp}(h_s) = \frac{1}{m}\sum_{i=1}^{m} \ell_{\exp}(x_i, y_i) = \frac{1}{m}\sum_{i=1}^{m} \mathrm{e}^{-y_i \sum_{k=1}^{t} \alpha_k h_k(x_i)}$
- The exponential loss is a proxy to the zero-one loss: $\mathcal{L}_{\exp}(h_s) \geq \mathcal{L}_{0/1}(h_s)$



AdaBoost optimizes the exponential loss

- Goal: Show AdaBoost optimizes $\mathcal{L}_{\exp}(h_s) = \frac{1}{m}\sum_{i=1}^{m} \exp\{-y_i \sum_{k=1}^{t} \alpha_k h_k(x_i)\}$.
- Assume: $h_1, \alpha_1, \dots, h_{t-1}, \alpha_{t-1}$ were already chosen.
- Question: How to choose h_t, α_t to minimize $\mathcal{L}_{\exp}(h_s)$?
- Lemma: $\mathcal{L}_{exp}^{(t)} \propto e^{-\alpha_t} + (e^{\alpha_t} - e^{-\alpha_t}) \underbrace{\sum_i D_i^{(t)} \mathbf{1}_{h_t(x_i)\neq y_i}}_{\triangleq \epsilon_t}$

  (will be proven next)
- Answer: Use "two steps" greedy optimization:
  - Choose h_t minimizing the weighted error ϵ_t, *e.g.*, with ERM *w.r.t.* $D^{(t)}$.
  - Choose α_t minimizing $\mathcal{L}_{exp}^{(t)}$ given $h_1, \alpha_1, \dots, h_{t-1}, \alpha_{t-1}$ and h_t.

Initialize $D^{(1)} = \left(\frac{1}{m}, \dots, \frac{1}{m}\right)$

For t=1, …, T:

$$h_t = \mathcal{A}\left(S, D^{(t)}\right)$$

$$\epsilon_t = \sum_i D_i^{(t)} \cdot \mathbf{1}_{h_t(x_i)\neq y_i}$$

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

$$D_i^{(t+1)} = \frac{D_i^{(t)} \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

$$h_s(x) = \mathrm{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

1. Derive $\frac{\partial}{\partial \alpha_t}\mathcal{L}_{exp}^{(t)}$ and use it to prove that the choice $\alpha_t = \frac{1}{2}\log\left(\frac{1}{\epsilon_t} - 1\right)$ is optimal.

AdaBoost optimizes the exponential loss

- Goal: Show AdaBoost optimizes $\mathcal{L}_{\exp}(h_s) = \frac{1}{m}\sum_{i=1}^{m} \exp\{-y_i \sum_{k=1}^{t} \alpha_k h_k(x_i)\}$.
- Assume: $h_1, \alpha_1, \dots, h_{t-1}, \alpha_{t-1}$ were already chosen.
- Question: How to choose h_t, α_t to minimize $\mathcal{L}_{\exp}(h_s)$?
- Lemma: $\mathcal{L}_{exp}^{(t)} \propto e^{-\alpha_t} + (e^{\alpha_t} - e^{-\alpha_t}) \underbrace{\sum_i D_i^{(t)} \mathbf{1}_{h_t(x_i)\neq y_i}}_{\triangleq \epsilon_t}$
- Proof:
  2. Extra: Show that $D_i^{(t)} = c \cdot \exp\left(-y_i \sum_{k=1}^{t-1} \alpha_k h_k(x_i)\right)$, for $c > 0$

$$D_i^{(t)} = \frac{1}{Z_{t-1}} D_i^{(t-1)} \exp\left(-\alpha_{t-1} y_i h_{t-1}(x_i)\right)$$

$$= \frac{1}{Z_{t-1} Z_{t-2}} D_i^{(t-2)} \exp\left(-\alpha_{t-2} y_i h_{t-2}(x_i)\right) \exp\left(-\alpha_{t-1} y_i h_{t-1}(x_i)\right)$$

$$= \frac{1}{Z_{t-1} Z_{t-2}} D_i^{(t-2)} \exp\left(-y_i\left(\alpha_{t-2} h_{t-2}(x_i) - \alpha_{t-1} h_{t-1}(x_i)\right)\right)$$

$$= \cdots = \underbrace{\frac{1}{\prod_{k=1}^{t-1} Z_k} \underbrace{D_i^{(0)}}_{=1/m}}_{\triangleq c} \exp\left(-y_i \sum_{k=1}^{t-1} \alpha_k h_k(x_i)\right) \quad \blacksquare$$

Initialize $D^{(1)} = \left(\frac{1}{m}, \dots, \frac{1}{m}\right)$

For t=1, …, T:

$$h_t = \mathcal{A}\left(S, D^{(t)}\right)$$

$$\epsilon_t = \sum_i D_i^{(t)} \cdot \mathbf{1}_{h_t(x_i)\neq y_i}$$

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

$$D_i^{(t+1)} = \frac{D_i^{(t)} \exp\left(-\alpha_t y_i h_t(x_i)\right)}{Z_t}$$

$$h_s(x) = \mathrm{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

AdaBoost optimizes the exponential loss

- Goal: Show AdaBoost optimizes $\mathcal{L}_{\exp}(h_s) = \frac{1}{m}\sum_{i=1}^{m} \exp\{-y_i \sum_{k=1}^{t} \alpha_k h_k(x_i)\}$.
- Assume: $h_1, \alpha_1, \dots, h_{t-1}, \alpha_{t-1}$ were already chosen.
- Question: How to choose h_t, α_t to minimize $\mathcal{L}_{\exp}(h_s)$?
- Lemma: $\mathcal{L}_{exp}^{(t)} \propto e^{-\alpha_t} + (e^{\alpha_t} - e^{-\alpha_t}) \underbrace{\sum_i D_i^{(t)} \mathbf{1}_{h_t(x_i)\neq y_i}}_{\triangleq \epsilon_t}$
- Proof:
  2. Extra: Show that $D_i^{(t)} = c \cdot \exp\left(-y_i \sum_{k=1}^{t-1} \alpha_k h_k(x_i)\right)$, for $c > 0$
  3. Show that $\mathcal{L}_{exp}^{(t)} \propto \sum_i D_i^{(t)} \exp\{-y_i \alpha_t h_t(x_i)\}$

Initialize $D^{(1)} = \left(\frac{1}{m}, \dots, \frac{1}{m}\right)$

For t=1, …, T:

$$h_t = \mathcal{A}\left(S, D^{(t)}\right)$$

$$\epsilon_t = \sum_i D_i^{(t)} \cdot \mathbf{1}_{h_t(x_i)\neq y_i}$$

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

$$D_i^{(t+1)} = \frac{D_i^{(t)} \exp\left(-\alpha_t y_i h_t(x_i)\right)}{Z_t}$$

$$h_s(x) = \mathrm{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

$$\mathcal{L}_{exp}^{(t)} = \sum_i \exp\left\{-y_i \sum_{k=1}^{t} \alpha_k h_k(x_i)\right\} = \sum_i \exp\left\{\left(-y_i \sum_{k=1}^{t-1} \alpha_k h_k(x_i)\right) - y_i \alpha_t h_t(x_i)\right\}$$

$$= \sum_i \underbrace{\exp\left\{-y_i \sum_{k=1}^{t-1} \alpha_k h_k(x_i)\right\}}_{=\frac{1}{c}\cdot D_i^{(t)}} \exp\{-y_i \alpha_t h_t(x_i)\} \propto \sum_i D_i^{(t)} \exp\{-y_i \alpha_t h_t(x_i)\}$$

AdaBoost optimizes the exponential loss

- Goal: Show AdaBoost optimizes $\mathcal{L}_{\exp}(h_s) = \frac{1}{m}\sum_{i=1}^{m} \exp\{-y_i \sum_{k=1}^{t} \alpha_k h_k(x_i)\}$.
- Assume: $h_1, \alpha_1, \dots, h_{t-1}, \alpha_{t-1}$ were already chosen.
- Question: How to choose h_t, α_t to minimize $\mathcal{L}_{\exp}(h_s)$?
- Lemma: $\mathcal{L}_{exp}^{(t)} \propto e^{-\alpha_t} + (e^{\alpha_t} - e^{-\alpha_t}) \underbrace{\sum_i D_i^{(t)} \mathbf{1}_{h_t(x_i)\neq y_i}}_{\triangleq \epsilon_t}$
- Proof:
  2. Extra: Show that $D_i^{(t)} = c \cdot \exp\left(-y_i \sum_{k=1}^{t-1} \alpha_k h_k(x_i)\right)$, for $c > 0$
  3. Show that $\mathcal{L}_{exp}^{(t)} \propto \sum_i D_i^{(t)} \exp\{-y_i \alpha_t h_t(x_i)\}$
  4. Extra: Prove the following form of the exp. loss after t rounds is

$$\mathcal{L}_{exp}^{(t)} \propto e^{-\alpha_t} \sum_i D_i^{(t)} + (e^{\alpha_t} - e^{-\alpha_t}) \sum_i D_i^{(t)} \mathbf{1}_{h_t(x_i)\neq y_i}$$

Do so by filling in the blanks:

$$\mathcal{L}_{exp}^{(t)} \propto \sum_i D_i^{(t)} \exp\{-y_i \alpha_t h_t(x_i)\} = \cdots = e^{-\alpha_t} \sum_{i:y_i = h_t(x_i)} D_i^{(t)} + e^{\alpha_t} \sum_{i:y_i \neq h_t(x_i)} D_i^{(t)}$$

$$= \cdots = e^{-\alpha_t} \sum_i D_i^{(t)} + (e^{\alpha_t} - e^{-\alpha_t}) \sum_i D_i^{(t)} \mathbf{1}_{h_t(x_i)\neq y_i}$$

Initialize $D^{(1)} = \left(\frac{1}{m}, \dots, \frac{1}{m}\right)$

For t=1, …, T:

$$h_t = \mathcal{A}\left(S, D^{(t)}\right)$$

$$\epsilon_t = \sum_i D_i^{(t)} \cdot \mathbf{1}_{h_t(x_i)\neq y_i}$$

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

$$D_i^{(t+1)} = \frac{D_i^{(t)} \exp(-\alpha_t y_i h_t(x_i))}{Z_t}$$

$$h_s(x) = \mathrm{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

AdaBoost optimizes the exponential loss

- Goal: Show AdaBoost optimizes $\mathcal{L}_{\exp}(h_s) = \frac{1}{m}\sum_{i=1}^{m} \exp\{-y_i \sum_{k=1}^{t} \alpha_k h_k(x_i)\}$.
- Assume: $h_1, \alpha_1, \dots, h_{t-1}, \alpha_{t-1}$ were already chosen.
- Question: How to choose h_t, α_t to minimize $\mathcal{L}_{\exp}(h_s)$?
- Lemma: $\mathcal{L}_{exp}^{(t)} \propto e^{-\alpha_t} + (e^{\alpha_t} - e^{-\alpha_t}) \underbrace{\sum_i D_i^{(t)} \mathbf{1}_{h_t(x_i) \neq y_i}}_{\triangleq \epsilon_t}$
- Proof:
  2. Extra: Show that $D_i^{(t)} = c \cdot \exp\left(-y_i \sum_{k=1}^{t-1} \alpha_k h_k(x_i)\right)$, for $c > 0$
  3. Show that $\mathcal{L}_{exp}^{(t)} \propto \sum_i D_i^{(t)} \exp\{-y_i \alpha_t h_t(x_i)\}$
  4. Extra: Prove the following form of the exp. loss after t rounds is

$$\mathcal{L}_{exp}^{(t)} \propto e^{-\alpha_t} \underbrace{\sum_i D_i^{(t)}}_{=?} + (e^{\alpha_t} - e^{-\alpha_t}) \sum_i D_i^{(t)} \mathbf{1}_{h_t(x_i) \neq y_i}$$

Initialize $D^{(1)} = \left(\frac{1}{m}, \dots, \frac{1}{m}\right)$

For t=1, …, T:

$$h_t = \mathcal{A}\left(S, D^{(t)}\right)$$

$$\epsilon_t = \sum_i D_i^{(t)} \cdot \mathbf{1}_{h_t(x_i) \neq y_i}$$

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

$$D_i^{(t+1)} = \frac{D_i^{(t)} \exp\left(-\alpha_t y_i h_t(x_i)\right)}{Z_t}$$

$$h_s(x) = \mathrm{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

AdaBoost optimizes the exponential loss

- Goal: Show AdaBoost optimizes $\mathcal{L}_{\exp}(h_s) = \frac{1}{m}\sum_{i=1}^{m} \exp\{-y_i \sum_{k=1}^{t} \alpha_k h_k(x_i)\}$.
- Assume: $h_1, \alpha_1, \dots, h_{t-1}, \alpha_{t-1}$ were already chosen.
- Question: How to choose h_t, α_t to minimize $\mathcal{L}_{\exp}(h_s)$?
- Lemma: $\mathcal{L}_{exp}^{(t)} \propto e^{-\alpha_t} + (e^{\alpha_t} - e^{-\alpha_t}) \underbrace{\sum_i D_i^{(t)} \mathbf{1}_{h_t(x_i) \neq y_i}}_{\triangleq \epsilon_t}$

  (proven)
- Answer: Use "two steps" greedy optimization:
  - Choose h_t minimizing the weighted error ϵ_t, *e.g.*, with ERM *w.r.t.* $D^{(t)}$.
  - Choose α_t minimizing $\mathcal{L}_{exp}^{(t)}$ given $h_1, \alpha_1, \dots, h_{t-1}, \alpha_{t-1}$ and h_t.

Initialize $D^{(1)} = \left(\frac{1}{m}, \dots, \frac{1}{m}\right)$

For t=1, …, T:

$$h_t = \mathcal{A}\left(S, D^{(t)}\right)$$

$$\epsilon_t = \sum_i D_i^{(t)} \cdot \mathbf{1}_{h_t(x_i) \neq y_i}$$

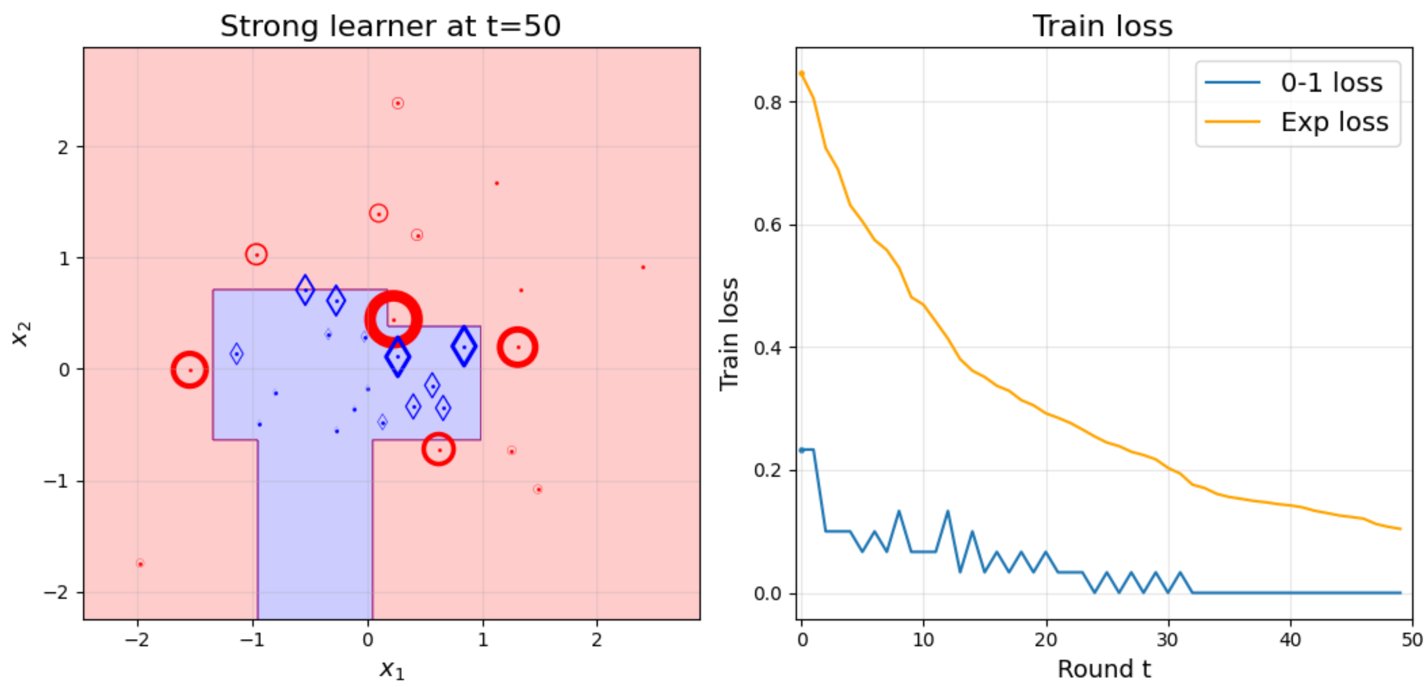
$$\alpha_t = \frac{1}{2}\log\left(\frac{1}{\epsilon_t} - 1\right)$$

$$D_i^{(t+1)} = \frac{D_i^{(t)} \exp\left(-\alpha_t y_i h_t(x_i)\right)}{Z_t}$$

$$h_s(x) = \operatorname{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

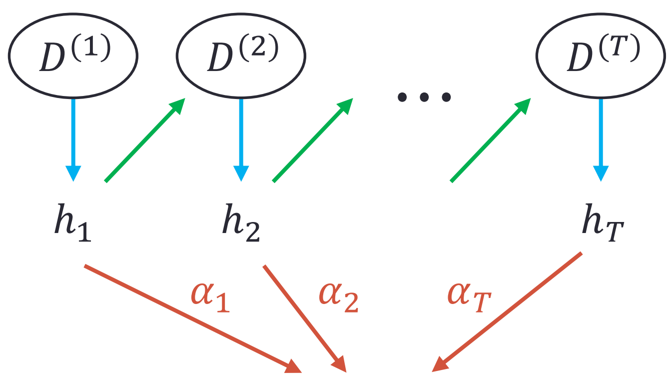
AdaBoost optimizes the exponential loss

- The simulation from earlier can also visualize the decaying exponential loss.
- After 0 classification training loss, the exponential loss keeps decreasing



Summary

- In boosting we create a strong classifier by combining multiple weak classifiers
- Each weak classifier is trained on an updated distribution where misclassified data points get larger probability mass



$$h_s(x) = \mathrm{sign}\left(\sum_{t=1}^{T} \alpha_t h_t(x)\right)$$

- AdaBoost is a boosting algorithm that greedily optimizes the exponential loss