Boosting

Lecture 9

Last Time

- A held-out ***validation set*** is a critical tool for model selection
- It helps assess where on the bias-complexity tradeoff a hypothesis is
- ***Regularizers*** like L2 regularization give us a knob λ to adjust bias-complexity tradeoff for a fixed hypothesis class
- Textbook: chapters 11.0, 11.2, 11.3, 13.0, 13.1, 13.4

This Class

- A new hypothesis class: “boost” a fixed class into a more complex one
- Textbook: chapter 10

Motivation

Increasing Hypothesis Class Complexity

- Regularization allows us to restrict the complexity of a hypothesis class

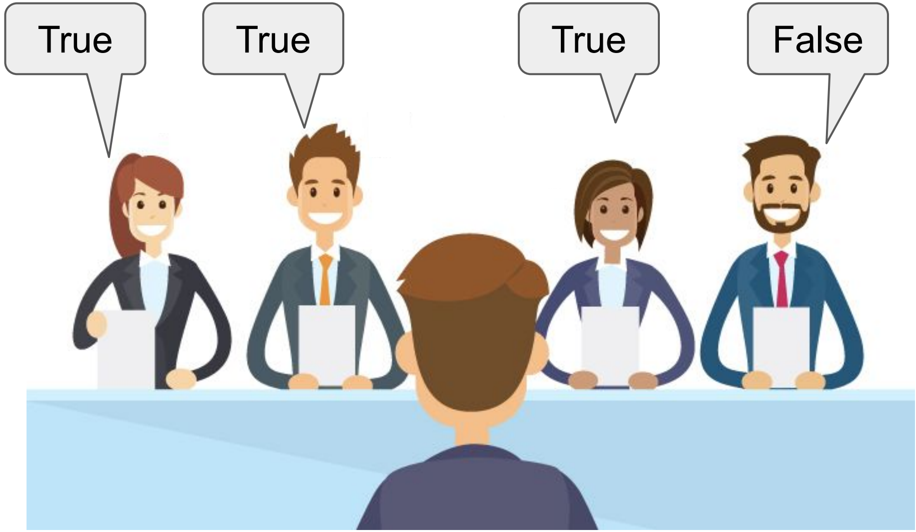
- What if we set $\lambda = 0$ and we still have high approximation error?

- How can we further *increase* the complexity of a hypothesis class?

Main Idea

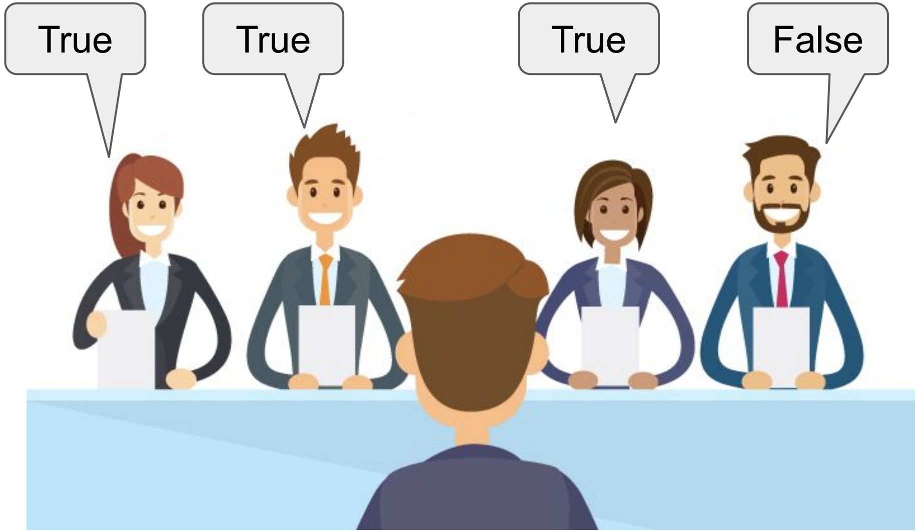
- One option is to get a set of hypotheses and have them vote

- A set of hypotheses, even from different classes, is called an "ensemble"



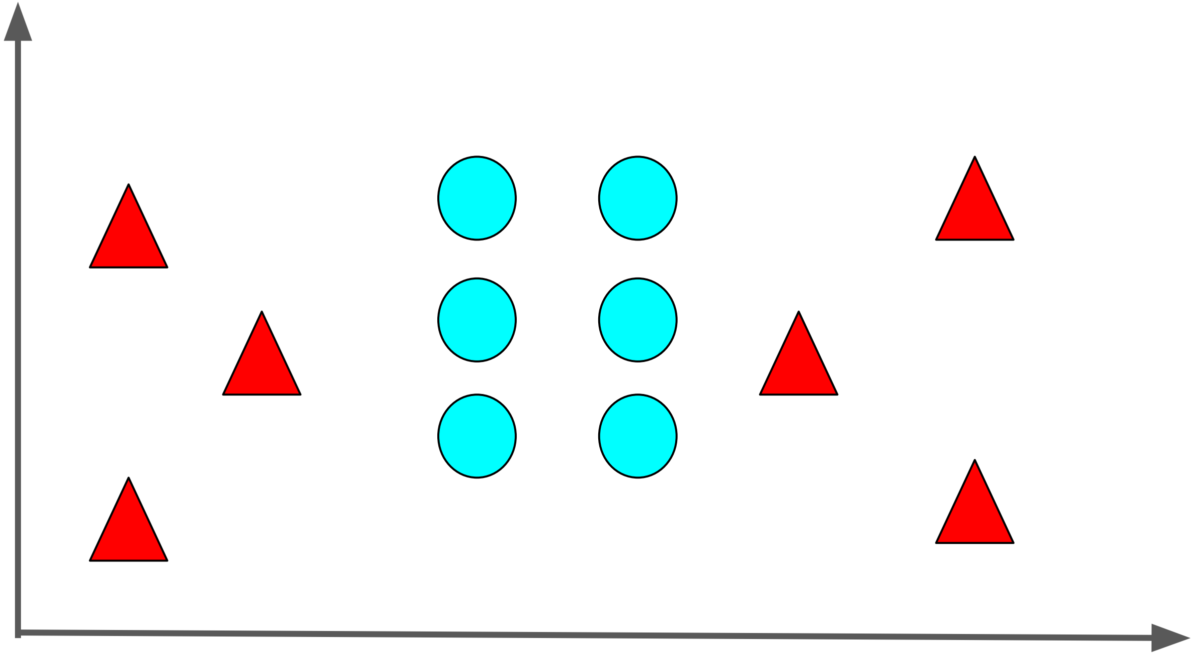
Why Might We Do This?

- Spread out the “expertise”
- Different hypotheses in the ensemble might be able to focus on different aspects of the problem
- Non-Realizability - no one hypothesis can solve the whole problem



Example

Halfspaces in 2-d

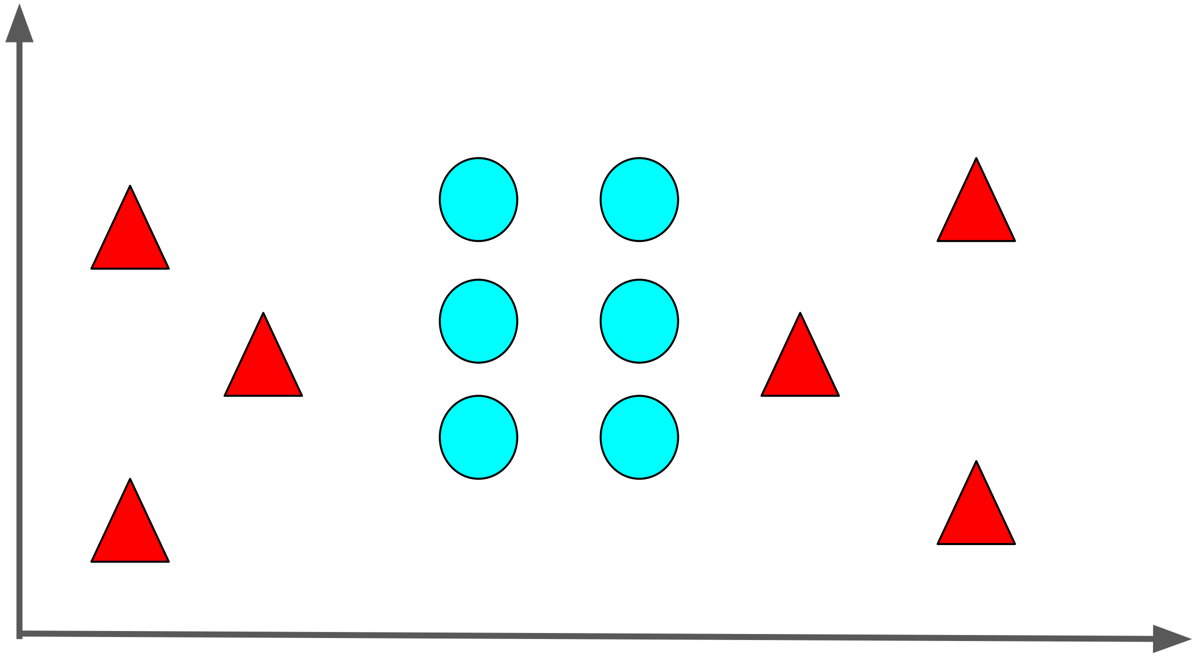


$$\mathcal{X} = \mathbb{R}^2$$

$$\mathcal{Y} = \{1\ (\bigcirc), 0\ (\triangle)\}$$

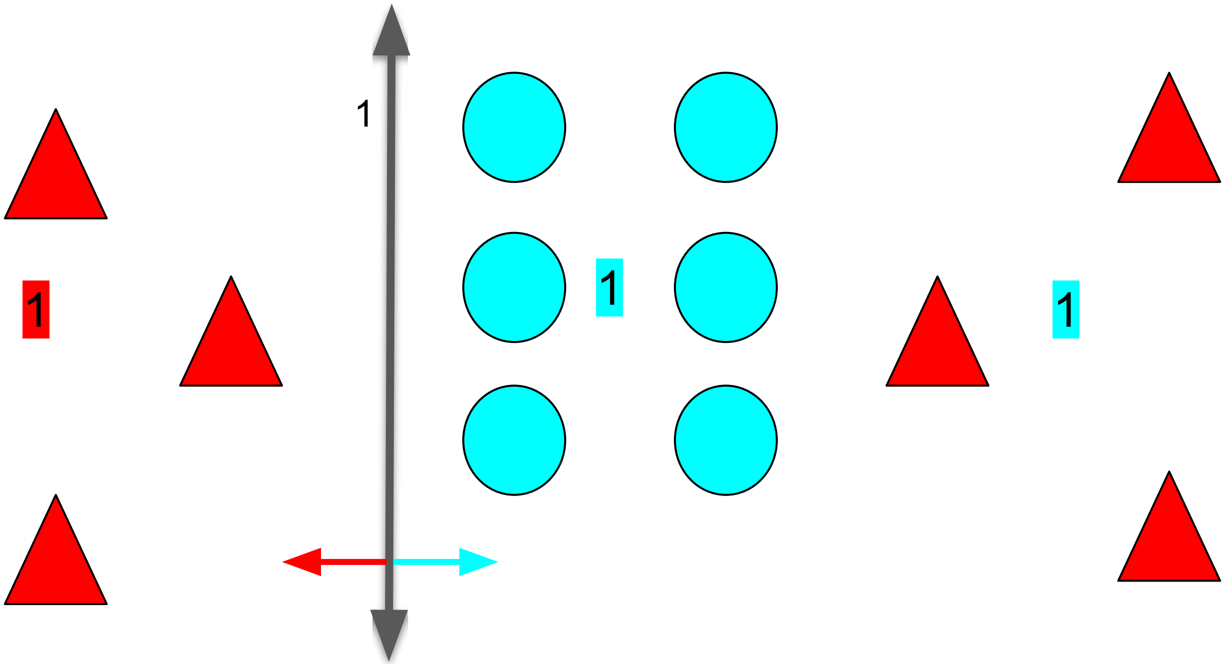
Not realizable!

Boosted Halfspaces in 2-d

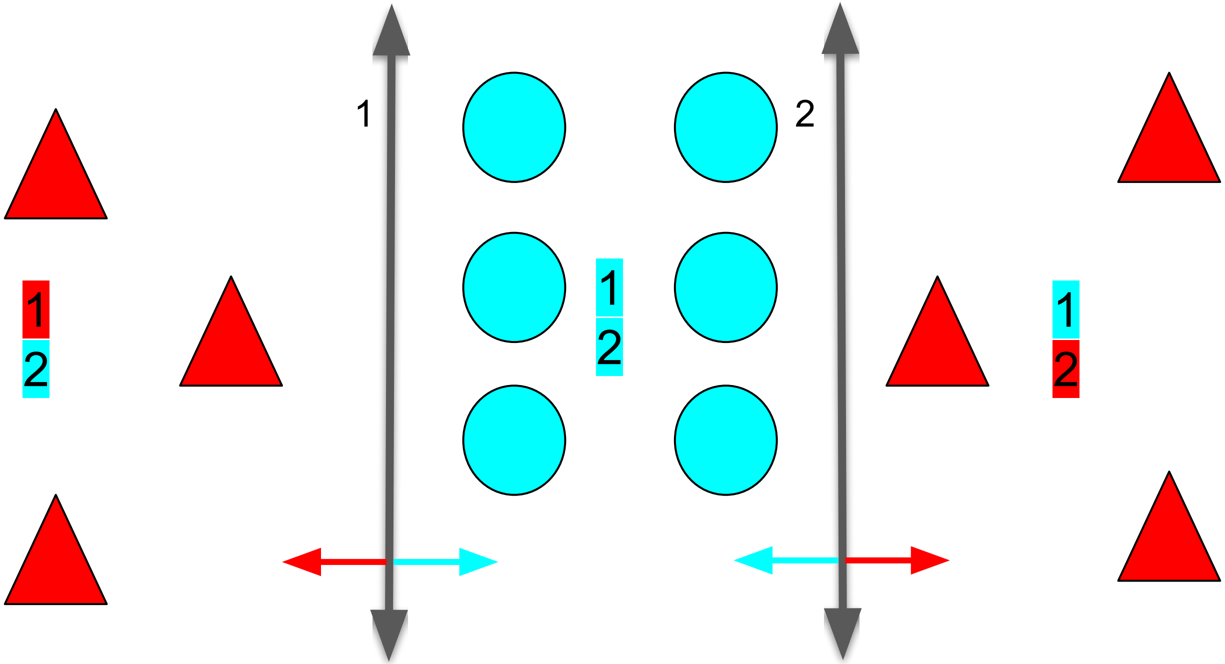


What if we took the majority vote of an ensemble of halfspaces?

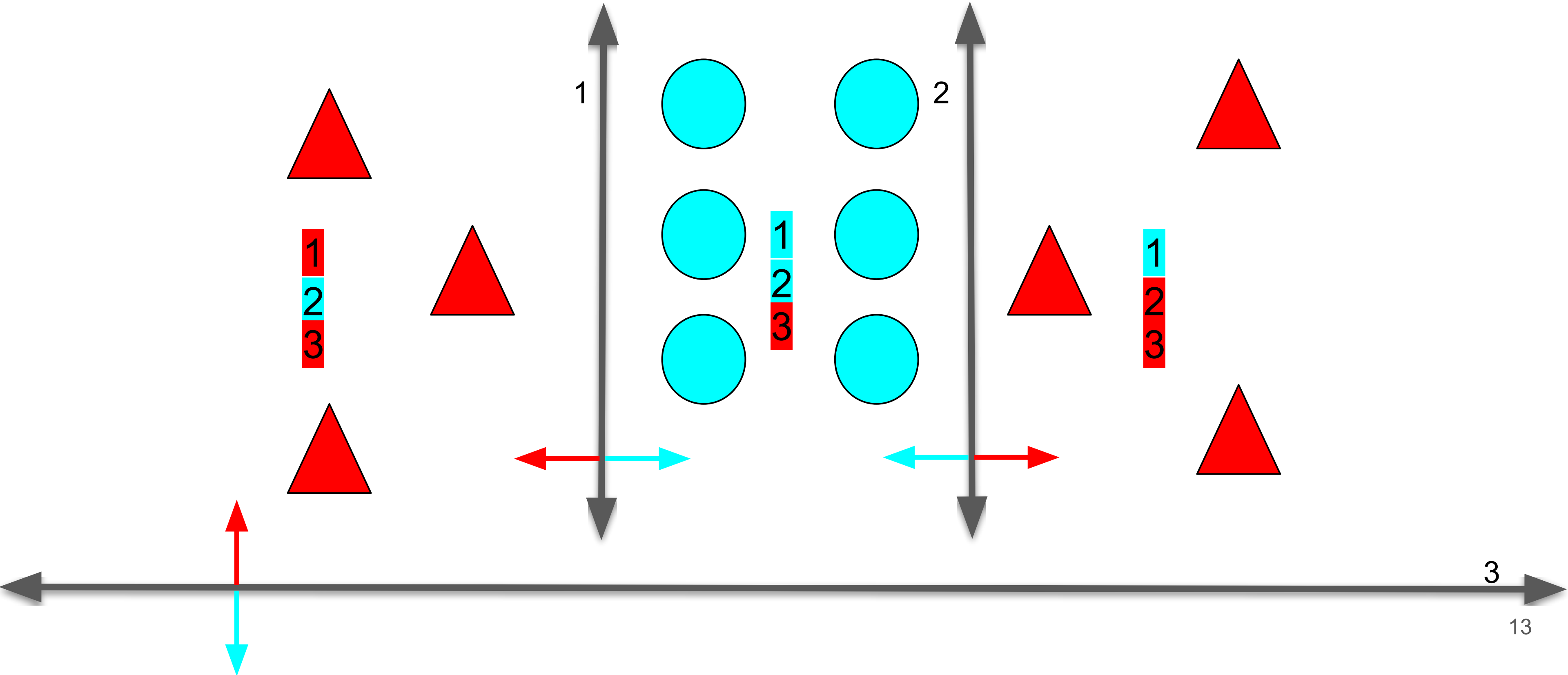
Separation with Boosted Halfspaces



Separation with Boosted Halfspaces



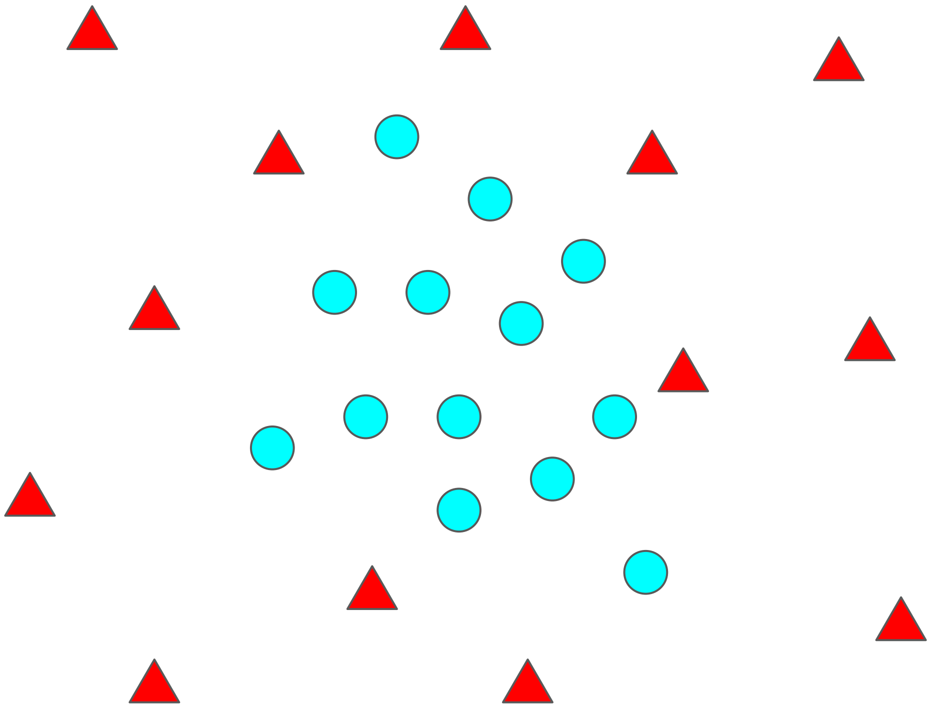
Separation with Boosted Halfspaces



Question

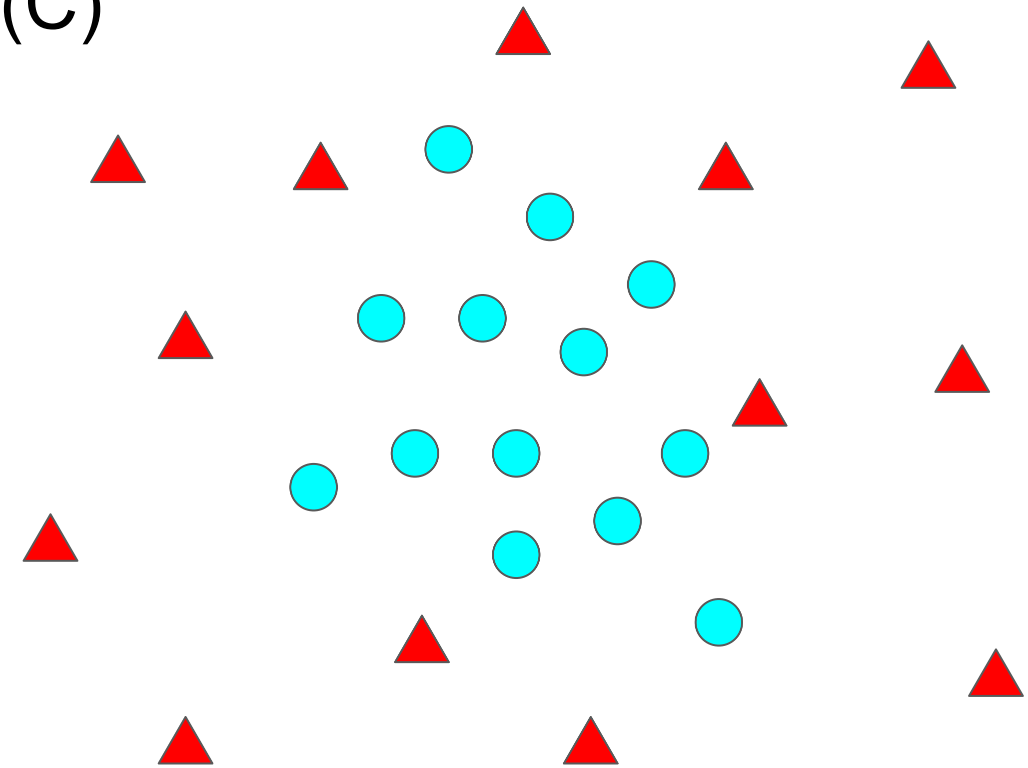
Minimum Number of Halfspaces for Separation?

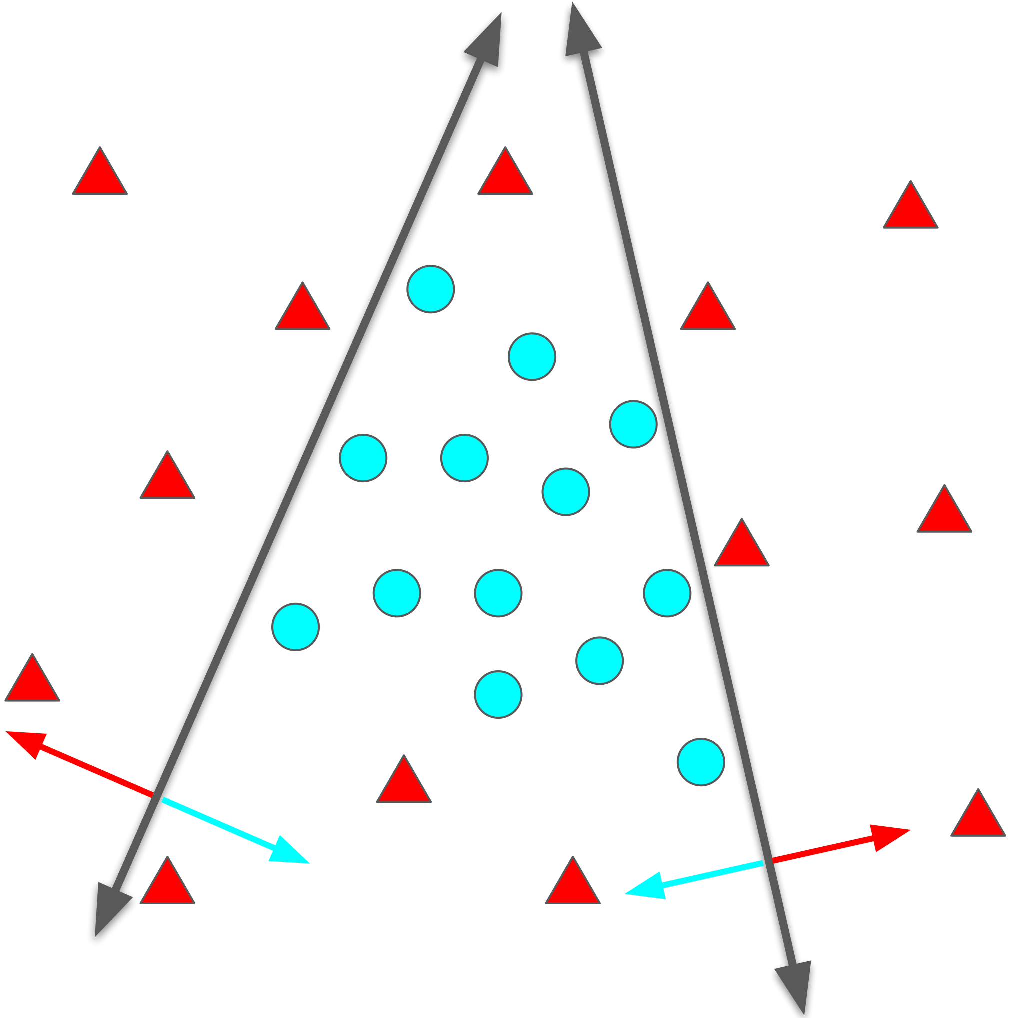
A) 4
B) 6
C) 7
D) 8
E) Impossible

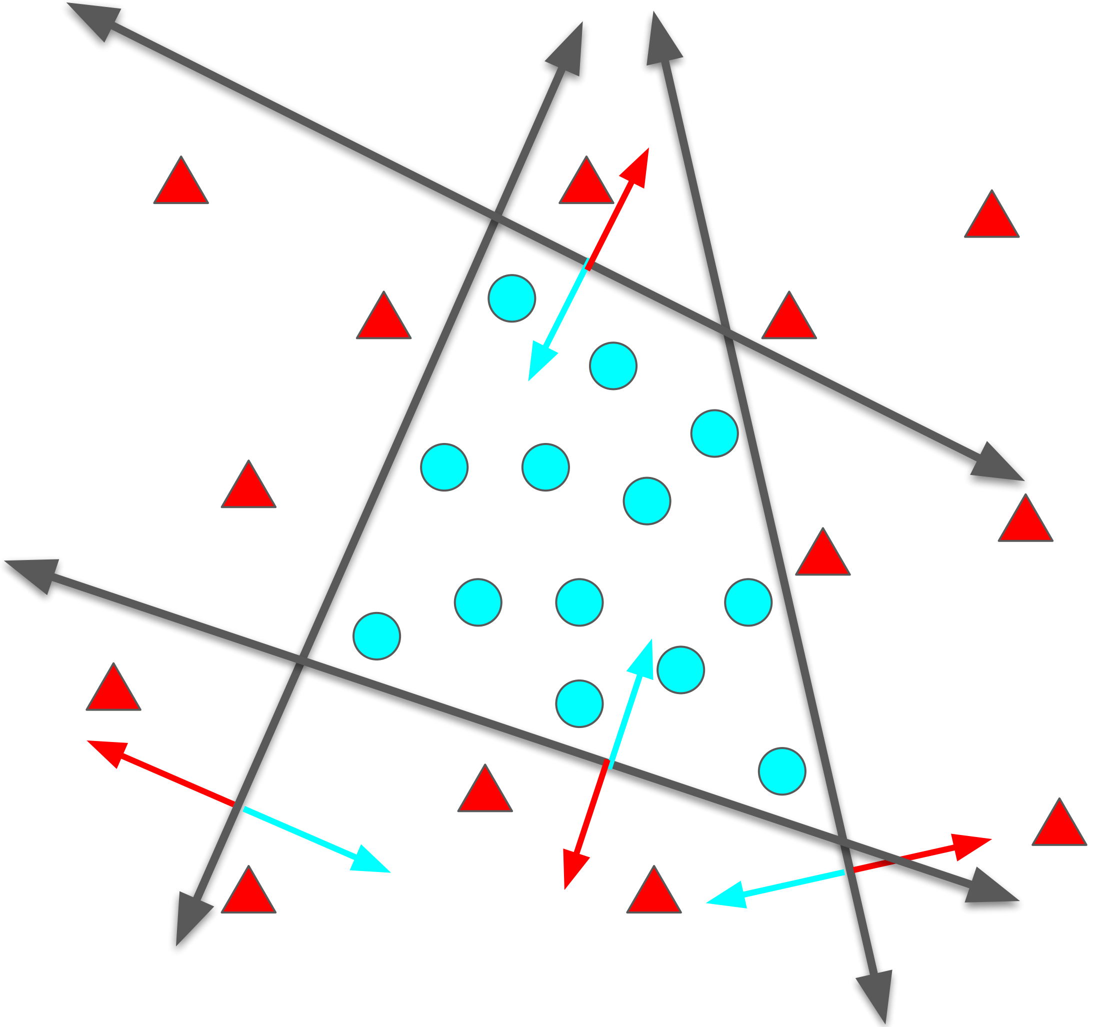


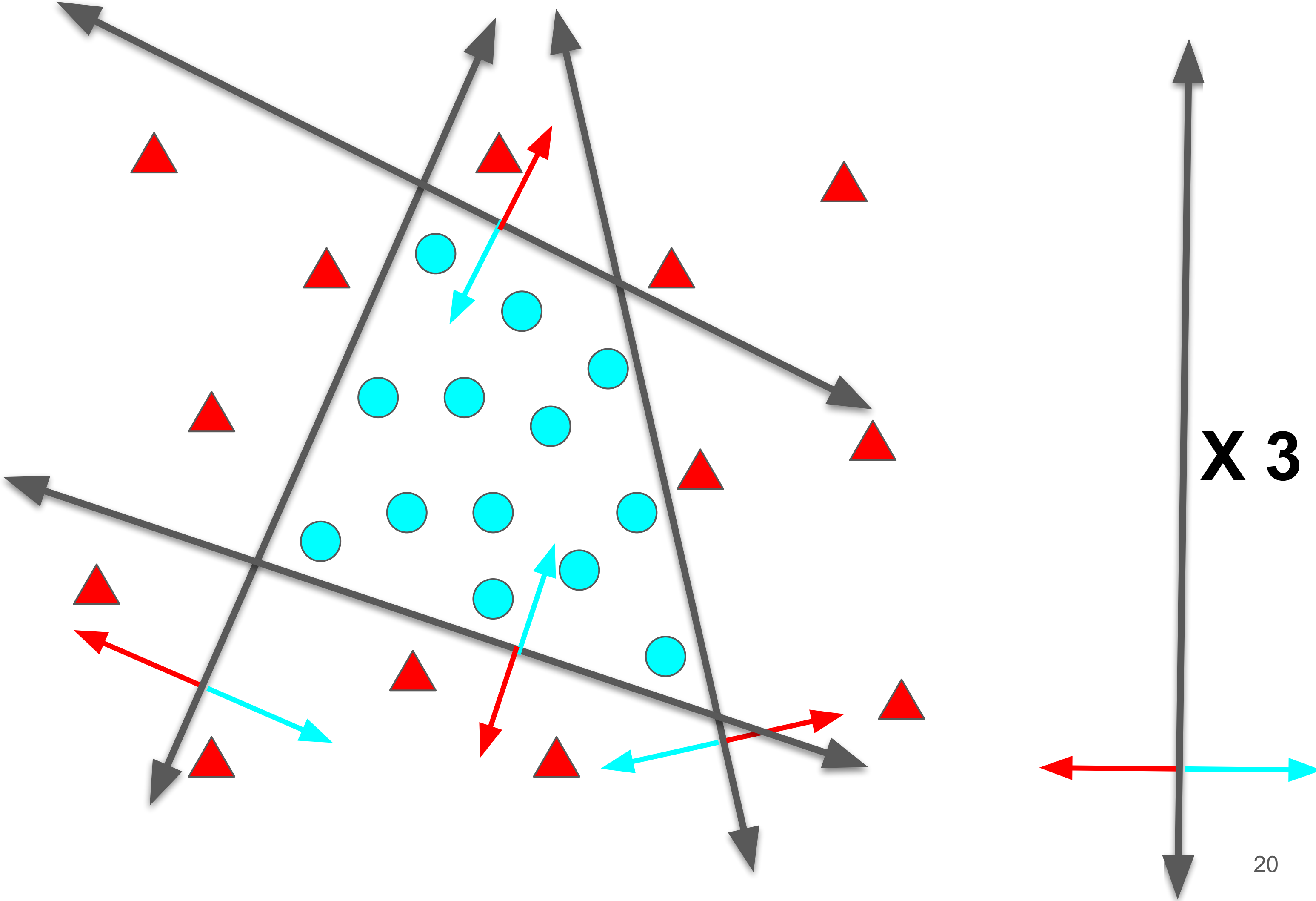
Answer

Answer: 7 (C)








X 3

Boosting

AdaBoost (Adaptive Boosting)

- Freund, Yoav, and Robert E. Schapire. "A decision-theoretic generalization of on-line learning and an application to boosting." Journal of computer and system sciences 55.1 (1997): 119-139.

- 15,000+ citations

- Won the 2003 Gödel Prize (for theoretical computer science)

- One of the best “out of the box” classifiers

ML algorithm = representation
+ loss function + optimizer

Representation

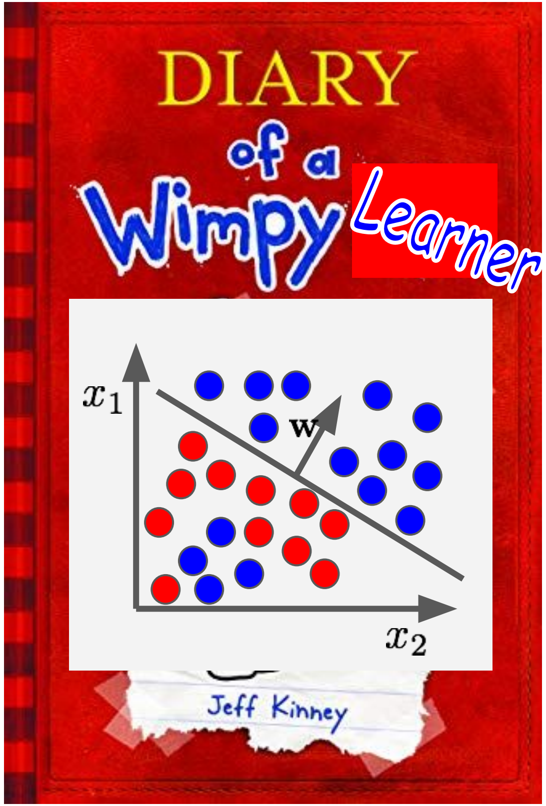
- Let $\mathcal{H}$ be the class of base, i.e., not-boosted hypotheses (book calls this B)
- Let $E(\mathcal{H}, T)$ be the class of ensemble hypotheses built using T elements of $\mathcal{H}$ (book calls this L(B, T))

$$E(\mathcal{H}, T) = \left\{ \mathbf{x} \mapsto \operatorname{sign}\left(\sum_{t=1}^{T} w_t h_t(\mathbf{x}) \right) : \mathbf{w} \in \mathbb{R}^T, \quad \forall t \;\; h_t \in \mathcal{H} \right\}$$

- Rest of representation ($\mathcal{X}, \mathcal{Y}$, etc.) same as for $\mathcal{H}$

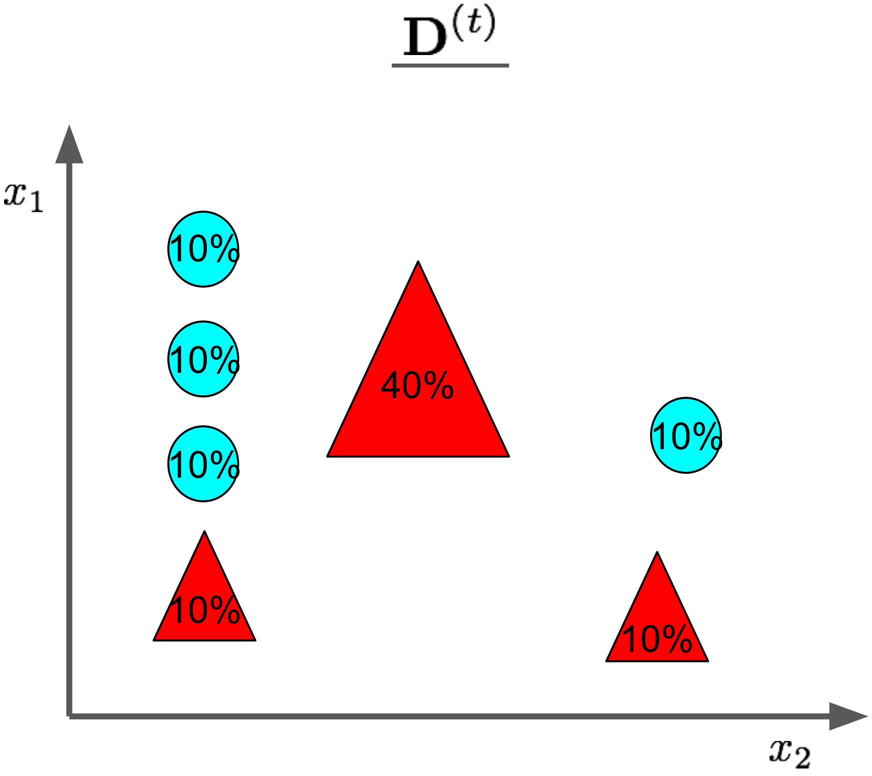
Representation: Weak Learners

- Our base hypotheses don't have to be PAC learnable!
- We'll see that boosting works as long as they're γ-*weakly learnable*
- γ-weakly learnable is just like PAC learnable, except instead of arbitrarily small ϵ we only require reaching error $\frac{1}{2} - \gamma$ for fixed $0 < \gamma < \frac{1}{2}$



Representation: Distributions over Examples

- To get different base learners to learn different things, we'll weight the m examples in S
- Let $\mathbf{D}^{(t)}$ be a distribution over the m examples in S for the t-th base learner



Loss Function

- Overall loss function is just 0-1 loss:

$$L_S(h_s) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{[h_s(\mathbf{x}_i) \neq y_i]}$$

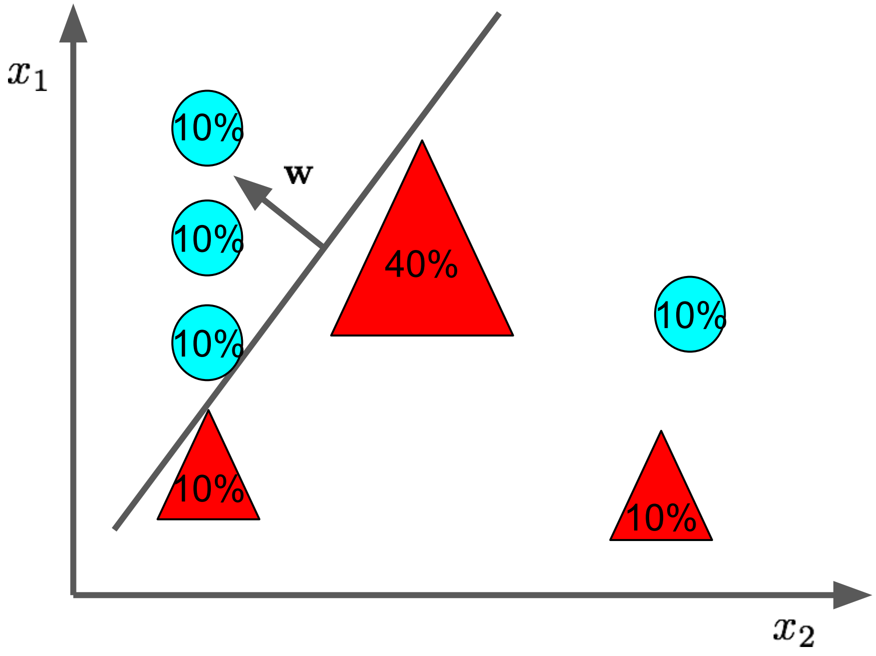
- But we redefine the loss on training examples for the t-th base hypothesis:

$$\epsilon_t \stackrel{\text{def}}{=} L_{\mathbf{D}^{(t)}}(h_t) \stackrel{\text{def}}{=} \sum_{i=1}^{m} D_i^{(t)} \mathbb{1}_{[h_t(\mathbf{x}_i) \neq y_i]} \quad \text{where} \quad \mathbf{D}^{(t)} \in \mathbb{R}^m$$

Optimizer: Base Hypotheses

- We assume that our base hypotheses can be selected with a weak learner

- In practice, we just run ERM with the training example weights $\mathbf{D}^{(t)}$



Optimizer: Ensemble

AdaBoost

input:

training set $S = (\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_m, y_m)$

weak learner WL

number of rounds T

initialize $\mathbf{D}^{(1)} = (\frac{1}{m}, \ldots, \frac{1}{m})$.

for $t = 1, \ldots, T$:

invoke weak learner $h_t = \mathrm{WL}(\mathbf{D}^{(t)}, S)$

compute $\epsilon_t = \sum_{i=1}^{m} D_i^{(t)} \mathbb{1}_{[y_i \neq h_t(\mathbf{x}_i)]}$

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

update $D_i^{(t+1)} = \dfrac{D_i^{(t)} \exp(-w_t y_i h_t(\mathbf{x}_i))}{\sum_{j=1}^{m} D_j^{(t)} \exp(-w_t y_j h_t(\mathbf{x}_j))}$ for all $i = 1, \ldots, m$

output the hypothesis $h_s(\mathbf{x}) = \mathrm{sign}\left(\sum_{t=1}^{T} w_t h_t(\mathbf{x})\right)$.

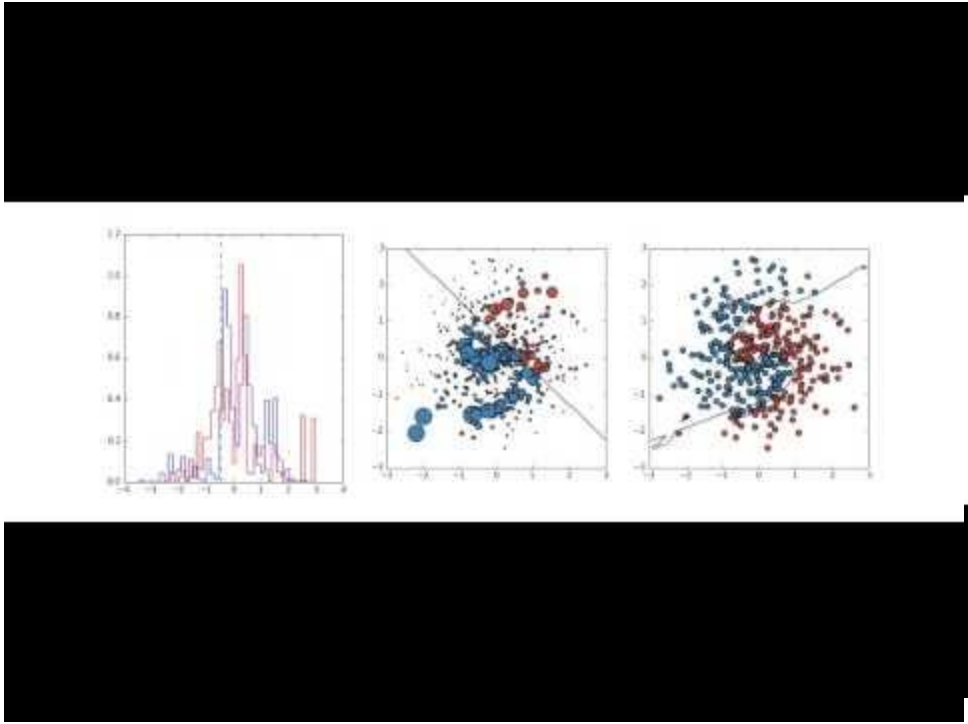
At first, example weights are all same

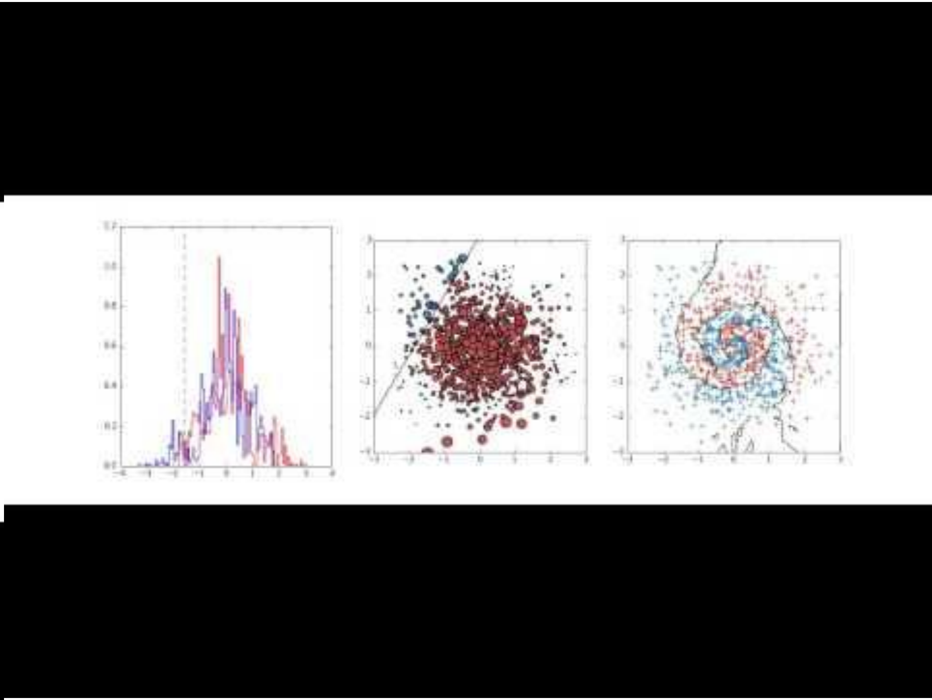
At each round, select a base hypothesis using example weights

Weight influence in the ensemble via its error

Update example weights via the ensemble's mistakes

Some Demonstrations:





Why does it work? Intuition

1. Better hypotheses get more weight.
2. Hard training examples get more attention.
3. If at first you don’t succeed…

   … use a weak PAC learner until you do!

   (Assume distribution-independent bounded error is a powerful assumption.)

$$\epsilon_{t+1} \leq \frac{1}{2} - \gamma$$

Why does it work? Proof Sketch

1. Place an upper bound (called Z) on the ensemble's loss.

2. Show that at each time step, Z shrinks by a multiplicative factor.

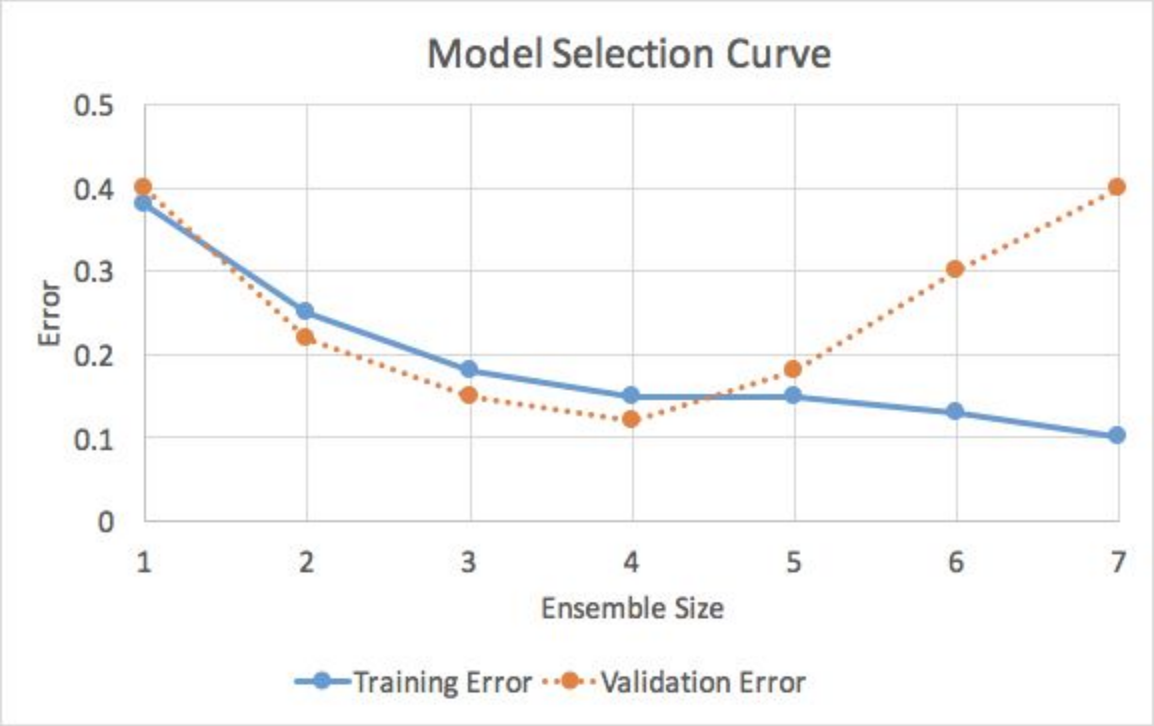
3. Therefore, the empirical training risk shrinks exponentially with T

THEOREM 10.2 *Let S be a training set and assume that at each iteration of AdaBoost, the weak learner returns a hypothesis for which $\epsilon_t \leq 1/2 - \gamma$. Then, the training error of the output hypothesis of AdaBoost is at most*

$$L_S(h_s) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{[h_s(\mathbf{x}_i) \neq y_i]} \leq \exp(-2\gamma^2 T) .$$

Understanding Machine Learning. Shalev-Shwartz and Ben-David, 2014.

The Bias-Complexity Tradeoff Strikes Again!



Ensembles in ML Today

Gradient Boosting

- In AdaBoost, we want to correct the mistakes of the ensemble, so we learn a new hypothesis focused on the misclassified examples

- However, a more direct solution would be to learn to predict what needs to be added to the ensemble to fix those mistakes, i.e., the residuals

- So the training examples at iteration t are $y_i - h_S^{t-1}(\mathbf{x}_i)$

- **Effect:** added ensemble members minimize the loss by following its gradient

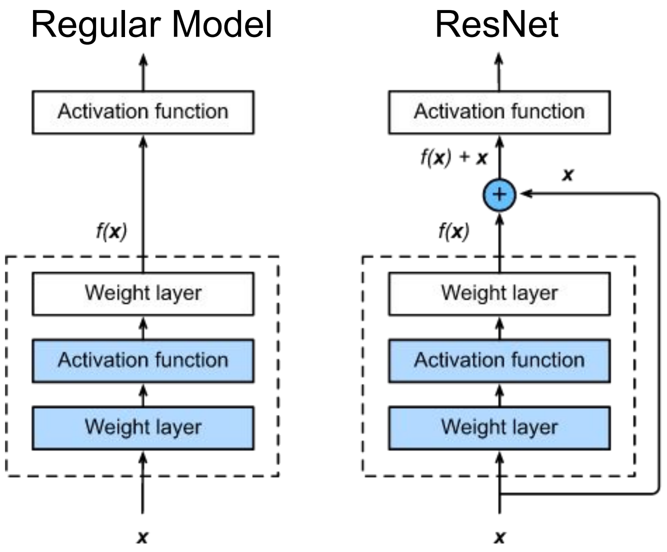
ResNets

Ensemble-like method in deep learning

- ~~7,500+~~ 40,000+ citations since 2016

- Use a big ordered, ensemble (~1000 members) where intermediate members are trying to predict the residual on the previous ones

- Similar in spirit to gradient boosting



https://d2l.ai/chapter_convolutional-modern/resnet.html

He, Kaiming, et al. "Deep residual learning for image recognition." Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2016.

Review

- ***Boosting*** is an algorithmic framework for extending a “base” hypothesis class into a more complex one

- ***AdaBoost*** (adaptive boosting) learns an ensemble of base hypotheses that vote to make predictions. Its complexity is only limited by the ensemble size.

- Textbook: chapter 10

Next Class

- Another useful class of hypotheses: decision trees
- Textbook: chapter 18