11 April 2023

CIS 5200: MACHINE LEARNING

ONLINE LEARNING

Surbhi Goel

Content here draws from material by Nike Haghtalab (UC Berkeley)

Spring 2023

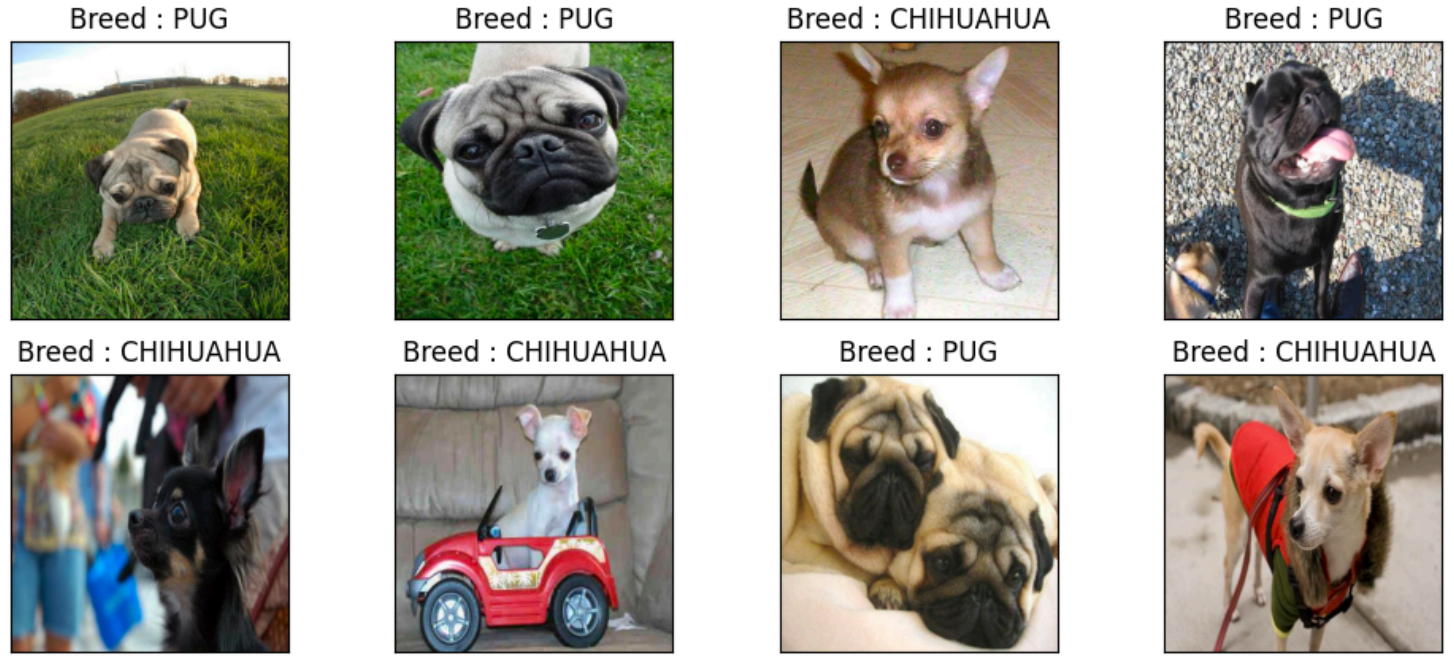
OUTLINE - TODAY

- Online Learning
  - Setup
- Mistake Bound
  - Having Algorithm
- Regret
  - Weighted Majority Algorithm

SUPERVISED LEARNING - RECAP

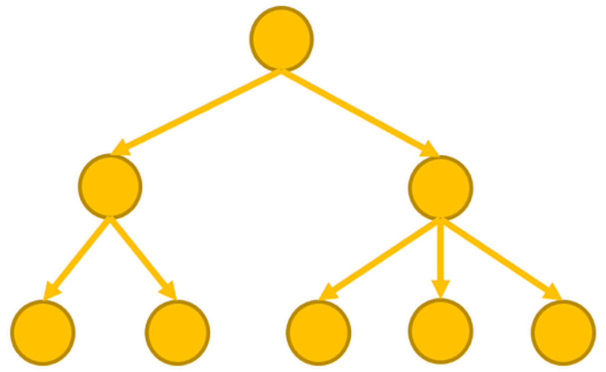
Training dataset

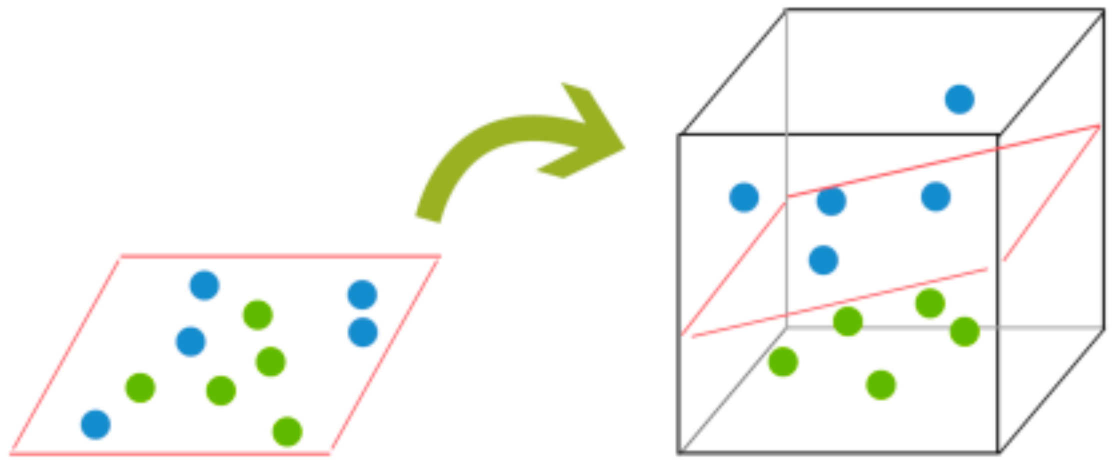
$\mathcal{S} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_m, y_m)\}$



Machine Learning Method

Prediction function $\hat{f}$



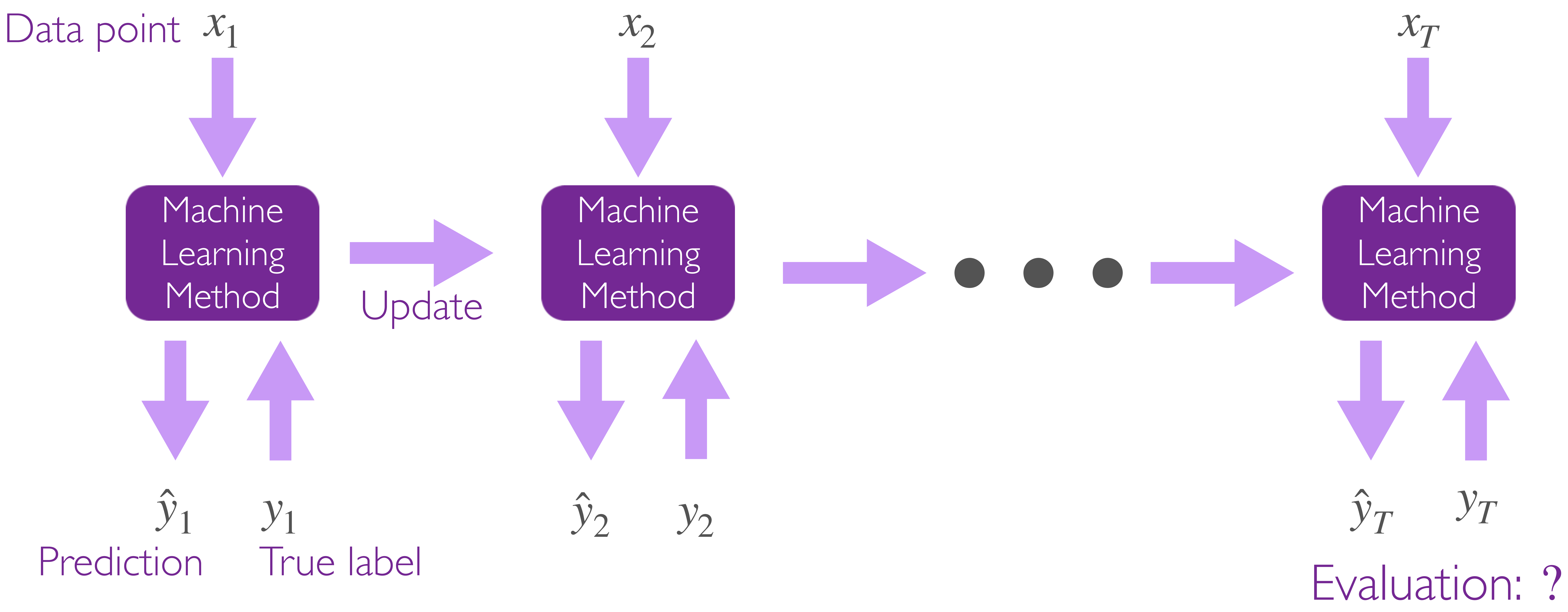


Evaluation: $R(\hat{f}) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\ell(\hat{f}(x), y)\right]$

- Uses the entire dataset to make prediction on new test example
- Assumption: data is drawn i.i.d. from some unknown distribution $\mathcal{D}$

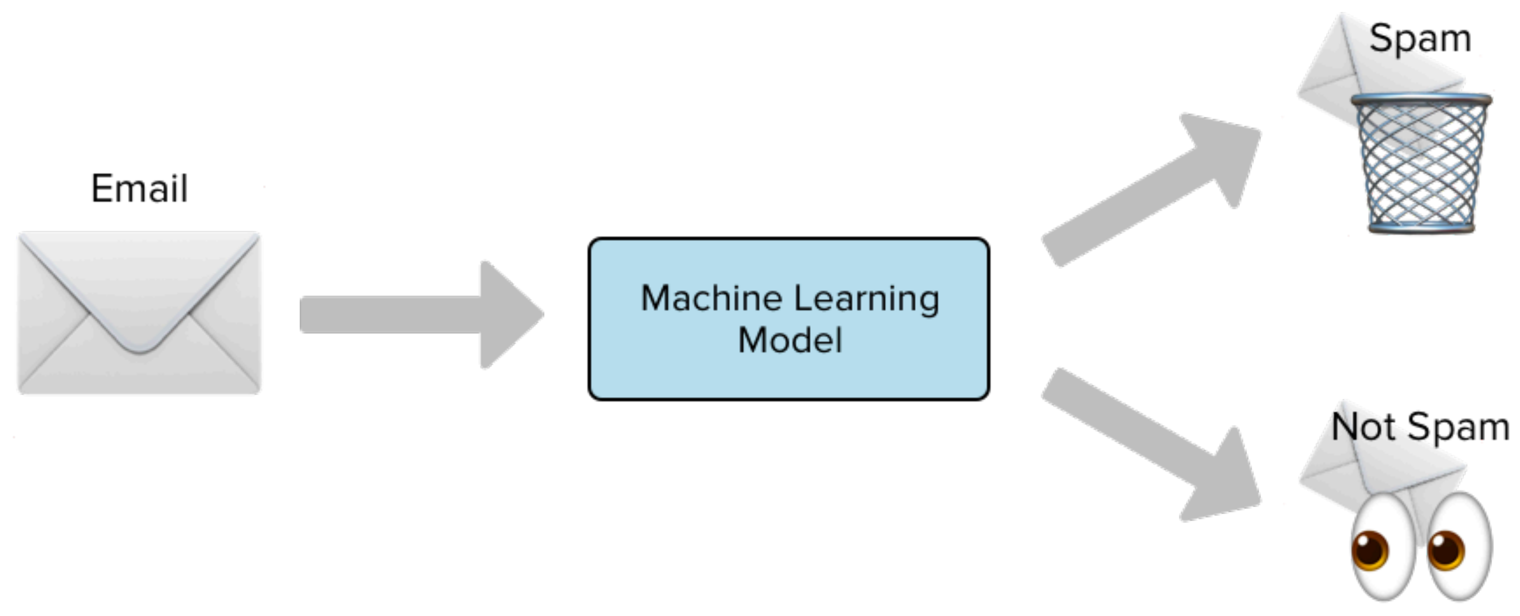
Relates training data to test data

ONLINE LEARNING - ONE AT A TIME



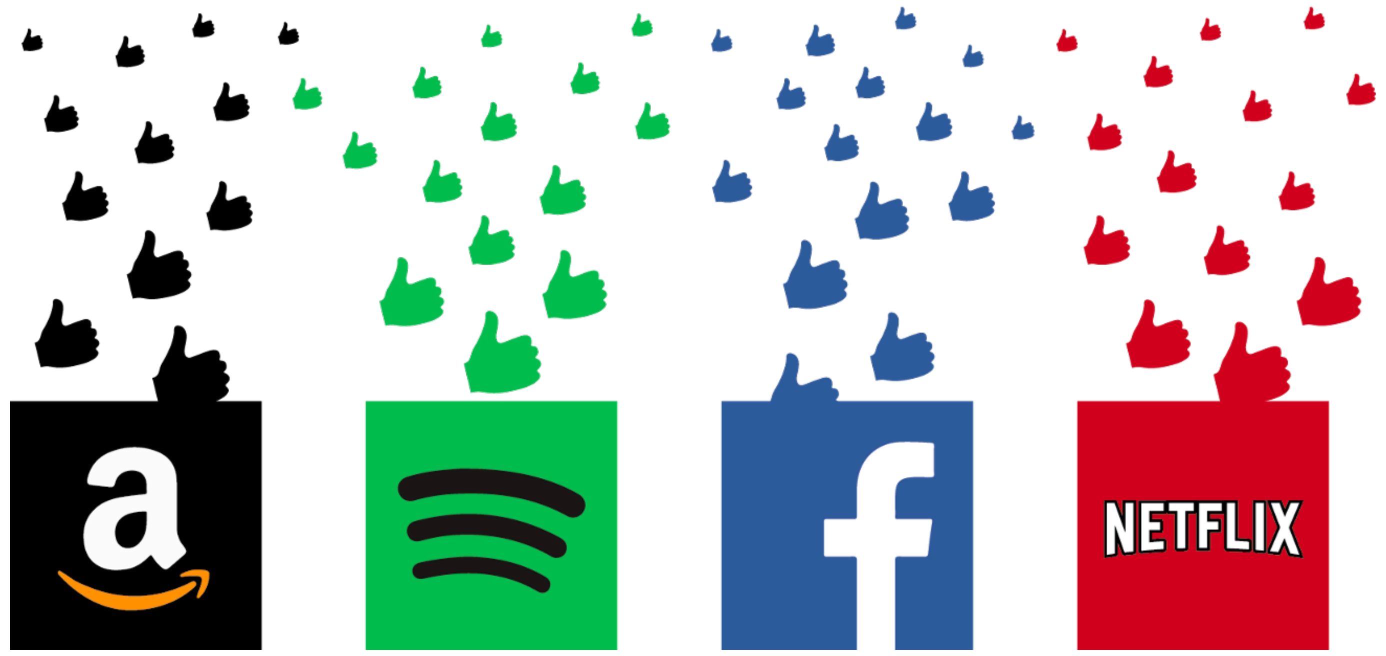
- Receive one data point at a time, predict, receive label and update model
- Sequence can be deterministic, stochastic, or adaptively adversarial

ONLINE LEARNING - EXAMPLES



Spam classification

Investment



Recommender Systems

- Need to make real-time decisions and update the model
- Handle changing distributions

ONLINE LEARNING - SETUP

We will focus on binary classification

- Learner is given an instance $x_t \in \mathcal{X}$ either from the environment or an adversary
- Learner makes a prediction $\hat{y}_t \in \{-1,1\}$
- Learner observes actual label $y_t \in \{-1,1\}$
- Learner suffers a loss $\ell(\hat{y}_t, y_t)$

  0-1 loss

Goal: Minimize the total loss that the learner suffers.

Not clear if this is possible, learner needs to be able to deduce something about the future from the past!

MISTAKE BOUND - REALIZABILITY

Assumption: Data satisfies $y_t = f(x_t)$ for some $f \in \mathcal{F}$ (no noise)

- We can hope that the learner can learn this f eventually
- Can count the total number of mistakes any learner makes in the ***worst-case***

$$M_{\mathcal{L}}(\mathcal{F}) := \max_{\substack{f \in \mathcal{F}, T \\ x_1, \ldots, x_T \in \mathcal{X}}} \sum_{t=1}^{T} 1[f(x_t) \neq \hat{y}_t]$$

Mistake Bound

Function class $\mathcal{F}$ if **online learnable** with mistake bound B if $M_{\mathcal{L}}(\mathcal{F}) \leq B < \infty$

CONSISTENT LEARNER

Forget about computational efficiency for now

Algorithm 2: Consistent

Initialize $\mathcal{V}_1 = \mathcal{F}$

for $t = 1, 2, \ldots$ **do**

  Receive x_t

  Choose any $f_t \in \mathcal{V}_t$

  Predict $\hat{y}_t = f_t(x_t)$

  Receive true label y_t

  Update $\mathcal{V}_{t+1} = \mathcal{V}_t \setminus \{f_t\}$

end

Each function is an expert, remove the expert that makes an error

CONSISTENT LEARNER

Theorem:

Let $\mathcal{F}$ be a finite hypothesis class. The Consistent algorithm enjoys the mistake bound

$$M_{\text{Consistent}}(\mathcal{F}) \leq |\mathcal{F}| - 1.$$

Each mistake, we remove one hypothesis!

In PAC learning, any ERM was good enough for our guarantee. Not in OL!

HALVING

Algorithm 1: Halving

Initialize $\mathcal{V}_1 = \mathcal{F}$

for $t = 1, 2, \ldots$ **do**

Receive x_t

Predict $\hat{y}_t = \arg\max_{y \in \{-1,1\}} |\{f \in \mathcal{V}_t : f(x_t) = y\}|$ If tie, predict 1

Receive true label y_t

Update $\mathcal{V}_{t+1} = \{f \in \mathcal{V}_t : f(x_t) = y_t\}$ Version space of all functions that are consistent with the inputs so far

end

Predicting based on majority vote among experts (each classifier is an expert)

HALVING

Theorem:

Let $\mathscr{F}$ be a finite hypothesis class. The Halving algorithm enjoys the mistake bound

$$M_{\text{Halving}}(\mathscr{F}) \leq \log(|\mathscr{F}|).$$

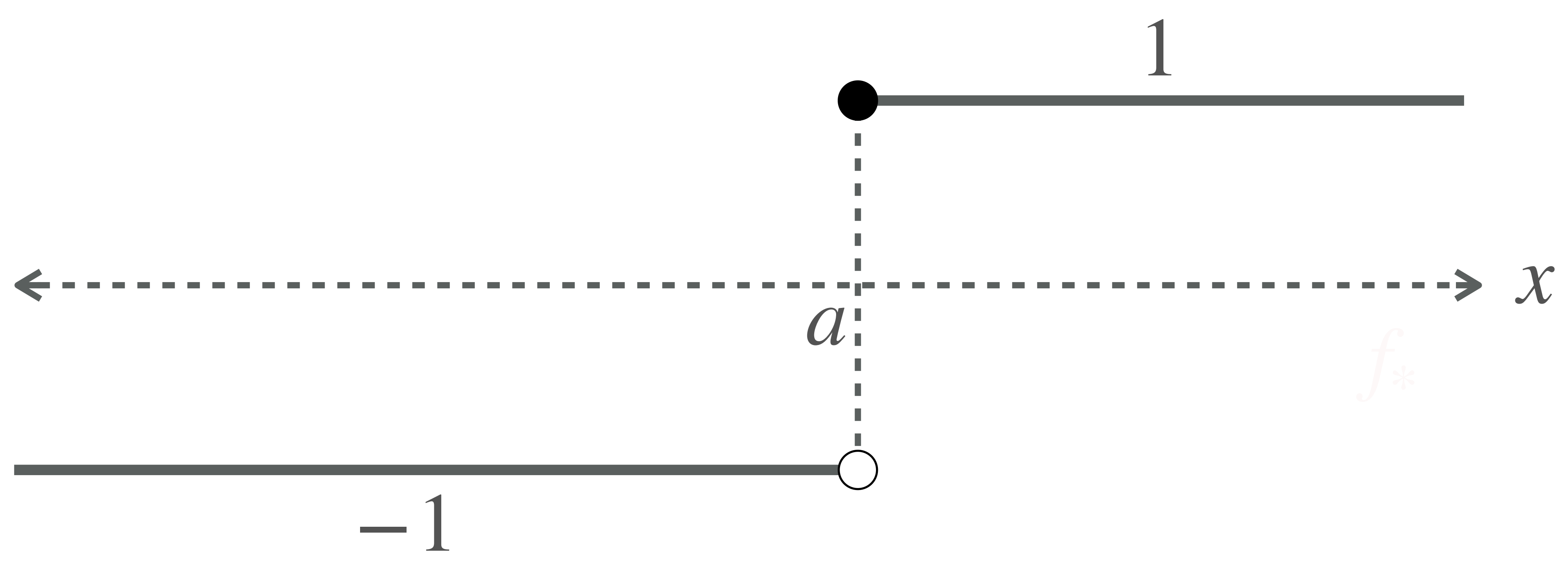
Halving comes from the fact that the version class is halved at each mistake

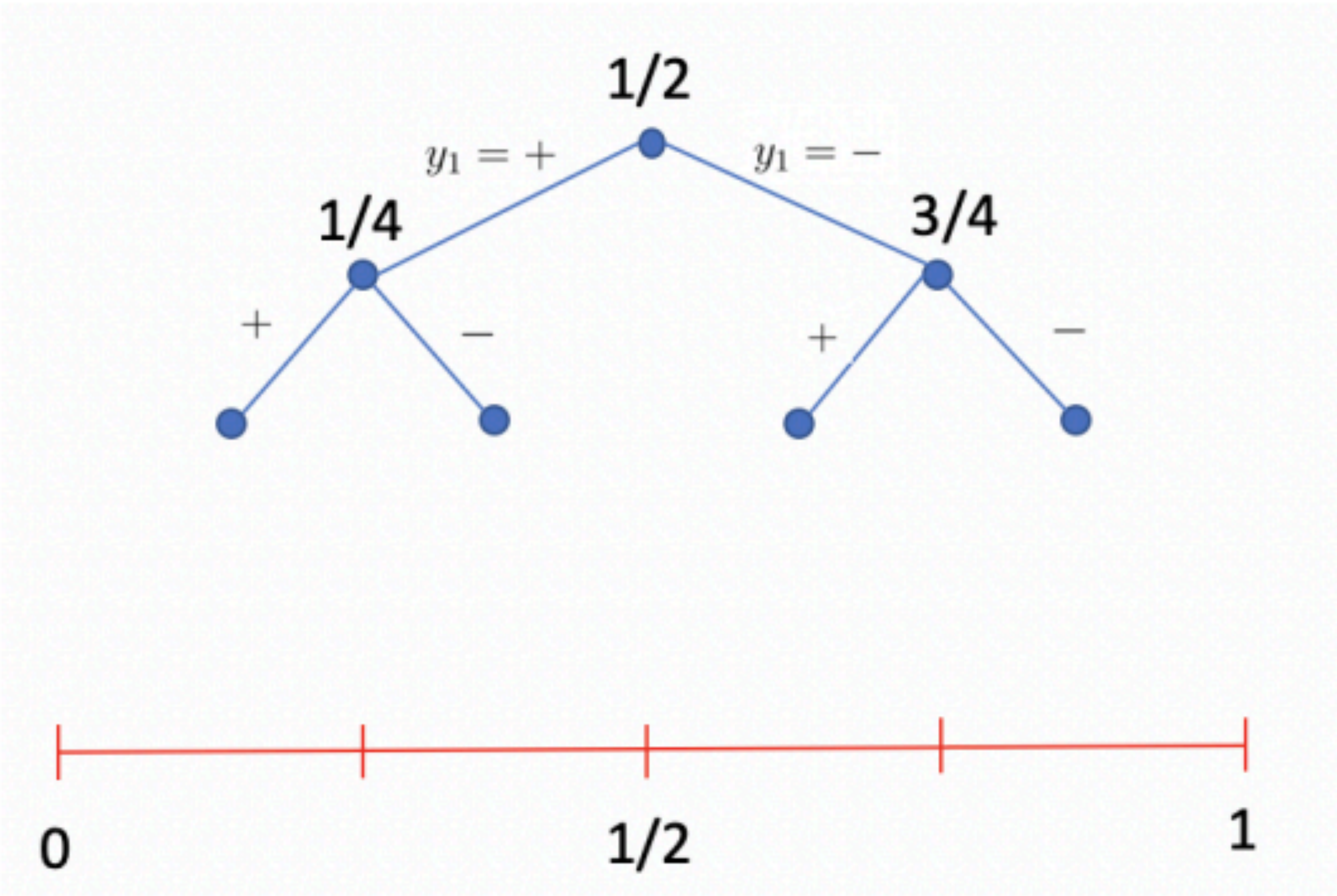
This ERM behaves much better!

BEYOND FINITE CLASS

Is VC dimension a good measure here?

$$f_a(x) = \begin{cases} 1 & \text{if } x \geq a \\ -1 & \text{otherwise.} \end{cases}$$





Can get $T/2$ mistakes in expectation!

There is a notion of Littlestone dimension that captures the complexity

EXAMPLE - PERCEPTRON

Same idea as the offline (batch) perceptron

Algorithm 3: Perceptron

Initialize $w_1 = 0$

for $t = 1, 2, \ldots$ **do**

  Receive x_t

  Predict $\hat{y}_t = \text{sign}\left(w_t^\top x_t\right)$

  Receive true label y_t

  if $\hat{y}_t \neq y_t$ **then** Update $w_{t+1} = w_t + y_t x_t$

  else Update $w_{t+1} = w_t$

end

Gets mistake bound $1/\gamma^2$ for margin γ and norm-1 bounded inputs

BEYOND REALIZABILITY

Is it possible to always get small mistake bound?

$$\text{Regret}_{\mathscr{L}}(\mathscr{F}, T) = \sum_{t=1}^{T} \ell(\hat{y}_t, y_t) - \min_{f \in \mathscr{F}} \sum_{t=1}^{T} \ell(f(x_t), y_t) .$$

Only need to do as well as the best classifier (expert) in hindsight

Function class $\mathscr{F}$ if **online learnable** for any sequence if $\lim_{T \to \infty} \frac{\text{Regret}_{\mathscr{L}}(\mathscr{F}, T)}{T} = 0.$

Example: $\text{Regret}_{\mathscr{L}}(\mathscr{F}, T) = \sqrt{T}$ or $\text{Regret}_{\mathscr{L}}(\mathscr{F}, T) = \log T$

WEIGHTED MAJORITY - GENERALIZATION TO HALVING

How can we use the idea of halving?

Algorithm 4: Weighted Majority

Initialize $w_{1,i} = 1$ for all $i \in [n]$
for $t = 1, 2, \ldots$ **do**
  Receive x_t
  Predict $\hat{y}_t = \text{sign}\left(\sum_{i=1}^{n} w_{t,i} f_i(x_t)\right)$
  Receive true label y_t
  Define $E_t = \{i : f_i(x_t) \neq y_t\}$ (set of all incorrect experts)
  if $i \in E_t$ **then** Update $w_{t+1,i} = w_{t,i}/2$
  else Update $w_{t+1,i} = w_{t,i}$
end

Down-weight the prediction whenever a classifier (expert) is making a mistake

WEIGHTED MAJORITY LEARNER

Theorem:

Let $\mathcal{F}$ be a finite hypothesis class. Let M be the total number of mistakes made by the Weighted Majority algorithm, and let M^* be the number of mistakes made by the best expert, then

$$M \leq 2.41(M^* + \log|\mathcal{F}|).$$

Not exactly the regret bound we wanted, but can improve to regret $O\left(\sqrt{T\log|\mathcal{F}|}\right)$

What happens when we add more good/bad experts (classifiers)?

ONLINE VERSUS BATCH LEARNING

Online Learning	**Batch Learning**
• Define function class	• Define function class
• Define loss function	• Define loss function
• Have inputs and corresponding labels	• Have inputs and corresponding labels
• Learning in each round, no difference between test and train	• Learn a model first using training data, then test
• Data can be adversarial	• Data is i.i.d.

MORE CHALLENGING ONLINE SETTINGS

- Limited feedback, only know the outcome of the choice we made
- Our choices change the environment

Next Lecture: Reinforcement Learning!