Introduction to Machine Learning

Learnability and Generalization Bounds

Overview

1. Probably approximately correct (PAC) model

2. Basic generalization bounds
   - Finite hypothesis class
   - Infinite hypothesis class

Reminder

Before we start... What did we learn so far?

- **Bayesian Inference.**
- **Gaussian mixture model.**
- **Expectation maximization.**

They all concentrate on **estimating the distribution**, **rather than** the **descision rule**.

Today we focus on learning the descision rule!

Reminder

Descision rule?

- **Naïve Bayes.**
- **K-NN for binary classification.**

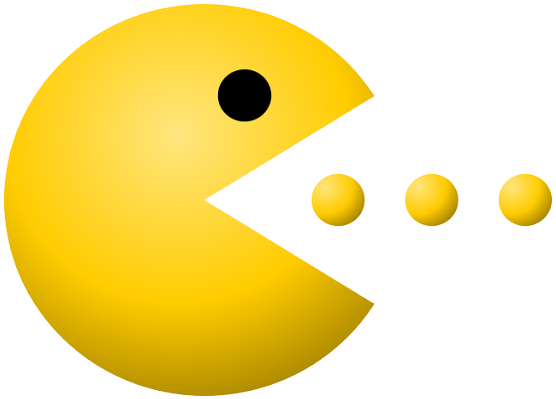
Motivation

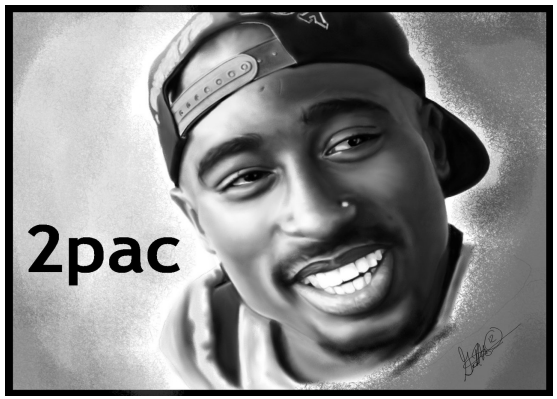
- We would like to define **learnability**.
- To identify between learnable and unlearnable tasks.
- What are the **requirements** for learnability? (assumptions, learning method, etc).
- What about the involved **complexities**? (size of data, runtime, etc).

Note: we focus only on **binary classification**.

Motivation

The solution is called... **PAC learning**!





Probably **A**pproximately **C**orrect.

Motivating example (PAC)

Before we give a very formal definition of learnability.. let's start with a simple toy example!

It will help us to identify the necessities in the model with some illustrations.

Learning average body-size person

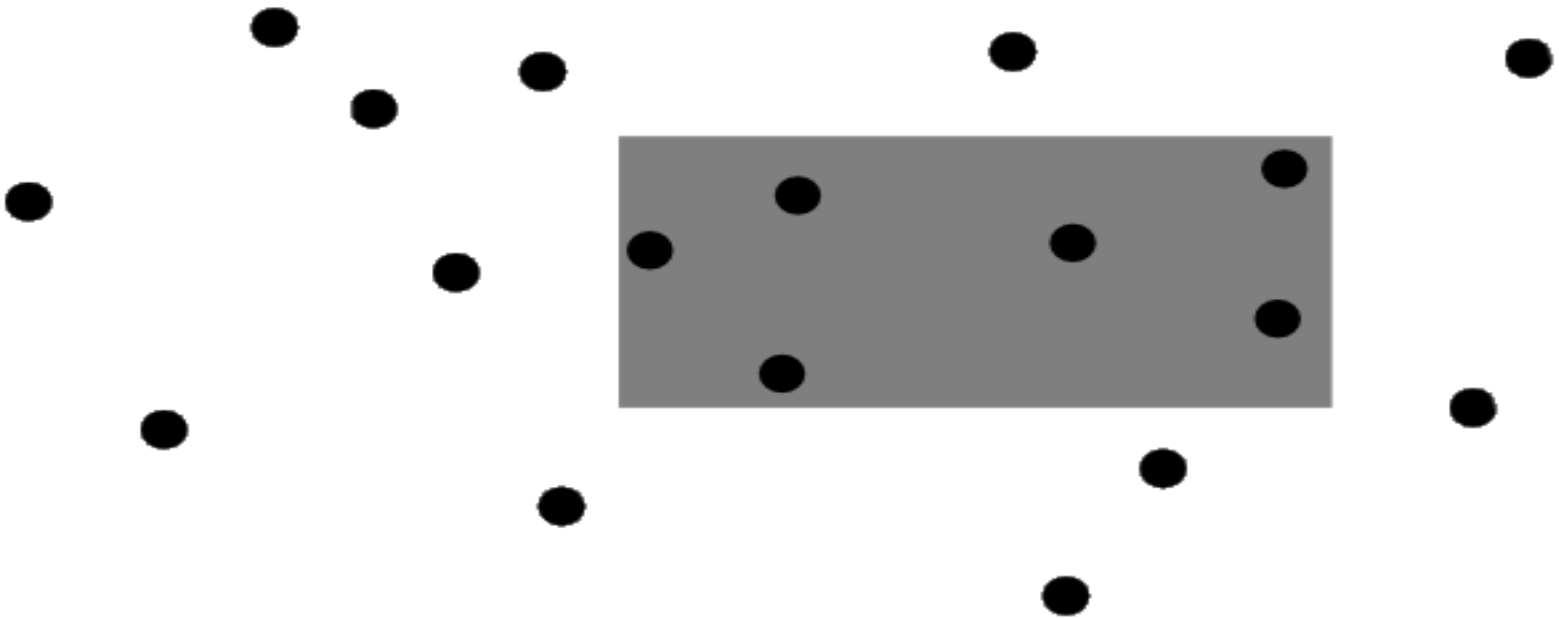
EXAMPLE: learn the average body-size of a person.

- Examples: (x, y), x is a height-weight tuple and $y \in \{1, -1\}$ determining if average body size or not.
- Assumption: **average body-size vectors lie in a rectangle.**

Motivating example (PAC)

First step.

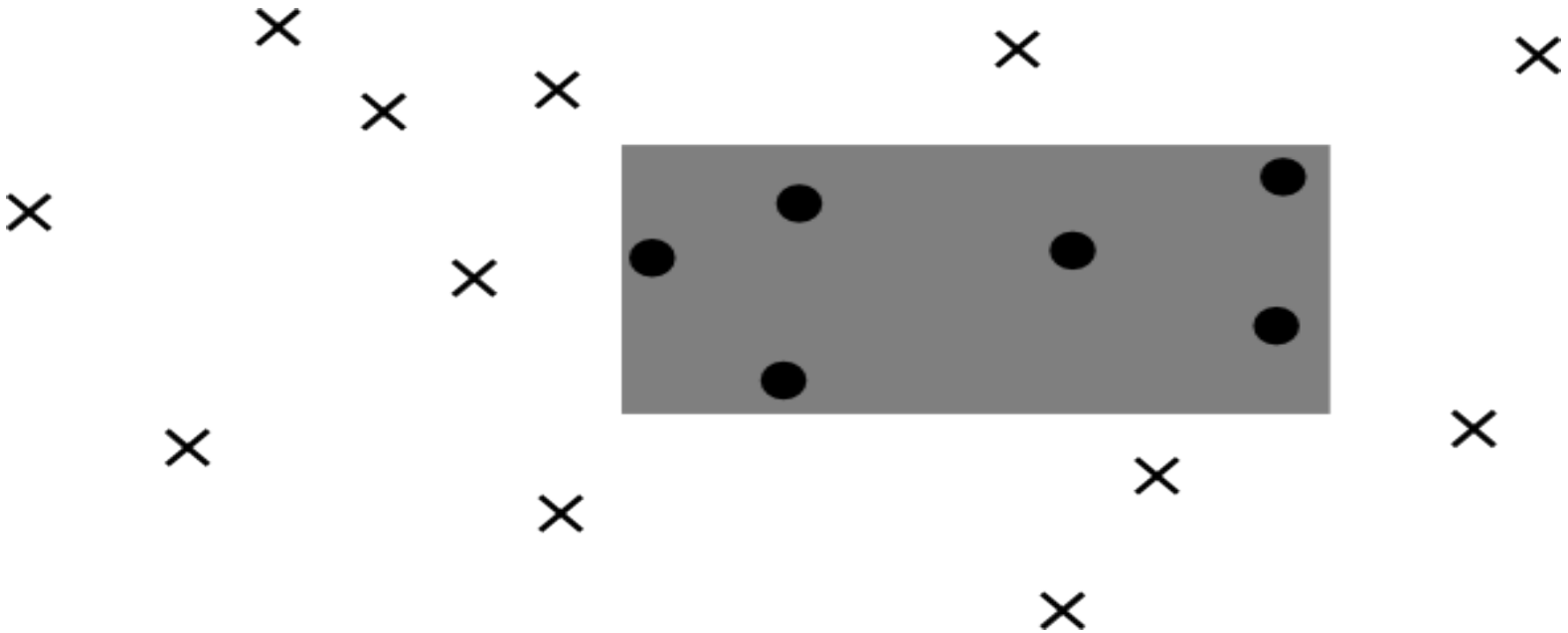
Select m i.i.d samples $\{x_i\}_{i=1}^m$ according to an unknown distribution D.



Motivating example (PAC)

Second step.

$c_t \in \mathcal{C}$ is selected and labels the samples, i.e, $(x_i, c_t(x_i))_{i=1}^m$.

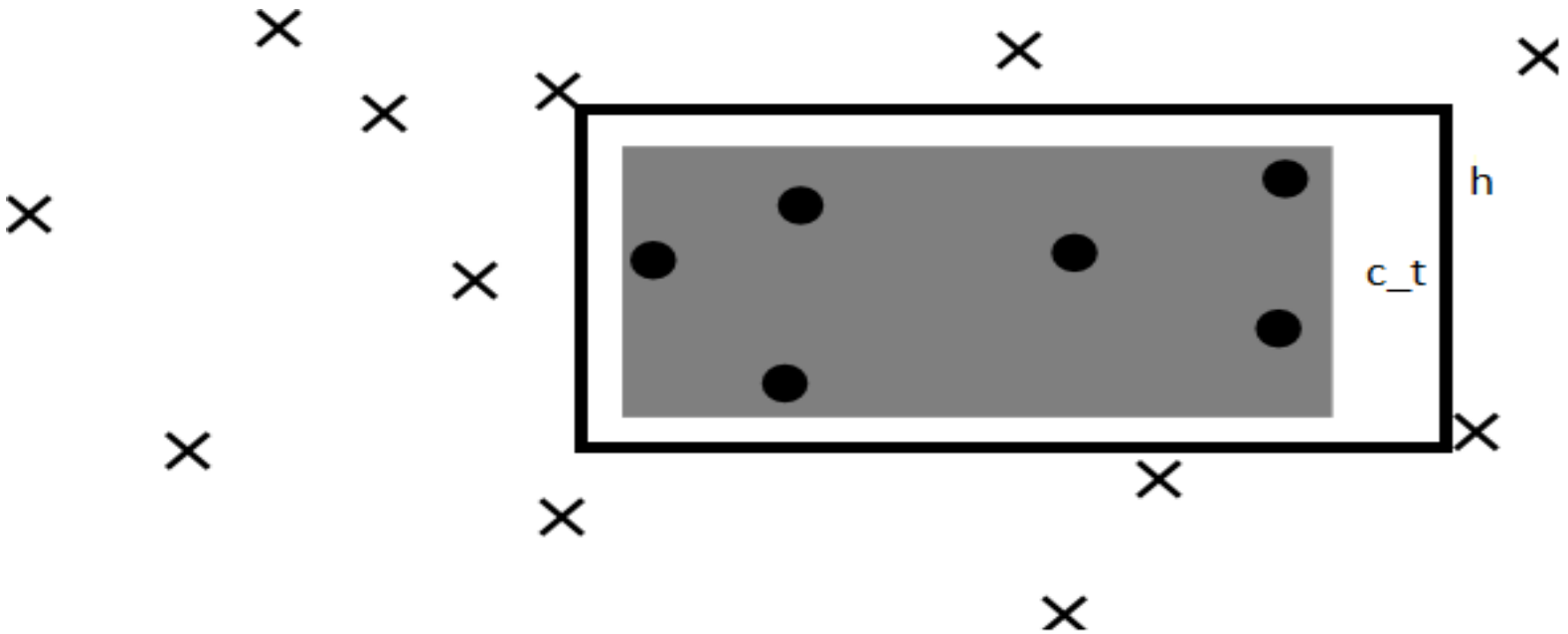


- X = negative example.
- O = positive example.

Motivating example (PAC)

Third step.

Algorithm A with access to the data examples $(x_i, c_t(x_i))_{i=1}^m$, selects $h \in \mathcal{H}$ to approximate c_t.



- X = negative example.
- O = positive example.

Motivating example (PAC)

- Assumption: target concept (the function we learn) is a rectangle.
- Goal: find a rectangle that "well" approximates the target.

What is a **good approximation**? Any ideas?

Motivating example (PAC)

The solution is simple: we would like to output a rectangle that has **small error** (w.r.t the target) with **high probability** over the selection of the data.

Sure.. but how do we measure this error rate? Why only with high probability?

Modeling PAC

- Assumption: samples are i.i.d according to some unknown distribution.
  - Identically.
  - Independently.
  - Distributed.
- The error rate = probability of mismatch, i.e,

$$error(h) = \mathbb{P}_{x \sim D}[h(x) \neq c_t(x)]$$

- Goal: **output a function that has low error rate with high probability over the selection of the data set**.

PAC approach

The PAC model is a general solution. As a consequence it is a **worst-case analysis**.

- Samples are drawn i.i.d according to **any arbitrary** unknown distribution.
- **Concentrate on the descision rule rather than the distribution**.

Why *any arbitrary distribution*?

Revisiting the example

We return to the toy example..

- Task: learn a target rectangle c_t from samples.
- Input: data set of m examples $(x_i, c_t(x_i))_{i=1}^m$ such that $x_i \sim D$.
- Goal: compute h that is a **good approximation** of c_t.

The **error** of h is measured by the **probability of mismatch**

$$error(h) = \mathbb{P}_{x \sim D}[h(x) \neq c_t(x)]$$

The PAC model for the example

- Task: learn a target rectangle c_t from samples.
- Input: data set of m examples $(x_i, c_t(x_i))_{i=1}^m$ such that $x_i \sim D$.
- Goal: compute h such that,

$$\mathbb{P}_{S \sim D^m}[error(h) \leq \epsilon] \geq 1 - \delta$$

Here, $S = \{x_i\}_{i=1}^m$.

Differently said, **the algorithm returns a hypothesis that has small error with high probability over the selection of the data**.

The PAC model: binary classification

Let's model PAC-learning!

- Target concept out of a concept class: $c_t \in \mathcal{C}$.
  - $c_t : X \to \{-1, 1\}$.
- Hypothesis out of a hypothesis class: $h \in \mathcal{H}$.
  - $h : X \to \{-1, 1\}$.
- Assume: a fixed unknown distribution D over X.
- Error is measured as follows:

$$error(h) = \mathbb{P}_{x \sim D}[h(x) \neq c_t(x)]$$

The PAC model: binary classification (contd')

DEFINITION:

- $\mathcal{C}$ is PAC-learnable with $\mathcal{H}$ if
- There is an algorithm A such that:
- For any distribution D over X and $c_t \in \mathcal{C}$,
- For every ϵ and δ there is m such that
- A outputs h in $\mathcal{H}$ using data set $(x_i, c(x_i))_{i=1}^m$ for $S = \{x_i\}_{i=1}^m \sim D^m$
- With probability $\geq 1 - \delta$ over S the error is $error(h) \leq \epsilon$.

The size of data, m is allowed to **depend** only on ϵ, δ.

Sleeping?

Did you fall asleep? Keep up!

The PAC model:dictionary

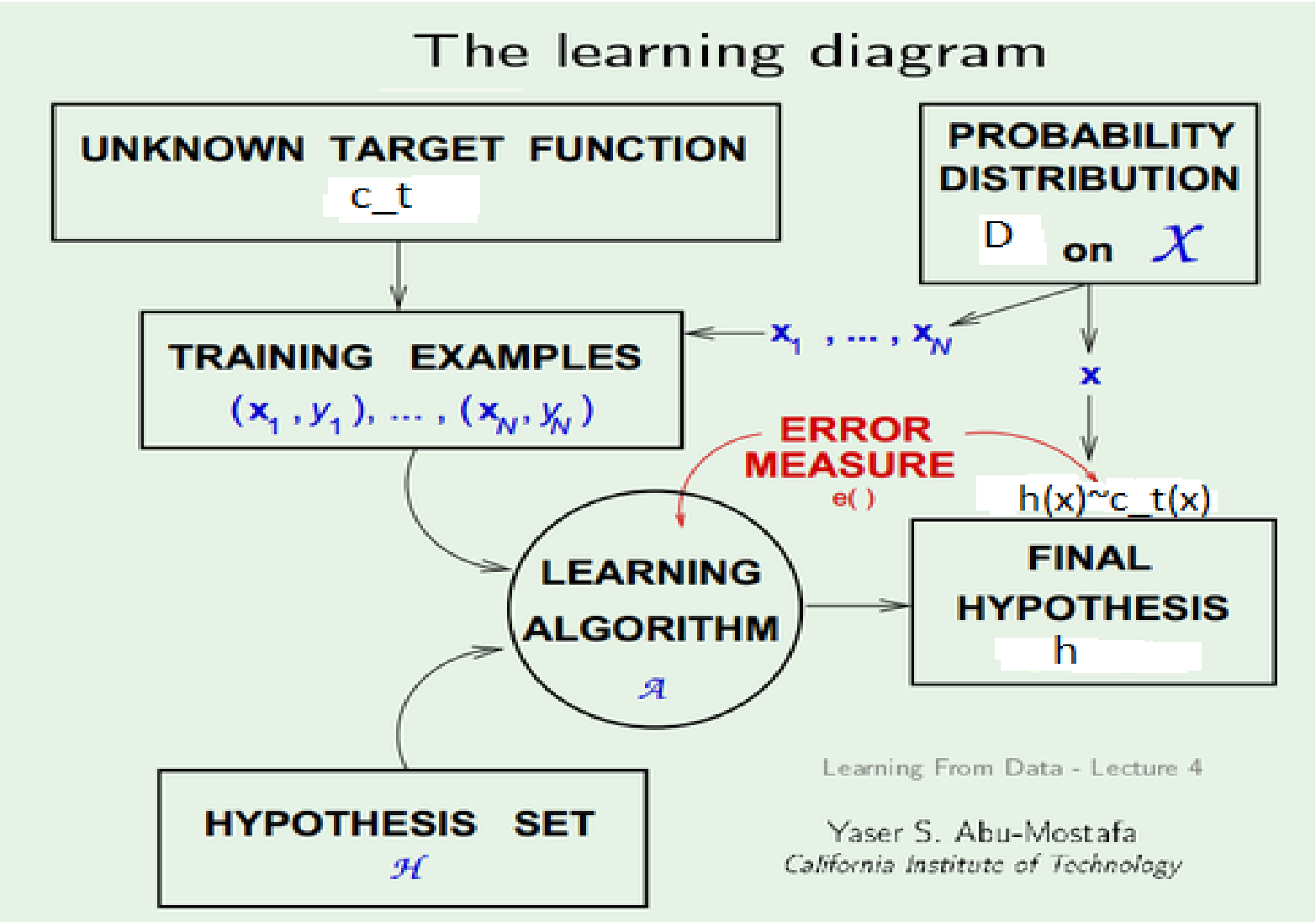
1. c: concept; a function.
2. c_t: target concept; the function we want to learn.
3. h: a hypothesis; a candidate approximation to c_t.
4. $\mathcal{C}$: concept class; a set of concepts.
5. $\mathcal{H}$: hypothesis class; a set of hypotheses.
6. D: a distribution over X.
7. A: a learning algorithm; function from data to $\mathcal{H}$.
8. ϵ: accuracy rate.
9. δ: confidence rate.

The PAC model: conclusive remarks

- The only one assumption is that samples are i.i.d.
- We have two independent parameters ϵ and δ.
- The output is tested on the same distribution.

The PAC model: conclusive remarks

Let's conclude the learning protocol with a diagram.



Generalization bounds

All generalizations are false,
including this one.
Marc Twain.

Generalization bounds

In the first part of the lecture we discussed learnability.

Reminder: *"A concept class is learnable if there is an algorithm such that for any accuracy ϵ and confidence δ rates, with enough data returns a hypothesis h that has error $\leq \epsilon$ with probability at least $1 - \delta$".*

In this part of the lecture we **focus on the size of the data**. We would like to provide sufficient bounds for m.

Finite realizable case

Let's see what happens when $\mathcal{C} = \mathcal{H}$ is finite.

Learnable? Unlearnable? Any suggestion for an algorithm?

Note: whenever $\mathcal{C} = \mathcal{H}$ holds, we call it **realizable case**.

Finite realizable case

Actually, it is **learnable by a very simple algorithm**.

Sample: produce $(x_i, c_t(x_i))_{i=1}^m$.

Output: a function $h \in \mathcal{C}$ that is consistent with data.

Finite realizable case

Why does it work? Let's do some simple analysis!

Goal: for ϵ, δ introduce m such that

$$\mathbb{P}[error(h_{output}) \leq \epsilon] \geq 1 - \delta$$

We say that h is ϵ-bad if $error(h) > \epsilon$.

Our algorithm returns any hypothesis that is consistent with data. Therefore, we would like to say that the event "$\exists h$ that is ϵ-bad and consistent" has small probability.

Finite realizable case

If h is ϵ-bad we have:

$$\mathbb{P}_{x \sim D}[h(x) = c_t(x)] \leq 1 - \epsilon$$

Therefore,

$$\mathbb{P}_{x_1,\ldots,x_m \sim D^m}[h \text{ is } \epsilon - \text{bad and } \forall i : h(x_i) = c_t(x_i)] \leq (1-\epsilon)^m < e^{-\epsilon m}$$

Finite realizable case

In addition,

$$
\begin{aligned}
&\mathbb{P}[A \text{ outputs an } \epsilon - \text{bad output}] \\
&\leq \mathbb{P}[\exists h \ \ \epsilon - \text{bad and } \forall i : h(x_i) = c_t(x_i)] \\
&\leq \sum_{h \ \ \epsilon-\text{bad}, h \in \mathcal{C}} \mathbb{P}[\forall i : h(x_i) = c_t(x_i)] \\
&\leq \sum_{h \ \ \epsilon-\text{bad}, h \in \mathcal{C}} \exp(-\epsilon m) \\
&= |\{h \text{ is } \epsilon - \text{bad}, h \in \mathcal{C}\}| \exp(-\epsilon m) \leq |\mathcal{C}| \cdot \exp(-\epsilon m)
\end{aligned}
$$

Finite realizable case

- We bound the estimation by δ and **wish for good**!

$$\mathbb{P}[A \text{ outputs an } \epsilon - \text{bad output}] \leq |\mathcal{C}| \cdot \exp(-\epsilon m) \leq \delta$$

- Therefore, for $m \geq \frac{1}{\epsilon} \log \frac{|\mathcal{C}|}{\delta}$ we have,

$$\mathbb{P}[A \text{ outputs an } \epsilon - \text{bad output}] \leq \delta$$

- Alternatively,

$$\mathbb{P}[A \text{ outputs } h : \mathit{error}(h) \leq \epsilon] \geq 1 - \delta$$

- QED.

Any questions?

Example 1: learning OR of literals

What about a nice example?

Consider the following learning problem.

Task: $\mathcal{C} = \{z_1 \vee ... \vee z_n\}$ where z_i is a literal (i.e, x_i or $\bar{x}_i$).

We discussed that returning a hypothesis is consistent with data is a sucessful learning method in the finite realizable case.

How to implement this method algorithmically? Any ideas?

Example 1: learning OR of literals

ELIM algorithm.

- Keep a list of all literals.
- For every example (x, y) such that $y = 0$, remove all literals that are 1 in x.
- Return the $\bigvee$ of the remaninig literals.

Sample size $m \geq \frac{1}{\epsilon} \log(\frac{3^n}{\delta}) = O((n + \log(1/\delta))/\epsilon)$.

Example 1: learning OR of literals

Correctness.

- We have to show consistency with data.
- Our set of literals includes the target OR literals (by induction on the iteration).
  Therefore it outputs 1 whenever the target function does.
- For every $y = 0$: it is obviously consistent by definition.

Finite unrealizable case

What about the **unrealizable case**, i.e, $\mathcal{C} \neq \mathcal{H}$? (both still finite).

In this case the goal is to return h such that
$error(h) \leq \min_{h \in \mathcal{H}} error(h') + \epsilon$ (with probability $\geq 1 - \delta$).

i.e, to return h that is ϵ-close to optimal.

Finite unrealizable case

In the unrealizable case, it is still **learnable by the same simple algorithm**.

Sample: produce $(x_i, c_t(x_i))_{i=1}^m$.

Output: a function $h \in \mathcal{H}$ that has minimal error on data.

Finite unrealizable case

What is the strategy?

We have two statements:
1. Proof idea: for enough samples, for all $h \in \mathcal{H}$, **the error** of h **on data** is **close to** it's **true error**!

2. It sounds like a promising approach to return h that **minimizes the error on data**.

Finite unrealizable case

Let's outline the proof?

1. For all $h \in \mathcal{H}$, with high probability, **the error** of h **on data** is **close to** it's **true error**.
2. With high probability, for all $h \in \mathcal{H}$, **the error** of h **on data** is **close to** it's **true error**.
3. It is sufficient for proving that the algorithm successfully learns the task.

Finite unrealizable case

And even more formal...

$$\textbf{empirical error of } h = error_S(h) = \frac{1}{m}\sum_{i=1}^{m} I[h(x_i) = c_t(x_i)]$$

Each $I[h(x_i) = c_t(x_i)]$ determines if h classifies or misclassifies x_i (i.e, returns 0 if $h(x_i) = c_t(x_i)$ and 1 otherwise). Their average measures the error of h on the data.

1. For all $h \in \mathcal{H}$, with high probability, $|error_S(h) - error(h)| \leq \epsilon$.
2. for $m(\epsilon, \delta)$, with probability $\geq 1 - \delta$, for all $h \in \mathcal{H}$,

$$|error_S(h) - error(h)| \leq \epsilon$$

Finite unrealizable case

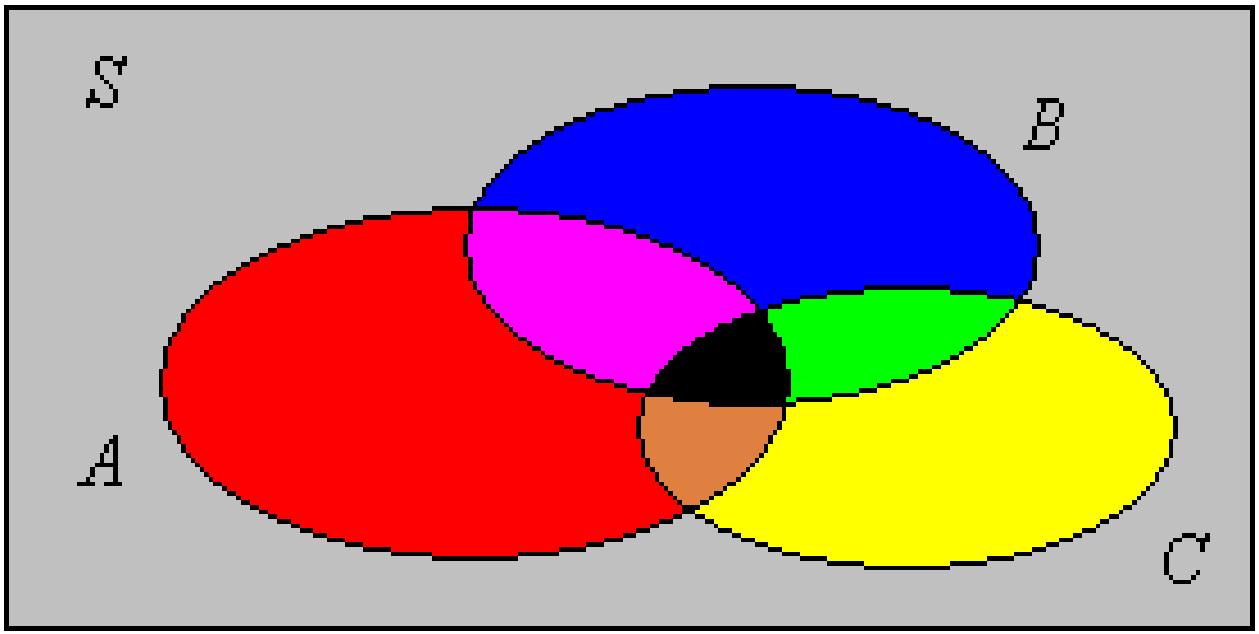
We will use two probabilistic bounds.

- Hoeffding inequality: to estimate the probability that the empirical error and the true error are close **for a given** h.
- Union bound: to **combine** all results **for all** $h \in \mathcal{H}$.

Union bound: reminder

Union bound.

$$\mathbb{P}[A \cup B] \leq \mathbb{P}[A] + \mathbb{P}[B]$$



Hoeffding's inequality: reminder

Hoeffding's inequality.

Let $X_1, ..., X_m$ be independent random variables in [0,1] (i.e, $X_i \in [0, 1]$), $\bar{X}$ their average and $\epsilon > 0$, then,

$$\mathbb{P}[|\bar{X} - \mathbb{E}[\bar{X}]| \geq \epsilon] \leq 2\exp(-2m\epsilon^2)$$

Hoeffding's inequality: reminder

What if $\{X_i\}_{i=1}^m$ are i.i.d?

Denote μ their expectation, then,

$$\mathbb{P}[|\bar{X} - \mu| \geq \epsilon] \leq 2\exp(-2m\epsilon^2)$$

i.e, the probability that the average is ϵ-far from the expected value decreases exponentially in the number of samples.

Finite unrealizable case

We would like to show that for any h with high probability,

$$|error_S(h) - error(h)| \leq \epsilon$$

Consider the random variables

$$X_i = I[h(x_i) = c_t(x_i)]$$

Their **average is the empirical error**!

X_i are i.i.d, right? What is the expectation of each one?

$$\mathbb{E}[X_i] = error(h)$$

So.. we can apply **Hoeffding's inequality**!

Finite unrealizable case

- By Hoeffding's inequality, for any $h \in \mathcal{H}$,

$$\mathbb{P}[|error(h) - error_S(h)| > \epsilon/2] \leq 2\exp(-\epsilon^2 m/2)$$

- By union bound for all h,

$$\mathbb{P}[\exists h \in \mathcal{H} : |error(h) - error_S(h)| > \epsilon/2] \leq 2|\mathcal{H}|\exp(-\epsilon^2 m/2)$$

- Which is bounded by δ for $m > 2/\epsilon^2 \log(2|\mathcal{H}|/\delta)$.

Finite unrealizable case

- Assume that,

$$\forall h \in \mathcal{H} : |error(h) - error_S(h)| \leq \epsilon/2$$

- In particular, $error_S(h_{optimal}) \leq error(h_{optimal}) + \epsilon/2$.
- And also, $error(h_{output}) \leq error_S(h_{output}) + \epsilon/2$.
- Therefore, since $error_S(h_{output}) \leq error_S(h_{optimal})$,

$$error(h_{output}) \leq error_S(h_{optimal}) + \epsilon/2 \leq error(h_{optimal}) + \epsilon$$

- Combining both results, for $m > 2/\epsilon^2 \log(2|\mathcal{H}|/\delta)$ we have,

$$\mathbb{P}[error(h_{output}) \leq error(h_{optimal}) + \epsilon] \geq 1 - \delta$$

Finite case: conclusions

- Returning a hypothesis that has minimal empirical error (error on data) is a good learner (**VERY SIMPLIFYING**, RIGHT!?).
- Realizable finite case sample complexity: $O((1/\epsilon)\log(|\mathcal{H}|/\delta))$.
- Unrealizable finite case sample complexity: $O((1/\epsilon)^2\log(|\mathcal{H}|/\delta))$.

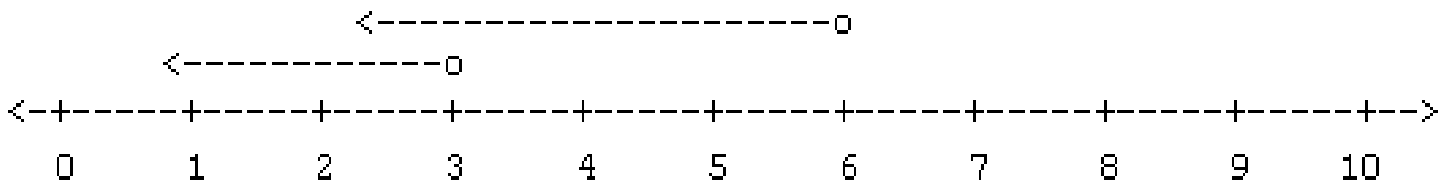
Interesting huh? Why is there a **quadratic blowup**? Any thought?

Actually, it is a very deep question in learning theory...

Example 2: learning thresholds

What about **infinite concept classes**?

Task: $\mathcal{C} = \{f_\theta(x) = I[x \in (-\infty, \theta)] : \theta \in [0, 1]\}$.



Is it learnable? How many samples?

Example 2: learning thresholds

Algorithm.

- Return the minimal interval (∞, θ) that includes all positive examples.

Illustrations on board.

Example 2: learning intervals

Correctness.

- Denote the target θ^* and the output θ.
- The error is $error(f_\theta) = D([\theta, \theta^*])$.
- We prove that $\mathbb{P}[D([\theta, \theta^*]) \leq \epsilon] \geq 1 - \delta$.
- We have,

$$\begin{aligned}
&\mathbb{P}[D([\theta, \theta^*]) > \epsilon] \\
&\leq \mathbb{P}[\text{All samples didn't fall in } [\theta, \theta^*)] \\
&\leq (1-\epsilon)^m < \exp(-\epsilon m) < \delta
\end{aligned}$$

- For appropriate m function of ϵ, δ.
- Therefore, $\mathbb{P}[D([\theta, \theta^*]) \leq \epsilon] \geq 1 - \delta$ as desired.

Where did we use i.i.d'ness? Where did we use the fact that $D([\theta, \theta^*]) > \epsilon$?

Model selection

We learn $\mathcal{C}$ with a different class $\mathcal{H}$, right? How do we select $\mathcal{H}$ appropriately?

For a given c_t we define $\epsilon = \inf_{h \in \mathcal{H}} error(h)$. This is the optimal error rate of the class $\mathcal{H}$.

In the PAC-learning model, with data of size m we can obtain error at most $\epsilon + \epsilon_{\mathcal{H}}(m)$.

Model selection

Model selection asks what are the conditions to reduce ϵ.

$\mathcal{H}$ is viewed as the model. How to select a good model?

Idea: in order to reduce ϵ we can enlarge the class $\mathcal{H}$ (i.e, to take a very complex one).

Model selection

That's might be a good approach..
But what about $\epsilon_{\mathcal{H}}(m)$?

It is claimed that when $\mathcal{H}$ is very complex, this quantity is larger... too bad...

It seems there is a **tradeoff** between the **ability to learn** and the **expressivity** of the hypothesis class.

$$\epsilon + \epsilon_{\mathcal{H}}(m)$$

Model selection: overfitting

For instance, a huge hypothesis class $\mathcal{H}$ might have $\epsilon = 0$.
Nevertheless, $\epsilon_{\mathcal{H}}(m)$ may be large and decrease very slow...

This is called overfitting.. The hypothesis class fits too much ;)

Model selection: overfitting

So... What shall we do?!

We emply a penalty term on the complexity.

We can take a **long chain of classes** $\mathcal{H}_1 \subset \mathcal{H}_2 \subset ... \subset \mathcal{H}_\infty = \mathcal{H}$ and define a **penalty term** $P(g, m)$ for $g \in \mathcal{H}$ that **depends on the complexity** of g (depends on $\arg\min_i g \in \mathcal{H}_i$) and on the **number of samples** m.
Select the hypothesis that minimizes the empirical error along to the penalty on the complexity:

$$h = \arg\min_{h \in \mathcal{H}} error_S(h) + P(g, m)$$

Model selection: regularization

$P =$ **regularization**.

How to select the regularization? Any ideas?

P depends both on the complexity and the number of samples!
Why? because with more data it becomes easier to learn!

Model selection: regularization

Actually there are theorems (generalization bounds) that show how to select the regularization term.

EXAMPLES:

$$\frac{1}{\delta}\sqrt{\frac{2VC\log(2em/VC)}{m}},\ \text{where } VC \text{ is a "dimension" of the class}$$

$$\sqrt{\frac{|h|+\log(2/\delta)}{2m}},\ \text{where } |h| \text{ is a "description length"}$$

$$\sqrt{\frac{||h||^2+\log(m/\delta)}{2(m-1)}},\ \text{where } ||h|| \text{ is a norm}$$

$$\lambda_m||h||^2$$