CS60050 Machine Learning
IIT Kharagpur

PAC Learning

Somak Aditya, Sudeshna Sarkar

Goal of Learning Theory

To understand

- What kinds of tasks are ***learnable***?
- What kind of data is required for learnability?
- What are the (space, time) requirements of the learning algorithm.?

To develop and analyze models

- Develop algorithms that provably meet desired criteria
- Prove guarantees for successful algorithms

Goal of Learning Theory

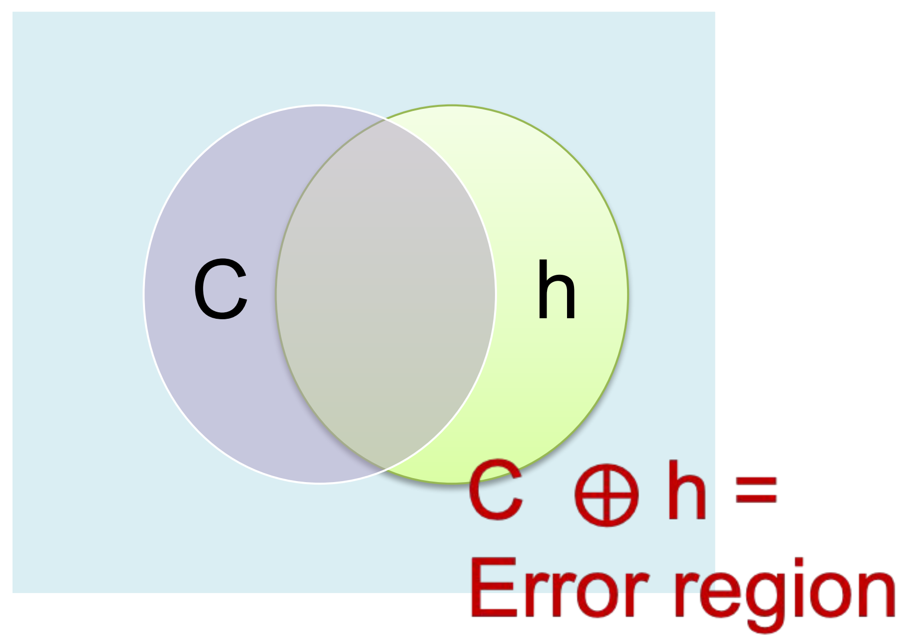
Two core aspects of ML

- Algorithm Design. How to optimize?
- Confidence for rule effectiveness on future data.

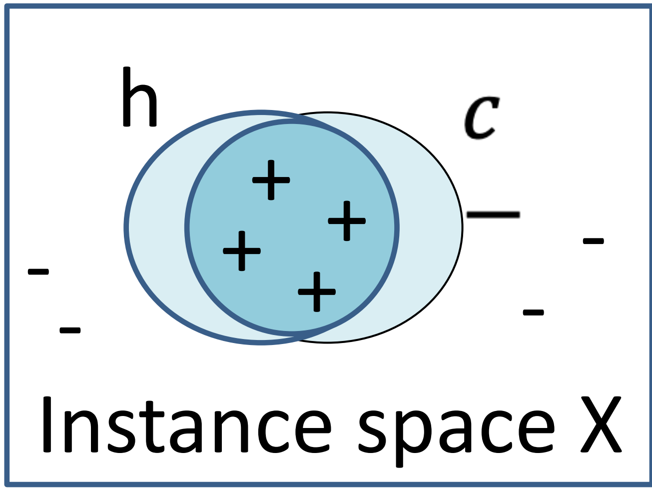
We need particular settings (models)

- Probably Approximately Correct (PAC)

Approximately correct $(P(c \oplus h) \leq \epsilon)$



Prototypical Concept Learning Task



Given

- Instance Space X
  - (e.g., $X = R^d$ or $X = \{0,1\}^d$
- Distribution $\mathcal{D}$ over X
- Target concept c
  - *Concept Space C, set of possible target functions*
- Hypothesis Space $\mathcal{H}$
- Training Instances S = $\{(x_i, c(x_i))\}$ x_i i.i.d. from $\mathcal{D}$

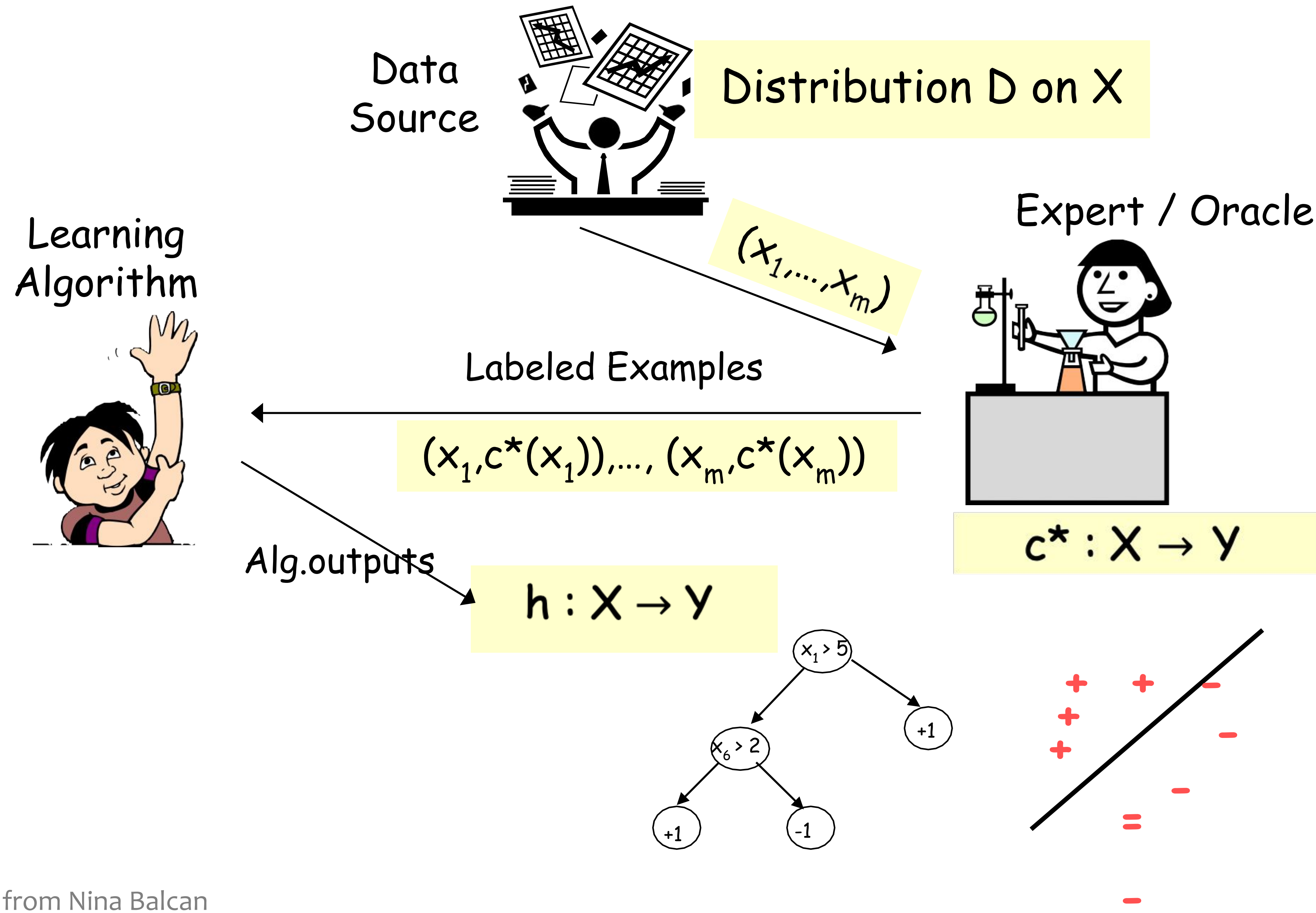
Determine

- A hypothesis h ∈ $\mathcal{H}$ s.t. $h(x) = c(x)$ for all x in S? consistent
- A hypothesis h ∈ $\mathcal{H}$ s.t. $h(x) = c(x)$ for all x in X?

ML--> An algorithm does optimization over S, find hypothesis h.

Goal: Find h which has small error over $\mathcal{D}$

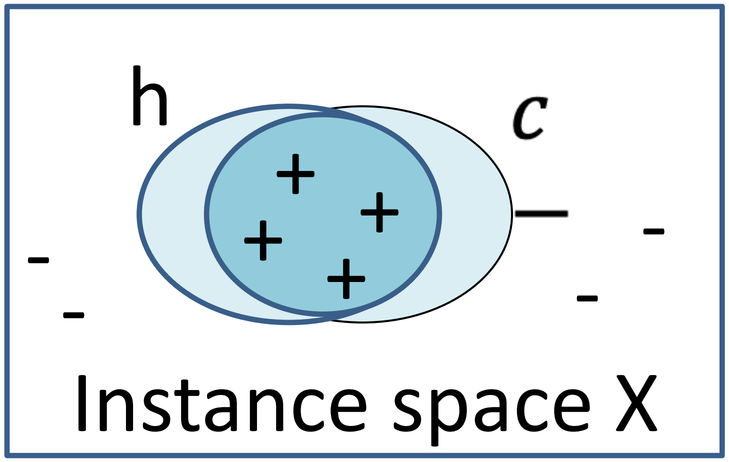
PAC/SLT models for Supervised Learning



Computational Learning Theory

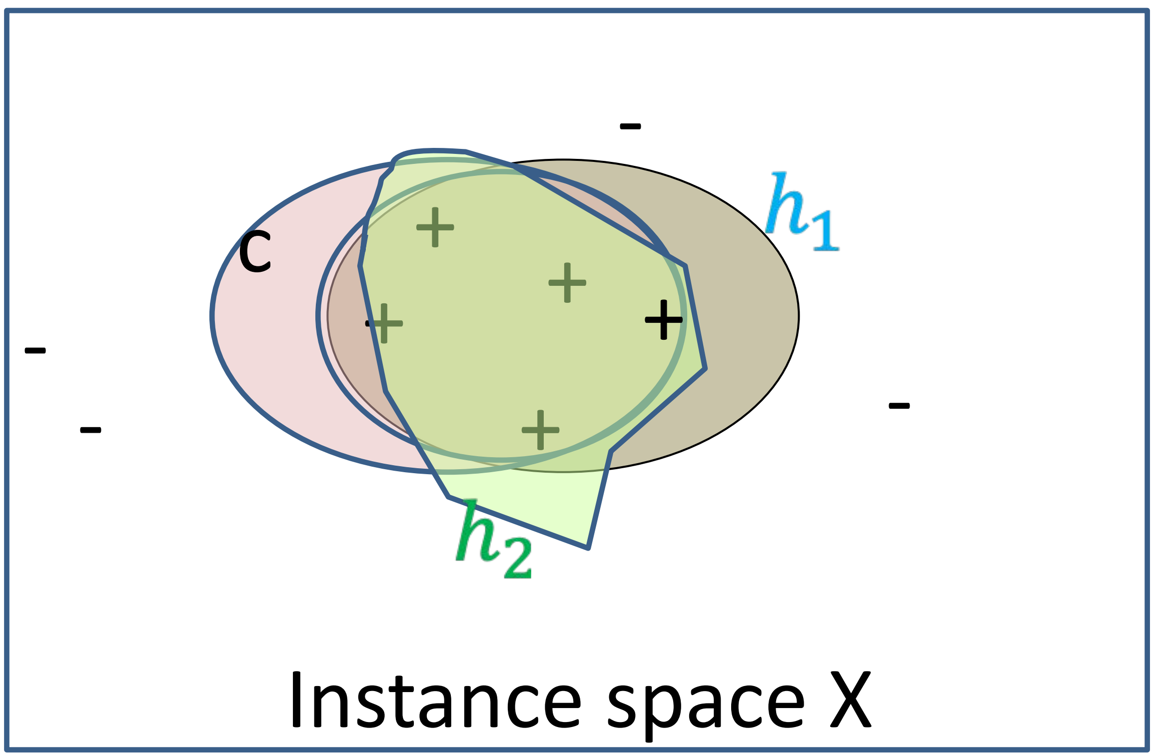
- *Can we be certain about how the learning algorithm generalizes?*
- *We would have to see all the examples.*

- *Inductive inference – generalizing beyond the training data is impossible unless we add more assumptions (e.g., priors over H)*
  - We need a bias!



Function Approximation

- *How many labeled examples in order to determine which of the 2^{2^N} hypothesis is the correct one?*
- *All 2^N instances in X must be labeled!*
- *Inductive inference: generalizing beyond the training data is impossible unless we add more assumptions (e.g., bias)*



$H = \{h: X \to Y\}$

$||H|=2^{|X|} = 2^{2^N}$

Expectations of learning

We cannot expect a learner to learn a concept exactly

- There will generally be multiple concepts consistent with the available data (which represent a small fraction of the available instance space)

> The only realistic expectation of a good learner is that with high probability it will learn a close approximation to the target concept

We cannot always expect to learn a close approximation to the target concept

Sometimes (hopefully only rarely) the training set will not be representative (will contain uncommon examples)

Error of a hypothesis

*The **true error** of hypothesis h, with respect to the target concept c and observation distribution $\mathcal{D}$ is the probability that h will misclassify an instance drawn according to $\mathcal{D}$*

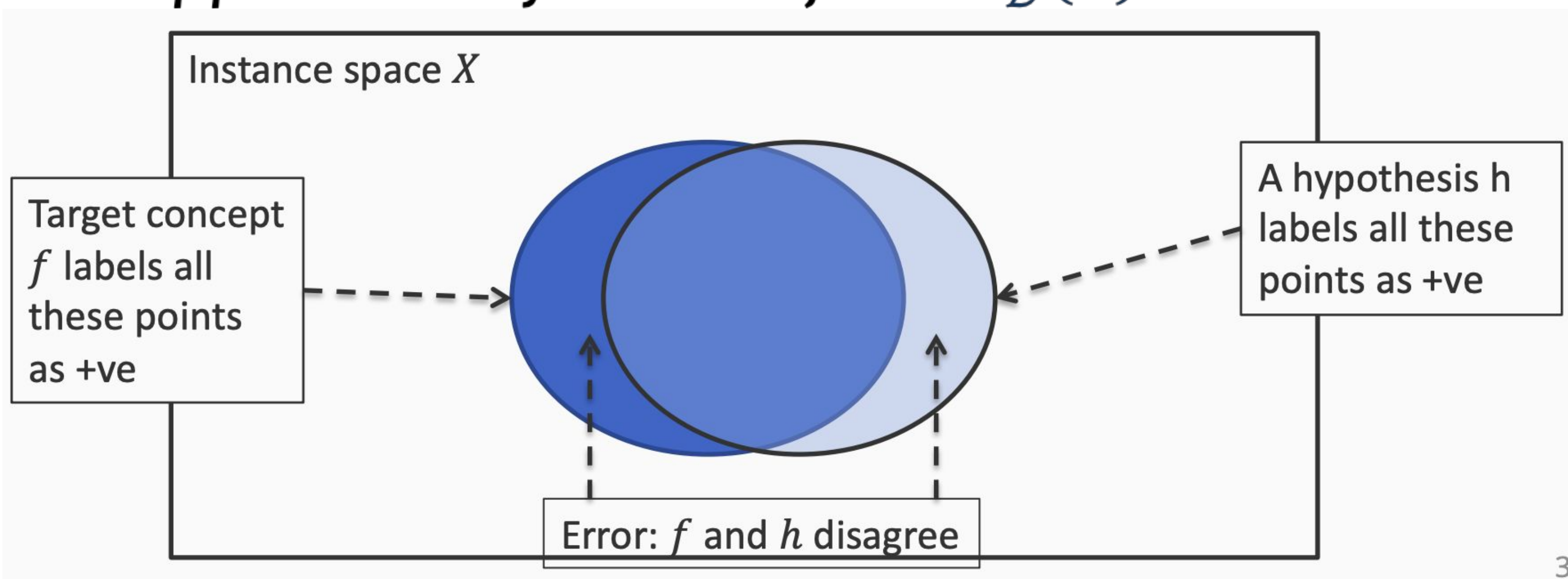
$$error_{\mathcal{D}}(h) = Pr_{x\sim\mathcal{D}}[c(x) \neq h(x)]$$

In a perfect world, we'd like the true error to be 0.

Bias: Fix hypothesis space H

c may not be in H => Find h close to c

A hypothesis h is approximately correct if $error_{\mathcal{D}}(h) \leq \varepsilon$



The need of a PAC model

- *Goal: h has small error over D.*
- True error: $error_D(h) = \quad Pr_{x\sim D}\ \big(h(x) \neq c^*(x)\big)$
  - How often $h(x) \neq c^*(x)$ over future instances drawn at random from D
- *But, can only measure:*

$$\text{Training error: } error_S(h) = \frac{1}{m}\sum_i I(h(x_i) \neq c^*(x))$$

How often $h(x) \neq c^*(x)$ over training Instances

Sample Complexity: bound $error_D(h)$ in terms of $error_S(h)$

Probably Approximately Correct Learning

PAC Learning concerns efficient learning

We would like to prove that

With high ***probability*** an (efficient) learning algorithm will find a hypothesis that is ***approximately*** identical to the hidden target concept.

We specify two parameters, ε and δ and require that

- *with probability at least $(1-\delta)$*
- *a system learn a concept with error at most ε.*

PAC Learnability

Consider a concept class C defined over an instance space X (containing instances of length n), and a learner L using a hypothesis space H

The concept class C is PAC learnable by L using H if
for all $f \in C$,
for all distribution D over X, and fixed $0 < \epsilon, \delta < 1$,
given m examples sampled independently according to D, with probability at least $(1 - \delta)$, the algorithm L produces a hypothesis $h \in H$ that has error at most ϵ,

PAC Learnability

Consider a concept class C defined over an instance space X (containing instances of length n), and a learner L using a hypothesis space H

The concept class C is PAC learnable by L using H if

for all $f \in C$,

for all distribution D over X, and fixed $0 < \epsilon, \delta < 1$,

$\boxed{\text{given } m \text{ examples}}$ sampled independently according to D, with probability at least $(1 - \delta)$, the algorithm L produces a hypothesis $h \in H$ that has error at most ϵ,

where $\boxed{m \text{ is } \textit{polynomial} \text{ in } 1/\epsilon, 1/\delta, n \text{ and } size}$

Given a small enough number of examples

PAC Learnability

Consider a concept class C defined over an instance space X (containing instances of length n), and a learner L using a hypothesis space H

The concept class C is PAC learnable by L using H if
for all $f \in C$,
for all distribution D over X, and fixed $0 < \epsilon, \delta < 1$,
given m examples sampled independently according to D, with probability at least $(1 - \delta)$, the algorithm L produces a hypothesis $h \in H$ that has error at most ϵ,
where m is *polynomial* in $1/\epsilon$, $1/\delta$, n and $size(H)$.

Given a small enough number of examples

PAC Learnability

Consider a concept class C defined over an instance space X (containing instances of length n), and a learner L using a hypothesis space H

The concept class C is PAC learnable by L using H if

for all $f \in C$,

for all distribution D over X, and fixed $0 < \epsilon, \delta < 1$,

given m examples sampled independently according to D, with probability at least $(1 - \delta)$, $\boxed{\text{the algorithm } L \text{ produces a hypothesis } h \in H \text{ that has error at most } \epsilon,}$

where m is *polynomial* in $1/\epsilon$, $1/\delta$, n and $size(H)$.

Given a small enough number of examples

with high probability

the learner will produce a "good enough" classifier.

PAC Learnability

Consider a concept class C defined over an instance space X (containing instances of length n), and a learner L using a hypothesis space H

The concept class C is PAC learnable by L using H if

for all $f \in C$,

for all distribution D over X, and fixed $0 < \epsilon, \delta < 1$,

given m examples sampled independently according to D, with probability at least $(1-\delta)$, the algorithm L produces a hypothesis $h \in H$ that has error at most ϵ,

where m is *polynomial* in $1/\epsilon$, $1/\delta$, n and $size(H)$.

recall that $Err_D(h) = Pr_{x\sim D}[f(x) \neq h(x)]$

Given a small enough number of examples

with high probability

the learner will produce a "good enough" classifier.

PAC Learnability

Consider a concept class C defined over an instance space X (containing instances of length n), and a learner L using a hypothesis space H

The concept class C is PAC learnable by L using H if
for all $f \in C$,
for all distribution D over X, and fixed $0 < \epsilon, \delta < 1$,
given m examples sampled independently according to D, with probability at least $(1 - \delta)$, the algorithm L produces a hypothesis $h \in H$ that has error at most ϵ,
where m is *polynomial* in $1/\epsilon$, $1/\delta$, n and $size(H)$.

The concept class C is *efficiently learnable* if L can produce the hypothesis in time that is polynomial in $1/\epsilon$, $1/\delta$, n and $size(H)$.

Sample Complexity for Supervised Learning

Theorem

$$m \geq \frac{1}{\in}\left[In(|H|) + In\left(\frac{1}{\delta}\right)\right]$$

labeled examples are sufficient so that with prob. $1-\delta$, *all* $h \in H$ *with* $error_{\mathcal{D}}(h) \geq \in$ *have* $error_{S}(h) > 0$.

- *inversely linear in* ϵ
- *logarithmic in* $|H|$
- ϵ *error parameter: D might place low weight on certain parts of the space*
- δ *confidence parameter: there is a small chance the examples we get are not representative of the distribution*

Sample Complexity for Supervised Learning

Theorem: $m \geq \frac{1}{\epsilon}\left[In(|H|) + In\left(\frac{1}{\delta}\right)\right]$ *labeled examples are sufficient so that with prob.* $1-\delta$, *all* $h \in H$ *with* $error_{\mathcal{D}}(h) \geq \epsilon$ *have* $error_S(h) > 0$.

Proof: Assume k bad hypotheses $H_{bad}=\{h_1, h_2, \dots, h_k\}$ with

$$err_{\mathcal{D}}(h_i) \geq \epsilon$$

- *Fix* h_i. *Prob.* h_i *consistent with first training example is* $\leq 1 - \epsilon$. *Prob.* h_i *consistent with first m training examples is* $P(error_S(h) = 0) \leq (1 - \epsilon)^m$.
- *Prob. that at least one* h_i *consistent with first m training examples is* $P(\cup_{h\in \mathrm{H}}\, error_S(h) = 0) \leq k(1 - \epsilon)^m \leq |H|(1 - \epsilon)^m$.
  - This is the probability that there exists consistent yet bad hypothesis. This is a bad situation, we want to avoid and minimize the possibility of this.
- *So, we want* $|H|(1 - \epsilon)^m \leq \delta$

Sample Complexity for Supervised Learning

The probability that there is a hypothesis $h \in H$ that:

1. is Consistent with m examples, and
2. has $Err_D(h) > \epsilon$

is less than $|H|(1-\epsilon)^m$

This situation is a bad one. Let us try to see what we need to do to ensure that this situation is rare.

We want to make this probability small, say smaller than δ

$$|H|(1-\epsilon)^m < \delta$$

$$\log(|H|) + m\log(1-\epsilon) < \log\delta$$

If δ is small, then the probability that there is a consistent, yet bad hypothesis would also be small (because of this inequality)

- Use the fact that $1 - x \leq e^{-x}$, sufficient to set $|H|e^{-\epsilon m} \leq \delta$

$$P(\text{consist}(H_{bad}, D)) \le |H| e^{-\varepsilon m} \le \delta$$

$$e^{-\varepsilon m} \le \frac{\delta}{|H|}$$

$$-\varepsilon m \le \ln\left(\frac{\delta}{|H|}\right)$$

$$m \ge \left(-\ln\frac{\delta}{|H|}\right) / \varepsilon \quad \text{(flip inequality)}$$

$$m \ge \left(\ln\frac{|H|}{\delta}\right) / \varepsilon$$

$$m \ge \left(\ln\frac{1}{\delta} + \ln|H|\right) / \varepsilon$$

Therefore, the probability of getting a bad hypothesis is small, bounded by δ

Consistent hypotheses

The probability that there is a hypothesis $h \in H$ that:

Then, this is improbable

is less than $|H|(1-\epsilon)^m$

We want to make this probability small, say smaller than δ

Then, this holds

$$|H|(1-\epsilon)^m < \delta$$
$$\log(|H|) + m \log(1-\epsilon) < \log \delta$$

We know that $e^{-x} = 1 - x + \frac{x}{2} - \frac{x}{3} \ldots < 1 - x$

Let's use $\log(1-\epsilon) < -\epsilon$ to get a safer δ

That is, if $m > \frac{1}{\epsilon}\left(\ln|H| + \ln\left(\frac{1}{\delta}\right)\right)$ then, the probability of getting a bad hypothesis is small

If this is true

Occam's Razor for consistent hypotheses

Let H be any hypothesis space.

With probability δ, a hypothesis $h \in H$ is consistent yet bad, AND

With probability $1 - \delta$, a hypothesis $h \in H$ that is consistent with a training set of size m will have an error $< \epsilon$ on future examples if

$$m > \boxed{\frac{1}{\epsilon}} \left(\boxed{\ln |H|} + \boxed{\ln \frac{1}{\delta}} \right)$$

1. Expecting lower error increases sample complexity (i.e more examples needed for the guarantee)

2. If we have a larger hypothesis space, then we will make learning harder (i.e higher sample complexity)

3. If we want a higher confidence in the classifier we will produce, sample complexity will be higher.

Sample Complexity: Finite Hypothesis Spaces Realizable Case

PAC: How many examples suffice to guarantee small error whp.
Theorem

$$m \geq \frac{1}{\epsilon}\left[ln(|H|) + ln\left(\frac{1}{\delta}\right)\right]$$

labeled examples are sufficient so that with prob. $1-\delta$, *all* $h \in H$ *with* $err_{\mathcal{D}}(h) \geq \epsilon$ *have* $err_S(h) > 0$.

Statistical Learning Way:

With probability at least $1-\delta$, *all* $h \in H$ *s.t.* $err_S(h) = 0$ *we have*

$$err_D(h) \leq \frac{1}{m}\left[ln(|H|) + ln\left(\frac{1}{\delta}\right)\right]$$

Agnostic Learning: Inconsistent Hypothesis

Assumptions so far:

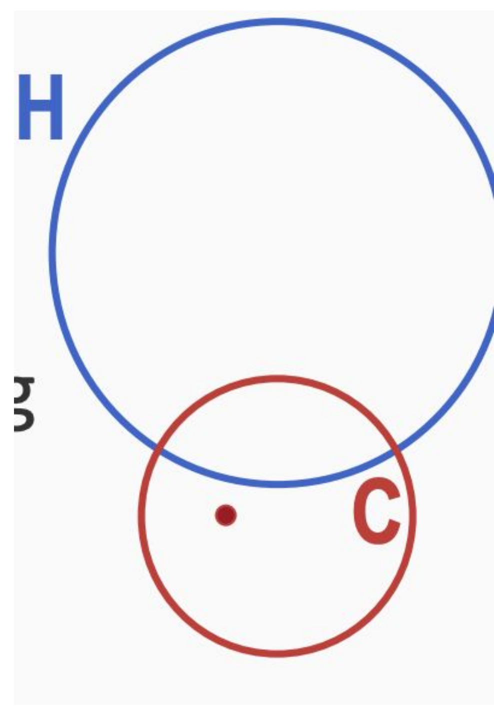
1. *Training and test examples come from the same distribution*
2. *The hypothesis space is finite.*
3. *For any concept, there is some function in the hypothesis space that is consistent with the training set*

What if: We are trying to learn a concept f using hypotheses in H, but $f \notin H$

– That is C is not a subset of H

– This setting is called agnostic learning

– Can we say something about sample complexity?



Agnostic Learning

Are we guaranteed that training error will be zero?
– No. There may be no consistent hypothesis in the hypothesis space!

We can find a classifier $h \in \mathrm{H}$ *that has low* training error $error_S(h)$
-- Can this error tell something about the generalization error $error_D(h)$

Bounding Probabilities

Law of large numbers: As we collect more samples, the empirical average converges to the true expectation

- Suppose we have an unknown coin and we want to estimate its bias (i.e. probability of heads)
- Toss the coin m times

$$\frac{\text{number of heads}}{m} \rightarrow \mathrm{P}(\text{heads})$$

As m increases, we get a better estimate of $\mathrm{P}(\text{heads})$

What can we say about the gap between these two terms?

Hoeffding's Inequality

Upper bounds on how much the sum of a set of random variables differs from its expected value

$$P[p > \bar{p} + \epsilon] \leq e^{-2m\epsilon^2}$$

True mean (Eg. For a coin toss, the probability of seeing heads)

Empirical mean, computed over m independent trials

What this tells us: The empirical mean will not be too far from the expected mean if there are many samples.

And, it quantifies the convergence rate as well.

What we want: To bound sums of random variables – Why? Because the training error depends on the number of errors on the training set

Sample complexity: inconsistent finite $|\mathcal{H}|$

For a single hypothesis to have misleading training error

$$\Pr[error_{\mathcal{D}}(f) \leq \varepsilon + error_S(f)] \leq e^{-2m\varepsilon^2}$$

We want to ensure that the best hypothesis has error bounded in this way

So consider that any one of them could have a large error

$$\Pr[(\exists f \in \mathcal{H}) error_{\mathcal{D}}(f) \leq \varepsilon + error_S(f)] \leq |\mathcal{H}| e^{-2m\varepsilon^2}$$

From this we can derive the bound for the number of samples needed.

$$m \geq \frac{1}{2\varepsilon^2}(\ln|\mathcal{H}| + \ln(\frac{1}{\delta}))$$

Sample Complexity: Finite Hypothesis Spaces

Consistent Case

Theorem

$$m \geq \frac{1}{\epsilon}\left[In(|H|) + In\left(\frac{1}{\delta}\right)\right]$$

labeled examples are sufficient so that with prob. $1-\delta$*, all* $h \in H$ *with* $err_{\mathcal{D}}(h) \geq \epsilon$ *have* $err_S(h) > 0$.

Inconsistent Case

What if there is no perfect h?

Theorem: After m examples, with probability $\geq 1-\delta$*, all* $h \in H$ *have* $|err_{\mathcal{D}}(h) - err_S(h)| < \epsilon$*, for*

$$m \geq \frac{2}{2\epsilon^2}\left[In(|H|) + In\left(\frac{2}{\delta}\right)\right]$$

Sample complexity: example

$\mathcal{C}$: Conjunction of n Boolean literals. Is $\mathcal{C}$ PAC-learnable?

$$|\mathcal{H}| = 3^n$$

$$m \geq \frac{1}{\varepsilon}(n \ \ln 3 + \ln(\frac{1}{\delta}))$$

Concrete examples:

δ=ε=0.05, *n*=10 gives 280 examples

δ=0.01, ε=0.05, *n*=10 gives 312 examples

δ=ε=0.01, *n*=10 gives 1,560 examples

δ=ε=0.01, *n*=50 gives 5,954 examples

Result holds for any consistent learner, such as Find-S.

Sample Complexity of Learning Arbitrary Boolean Functions

Consider any boolean function over n boolean features such as the hypothesis space of DNF or decision trees. There are 2^{2^n} of these, so a sufficient number of examples to learn a PAC concept is:

$$m \geq \frac{1}{\varepsilon}(\ln 2^{2^n} + \ln(\frac{1}{\delta})) = \frac{1}{\varepsilon}(2^n \ln 2 + \ln(\frac{1}{\delta}))$$

$\delta=\varepsilon=0.05$, $n=10$ gives 14,256 examples
$\delta=\varepsilon=0.05$, $n=20$ gives 14,536,410 examples
$\delta=\varepsilon=0.05$, $n=50$ gives $1.561x10^{16}$ examples

PAC Learnability

We impose two limitations

- Polynomial *sample complexity* (information theoretic constraint)
  – Is there enough information in the sample to distinguish a hypothesis h that approximates f ?
- Polynomial *time complexity* (computational complexity)
  – Is there an efficient algorithm that can process the sample and produce a good hypothesis h ?

To be PAC learnable, there must be a hypothesis $h \in H$ with arbitrary small error for every $f \in C$. We assume $H \supseteq C$. (*Properly* PAC learnable if H=C)

Worst Case definition: the algorithm must meet its accuracy

– for every distribution (The distribution free assumption)
– for every target function f in the class C

Questions For Today

1. Given a classifier with zero training error, what can we say about generalization error? (Sample Complexity, Realizable Case)

2. Given a classifier with low training error, what can we say about generalization error? (Sample Complexity, Agnostic Case)

3. Is there a theoretical justification for regularization to avoid overfitting? (Structural Risk Minimization)

Sample Complexity Results

Definition 0.1. The **sample complexity** of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four Cases we care about…

We started with the finite case…

	Realizable	Agnostic
Finite $\|\mathcal{H}\|$		
Infinite $\|\mathcal{H}\|$		

Sample Complexity Results

Definition 0.1. The **sample complexity** of a learning algorithm is the number of examples required to achieve arbitrarily small error (with respect to the optimal hypothesis) with high probability (i.e. close to 1).

Four Cases we care about…

	Realizable	Agnostic
Finite $\|\mathcal{H}\|$	$N \geq \frac{1}{\epsilon}\left[\log(\|\mathcal{H}\|) + \log(\frac{1}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $R(h) \geq \epsilon$ have $\hat{R}(h) > 0$.	
Infinite $\|\mathcal{H}\|$		

Learning Theory Objectives

You should be able to…

- Identify the properties of a learning setting and assumptions required to ensure low generalization error
- Distinguish true error, train error, test error
- Define PAC and explain what it means to be approximately correct and what occurs with high probability
- Apply sample complexity bounds to real-world learning examples
- Distinguish between a large sample and a finite sample analysis
- Theoretically motivate regularization

Example: Conjunctions

In-Class Quiz:

Suppose H = class of conjunctions over **x** in $\{0,1\}^M$

If M = 10, ϵ = 0.1, δ = 0.01, how many examples suffice?

	Realizable	Agnostic
Finite $\|\mathcal{H}\|$	$N \geq \frac{1}{\epsilon}\left[\log(\|\mathcal{H}\|) + \log(\frac{1}{\delta})\right]$ labeled examples are sufficient so that with probability $(1-\delta)$ all $h \in \mathcal{H}$ with $R(h) \geq \epsilon$ have $\hat{R}(h) > 0$.	
Infinite $\|\mathcal{H}\|$		

Concept Learning Task

"Days in which Aldo enjoys swimming"

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

- Hypothesis Representation: Conjunction of constraints on the 6 instance attributes
  - "?" : any value is acceptable
  - specify a single required value for the attribute
  - "∅" : that no value is acceptable

Concept Learning

h = (?, Cold, High, ?, ?, ?)

indicates that Aldo enjoys his favorite sport on cold days with high humidity

Most general hypothesis: (?, ?, ?, ?, ?, ?)

Most specific hypothesis: (∅, ∅, ∅, ∅, ∅, ∅)

Find-S Algorithm

1. *Initialize h to the most specific hypothesis in $\mathcal{H}$*
2. *For each positive training instance x*

   For each attribute constraint a_i in *h*

   IF the constraint a_i in *h* is satisfied by *x*

   THEN do nothing

   ELSE replace a_i in h by next more general constraint satisfied by *x*
3. *Output hypothesis h*

Concept Learning

Example	Sky	AirTemp	Humidity	Wind	Water	Forecast	EnjoySport
1	Sunny	Warm	Normal	Strong	Warm	Same	Yes
2	Sunny	Warm	High	Strong	Warm	Same	Yes
3	Rainy	Cold	High	Strong	Warm	Change	No
4	Sunny	Warm	High	Strong	Cool	Change	Yes

Finding a Maximally Specific Hypothesis

Find-S Algorithm

$h_1 \leftarrow (\varnothing, \varnothing, \varnothing, \varnothing, \varnothing, \varnothing)$
$h_2 \leftarrow$ (Sunny, Warm, Normal, Strong, Warm, Same)
$h_3 \leftarrow$ (Sunny, Warm, ?, Strong, Warm, Same)
$h_4 \leftarrow$ (Sunny, Warm, ?, Strong, ?, ?)

Back