SHIRAZ UNIVERSITY
Computer Science and Engineering Department
Machine Learning Lab

Learning Theory

Sattar Hashemi

Based on:

Mitchell, Tom M. "Machine learning, Chapter 7
NG, Andrew. Machine learning Lecture Notes, Learning Theory
Rudin, Walter. "Principles of Mathematical Analysis." (1976).
Vapnik, Vladimir. "Statistical learning theory" (1998).

Inductive Learning

Definition [1]

The inductive learning hypothesis: Any hypothesis found to approximate the target function well over a sufficiently large set of training examples will also approximate the target function well over other unobserved examples.

- The problem of **inducing** general functions from specific training examples is central to learning
- Human as a Inductive Learner:
  - Concept Learning: Example:
    - Game : (Football,+),(Chess,+),(Teaching,-),(Blackjack,+),(Business, -)
    - Test: (Language,?)
- Machine as a inductive Learner: Algorithms:
  - Concept learning: FIND-S, CANDIDATE-ELIMINATE
  - Other supervised learning: SVM algorithm, Perceptron algorithm

*Based on[1]: Mitchel, Tom M. “Machine Learning. WCB.” (1997)

Function Approximation Review

Function Approximation: Compute a function that hopes to interpolate or generalize from the training patterns.

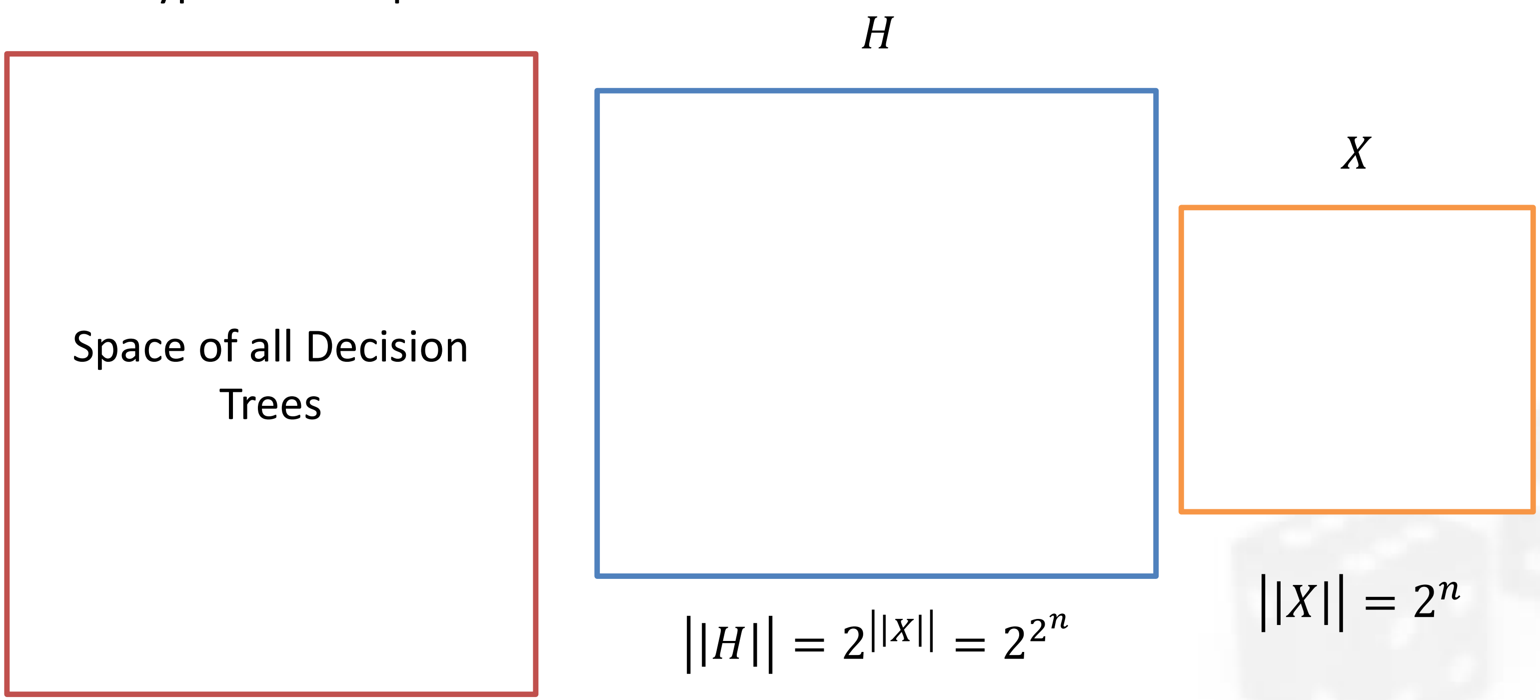
- **Input space** (X): set of all instances can be presented specified by X
- **Target Function**(c): a **Boolean-valued** function $c: X \rightarrow \{0,1\}$
- **Hypothesis**(h): function $h: X \rightarrow \{0,1\}$ that is approximated target function
- **Hypothesis space**(H): $H = \{h |\ h: X \rightarrow \{0,1\} \wedge described\ by\ learner\}$
- **Training examples**(D): **sequence** of $\big(x, c(x)\big)$ by which Learner approximate c

- **Hypothesis Representor :** MLP, Logistic Regression, Decision Tree
- Input space can be **continues** or **discrete** :
- Example :
  - $\mathrm{X} = \{x | x \in \{0,1\}^n\}$
  - $\mathrm{X} = \{x | x \in N^n\}$
  - $\mathrm{X} = \{x | x \in R^n\}$

PAC Learnability

Informal Example

- **Simplest dataset ever!**
  - $X = \{x \mid x \in \{0,1\}^n\}$
  - $c: \{0,1\}^n \rightarrow \{0,1\}$
  - Hypothesis representor: decision tree



PAC assumption

- $\boldsymbol{X}$ and $\boldsymbol{H}$ are given.
- Instances $\boldsymbol{x}$ are drawn from distribution $\boldsymbol{\mathcal{D}}$
- Teacher provides target value for each $\boldsymbol{x}$ **determinately** (without any noise in labeling)
- Learner must output a hypothesis $\boldsymbol{h}$ estimating $\boldsymbol{c}$
- $\boldsymbol{h}$ is evaluating by its performance on subsequence instances drawn according to $\boldsymbol{\mathcal{D}}$

- for sake of simplicity we add this assumption too
  - $\boldsymbol{c}$ is Boolean-valued function $\boldsymbol{c}: \boldsymbol{X} \rightarrow \{\mathbf{0}, \mathbf{1}\}$
  - Noise free classification: training instances is sampled independently from $\boldsymbol{\mathcal{D}}$ without noise.

PAC Learnability

PAC Learning

- **Computational complexity***:
  how much computational effort is needed for a learner to converge (with high probability) to a successful learner?

- **Sample complexity**:
  How many training examples are needed for a leaner to converge (with high probability) to a successful hypothesis?

PAC Learnability

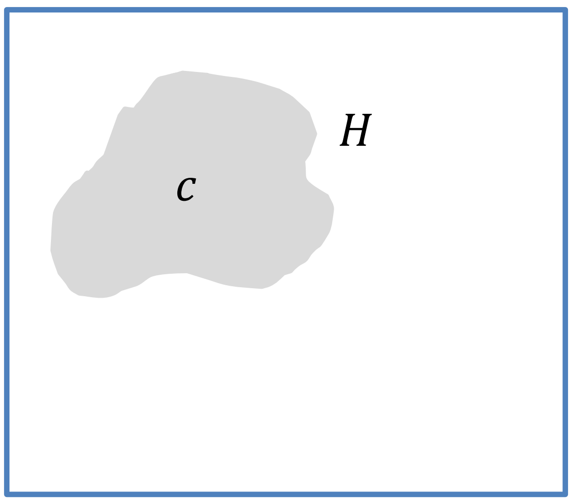
*Computational complexity is the main contribution of PAC learnability from the computer scientific view.

Successful learner : Informal Definition

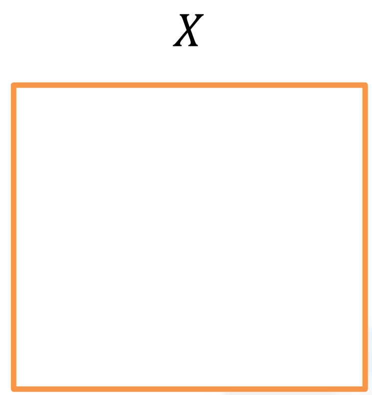
- $X = \{x | x \in \{0,1\}^n\}$
- $c: \{0,1\}^n \to \{0,1\}$

- Unrealistic definition:

  The learner is successful if it gives **one hypothesis** h with $error_{\mathcal{D}}(h) = 0$



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



$$||X|| = 2^n$$

Successful learner

- **Two relaxations:**
  - ✓ we want hypothesis that its true error is bound by small constant ϵ (**approximately correct** part)

    $$error_{\mathcal{D}}(h) < \epsilon$$

  - ✓ We want to achieve above hypothesis, the probability of success would be at least $1 - \delta$ (**probability** part)

- In short, we require the learner **probably** learns a hypothesis that is **approximately correct**

PAC Learnability

PAC Learnability

- **Input measure:**
  - Complexity measure of input space? $\boldsymbol{n}$
  - How good should be trained hypothesis? $\frac{1}{\epsilon}$
  - Probability of success? $\frac{1}{\delta}$

- **PAC Learnability** *
  - Running time of learner algorithm with PAC assumptions: $T_{PAC}\left(n, \frac{1}{\epsilon}, \frac{1}{\delta}\right)$
  - $\forall c \in C, \mathcal{D}, \epsilon: 0 < \epsilon < \frac{1}{2}, \delta: 0 < \delta < \frac{1}{2}$

$$T_{PAC}\left(n, \frac{1}{\epsilon}, \frac{1}{\delta}\right) \in O\left(p\left(n, \frac{1}{\epsilon}, \frac{1}{\delta}\right)\right) \quad s.t.\ p(.,.,.) is\ a\ polynomial\ function$$

*[3],[4]based on Haussler, David, “Overview of the Probably Approximately Correct(PAC) Learning Framework” and Valiant, Leslie G. "A theory of the learnable."

PAC Learnability

Version Space

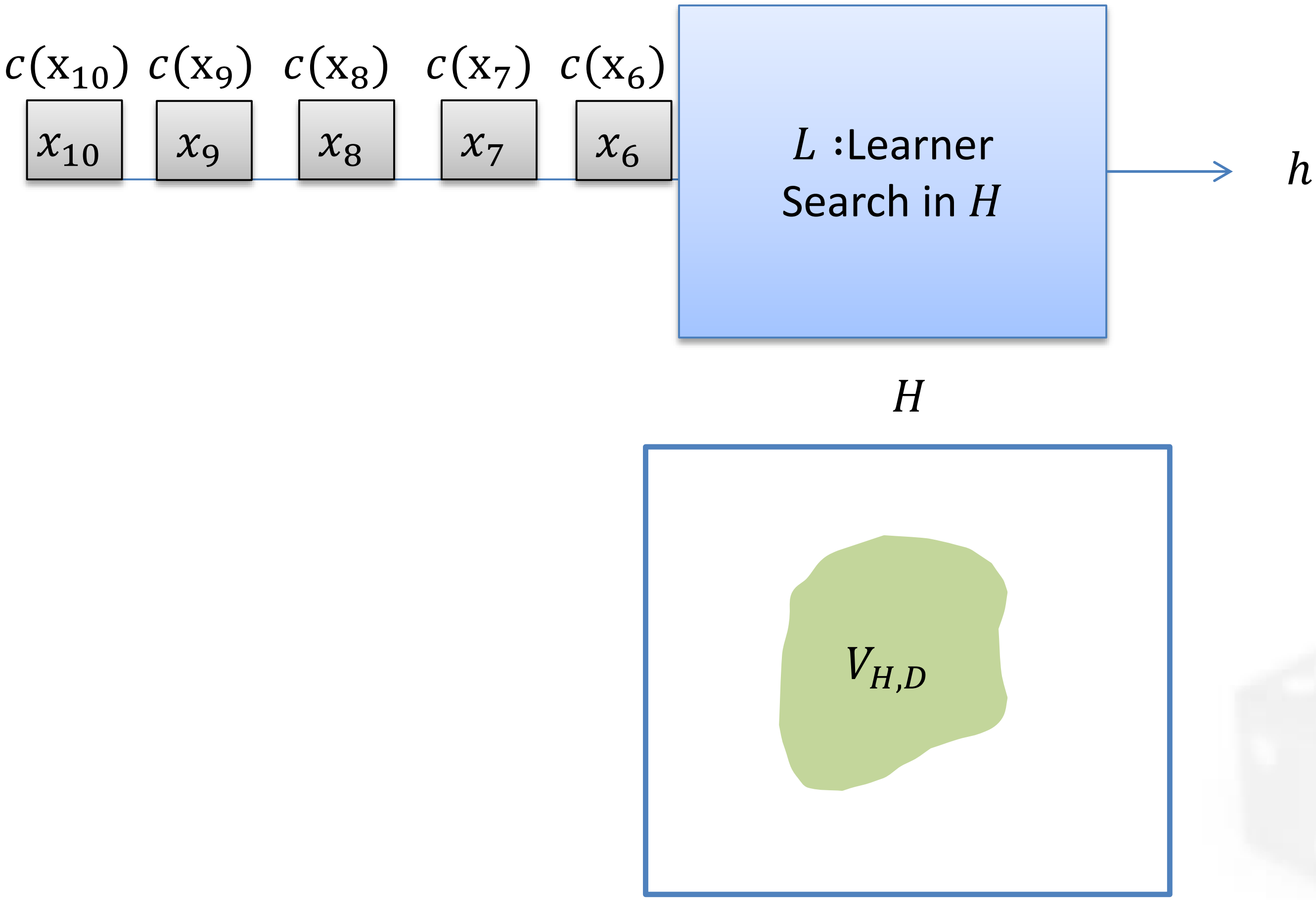
Definition: A hypothesis h is consistent with a sequence of training examples D of target concept c, **if and only if** $\forall\left(x, c(x)\right) \in D \quad h(x) = c(x)$

$$Consistent(h, D) \equiv \left(\forall\left(x, c(x)\right) \in D\right) \quad h(x) = c(x)$$

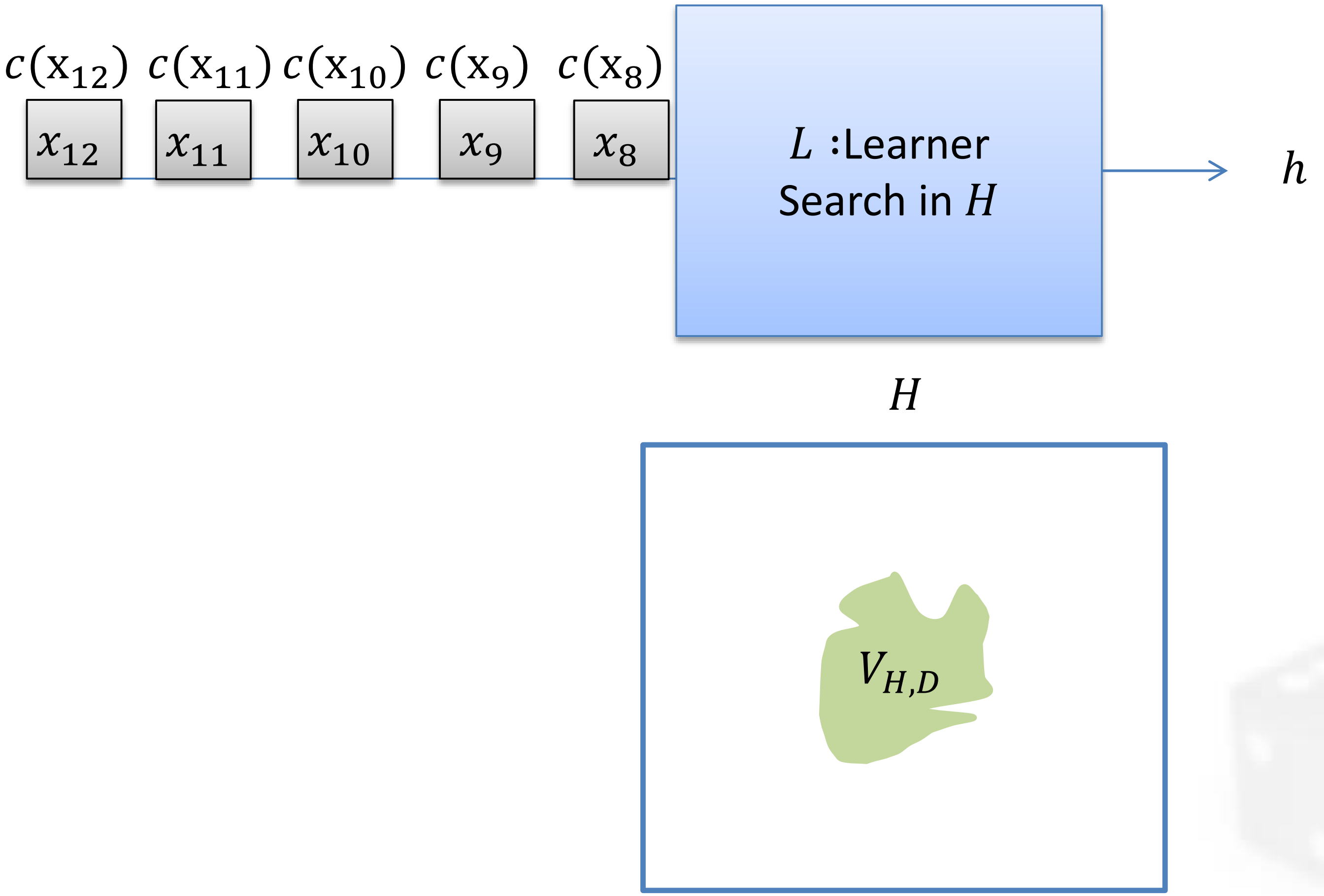
Definition: The version Space $VS_{H,D}$ with respect to Hypothesis Space H training example D, is the subset of Hypothesis from H consistent with all training examples in D

$$VS_{H,D} \equiv \{h \in H | Consistent\ (h, D)\}$$

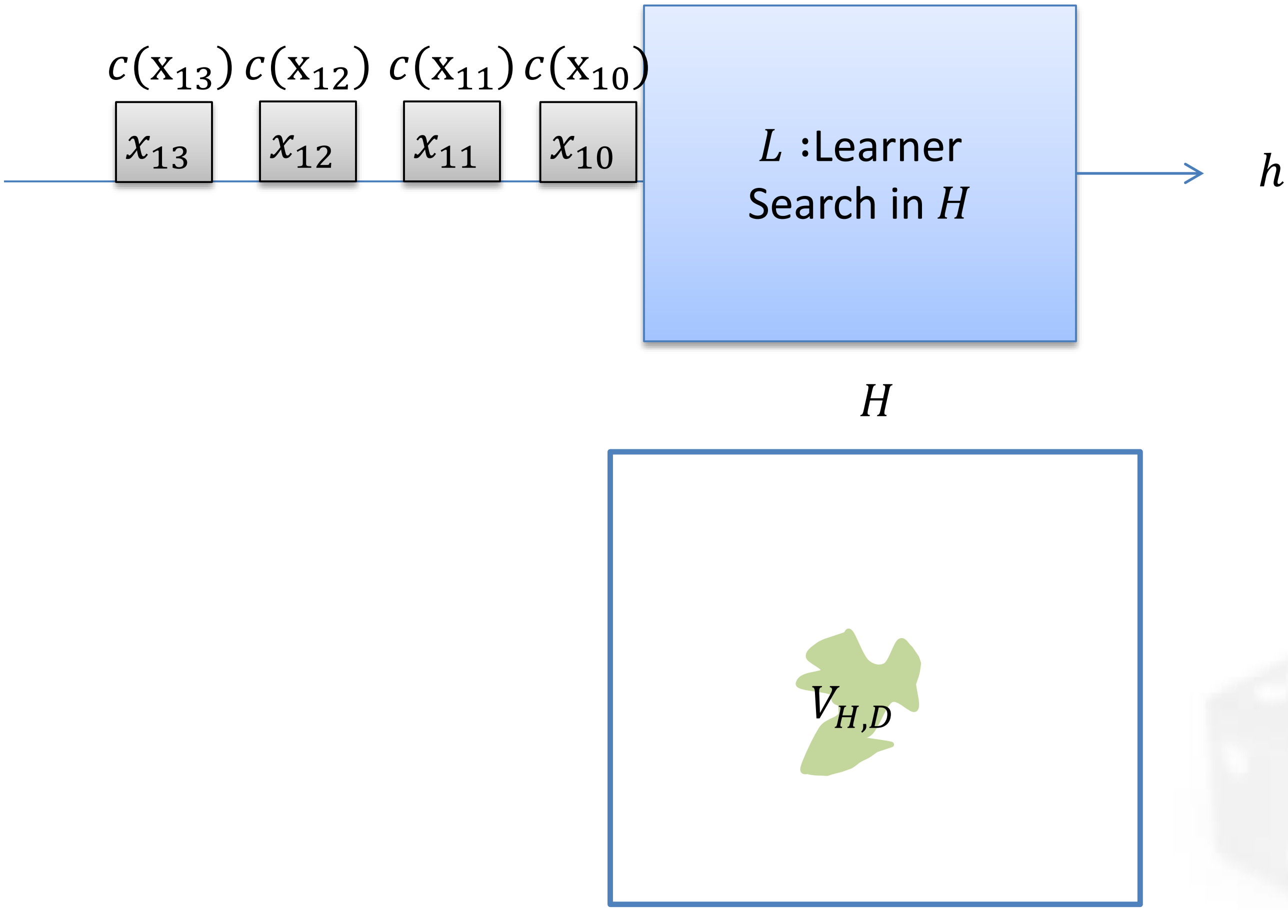
Mitchell's point of view



Mitchell's point of view



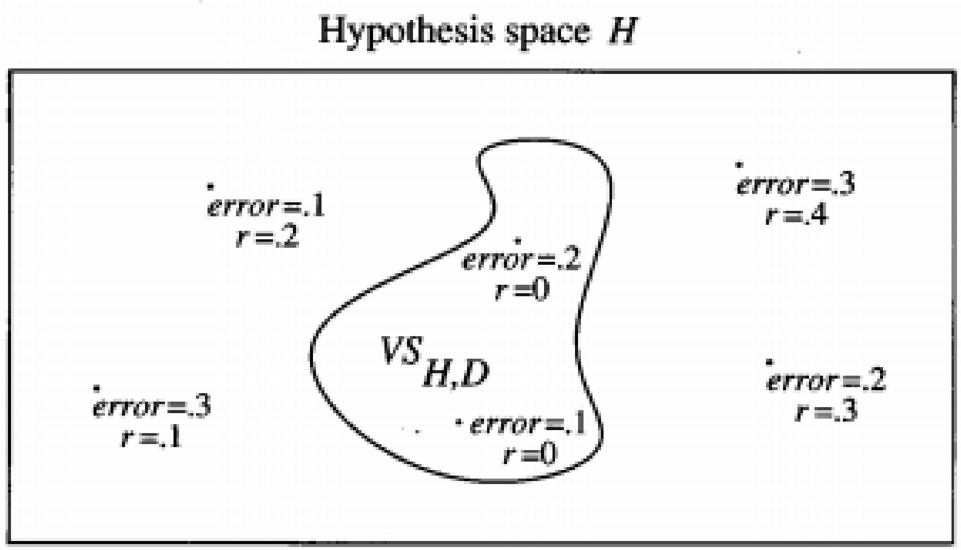
Mitchell’s point of view



Exhausting The Version Space

Definition: The version Space $VS_{H,D}$ is said to be $\epsilon - \boldsymbol{exhausted_{c,D}}$ if every hypothesis h in $VS_{H,D}$ has true error less than ϵ with respect to c and D

$$(h \in VS_{H,D}) \quad error_{\mathcal{D}}(h) < \epsilon$$



Haussler Theorem

Theorem: [Haussler 1988]
if the Hypothesis space H is **finite,** and D is a sequence of $m \geq 1$ independent random examples of some target concept c, then for any $0 \leq \epsilon \leq 1$, the probability that the $VS_{H,D}$ is **not** $\epsilon - \boldsymbol{exhasetd_{c,D}}$ is less than $|\boldsymbol{H}|\boldsymbol{e}^{-\epsilon m}$

This bounds the probability that any consistent learner will output a hypothesis $\boldsymbol{h}$ with $\boldsymbol{error_{\mathcal{D}}} \geq \boldsymbol{\epsilon}$

Proof

Sample Complexity for Finite H

$\{h_1, h_2, \dots, h_k\} \subset H \quad s.t. \forall i \quad error_{\mathcal{D}}(h_i) > \epsilon$

$\forall h \in H \quad P(\neg Consistent(h, \{x_1\}) = \left(error_{\mathcal{D}}(h)\right)$

$\forall i \in k \quad P(Consistent(h_i, \{x_1\}) \leq (1 - \epsilon)$

$\forall i \in k \quad P(Consistent(h_i, \{x_1, x_2, \dots . x_m\}) \leq (1 - \epsilon)^m$

$P(Consistent(\{h_1, h_2, \dots, h_k\}, \{x_1, x_2, \dots . x_m\}) \leq ?$

$Lemma\ (The\ union\ bound)$: *Let* $A_1, A_2, \dots . A_k$ *be* k *deferent events (that may be not independent) then*

$P(A_1 \cup A_2 \cup \ \dots A_k) \leq P(A_1) + P(A_2) + \cdots + P(A_3)$

Proof

$\{h_1, h_2, \dots, h_k\} \subset H \quad s.t. \forall i \quad error_{\mathcal{D}}(h_i) > \epsilon$

$\forall h \in H \quad P(\neg Consistent(h, \{x_1\}) = \left(error_{\mathcal{D}}(h)\right)$
$\forall i \in k \quad P(Consistent(h_i, \{x_1\}) \leq (1 - \epsilon)$
$\forall i \in k \quad P(Consistent(h_i, \{x_1, x_2, \dots . x_m\}) \leq (1 - \epsilon)^m$

$P(Consistent(\{h_1, h_2, \dots, h_k\}, \{x_1, x_2, \dots . x_m\}) \leq ?$

Lemma (The union bound): Let $A_1, A_2, \dots . A_k$ be k deferent events (that may be not independent) then
$P(A_1 \cup A_2 \cup \ \dots A_k) \leq P(A_1) + P(A_2) + \cdots + P(A_3)$

$P(Consistent(\{h_1, h_2, \dots, h_k\}, \{x_1, x_2, \dots . x_m\}) \leq k(1 - \epsilon)^m$
$k(1 - \epsilon)^m \leq |H| \, (1 - \epsilon)^m \leq |H| e^{-\epsilon m}$

$* \forall \ 0 < \epsilon < 1 \qquad 1 - \epsilon < e^{-\epsilon}$

Sample Complexity

Sample Complexity for Finite H

probability that the version space is not $\epsilon - exhauseted$ after m training examples is at most $|\boldsymbol{H}|\boldsymbol{e}^{-\boldsymbol{\epsilon m}}$

$$P[(\exists h \in H)\ s.t.(error_D(h) = 0) \wedge (error_{\mathcal{D}}(h) > \epsilon)] \leq |H|e^{-\epsilon m}$$

Suppose we want this probability at most δ

1. How many training example suffice?

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

2. If $error_D(h) = 0$ then with probability at least $1 - \delta$

$$error_{\mathcal{D}}(h) \leq \frac{1}{m}(\ln(|H|) + \ln(\frac{1}{\delta}))$$

H is Conjunction of Boolean Literals

consider classification problem:

- instances $X = (X_1\ X_2\ X_3\ X_4\ )$ where each X_i is Boolean.
- Learned hypothesis are rules of the form
  - If $(X_1\ X_2\ X_3\ X_4\ ) = (0, ?\ , 1, ?\ )$ then Class=1 else Class=0
  - i.e. rules constrain any subset of the X_i

How many training examples m suffice to assure that with probability at least 0.99, any consistent learner will output a hypothesis with true error at most 0.05?

H is Conjunction of Boolean Literals

consider classification problem:

- instances $X = (X_1\ X_2\ X_3\ X_4\ )$ where each X_i is Boolean.
- Learned hypothesis are rules of the form
  - If $(X_1\ X_2\ X_3\ X_4\ ) = (0, ?\,, 1, ?\,)$ then Class=1 else Class=0
  - i.e. rules constrain any subset of the X_i

How many training examples m suffice to assure that with probability at least 0.99, any consistent learner will output a hypothesis with true error at most 0.05?

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

$$m \geq \frac{1}{0.05}\left(\ln(3^4) + \ln\left(\frac{1}{0.01}\right)\right)$$

Sample Complexity for Finite H

PAC learnability of Conjunction of Boolean Literals

consider classification problem:

- instances $X = (X_1\ X_2\ X_3\ X_4\ \ldots X_n)$ where each X_i is Boolean.
- Learned hypothesis are rules of the form
  - Some rule…
  - i.e. rules constrain any subset of the X_i
- If $c \in H$ then $m \in O\left(\frac{1}{\epsilon}, \log\left(\frac{1}{\delta}\right)\right)$

How many training examples m suffice to assure that with probability at least $1 - \delta$, any consistent learner will output a hypothesis with true error at most ϵ?

$$m \geq \frac{1}{\epsilon}\left(\ln(3^n) + \ln\left(\frac{1}{\delta}\right)\right)$$

$m \in O(n) \rightarrow$ Conjunction of Boolean Literals is **PAC Learnable**

(7.7)H is Decision tree with depth 2

consider classification problem:

- instances $X = (X_1\ X_2\ X_3\ X_4\ )$ where each X_i is Boolean.
  - Learned hypothesis are decision trees of depth 2, using only two variables

How many training examples m suffice to assure that with probability at least 0.99, any consistent learner will output a hypothesis with true error at most 0.05?