Computational Learning Theory

Sinh Hoa Nguyen, Hung Son Nguyen

Polish-Japanese Institute of Information Technology
Institute of Mathematics, Warsaw University

February 14, 2006

Outline

1. Introduction
2. PAC Learning
3. VC Dimension

Computational Learning Theory

What general laws constrain inductive learning?

We seek theory to relate:

- Probability of successful learning
- Number of training examples
- Complexity of hypothesis space
- Accuracy to which target concept is approximated
- Manner in which training examples presented

$$1^2 + 2^2 + ... + n^2 = \frac{n(n+1)(2n+1)}{6}$$

Prototypical Concept Learning Task

- **Given:**
  - Instances X: Possible days, each described by the attributes *Sky, AirTemp, Humidity, Wind, Water, Forecast*
  - Target function c: $EnjoySport : X \to \{0,1\}$
  - Hypotheses H: Conjunctions of literals. E.g.

$$\langle ?, Cold, High, ?, ?, ? \rangle.$$

  - Training examples D: Positive and negative examples of the target function

$$\langle x_1, c(x_1) \rangle, \ldots \langle x_m, c(x_m) \rangle$$

- **Determine:**
  - A hypothesis h in H such that $h(x) = c(x)$ for all x in D?
  - A hypothesis h in H such that $h(x) = c(x)$ for all x in X?

Sample Complexity

How many training examples are sufficient to learn the target concept?

1. If learner proposes instances, as queries to teacher
   - Learner proposes instance x, teacher provides $c(x)$
2. If teacher (who knows c) provides training examples
   - teacher provides sequence of examples of form $\langle x, c(x) \rangle$
3. If some random process (e.g., nature) proposes instances
   - instance x generated randomly, teacher provides $c(x)$

Sample Complexity: 1

Learner proposes instance x, teacher provides $c(x)$
(assume c is in learner's hypothesis space H)

Optimal query strategy: play 20 questions

- pick instance x such that half of hypotheses in VS classify x positive, half classify x negative
- When this is possible, need $\lceil \log_2 |H| \rceil$ queries to learn c
- when not possible, need even more

Sample Complexity: 2

Teacher (who knows c) provides training examples
(assume c is in learner's hypothesis space H)

Optimal teaching strategy: depends on H used by learner

Consider the case $H =$ conjunctions of up to n boolean literals and their negations

> e.g., $(AirTemp = Warm) \wedge (Wind = Strong)$, where $AirTemp, Wind, \ldots$ each have 2 possible values.

- if n possible boolean attributes in H, $n + 1$ examples suffice
- why?

Sample Complexity: 3

Given:

- set of instances X
- set of hypotheses H
- set of possible target concepts C
- training instances generated by a fixed, unknown probability distribution $\mathcal{D}$ over X

Learner observes a sequence D of training examples of form $\langle x, c(x) \rangle$, for some target concept $c \in C$

- instances x are drawn from distribution $\mathcal{D}$
- teacher provides target value $c(x)$ for each

Learner must output a hypothesis h estimating c

- h is evaluated by its performance on subsequent instances drawn according to $\mathcal{D}$

Note: probabilistic instances, noise-free classifications

True Error of a Hypothesis

Definition

The **true error** (denoted $error_{\mathcal{D}}(h)$) of hypothesis h with respect to target concept c and distribution $\mathcal{D}$ is the probability that h will misclassify an instance drawn at random according to $\mathcal{D}$.

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

With probability $(1-\varepsilon)$ one can estimate

$$|er^c_{\mathcal{D}} - er^c_D| \leqslant s_{\frac{\varepsilon}{2}} \sqrt{\frac{er^c_D(1-er^c_D)}{|D|}}$$

Two Notions of Error

Training error of hypothesis h with respect to target concept c

- How often $h(x) \neq c(x)$ over training instances

True error of hypothesis h with respect to c

- How often $h(x) \neq c(x)$ over future random instances

Our concern:

- Can we bound the true error of h given the training error of h?
- First consider when training error of h is zero (i.e., $h \in VS_{H,D}$)

No Free Lunch Theorem

No search or learning algorithm can be the best on all possible learning or optimization problems.

- In fact, every algorithm is the best algorithm for the same number of problems.
- But only some problems are of interest.
- For example: a random search algorithm is perfect for a completely random problem (the "white noise" problem), but for any search or optimization problem with structure, random search is not so good.

Outline

Exhausting the Version Space

Definition

The version space $VS_{H,D}$ is said to be ε-**exhausted** with respect to c and $\mathcal{D}$, if every hypothesis h in $VS_{H,D}$ has error less than ε with respect to c and $\mathcal{D}$.

$$(\forall h \in VS_{H,D})\ error_{\mathcal{D}}(h) < \varepsilon$$

How many examples will ε-exhaust the VS?

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 $0 \leq \varepsilon \leq 1$, the probability that the version space with respect to H and D is not ε-exhausted (with respect to c) is less than

$$|H|e^{-\varepsilon m}$$

- Interesting! This bounds the probability that any consistent learner will output a hypothesis h with $error(h) \geq \varepsilon$
- If we want to this probability to be below δ

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

then

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

Learning Conjunctions of Boolean Literals

How many examples are sufficient to assure with probability at least $(1-\delta)$ that

every h in $VS_{H,D}$ satisfies $error_{\mathcal{D}}(h) \leq \varepsilon$

Use our theorem:

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

Suppose H contains conjunctions of constraints on up to n boolean attributes (i.e., n boolean literals). Then $|H| = 3^n$, and

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

or

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

How About $EnjoySport$?

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

If H is as given in $EnjoySport$ then $|H| = 973$, and

$$m \geq \frac{1}{\varepsilon}(\ln 973 + \ln(1/\delta))$$

... if want to assure that with probability 95%, VS contains only hypotheses with $error_{\mathcal{D}}(h) \leq .1$, then it is sufficient to have m examples, where

$$m \geq \frac{1}{.1}(\ln 973 + \ln(1/.05))$$

$$m \geq 10(\ln 973 + \ln 20)$$

$$m \geq 10(6.88 + 3.00)$$

$$m \geq 98.8$$

PAC Learning

Consider a class C of possible target concepts defined over a set of instances X of length n, and a learner L using hypothesis space H.

Definition

C is **PAC-learnable** by L using H if for all $c \in C$, distributions $\mathcal{D}$ over X, ε such that $0 < \varepsilon < 1/2$, and δ such that $0 < \delta < 1/2$, learner L will with probability at least $(1 - \delta)$ output a hypothesis $h \in H$ such that $error_{\mathcal{D}}(h) \leq \varepsilon$, in time that is polynomial in $1/\varepsilon$, $1/\delta$, n and $size(c)$.

Example

- Unbiased learner: $|H| = 2^{2^n}$

$$\begin{aligned} m &\geq \frac{1}{\varepsilon}(\ln|H| + \ln(1/\delta)) \\ &\geq \frac{1}{\varepsilon}(2^n \ln 2 + \ln(1/\delta)) \end{aligned}$$

Example

- Unbiased learner: $|H| = 2^{2^n}$

$$
\begin{aligned}
m &\geq \frac{1}{\varepsilon}(\ln|H| + \ln(1/\delta)) \\
&\geq \frac{1}{\varepsilon}(2^n \ln 2 + \ln(1/\delta))
\end{aligned}
$$

- k–term DNF:

$$T_1 \vee T_2 \vee \ldots \vee T_k$$

We have $|H| \leq (3^n)^k$, thus

$$
\begin{aligned}
m &\geq \frac{1}{\varepsilon}(\ln|H| + \ln(1/\delta)) \\
&\geq \frac{1}{\varepsilon}(kn \ln 3 + \ln(1/\delta))
\end{aligned}
$$

So are k-DNFs PAC learnable?

Agnostic Learning

So far, assumed $c \in H$

Agnostic learning setting: don't assume $c \in H$

- What do we want then?
  - The hypothesis h that makes fewest errors on training data
- What is sample complexity in this case?

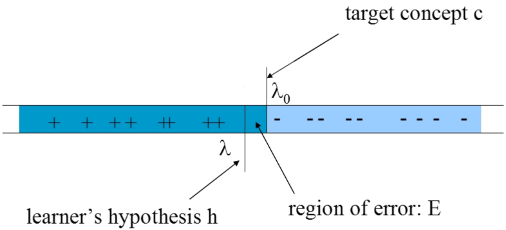
$$m \geq \frac{1}{2\varepsilon^2}(\ln |H| + \ln(1/\delta))$$

derived from Hoeffding bounds:

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

Outline

Discretization problem



- $er^c_D = \mu((\lambda, \lambda_0])$
- Let $\beta_0 = \sup\{\beta | \mu((\beta, \lambda_0]) < \varepsilon\}$. then $er^c_{\mathcal{D}}(f_{\lambda^*}) \leqslant \varepsilon \Leftrightarrow \lambda^* \leqslant \beta_0 \Leftrightarrow$ there exists an instance x_i which is belonging to $[\lambda_0, \beta_0]$;
- The probability that there is no instance that belongs to $[\beta, \lambda_0]$ is equal to $\leqslant (1-\varepsilon)^m$. Hence

$$\mu^m\{D \in \mathcal{S}(m, f_{\lambda_0}) | er_{\mathcal{D}}(L(D)) \leqslant \varepsilon\} \geqslant 1 - (1-\varepsilon)^m$$

- This probability is $> 1 - \delta$ if $m \geqslant m_0 = \left\lceil \frac{1}{\varepsilon} \ln \frac{1}{\delta} \right\rceil$

Shattering a Set of Instances

Definition

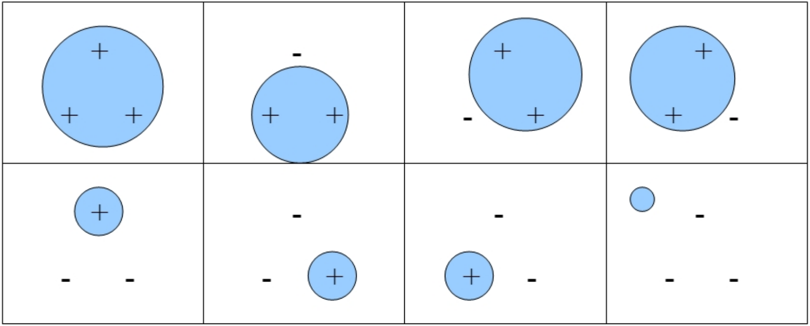
Definition: a **dichotomy** of a set S is a partition of S into two disjoint subsets.

Definition

A set of instances S is **shattered** by hypothesis space H if and only if for every dichotomy of S there exists some hypothesis in H consistent with this dichotomy.

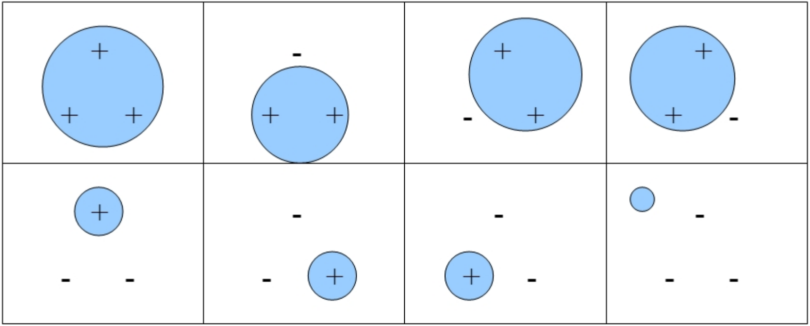
Three Instances Shattered

- Let $S = \{x_1, x_2, ...x_m\} \subset X$.



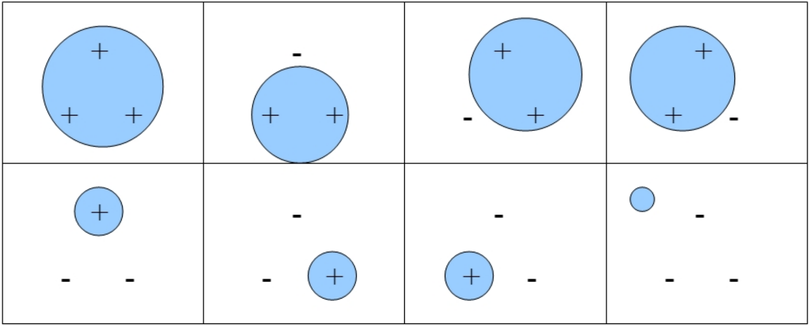
Three Instances Shattered

- Let $S = \{x_1, x_2, ...x_m\} \subset X$.
- Let $\Pi_{\mathbb{H}}(S) = |\{(h(x_1), ..., h(x_m)) \in \{0,1\}^m : h \in H\}| \leq 2^m$



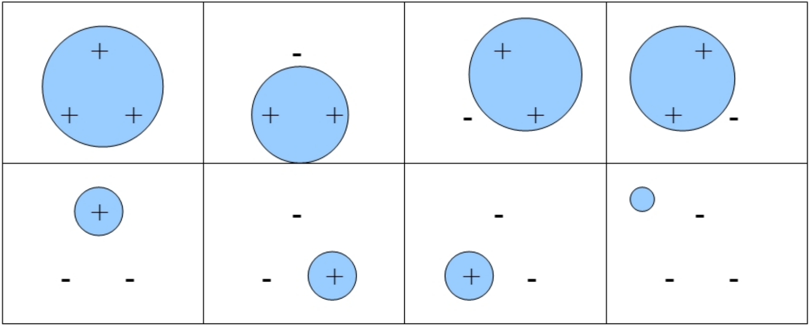
Three Instances Shattered

- Let $S = \{x_1, x_2, ...x_m\} \subset X$.
- Let $\Pi_{\mathbb{H}}(S) = |\{(h(x_1), ..., h(x_m)) \in \{0,1\}^m : h \in H\}| \leq 2^m$
- If $\Pi_{\mathbb{H}}(S) = 2^m$ then we say H shatters S.



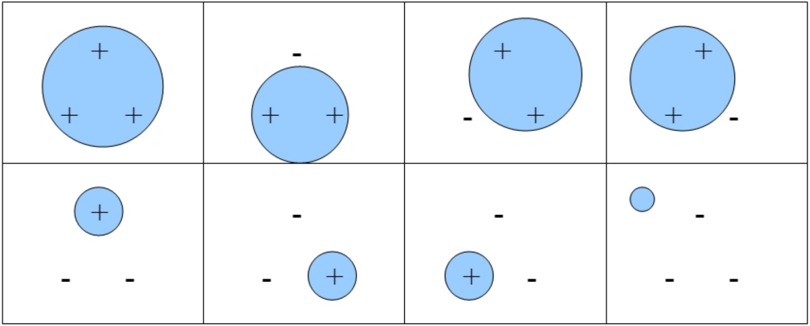
Three Instances Shattered

- Let $S = \{x_1, x_2, ...x_m\} \subset X$.
- Let $\Pi_{\mathbb{H}}(S) = |\{(h(x_1), ..., h(x_m)) \in \{0,1\}^m : h \in H\}| \leq 2^m$
- If $\Pi_{\mathbb{H}}(S) = 2^m$ then we say H shatters S.
- Let $\Pi_{\mathbb{H}}(m) = \max_{S \in X^m} \Pi_{\mathbb{H}}(S)$



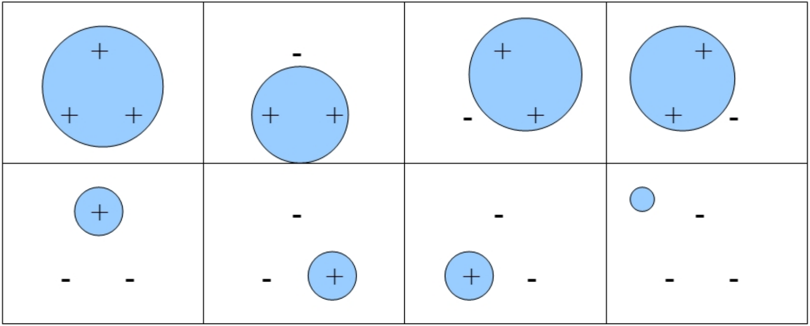
Three Instances Shattered

- Let $S = \{x_1, x_2, ... x_m\} \subset X$.
- Let $\Pi_{\mathbb{H}}(S) = |\{(h(x_1), ..., h(x_m)) \in \{0,1\}^m : h \in H\}| \leq 2^m$
- If $\Pi_{\mathbb{H}}(S) = 2^m$ then we say H shatters S.
- Let $\Pi_{\mathbb{H}}(m) = \max_{S \in X^m} \Pi_{\mathbb{H}}(S)$
- In previous example (space of radiuses) $\Pi_{\mathbb{H}}(m) = m + 1$.



Three Instances Shattered

- Let $S = \{x_1, x_2, ...x_m\} \subset X$.
- Let $\Pi_{\mathbb{H}}(S) = |\{(h(x_1), ..., h(x_m)) \in \{0,1\}^m : h \in H\}| \leq 2^m$
- If $\Pi_{\mathbb{H}}(S) = 2^m$ then we say H shatters S.
- Let $\Pi_{\mathbb{H}}(m) = \max_{S \in X^m} \Pi_{\mathbb{H}}(S)$
- In previous example (space of radiuses) $\Pi_{\mathbb{H}}(m) = m + 1$.
- In general it is hard to find a formula for $\Pi_{\mathbb{H}}(m)$!!!



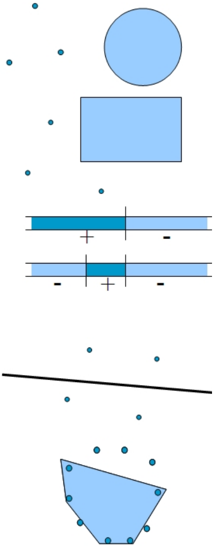
The Vapnik-Chervonenkis Dimension

Definition

The **Vapnik-Chervonenkis dimension**, $VC(H)$, of hypothesis space H defined over instance space X is the size of the largest finite subset of X shattered by H. If arbitrarily large finite sets of X can be shattered by H, then $VC(H) \equiv \infty$.

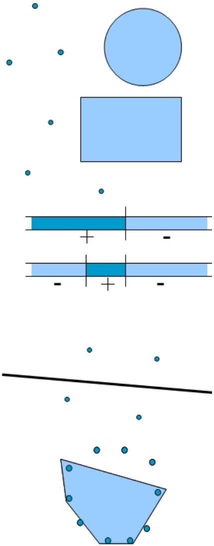
Examples of VC Dim

- $H = \{circles...\} \implies VC(H) = 3$



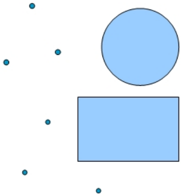
Examples of VC Dim

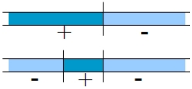
- $H = \{circles...\} \implies VC(H) = 3$
- $H = \{rectangles...\} \implies VC(H) = 4$

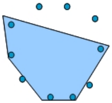


Examples of VC Dim

- $H = \{circles...\} \implies VC(H) = 3$
- $H = \{rectangles...\} \implies VC(H) = 4$
- $H = \{threshold\ functions...\} \implies$
  $VC(H) = 1$ if $+$ is always on the left;
  $VC(H) = 2$ if $+$ can be on left or right

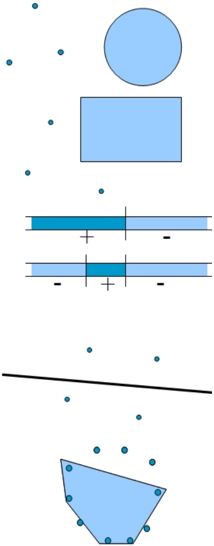






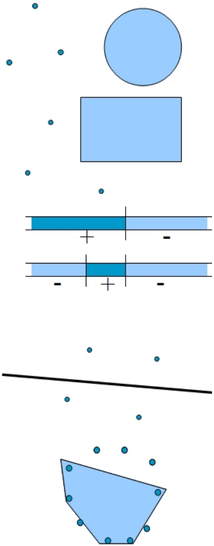
Examples of VC Dim

- $H = \{circles...\} \implies VC(H) = 3$
- $H = \{rectangles...\} \implies VC(H) = 4$
- $H = \{threshold\ functions...\} \implies$
  $VC(H) = 1$ if $+$ is always on the left;
  $VC(H) = 2$ if $+$ can be on left or right
- $H = \{intervals...\} \implies$
  VC(H) $= 2$ if $+$ is always in center
  VC(H) $= 3$ if center can be $+$ or -



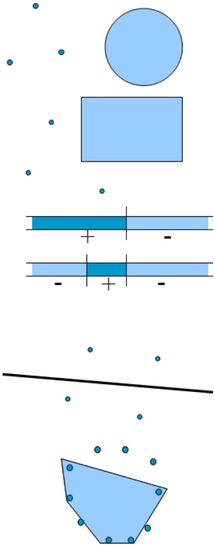
Examples of VC Dim

- $H = \{circles...\} \implies VC(H) = 3$
- $H = \{rectangles...\} \implies VC(H) = 4$
- $H = \{threshold\ functions...\} \implies$
  $VC(H) = 1$ if $+$ is always on the left;
  $VC(H) = 2$ if $+$ can be on left or right
- $H = \{intervals...\} \implies$
  VC(H) $=$ 2 if $+$ is always in center
  VC(H) $=$ 3 if center can be $+$ or -
- $H = \{$ linear decision surface in 2D ... $\}$
  $\implies VC(H) = 3$



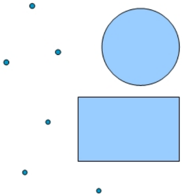
Examples of VC Dim

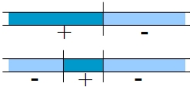
- $H = \{circles...\} \implies VC(H) = 3$
- $H = \{rectangles...\} \implies VC(H) = 4$
- $H = \{threshold\ functions...\} \implies$
  $VC(H) = 1$ if $+$ is always on the left;
  $VC(H) = 2$ if $+$ can be on left or right
- $H = \{intervals...\} \implies$
  VC(H) $= 2$ if $+$ is always in center
  VC(H) $= 3$ if center can be $+$ or -
- $H = \{$ linear decision surface in 2D ... $\}$
  $\implies VC(H) = 3$
- Is there an H with $VC(H) = \infty$?

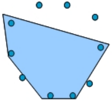


Examples of VC Dim

- $H = \{circles...\} \implies VC(H) = 3$
- $H = \{rectangles...\} \implies VC(H) = 4$
- $H = \{threshold\ functions...\} \implies$
  $VC(H) = 1$ if $+$ is always on the left;
  $VC(H) = 2$ if $+$ can be on left or right
- $H = \{intervals...\} \implies$
  VC(H) $= 2$ if $+$ is always in center
  VC(H) $= 3$ if center can be $+$ or -
- $H = \{$ linear decision surface in 2D ... $\}$
  $\implies VC(H) = 3$
- Is there an H with $VC(H) = \infty$?
- **Theorem** If $|\mathbb{H}| < \infty$ then $VCdim(\mathbb{H}) \leqslant \log |\mathbb{H}|$



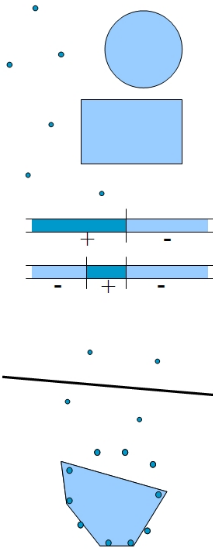




Examples of VC Dim

- $H = \{circles...\} \implies VC(H) = 3$
- $H = \{rectangles...\} \implies VC(H) = 4$
- $H = \{threshold\ functions...\} \implies$
  $VC(H) = 1$ if $+$ is always on the left;
  $VC(H) = 2$ if $+$ can be on left or right
- $H = \{intervals...\} \implies$
  VC(H) $= 2$ if $+$ is always in center
  VC(H) $= 3$ if center can be $+$ or -
- $H = \{$ linear decision surface in 2D ... $\}$
  $\implies VC(H) = 3$
- Is there an H with $VC(H) = \infty$?
- **Theorem** If $|\mathbb{H}| < \infty$ then $VCdim(\mathbb{H}) \leqslant \log |\mathbb{H}|$
- Let M_n = the set of all Boolean monomials of n variables. Since, $|M_n| = 3^n$ we have

$$VCdim(M_n) \leqslant n \log 3$$



Sample Complexity from VC Dimension

How many randomly drawn examples suffice to ε-exhaust $VS_{H,D}$ with probability at least $(1-\delta)$?

$$m \geq \frac{1}{\varepsilon}(4\log_2(2/\delta) + 8VC(H)\log_2(13/\varepsilon))$$

Potential learnability

- Let $D \in S(m, c)$

$$\mathbb{H}^c(D) = \{h \in \mathbb{H} | h(x_i) = c(x_i)(i = 1, ..., m)\}$$

- Algorithm L is consistent if and only if $L(D) \in \mathbb{H}^c(D)$ for any training sample D
- $\mathbb{B}^c_\varepsilon = \{h \in \mathbb{H} | er_\Omega(h) \geqslant \varepsilon\}$
- We say that H is potentially learnable if, given real numbers $0 < \varepsilon, \delta < 1$ there is a positive integer $m_0 = m_0(\varepsilon, \delta)$ such that, whenever $m \geqslant m_0$,

$$\mu^m\{D \in \mathcal{S}(m, c) | \mathbb{H}^c(D) \cap \mathbb{B}^c_\varepsilon = \emptyset\} > 1 - \delta$$

  for any probability distribution μ on X and $c \in \mathbb{H}$
- (Theorem:) If H is potentially learnable, and L is a consistent learning algorithm for H, then L is PAC.

Theorem

Haussler, 1988 Any finite hypothesis space is potentially learnable.

Proof: Let $h \in \mathbb{B}_\varepsilon$ then

$$\mu^m\{D \in \mathcal{S}(m,c) | er_D(h) = 0\} \leqslant (1-\varepsilon)^m$$

$$\Rightarrow \mu^m\{D : \mathbb{H}[D] \cap \mathbb{B}_\varepsilon \neq \emptyset\} \leqslant |\mathbb{B}_\varepsilon|(1-\varepsilon)^m \leqslant |\mathbb{H}|(1-\varepsilon)^m$$

It is enough to chose $m \geqslant m_0 = \left\lceil \frac{1}{\varepsilon} \ln \frac{|\mathbb{H}|}{\delta} \right\rceil$ to obtain $|\mathbb{H}|(1-\varepsilon)^m < \delta$

Fundamental theorem

Theorem

If a hypothesis space has infinite VC dimension then it is not potentially learnable. Inversely, finite VC dimension is sufficient for potential learnability

- Let $VCdim(\mathbb{H}) = d \geq 1$ Each consistent algorithm L is PAC with sample complexity

$$m_L(\mathbb{H}, \delta, \varepsilon) \leq \left\lceil \frac{4}{\varepsilon} \left(d \log \frac{12}{\varepsilon} + \log \frac{2}{\delta} \right) \right\rceil$$

- Lower bounds: for any PAC learning algorithm L for finite VC dimension space H,
  - $m_L(\mathbb{H}, \delta, \varepsilon) \geqslant d(1 - \varepsilon)$
  - If $\delta \leq 1/100$ and $\varepsilon \leq 1/8$, then $m_L(\mathbb{H}, \delta, \varepsilon) > \frac{d-1}{32\varepsilon}$
  - $m_L(\mathbb{H}, \delta, \varepsilon) > \frac{1-\varepsilon}{\varepsilon} \ln \frac{1}{\delta}$

Combine theory with practice

Theory is when we know everything and nothing works.

Combine theory with practice

Theory is when we know everything and nothing works.

Practice is when everything works and no one knows why.

Combine theory with practice

Theory is when we know everything and nothing works.

Practice is when everything works and no one knows why.

We combine theory with practice —

Combine theory with practice

Theory is when we know everything and nothing works.

Practice is when everything works and no one knows why.

We combine theory with practice —
nothing works and no one knows why.

Mistake Bounds

So far: how many examples needed to learn?

What about: how many mistakes before convergence?

Let's consider similar setting to PAC learning:

- Instances drawn at random from X according to distribution $\mathcal{D}$
- Learner must classify each instance before receiving correct classification from teacher
- Can we bound the number of mistakes learner makes before converging?

Mistake Bounds: Find-S

Consider Find-S when H = conjunction of boolean literals

FIND-S:

- *Initialize h to the most specific hypothesis* $l_1 \wedge \neg l_1 \wedge l_2 \wedge \neg l_2 \ldots l_n \wedge \neg l_n$
- *For each positive training instance x*
  - *Remove from h any literal that is not satisfied by x*
- *Output hypothesis h.*

How many mistakes before converging to correct h?

Mistake Bounds: Halving Algorithm

Consider the Halving Algorithm:

- Learn concept using version space CANDIDATE-ELIMINATION algorithm
- Classify new instances by majority vote of version space members

How many mistakes before converging to correct h?

- ... in worst case?
- ... in best case?

Optimal Mistake Bounds

Let $M_A(C)$ be the max number of mistakes made by algorithm A to learn concepts in C. (maximum over all possible $c \in C$, and all possible training sequences)

$$M_A(C) \equiv \max_{c \in C} M_A(c)$$

Definition: Let C be an arbitrary non-empty concept class. The **optimal mistake bound** for C, denoted $Opt(C)$, is the minimum over all possible learning algorithms A of $M_A(C)$.

$$Opt(C) \equiv \min_{A \in learning\ algorithms} M_A(C)$$

$$VC(C) \leq Opt(C) \leq M_{Halving}(C) \leq log_2(|C|).$$