Computational Learning Theory

- General laws constraining inductive machine learning

Theory on

- Classes of learning problems (independent of learning algorithm)
  - Difficulty
  - Computational complexity
- Probability of successful learning
- Number of training examples needed / errors committed for successful learning?
- Complexity of hypothesis space
- Accuracy to which target concept is approximated
- Manner in which training examples are presented

Successful learning: output hypothesis identical to target function

Outline

- Learning Scenarios:
  - *Setting 1*: learner poses queries to teacher
  - *Setting 2*: teacher chooses examples
  - *Setting 3*: randomly generated instances, labeled by teacher

- Probably Approximately Correct (PAC) learning
  → Sample & computational complexity
- Vapnik-Chervonenkis Dimension
  → Complexity of hypothesis space
- Mistake bounds

Prototypical Concept Learning Task

Given:

- Instances X: Possible days, each described by the attributes *Sky, AirTemp, Humidity, Wind, Water, Forecast*
- Target function c: $EnjoySport: X \rightarrow \{0,1\}$
- 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?

(Depends on the mode of providing training examples)

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: Case 1

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

Optimal query strategy:

- 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: Case 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, …* each have 2 possible values.
- if n possible boolean attributes in H, $n + 1$ examples suffice
- why? (by induction)

Exercise

Sample Complexity: Case 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 $< x, c(x) >$, for some target concept c in C

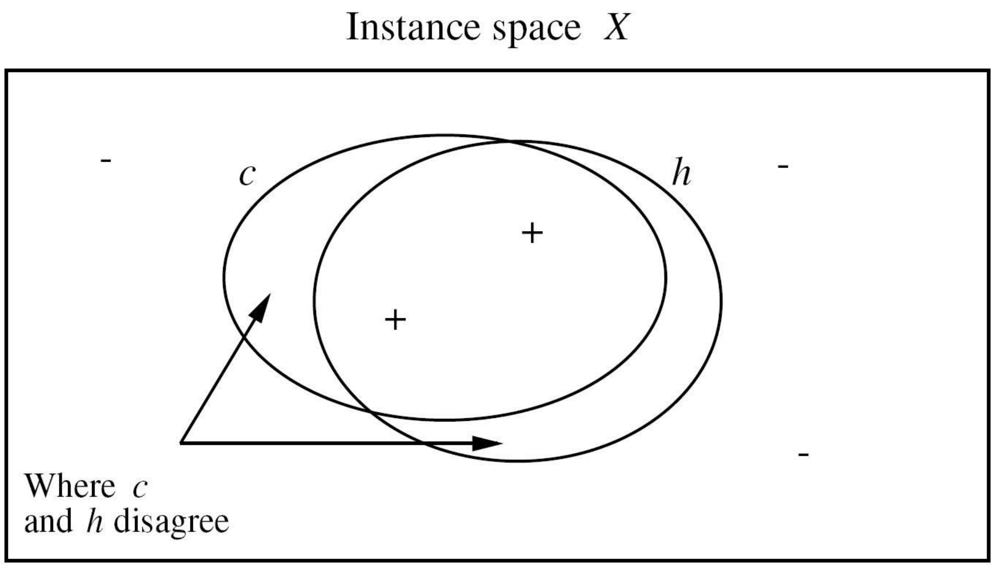
- Instances x are drawn from distribution $\mathcal{D}$
- teacher provides target value $c(x)$ for each

Learner must output a hypothesis h estimating c

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

Note: randomly drawn instances, noise-free classifications

True Error of a Hypothesis



The error of *h* with respect to *c* is the probability that a randomly drawn instance will fall into the region where h and c disagree

Expected error highly depends on $\mathcal{D}$: uniform vs. non-uniform

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 $\mathcal{D}$.

$$error_{\mathrm{D}}(h) = \Pr_{x \in \mathrm{D}}\left[c(x) \neq h(x)\right] = E_{\mathrm{D}}[c(x) \neq h(x)] = \int_{\mathrm{D}} p(x)[c(x) \neq h(x)]dx$$

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

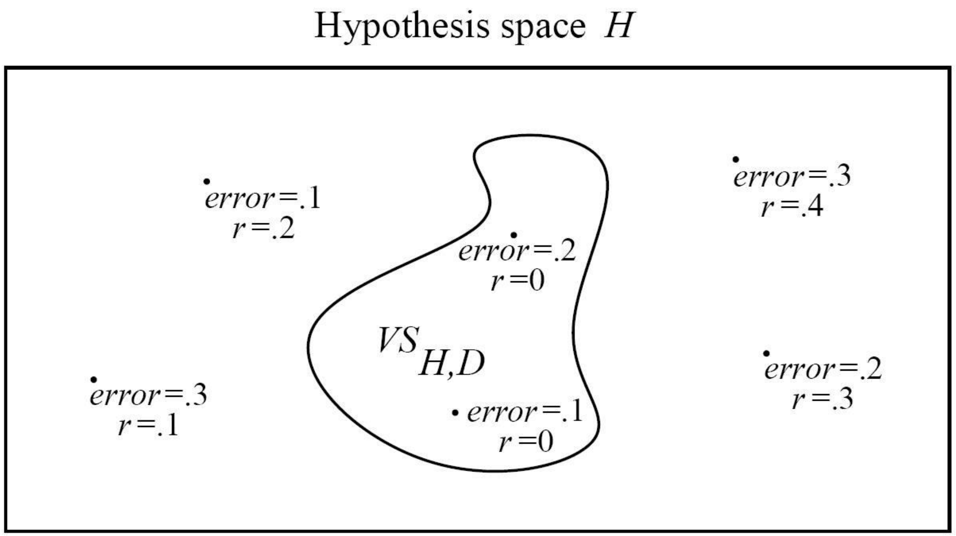
True error of hypothesis *h* with respect to *c*

- 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., $h \in VS_{H,D}$)

Exhausting the Version Space



(r = training error, $error$ = true error)

D = training examples
$\mathcal{D}$ = instance distribution

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 a true 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

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

<u>Interesting</u>: this bounds the probability that any consistent learner will output a hypothesis h with $error(h) \geq \varepsilon$

If we want this probability to be below δ, i.e.,

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

then

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

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:

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

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

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

If we want to learn a hypothesis for $n = 10$ with error less than .1 with 95% probability, we need 140 examples

How About *EnjoySport*?

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

$1 - \delta = 0.05$

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

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

... if we want to assure that with probability 95%, *VS* contains only hypotheses with $error_D(h) \leq .1$, then it is sufficient to have *m* examples, where

$\varepsilon = 0.1$

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

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

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

$$m \geq 98.8$$

Universität Augsburg University

Probably **A**pproximately **C**orrect:

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*).

New compared to previous slides.

Implicitly limits number of training examples (with some minimal processing time) to polynomial number!

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?

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

derived from Hoeffding bounds:

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

■ In addition compared to case $c \in H$

Sample Complexity of Infinite Hypothesis Spaces

VC-Dimension

Zweiteilung

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

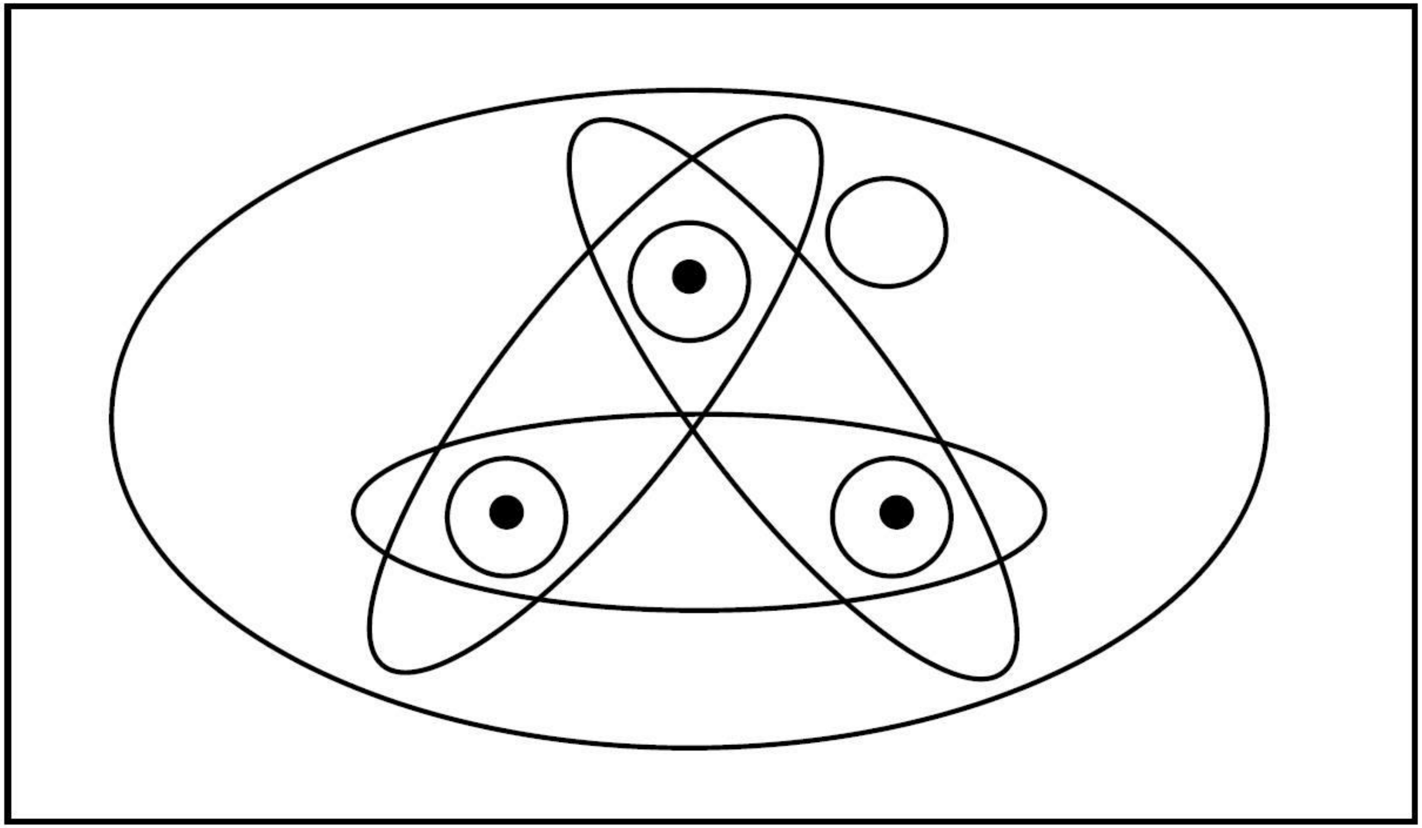
$$S \subseteq X$$
$$Y, \bar{Y} \subseteq S$$
$$S = Y + \bar{Y} = Y \cup \bar{Y}$$
$$Y \cap \bar{Y} = \emptyset$$

zersplittered

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

Instance space X

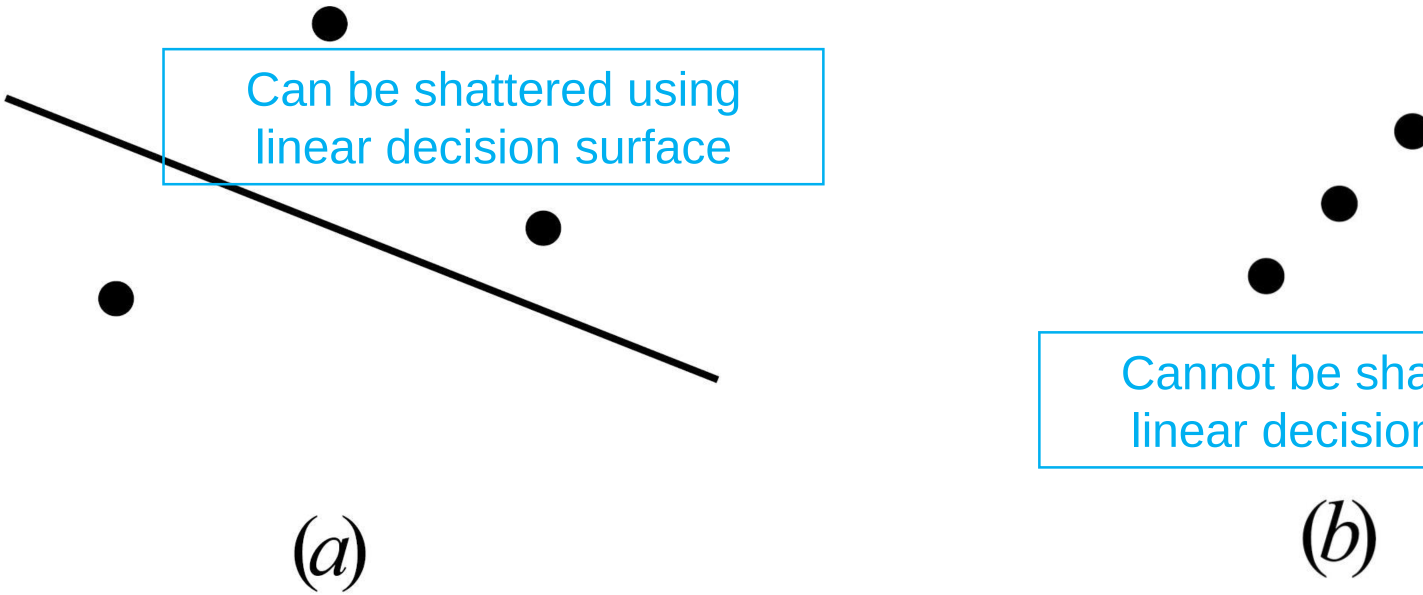


Definition: The **Vapnik-Chervonenkis dimension**, $VC(H)$, of hypothesis space H defined over instance space X is the size of the largest finite subset of X shattered by H. If arbitrarily large finite sets of X can be shattered by *H*, then $VC(H) \equiv \infty$

Determine $VC(H)$:

- Show that <u>any</u> set of n instances can be shattered by H
- Show that <u>no</u> set of $n + 1$ instances can be shattered by H
- Then, $VC(H) = n$

VC Dim. of Linear Decision Surfaces



In General: In the r-dimensional space $VC(H) = r + 1$ for linear decision surfaces

To show that $VC(H) < d$, we must show that no set of size d can be shattered! Def. of VC says that if we find *any* set of instances of size d that can be shattered, then $VC(H) \geq d$

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

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

Mistake Bounds

Mistake Bounds

The learner is evaluated by the total # of mistakes before it converges to the correct hypothesis.

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?

Used in actual systems, where the learning is done while the system is in use.

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?

Answer : n + 1 (worst case; target concept: $\forall x : c(x) = 1$)

Second Example: Halving Algorithm

Universität Augsburg University

Consider the Halving Algorithm:

- Learn concept using version space *Candidate-Elimination* algorithm
- Classify new instances by majority vote of version space members → a mistake can only happen if the majority of hypotheses in the current version space incorrectly classify the sample

How many mistakes before converging to correct *h?*

- ... in worst case? Answer: floor ($\log_2|H|$)
- ... in best case? Answer : 0

With every mistake, equal or more than half of instances are removed

Even when the majority vote is correct, the algorithm will remove the incorrect, minority hypotheses

Let $M_A(C)$ be the maximal 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 $M_A(C)$.

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

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