Chapter 7:
Computational Learning Theory

CS 536: Machine Learning

Littman (Wu, TA)

Computational Learning Theory

[Read Chapter 7]

[Suggested exercises: 7.1, 7.2, 7.5, 7.8]

- Computational learning theory
- 1: learner poses queries to teacher
- 2: teacher chooses examples
- 3: randomly generated instances
- PAC learning
- Vapnik-Chervonenkis Dimension
- Mistake bounds

COLT

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

Prototypical Learning Task

- **Given** (for concept learning):
  - 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.

    <?,*Cold*,*High*, ?, ?, ?>.
  - Training examples S : Positive and negative examples of the target function

    $<x_1, c(x_1)>, \ldots <x_m, c(x_m)>$

Prototypical Learning Task

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

Note: We’ll put statistical considerations on hold.

Sample Complexity

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

1. If learner proposes instances as queries to teacher

- Learner proposes *x*, teacher provides *c*(*x*)

2. If teacher (who knows *c*) provides training examples

- teacher provides example sequence <*x*, *c*(*x*)>

3. If some random process (e.g., nature) proposes instances

- *x* generated randomly, teacher provides *c*(*x*)

Sample Complexity: 1

Learner proposes instance *x*, teacher provides *c*(*x*) (assume *c* is known to be 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
- If this is always possible, $\log_2 |H|$ queries suffice to learn *c*
- When it’s not possible, need 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

ex., (*AirTemp* = *Warm*) ^ (*Wind* = *Strong*), where *AirTemp*,*Wind*, … 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 D over X

(Now, statistics are coming back in…)

Sample Complexity: 3

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

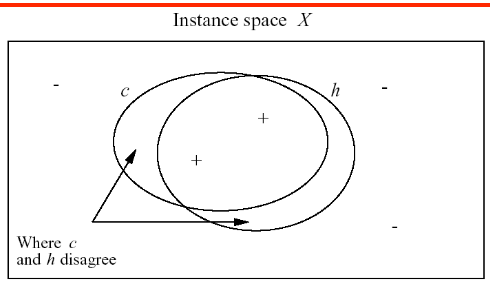
- instances x are drawn from distribution D
- teacher provides target values $c(x)$

Learner must output a hypothesis h estimating c

- h is evaluated by its performance on subsequent instances drawn from D

Note: randomly drawn instances, noise-free classifications

True Error of a Hypothesis



Definition: The **true error** (denoted $error_D(h)$) of hypothesis h with respect to target concept c and distribution D is the probability that h will misclassify an instance drawn at random via D.

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

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 S?

True error of hypothesis h with respect to c

- How often $h(x) \neq c(x)$ over future random instances drawn from D?

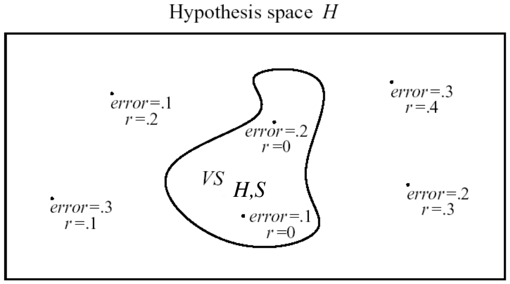
Our concern:

- Can we bound the true error of h given the training error of h?
- First, consider training error of h is zero.

A Word on Version Spaces

A bit of notation: $VS_{H,S}$ is the set of hypotheses in the hypothesis space H that are consistent with the training examples S.

Exhausting the Version Space



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

Definition: The version space $VS_{H,S}$ is said to be **ε–exhausted** with respect to c and S, if every hypothesis h in $VS_{H,S}$ has error less than ε with respect to c and S.

$$(\forall h \text{ in } VS_{H,S})\ error_D(h) < \varepsilon$$

Examples Needed?

Theorem: [Haussler, 1988].

If the hypothesis space H is finite, and S 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 S is *not* ε–exhausted (with respect to c) is less than $|H| e^{-\varepsilon m}$.

Implications

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

$$m \geq 1/\varepsilon\ (\ln|H| + \ln(1/\delta)).$$

Conjunctions of Literals

How many examples are sufficient to assure with probability at least $(1-\delta)$ that every h in $VS_{H,S}$ satisfies $error_D(h)$?

Use our theorem: $m \geq 1/\varepsilon\ (\ln|H| + \ln(1/\delta))$.

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

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

$$m \geq 1/\varepsilon\ (n \ln 3 + \ln(1/\delta)).$$

How About *EnjoySport*?

$$m \geq 1/\varepsilon\ (\ln|H| + \ln(1/\delta)).$$

If H is as given in *EnjoySport* then $|H| = 973$, and $m \geq 1/\varepsilon\ (\ln 973 + \ln(1/\delta))$.

... if 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

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

$= 10(\ln 973 + \ln 20) = 10(6.88 + 3.00)$

$= 98.8$

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 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 $error_D(h) \leq \varepsilon$, in time (samples) polynomial in $1/\delta$, $1/\varepsilon$, n and size(c).

Agnostic Learning Setting

So far, assumed c in H

Agnostic learning: 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?

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

derived from Hoeffding (Chernoff) bounds:

$$\Pr[error_D(h) > error_S(h) + \varepsilon] \leq \exp(-2m\varepsilon^2).$$

Shattering a Set

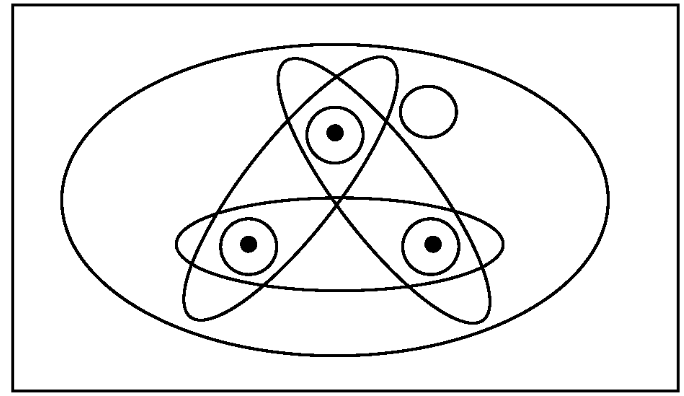
Definition: a **dichotomy** of a set S is a partition of S into two disjoint subsets.

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.

In other words: The instances can be classified in every possible way.

Three Instances Shattered

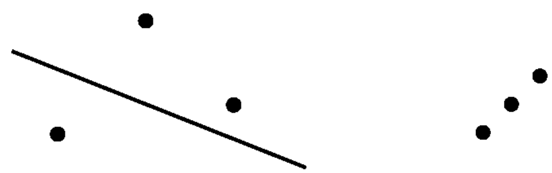
Instance space X



The VC 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 (but finite) sets of X can be shattered by H, then $VC(H) \equiv \infty$.

VC Dim. of Linear Decision Surfaces



(*a*) (*b*)

Is the VC dimension at least 3?

Sample Complexity and VC Dim.

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

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

The VC dimension plays an analogous role to $\ln |H|$.

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 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 most specific hypothesis

  $l_1 \wedge \neg l_1 \wedge l_2 \wedge \neg l_2 \wedge \ldots \wedge l_n \wedge \neg l_n$
- 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?

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 is over all possible c in C, and all possible training sequences)

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

All Together Now

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 $M_A(C)$.

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

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