Computational Learning Theory

Tian-Li Yu

Taiwan Evolutionary Intelligence Laboratory (TEIL)
Department of Electrical Engineering
National Taiwan University
tianliyu@ntu.edu.tw

Readings: ML Chapter 7 (AIMA 18.5 covers tiny little bit)

Outline

Computational Learning Theory

- What general laws constrain inductive learning?
- We seek theory to relate:
  - Complexity of hypothesis space considered by the learner
  - Accuracy to which target concept is approximated
  - Probability that the learner outputs a successful hypothesis
  - Manner in which training examples presented to the learner
- Goals:
  - Sample complexity: How many training examples are needed for successful learning?
  - Computational complexity: How much computational effort is needed for a learner to converge to a successful hypothesis?
  - Mistake bound: How many examples will the learner misclassify before the convergence?

Sample Complexity

- How many training examples are sufficient to learn the target concept?
- 3 settings:
  1. Learner proposes instances, as queries to teacher:
     Learner proposes instance x, teacher provides $c(x)$.
  2. Teacher provides training examples:
     Teacher provides sequence of examples of form $\langle x, c(x) \rangle$.
  3. Some random process (*e.g.*, nature) proposes instances:
     Instance x generated randomly, teacher provides $c(x)$.

Sample Complexity: Setting 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.

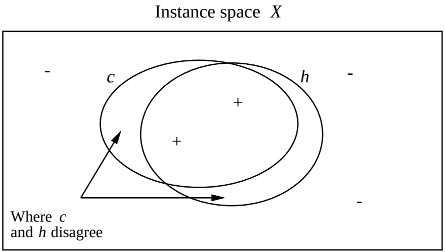
Sample Complexity: Setting 2

- 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 where H is conjunctions of up to n boolean literals (positive or negative).
    - e.g., $(AirTemp = Warm) \wedge (Wind = Strong)$, where $AirTemp, Wind, \ldots$ each has 2 possible values.
    - if n possible boolean attributes in H, $(n+1)$ examples suffice.
    - Why?

Sample Complexity: Setting 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 $\mathbb{D}$ 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 $\mathbb{D}$.
  - Teacher provides target value $c(x)$ for each x.
- Learner must output a hypothesis h estimating c
  - h is evaluated by its performance on subsequent instances drawn according to $\mathbb{D}$
- **Note:** randomly drawn instances, noise-free classifications.

True Error of a Hypothesis



Definition

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

$$error_{\mathbb{D}}(h) \equiv \Pr_{x \in \mathbb{D}} (c(x) \neq h(x))$$

Two Notions of Error

- Training error, denoted $error_D(h)$, of hypothesis h with respect to c: How often $h(x) \neq c(x)$ over training instances.
- True error, denoted $error_{\mathbb{D}}(h)$, of hypothesis h with respect to c: How often $h(x) \neq c(x)$ over future random instances.

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

PAC Learning

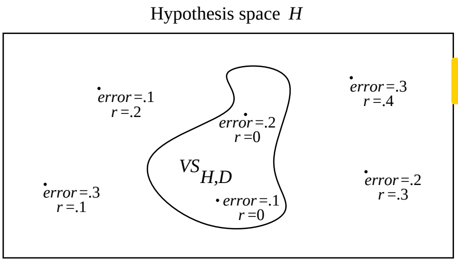
- Consider a class C of possible target concepts defined over a set of instances X of length n, and a learner L using hypothesis space H.
- We desire that the learner probably learns a hypothesis that is approximately correct.

Definition

C is **PAC-learnable** by L using H if for all $c \in C$, distributions $\mathbb{D}$ over X, ϵ such that $0 < \epsilon < 1/2$, 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 $error_{\mathbb{D}}(h) \leq \epsilon$, in time that is polynomial in $1/\epsilon$, $1/\delta$, n and $size(c)$.

- To prove any concept is PAC-learnable or not, we need to derive the sample complexity needed for setting 3.

Exhausting the Version Space



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

Definition

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

$$(\forall h \in VS_{H,D})\ error_{\mathbb{D}}(h) < \epsilon$$

Probability of Exhausting the Version Space

- How many examples ϵ-exhaust the VS?

Theorem (Haussler, 1988)

If H is finite, and D is a sequence of $m \geq 1$ independent random examples (from distribution $\mathbb{D}$) of some target concept c, then for any $0 \leq \epsilon \leq 1$, the probability that $VS_{H,D}$ is not ϵ-exhausted is less than or equal to

$$|H|e^{-\epsilon m}.$$

- The above theorem bounds the probability that any consistent learner will output a hypothesis h with $error_{\mathbb{D}}(h) \geq \epsilon$.
- If we want to this probability to be below δ

$$|H|e^{-\epsilon m} \leq \delta \quad \Rightarrow \quad m \geq \frac{1}{\epsilon}(\ln|H| + \ln(1/\delta))$$

Proof of ϵ-Exhausting

Proof: ϵ-exhausting the version space.

- Let $h_1, \cdots, h_k$ be all hypotheses in H with true errors greater than ϵ with respect to c.
- Fail to ϵ-exhausting the VS iff at least one of these hypotheses consistent with all m examples.
- Such prob. for a single hypothesis and a single random example is $(1-\epsilon)$; or $(1-\epsilon)^m$ for all m examples.
- The prob. that fail to ϵ-exhausting is at most $k(1-\epsilon)^m$.

$$k(1-\epsilon)^m \leq |H|(1-\epsilon)^m \leq |H|e^{-\epsilon m}$$

$\square$

Learning Conjunctions of Boolean Literals

- Recall that $m \geq \frac{1}{\epsilon}(\ln |H| + \ln(1/\delta))$ examples are sufficient to assure with probability at least $(1 - \delta)$ that every h in $VS_{H,D}$ satisfies $error_{\mathbb{D}}(h) \leq \epsilon$.
- Suppose H contains conjunctions of constraints on up to n boolean attributes.
  - $|H| = 3^n$.
  - $m \geq \frac{1}{\epsilon}(n \ln 3 + \ln(1/\delta))$
  - Boolean conjunctions is PAC-learnable!

EnjoySport Revisit

- Inn *EnjoySport*, if we consider only conjunctions, $|H| = 973$.

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

- If want to assure that with probability 95%, *VS* contains only hypotheses with $error_{\mathbb{D}}(h) \leq 0.1$, then it is sufficient to have m examples, where

$$m \geq \frac{1}{0.1}\left(\ln 973 + \ln\frac{1}{0.05}\right)$$

$$m \geq 98.8$$

Unbiased Learners

- Consider the unbiased concept class C over an instance space X.

$$|C| = 2^{|X|}$$

- If an instance contains n-boolean features: $|X| = 2^n$; $|C| = 2^{2^n}$

$$m \geq \frac{1}{\epsilon}\left(2^n \ln 2 + \ln \frac{1}{\delta}\right)$$

- In general, unbiased concepts are not PAC-learnable.

Agnostic Learning (Learning Inconsistent Hypotheses)

- The equation $m \geq \frac{1}{\epsilon}(\ln|H| + \ln(1/\delta))$ tells us how many training examples suffice to ensure that every hypotheses in H having zero training error will have true error of at most ϵ.
- However, if $c \notin H$, zero training error may not be achievable.
- We desire to know how many examples suffice to ensure $error_{\mathbb{D}}(h) \leq error_D(h) + \epsilon$.
- **Hoeffding bounds:**

$$\Pr\left(error_{\mathbb{D}}(h) > error_D(h) + \epsilon\right) \leq e^{-2m\epsilon^2}$$

- Sample complexity in this case:

$$\Pr\left((\exists h \in H)\ error_{\mathbb{D}}(h) > error_D(h) + \epsilon\right) \leq |H|e^{-2m\epsilon^2} \leq \delta$$

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

Infinite Hypothesis Space

- The above sample complexity has two drawbacks:
  1. Weak bounds.
  2. H has to be finite.
- We need another measure of the complexity of H.

Definition

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

Definition

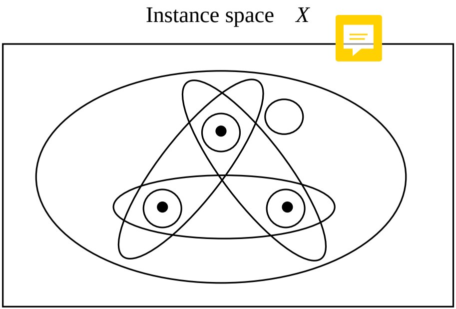
A set of instances S is **shattered** by hypothesis space H iff for every dichotomy of S there exists some hypothesis in H consistent with this dichotomy.

Shattering a Set of Instances

- S is a subset of instances, $S \subseteq X$; $2^{|S|}$ distinct dichotomies in total.
- Each $h \in H$ imposes a dichotomy on S:

$$\{x \in S | h(x) = 0\} \text{ and } \{x \in S | h(x) = 1\}$$

- H shatters S iff every dichotomy of S is represented by some $h \in H$.



The Vapnik-Chervonenkis (VC) Dimension

- The ability to shatter a set of instances is closely related to the inductive bias of the hypothesis space.
- An unbiased hypothesis space can represent every possible concept (dichotomy) over X: An unbiased hypothesis space shatters X.
- What if H cannot shatter X, but can shatter a subset S?
- Intuitively, the larger S is, the more expressive H is.

Definition

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

- Note that for any finite H, $VC(H) \leq \log_2 |H|$.

VC Dimension

- Instances are real numbers: $X = \mathbb{R}$
- Hypotheses are real intervals: $h_{ab} = a < x < b$; $H = \{\forall a, b\ h_{ab}\}$
- Consider $S = \{3.1, 5.7\}$. H shatters S, why?
- For any set of 3 instances: $S = \{x, y, z\}$, where $x < y < z$. There is no way for H to represent this dichotomy: $\{x, z\}$ and $\{y\}$.

$$VC(H) = 2$$

- For 2D points (X) and line separations (H), $VC(H) = 3$.

(a) (b)

VC Dimension and Sample Complexity

- How many randomly drawn examples suffice to ϵ-exhaust $VS_{H,D}$ with probability at least $(1-\delta)$? [Blumer *et al.*, 1989]

Upper bound on sample complexity

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

- Similarly, m grows with $\log(1/\delta)$.
- Now, m grows with $(1/\epsilon)\log(1/\epsilon)$ rather than linear.
- Most importantly, $\ln|H|$ is replaced by $VC(H)$. Recall that $VC(H) \leq \log_2|H|$.

VC Dimension and Sample Complexity

- How about lower bound? [Ehrenfeucht *et al.*, 1989]

Lower bound on sample complexity

Consider any concept C where $VC(C) \geq 2$, any learner L, any $0 < \epsilon < \frac{1}{8}$, and $0 < \delta < \frac{1}{100}$. There exists a distribution $\mathbb{D}$ and target concept in C such that if L observes fewer examples than

$$\max\left\{\frac{1}{\epsilon}\log_2(1/\delta), \frac{VC(C)-1}{32\epsilon}\right\}$$

then with prob. at least δ, L outputs a hypothesis h having $error_{\mathbb{D}}(h) > \epsilon$.

- Given the lower bound, we see that the upper bound in the previous slide is fairly tight.

Mistake Bounds

- So far: how many examples needed to learn?
- What about: how many mistakes before convergence?
  Similar setting to PAC learning:
  - Instances drawn at random from X according to distribution $\mathbb{D}$.
  - Learner must classify each instance before receiving correct classification from teacher.
  - Can we bound the number of mistakes learner makes before converging?

Mistake Bound for FIND-S

- Consider FIND-S when H are conjunctions of n boolean literals $\ell_1, \cdots, \ell_n$.

FIND-S

- Initialize h to the most specific hypothesis

$\emptyset$ = $\ell_1 \wedge \neg\ell_1 \wedge \ell_2 \wedge \neg\ell_2 \ldots \ell_n \wedge \neg\ell_n$

- 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?
  - Provided $c \in H$, FIND-S never misclassifies negative examples.
  - 1 The first positive example reduce the $2n$ literals to n.
  - +) n Then every misclassified positive examples removes at least one literal.
  - n + 1 At most $(n+1)$ mistakes.

Mistake Bound for HALVING Algorithm

- Consider the HALVING Algorithm:
  - Learn concept with version space such as the CANDIDATE-ELIMINATION algorithm
  - Classify new instances by majority vote of version space members
- How many mistakes before converging to correct h?
  - Worst case: $\lfloor \log_2 |H| \rfloor$, why?
  - Best case: 0, why?

Optimal Mistake Bound

- Interested in the optimal mistake bound for an arbitrary concept class C, assuming $H = C$.
- Define $M_A(c)$ as the maximum over all possible sequence of training examples of the number of mistakes made by algorithm A and the target concept c.
- For any nonempty concept class C, define $M_A(C) = \max_{c \in C} M_A(c)$.

Definition

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

$$Opt(C) = \min_A M_A(C)$$

Bounds for Optimal Mistake Bound

- $VC(C) \leq Opt(C) \leq \log_2 |C|$ (Littlestone, 1987)

Proof.

Right: $Opt(C) \leq M_{\text{HALVING}}(C) \leq \log_2 |C|$

Left (Adversarial):

1. Let $S = \{x_1, \ldots, x_{VC(C)}\} \subseteq X$ be a shattered set.
2. Suppose the environment reveals $x_i \in S$, and the algorithm outputs $\hat{y}_i$.
3. The environment selects a new target concept $c \in C$ such that $c(x_i) = y_i \neq \hat{y}_i$.
4. Since S is shattered by C, there always exists such c, and no way the algorithm can tell the difference.
5. Therefore, the algorithm makes at least $VC(C)$ mistakes. □

WEIGHTED-MAJORITY Algorithm

WEIGHTED-MAJORITY

a_i: prediction algorithms; w_i: weights, initialized to all 1; $0 \leq \beta < 1$

for each training example $\langle x, c(x) \rangle$

    $q_0 = 0$; $q_1 = 0$

    for each algorithm a_i

        If $a_i(x) == 0$ **then** $q_0 = q_0 + w_i$

        If $a_i(x) == 1$ **then** $q_1 = q_1 + w_i$

    If $q_0 > q_1$ **then** predict $\hat{c}(x) = 0$

    If $q_0 < q_1$ **then** predict $\hat{c}(x) = 1$

    If $q_0 == q_1$ **then** predict $\hat{c}(x) = 0$ or 1 at random

    for each algorithm a_i

        each $a_i(x) \neq c(x)$ **then** $w_i = \beta w_i$.

- Note that β is 0, WEIGHTED-MAJORITY reduces to HALVING.

Mistake Bound for WEIGHTED-MAJORITY

- For any sequence of training examples D, let A be any set of n prediction algorithms, and let k be the minimum number of mistakes made by any algorithm in A over D. The number of mistakes over D made by WEIGHTED-MAJORITY with $\beta = 1/2$ is at most

$$M \leq 2.4(k + \log_2 n).$$

取 lg

Proof.

- Let a_j be the best algorithm which yields k; its final weight $w_j = \frac{1}{2^k}$.
- Consider the sum $W = \sum_i w_i$. W initially n.
- Each mistake reduces W to at most $\frac{3}{4}W$.
- Let M be the total number of mistakes of WEIGHTED-MAJORITY.
- The final W is at most $n\left(\frac{3}{4}\right)^M$. So $\left(\frac{1}{2}\right)^k \leq n\left(\frac{3}{4}\right)^M$ □

Summary

- PAC considers algorithms that learns target concept using training examples randomly drawn from an unknown but fixed distribution.
- PAC: with high probability $(1-\delta)$, the learner outputs a hypothesis that is approximately correct (within error ϵ) within computational time polynomial in $1/\delta$, $1/\epsilon$, the size of instances, and the size of target concept.
- For finite hypothesis spaces, sample complexity can be derived for a consistent and agnostic learners, respectively.
- VC dimension measures the expressiveness of a hypothesis space, and an alternative (usually tighter, and for infinite hypothesis space) upper bound is derived using VC-dimension.
- Optimal mistake is bounded by VC-dimension and HALVING.
- The number of mistakes of WEIGHTED-MAJORITY is bounded by its best predictor.