Machine learning

Computational learning theory

Hamid Beigy

Sharif University of Technology

December 12, 2021

Table of contents

Introduction

Introduction

1. Computational learning theory seeks to answer questions such as
   - Is it possible to identify classes of learning problems that are inherently easy or difficult?
   - Can we characterize the number of training examples necessary or sufficient to assure successful learning?
   - How is this number affected if the learner is allowed to pose queries to the trainer?
   - Can we characterize the number of mistakes that a learner will make before learning the target function?
   - Can we characterize the inherent computational complexity of classes of learning problems?
2. General answers to all these questions are not yet known.
3. In this lecture, we want to answer some of the above questions for simple learning problems / algorithms?

Problem setting

1. Problem setting for concept learning
   - Domain Set of all possible instances over which target functions may be defined. Training and Testing instances are generated from X according some unknown distribution $\mathcal{D}$.

     We assume that $\mathcal{D}$ is stationary.
   - Set of labels In this model label set $\mathcal{T}$ will either be $\{0, 1\}$ or $\{1, +1\}$.
   - Concept class Set of target concepts that our learner might be called upon to learn. Target concept is a Boolean function $c : X \to \{0, 1\}$.
   - Hypothesis class Set of all possible hypotheses.

     The goal is producing hypothesis $h \in H$ which is an estimate of c.
   - Performance measure Performance of h measured over new samples drawn randomly using distribution $\mathcal{D}$.

Error of a classifier

Definition (Sample error)

The sample error (denoted $E_E(h)$) of hypothesis h with respect to target concept c and data sample S of size N is.

$$E_E(h) = \frac{1}{N} \sum_{x \in S} \mathbb{I}\left[c(x) \neq h(x)\right]$$

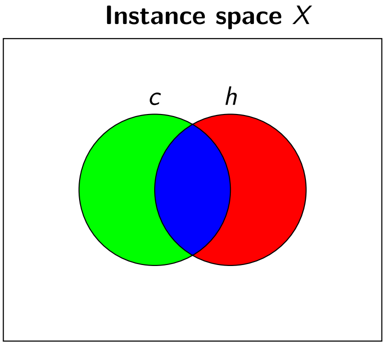
Definition (True error)

The true error (denoted $E(h)$) 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 distribution $\mathcal{D}$.

$$\begin{aligned} E(h) &= P_{x \sim \mathcal{D}}[c(x) \neq h(x)] \\ &= \sum_{c(x) \neq h(x)} \mathcal{D}(x) \end{aligned}$$

Notions of errors

1. True error is



2. $E(h)$ depends strongly of the $\mathcal{D}$.

Definition (Approximately correct)

Hypotesis h is approximately correct if $E(h) \leq \epsilon$.

Probably approximately correct (PAC) learning

Probably approximately correct (PAC) learning

1. We are trying to characterize the number of training examples needed to learn a hypothesis h for which $E(h) = 0$.
2. Is it possible?
   - May be multiple consistent hypotheses and the learner can not pickup one of them.
   - Since training set is chosen randomly, the true error may not be zero.
3. To accommodate these difficulties, we need
   - We will not require that the learner output a zero error hypothesis, we will require only that its error be bounded by some constant ϵ that can be made arbitrarily smal.
   - We will not require that the learner succeed for every sequence of randomly drawn training examples, we will require only that its probability of failure be bounded by some constant, δ, that can be made arbitrarily small.
   - δ is confidence parameter.

Probably approximately correct (PAC) learning

Definition (PAC Learnability)

Concept class C is PAC-learnable by learning algorithm L using hypotheses space H if for all concepts $c \in C$, distributions $\mathcal{D}$ over X, there exists

- an ϵ $(0 < \epsilon < \frac{1}{2})$, and
- a δ $(0 < \delta < \frac{1}{2})$,

with probability at least $(1 - \delta)$, learner L will output a hypothesis $h \in H$ such that

- $E(h) \leq \epsilon$, and
- in time that is polynomial in $(\frac{1}{\epsilon})$, $(\frac{1}{\delta})$, n, and $|C|$.

Probably approximately correct (PAC) learning

1. If L requires some minimum processing time per training example, then for C to be PAC-Learnable by L, L must learn from a polynomial number of training examples.

Definition (Sample complexity)

The growth in the number of required training examples with problem size.

2. The most limiting factor for success of a learner is the limited availability of training data.

Definition (Consistent learner)

A learner is consistent if it outputs hypotheses that perfectly fit the training data, whenever possible.

3. Our concern : Can we bound $E(h)$ given $E_E(h)$?

Probably approximately correct (PAC) learning

Theorem (Haussler, 1988)

Let H be a finite hypothesis class. Let A be an algorithm that for any target concept $c \in H$ and i.i.d. sample S returns a consistent hypothesis h_S. Then, for any $\epsilon, \delta > 0$, the inequality

$$P_{x \sim \mathcal{D}^m}[E(h_S) \leq \epsilon] \geq 1 - \delta$$

holds if $m \geq \frac{1}{\epsilon}\left(\log |H| + \log \frac{1}{\delta}\right)$.

Proof.

1. Bound the probability that any consistent learner will output a hypothesis h with $E(h) \geq \epsilon$.
2. Want this probability to be below a specified threshold δ, i.e. $|H|e^{-\epsilon m} \leq \delta$
3. To achieve, solve inequality for m such that $m \geq \frac{1}{\epsilon}\left(\ln |H| + \ln\left(\frac{1}{\delta}\right)\right)$

□

It is possible that $|H|e^{-\epsilon m} > 1$.

PAC learning (example)

1. Let H be conjunctions of constraints on up to n boolean attributes. Then

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

2. Thus this problem is PAC learnable.
3. Consider the following dataset, what is the sample complexity for $\epsilon = 0.1, \delta = 0.05$.

Example	Sky	Air Temp	Humidity	Wind	Water	Forecast	Enjoy Sport
0	Sunny	Warm	Normal	Strong	Warm	Same	**Yes**
1	Sunny	Warm	High	Strong	Warm	Same	**Yes**
2	Rainy	Cold	High	Strong	Warm	Change	**No**
3	Sunny	Warm	High	Strong	Cool	Change	**Yes**

4. In this case, $|H| = 973$ and

$$m \geq 1/0.1(ln\ 973 + ln(1/0.05)) \simeq 98.8$$

PAC learning (example)

1. Let H be the set of all funtions on up to n boolean attributes. Then

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

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

2. Sample complexity is exponential in n and thus this problem is not PAC learanable.
3. This is a unbiased learner. It has no assumption about the hypotheses space or search method.

Agnostic learner

Definition (Agnostic learner)

A learner that make no assumption that the target concept is representable by H and that simply finds the hypothesis with minimum error.

1. How hard is this?

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

2. Derived from Hoeffding bounds

$$P\left[E(h) > E_E(h) + \epsilon\right] \leq e^{-2m\epsilon^2}$$

Vapnik-Chervonekis dimension

Vapnik-Chervonekis dimension

1. Drawbacks of sample complexity
   The bound is not tight, when $|H|$ is large and the probability may be grater than 1.
   When $|H|$ is infinite.

Definition (Vapnik-Chervonekis dimension ($VC(H)$))

VC-dimension measures complexity of hypothesis space H,not by the number of distinct hypotheses $|H|$, but by the number of distinct instances from X that can be completely discriminated using H.

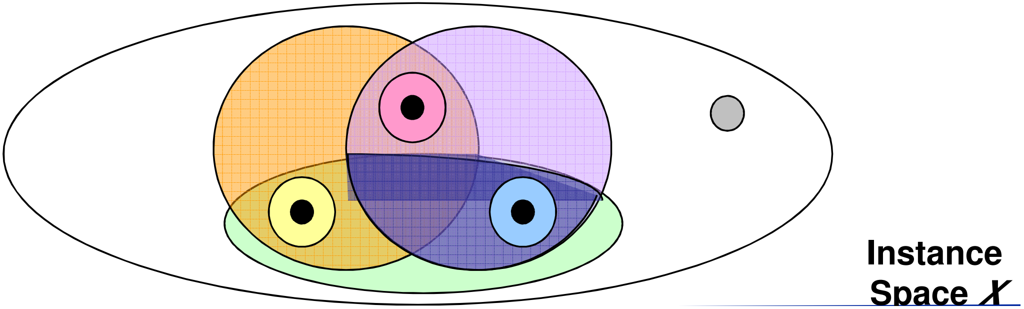
Definition (Dichotomy)

A dichotomy of a set S is a partition of S into two subsets S_1 and S_2.

Vapnik-Chervonekis dimension

Definition (Shattering)

A set S is shattered by hypothesis space H if and only if for every dichotomy (concept) of S, there exists a hypothesis in H consistent with this dichotomy



Vapnik-Chervonekis dimension

Definition (Vapnik-Chervonekis dimension ($VC(H)$))

$VC(H)$ of hypotheses space H defined over the instance space X is the size of largest finite subset of X shattered by H. If arbitrary large finite sets of X can be shattered by H, then $VC(H) = \infty$.

Lemma

For any finite H, we have $VC(H) \leq \log_2 |H|$.

Example

1. Let $X = \Re$ and $H = \{(a,b) | a < b\}$, then $VC(H) = 2$.
2. Let $X = \Re^2$ and H be the set of linear decision surfaces, then $VC(H) = 3$.
3. Let $X = \Re^n$ and H be the set of linear decision surfaces, then $VC(H) = n + 1$.
4. Let $X = \Re^2$ and H be the set of all axis aligned rectangles in $\Re^2$, then $VC(H) = 4$.
5. VC for a NN with linear activation and N free parameters is $O(N)$.
6. VC for a NN with threshold activation and N free parameters is $O(N \log N)$.
7. VC for a NN with sigmoid activation and N free parameters is $O(N^2)$.

Mistake bounds

Mistake bounds

Definition (Mistake bound)

How many mistakes will the learner make in its prediction before it learns the target concept?

1. Suppose H be conjunction of up to n Boolean literals and their negations.
2. Find-S algorithm
   - Initialize h to the most specific hypothesis

     $$(\bar{l}_1 \wedge l_1) \wedge (\bar{l}_2 \wedge l_2) \wedge \ldots \wedge (\bar{l}_n \wedge l_n)$$

   - For each positive training instance x remove from h any literal that is not satisfied by x.
   - Output hypothesis h

Mistake bounds

1. How many mistakes before converging to correct *h*?
   - Once a literal is removed, it is never put back
   - No false positives (started with most restrictive *h*), count only false negatives
   - First example will remove n candidate literals
   - Worst case: every remaining literal is also removed (incurring 1 mistake each)
   - Find-S makes at most $n + 1$ mistakes

Reading

Readings

1. Chapter 7 of Machine Learning Book (Mitchell 1997).

References i

Mitchell, Tom M. (1997). *Machine Learning*. McGraw-Hill.

Questions?