Introduction to Machine Learning

PAC Learning and VC Dimension

Varun Chandola

February 28, 2017

Outline

Contents

1 Measuring Sample Complexity

2 Types of Complexities for a Hypothesis Space

- **Mistake Bound** - How many mistakes before learning the target concept?
- **Sample Complexity Bound** - How many training examples are needed before learning the target concept (with high probability)?
- **Computational Complexity Bound** - How much computational effort is needed before learning the target concept (with high probability)?

Understanding Sample Complexity [1]

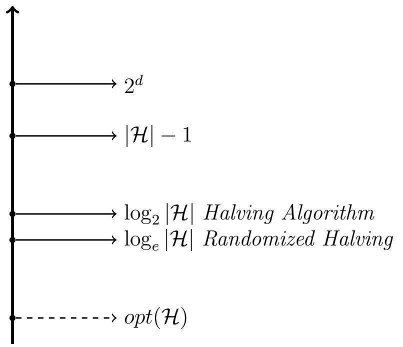
- How many training examples are sufficient to learn the target concept?

Note that this is different from the question that we asked in the mistake bound model - *How many mistakes will be made before learning the target concept*. The current question can be posed to batch learning as well as online learning. The question of sample complexity is important because, often, in real settings, training data is typically limited.

3 Mistake Bound Analysis

Mistake Bound Analysis is Too Strict

- An upper bound of total number of mistakes
- Issues?



2^d
$|\mathcal{H}| - 1$
$\log_2 |\mathcal{H}|$ *Halving Algorithm*
$\log_e |\mathcal{H}|$ *Randomized Halving*
$opt(\mathcal{H})$

There are two key issues with the mistake bound model for analyzing machine learning algorithms.

1. We can only say how many mistakes will be made, not when they will be made. It might be desirable to prove a learner's reliability after examining a certain number of examples. One learner might make all its mistakes in the very beginning and quickly become reliable (though still making a few mistakes). On the other hand, another learner might keep making a moderate number of mistakes for a long time, before converging to the target concept. Clearly, depending on the domain requirements, one learner might be preferred over another. But mistake bound model cannot differentiate between the two learners.
2. Mistake bound model assumes the worst possible training examples. In reality, this might not be the case.

These issues motivated the *Probably Approximately Correct* (PAC) model of learnability, proposed by Leslie Valiant in 1984 [1]. Valiant was awarded the 2010 ACM Turing Award for his work on PAC learnability and other aspects of learning and computing.

- Mistake bound analysis focuses on worst case performance
  - Learn the true concept that works perfectly on any unseen data
- Bounds are not very tight
- Relaxing the expectations
  - Focus on Probably Approximately Correct (PAC) learnability

4 Motivating the Need for Analysis

Algorithm Producer
Has an algorithm to **distinguish between malignant and benign tumors** that gives **0% training error** and **5% error on one test data set** after **learning from one training data set**

Buyer
Wants an algorithm that can **distinguish between arbitrary types of tumors** which gives **0% training error** and **5% error rate on any test data set** after **learning from any training data**

4.1 Version Spaces

- **Consistent Learner**: A learner that makes 0 mistakes on training data
- How much training data is needed for a consistent learner to be PAC learnable?

$$VS_{\mathcal{H},D} = \{h \in \mathcal{H} | \forall \langle x, c(x) \rangle \in D; h(x) = c(x)\}$$

- Set of hypotheses with zero training error
- **Observation:** If the version space contains all acceptable (less than ϵ error rate) hypotheses, then we will definitely learn an acceptable hypothesis.
- To get a bound on number of training examples needed by a consistent learner
  - Need bound on the number of training examples to ensure $VS_{H,D}$ contains no *unacceptable* hypotheses

ϵ-Exhausted Version Space

- $VS_{H,D}$ is ϵ-exhausted with respect to c and D:
  - $\forall h \in VS_{H,D}$, $error_{\mathcal{D}}(h) < \epsilon$

4.2 Bounds for ϵ-Exhausted Version Space

Theorem 1. *If $\mathcal{H}$ is finite and D is a sequence of m training examples (randomly sampled from $\mathcal{D}$), probability that version space $VS_{\mathcal{H},D}$ is* **not** ϵ**-exhausted** *with respect to c is less than or equal to:*

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

The proof of the above theorem can be sketched as follows:

Proof. Let $h_1, h_2, \ldots, h_k$ be all hypotheses in $\mathcal{H}$ whose true error is greater than ϵ with respect to c. If any of these hypotheses have a zero training error, i.e., they are in the version space, then the version space will fail to be ϵ-exhausted.

Let p_i be the probability that h_i is consistent with m independently drawn examples in D. The probability that at least one of the k hypotheses will be consistent with all m training examples will be less than $\sum_i p_i$. Since for each of these hypotheses, the true error rate is greater than ϵ, it means that the probability of h_i to be consistent with any independently drawn example is at most $(1-\epsilon)$. Therefore, $p_i \leq (1-\epsilon)^m$. Therefore, the probability that the version space is not ϵ-exhausted is at most

$$k(1-\epsilon)^m$$

But $k \leq |\mathcal{H}|$, and hence the above probability is at most $|\mathcal{H}|(1-\epsilon)^m$. Making use of a general inequality: if $0 \leq \epsilon \leq 1$, then $(1-\epsilon) \leq e^{-\epsilon}$, we get:

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

□

- The theorem proves an upper bound on the probability that version space is not ϵ-exhausted, in terms of m, ϵ, and $|\mathcal{H}|$
- Bounds the probability that m training examples will fail to eliminate “bad” hypotheses
- For this probability to be below δ:

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

- Rearranging:

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

Analyzing Specific Concept Classes

- For any **consistent learner** that learns over a finite hypothesis space
- Number of training examples needed under PAC-learnability requirements:

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

- Guarantee: Learner will have error less than ϵ with probability $1-\delta$
- For 20 binary attributes, $\epsilon = 10\%$, $\delta = 5\%$:
  - $m \geq 250$ (**conjunctive concepts only**)
- For all concepts:
  - $m \geq 7.2million$
  - $m \gg |X| \approx 1million$!!!

If the input space is $\{0,1\}^{20}$ and the hypothesis space is the set of conjuctive hypotheses, i.e., $|\mathcal{H}| = 3^{20}+1$, then number of training examples needed to train a learner with error rate under 10% with at least 95% probability:

$$m \geq \frac{1}{0.1}(ln(3^{20}+1) + ln(1/0.05)) \approx 250$$

5 Agnostic Learning

- A zero training error hypothesis cannot be found
- Output h with *minimum training error*
- **Agnostic Learner**

Sample Complexity for Agnostic Learner

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

If the input space is $\{0,1\}^{20}$ and the hypothesis space is the set of conjuctive hypotheses, i.e., $|\mathcal{H}| = 3^{20}+1$, then number of training examples needed to train a learner with error rate under 10% with at least 95% probability, without assuming that $c \in \mathcal{H}$:

$$m \geq \frac{1}{0.1^2}(ln(3^{20}+1) + ln(1/0.05)) \approx 1248$$

6 Infinite Hypothesis Spaces

- Sample complexity: $m \propto ln(|\mathcal{H}|)$
- What if $|\mathcal{H}| = \infty$?
  - Linear hyperplanes?
- This is an issue for large $|\mathcal{H}|$ too.

6.1 Vapnik-Chervonenkis Dimension of $\mathcal{H}$

- Alternate measure of complexity of $\mathcal{H}$
- $VC(\mathcal{H})$ instead of $|\mathcal{H}|$
- Can be computed for infinite hypothesis spaces
- Gives *tighter* sample complexity bounds

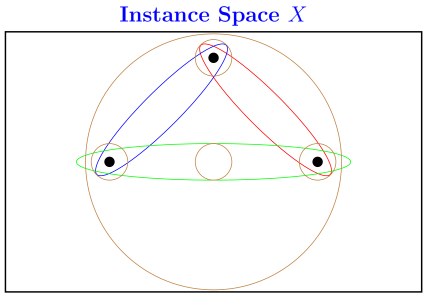
6.2 Understanding VC Dimension

- What does each $h \in \mathcal{H}$ do to a set of examples, $S \subseteq X$?
- How many ways can a set $S \subseteq X$ be partitioned?
- Is there an $h \in \mathcal{H}$ which represents a partition of S?

Each hypothesis labels the instances in a certain way, and essentially partitions S into two subsets: $\{x \in S : h(x) = 1\}$ and $\{x \in S : h(x) = 0\}$. A set S can be partitioned into two subsets in $2^{|S|}$ ways.

6.3 Shattering a Set of Instances

- $\mathcal{H}$ shatters $S \subseteq X$ iff every partitioning of S is represented by at least one $h \in \mathcal{H}$.



- Example 1:
  - Hypothesis space, $\mathcal{H}$ = All intervals (a, b) on a real number line
  - $S = \{3.4\}$?
  - $S = \{3.4, 4.7\}$?
  - $S = \{3.4, 7.2, 8.3\}$?
- $d = 1$
- $d = 2$

What Does Shattering Signify?

- Complexity of the hypothesis space
- If $\mathcal{H}$ does not shatter S, then there is a concept defined over S which cannot be "represented" by the hypothesis space.
- Closely related to the inductive bias of a hypothesis space.

Remember that an unbiased hypothesis space will shatter X because it contains all possible partitionings of X.

6.4 Definition of VC dimension for $\mathcal{H}$

- Size of the largest subset of X that can be shattered by $\mathcal{H}$

- Some common $VC(\mathcal{H})$:
  - For d-dimensional data and linear hyperplanes ($\mathcal{H}$):

$$VC(\mathcal{H}) = d + 1$$

Note that for any finite $\mathcal{H}$, $VC(\mathcal{H}) \leq \log_2 |\mathcal{H}|$. A data set with d instances can have 2^d possible partitions, each can be represented using one hypothesis. Thus for that hypothesis class, $VC(\mathcal{H}) = d$ and $2^d \leq |\mathcal{H}|$, or $d = VC(\mathcal{H}) \leq \log_2 |\mathcal{H}|$.

6.5 Sample Complexity and the VC Dimension

- Recall that:
$$m \geq \frac{1}{\epsilon}(ln|\mathcal{H}| + ln(\frac{1}{\delta}))$$
for the case where $c \in \mathcal{H}$.

- When $|\mathcal{H}| = \infty$:

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

6.6 Analyzing ML Algorithms

VC Dimension of Learning Lines (e.g., perceptrons)

- VC Dimension $= d + 1$

- Training data needed for PAC learnability:

$$m \geq \frac{1}{\epsilon}(8(d + 1)) \log_2(\frac{13}{\epsilon}) + 4log_2(\frac{2}{\delta}))$$

VC Dimension for "Neural Networks"

- For a acyclic layered network with s *perceptrons* with r inputs per perceptron
$$VC \leq 2(r + 1)s \log(es)$$

References

References

[1] L. G. Valiant. A theory of the learnable. *Commun. ACM*, 27(11):1134–1142, Nov. 1984.