18s1: COMP9417 Machine Learning and Data Mining

Lectures: Aspects of Learning Theory and Algorithm-Independent Machine Learning
Topic: Questions from lecture topics
Version: with answers

Last revision: Sun May 20 2018

Introduction

Some questions and exercises from the course lectures covering theoretical aspects of machine learning independent of particular algorithms. We start by generating a simple concept learning algorithm that is provably correct, then look at some of the ideas used to prove sample complexity in the PAC learning setting, then cover some results for the VC dimension of some concept classes, and finish with an example of the mistake bounds for aan attribute-efficient linear threshold classifier.

This tutorial note, together with the corresponding lecture notes, are intended to contain all the relevant material to enable you to answer all the questions, but you may wish to refer to the cited textbooks, for example [Blum et al., 2018], a draft copy of which is freely available online at `http://www.cs.cornell.edu/jeh/book.pdf` for additional background.

Question 1 ([Blum et al., 2018]) To get a sense of how learning theory characterises sample complexity we start by formulating a simple *consistent learner* to learn *disjunctions*, i.e., Boolean OR functions, of d variables. Recall that a consistent learner is just one that makes no mistake on the training data. The target concept c is assumed to be expressed as a disjunction of literals, where a literal is defined as some feature x_i being true (having value 1).

So an instance is just a set of literals, such as $\{\mathbf{x} | x_1 = 1 \vee x_3 = 1 \vee x_8 = 1\}$. For example, if the target concept was to distinguish between spam and non-spam emails, the presence in an email of any of the features x_1, x_3 or x_8 would be enough to classify it as spam, whereas the absence of all of them would mean non-spam.

Question 1a) The hypothesis space H is the set of all disjunctions of d features. What is the size of this hypothesis space ?

Answer
Since we can represent any disjunctive hypothesis as specified above simply as the set of the d features that are true in the hypothesis, the hypothesis space is the power set of literals, so it has size 2^d.

Question 1b) Give an algorithm for a consistent learner for such disjunctive concepts from a set of labelled noise-free training examples S. *HINT*: try adapting the basic approach of the Find-S algorithm to learn conjunctive concepts shown on slide 44 of the lecture notes.

Answer

Given the specification for learning disjunctive concepts, there is a straightforward algorithm.

Disjunctive Concept Learner:

- Initial hypothesis h is the set of all literals $x_i = 1$, $1 \leq i \leq d$
- For each negative instance x in S
  - Remove from h any literal $x_i = 1$
- Output concept that is the logical OR of the features remaining in h

---

Question 1c) Outline the steps in a proof that your disjunctive concept learning algorithm will find a consistent hypothesis h, i.e., that the error on sample S $error_S(h) = 0$.

Answer
Assume that the target concept c is in fact a disjunction. Then for any literal $x_i = 1$ in c, x_i will not be set to 1 in any negative example in S. So h will include $x_i = 1$. Since h will contain all such literals, h will correctly predict all positive examples in S. Furthermore, h will correctly predict negative on all negative examples in S since by design all features set to 1 in any negative example were discarded. Therefore, h is correct on all examples in S.

Question 1d) Analyse the sample complexity of the consistent learner for disjunctive concepts in the PAC learning setting, i.e., use the formula from slide 23 in the lecture notes.

Answer

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

Since we know from above that the size of the hypothesis space $|H|$ is 2^d we obtain

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

Question 2 Consider how you could prove the theorem bounding the probability that a consistent learner will output a hypothesis h with $error(h) \geq \epsilon$ that appears on slides 19–23 of the lecture notes.

Theorem:

> If the hypothesis space H is finite, and D is a sequence of $m \geq 1$ independently drawn random examples of some target concept c, then for any $0 \leq \epsilon \leq 1$, the probability that the version space with respect to H and D is not ϵ-exhausted (with respect to c) is less

than

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

Background Proofs of this theorem are in Chapter 7 of [Mitchell, 1997] and Chapter 5 of [Blum et al., 2018]. These proofs use the following facts:

- For events $A_1, A_2, \ldots, A_n$, the probability of the union of these events $Pr(A_1 \cup A_2 \cup \ldots \cup A_n) \leq \sum_{i=1}^{n} Pr(A_i)$ (this is called the "union bound")
- If $0 \leq \epsilon \leq 1$ then $(1 - \epsilon) \leq e^{-\epsilon}$ (from Taylor series)

Answer

Proof:
Suppose $h_1, h_2, \ldots, h_k$ are all hypotheses in the hypothesis space H with *true error* $error_{\mathcal{D}}(h_i) > \epsilon$.

Now the version space $VS_{H,D}$ is NOT ϵ-exhausted if *at least one* of the hypotheses h_i is consistent with all m examples in the training set D.

Since each hypothesis has true error $h_i > \epsilon$, an individual example of the target concept c is consistent with h_i with probability at most $1 - \epsilon$. So the probability of a hypothesis with true error $h_i > \epsilon$ being consistent with *all* m examples of the target concept is at most $(1 - \epsilon)^m$. Since we have k such hypotheses, this probability for all k hypotheses is at most $k(1 - \epsilon)^m$, by the union bound.

Now $k \leq |H|$, so we get $|H|(1 - \epsilon)^m$. Finally we use the inequality $(1 - \epsilon) \leq e^{-\epsilon}$ to get

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

which completes the proof.

Question 3 Refer to the VC dimension example for linear classifiers in the 2-dimensional x, y plane of slides 39–41 of the lecture notes. Answer the following:

1. give an intuitive argument for why the VC dimension must be at least 3;
2. suppose you have a set of 3 points that are collinear – does that change your argument ?
3. can the VC dimension be 4 ?

Answer

1. in the first 3 diagrams on slide 39 are shown sets of 3 points that can be shattered, i.e., a linear decision surface can be drawn for each of the 2^3 subsets of the points;
2. no – although a set of 3 collinear points cannot be shattered, as long as *at least one* set of 3 points can be shattered, the VC dimension must be at least 3;
3. to show that $VC(H) < d$, we must show that *no* set of size d can be shattered, and in this setting no sets of size four can be shattered, so $VC(H) = 3$ (there is always an XOR-type problem).

Question 4 With reference to slides 52–54 of the lecture notes, outline a version of the Halving Algorithm for Boolean functions, and wih reference to it give the worst-case mistake bounds, and an intuitive explanation of why this bound holds. Now repeat this for the best-case performance!

Background Key points about the mistake bounds framework:

- this is intuitively based on a "learning curve" idea
- it is based on an online-learning framework, but can be adapted for batch learning too
- it is closely related to PAC learning, boosting, and other theoretical frameworks

---

Answer

Without loss of generality, suppose that the hypothesis space H is a set of Boolean functions.

```
Halving Algorithm:

• Initialise the set of consistent hypotheses $C = H$
• Repeat get new instance $x$
    – let $\pi_0(C, x)$ be subset of $C$ that predict 0
    – let $\pi_1(C, x)$ be subset of $C$ that predict 1
    – if $|\pi_0(C, x)| > |\pi_1(C, x)|$ predict 0 else 1
    – if class($x$) = 0 then $C = \pi_0(C, x)$ else $C = \pi_1(C, x)$
```

Worst-case mistake bound: on every example x the majority vote prediction is the opposite of the actual class(x), so this is a mistake. However, on each prediction for x, the majority $\pi_0(C, x)$ (respectively $|\pi_1(C, x)|$) of functions are eliminated. Therefore the size of the set C will be (at least) reduced by half on every example x. That is, the number of mistakes $M_{Halving} \leq \log_2 |H|$.

However, in the best case the situation is reversed! On every example x the majority vote prediction is correct, so the number of mistakes is zero!

Question 5 Work through applying the Winnow 2 algorithm on slides 55–57 of the lecture notes to the examples below *in the order in which they appear.*

Use the settings: $\alpha = 2$, $\theta = 2$; and initialise all weights $w_i = 1$. Show all predictions, and whether a mistake has occurred or not. When the algorithm has passed through the examples, do you think the target concept has been learned ?

Example	$\mathbf{x_1}$	$\mathbf{x_2}$	$\mathbf{x_3}$	$\mathbf{x_4}$	$\mathbf{x_5}$	$\mathbf{x_6}$	$\mathbf{x_7}$	$\mathbf{x_8}$	$\mathbf{x_9}$	$\mathbf{x_{10}}$	**Class**
1)	0	1	0	1	0	1	1	0	1	1	$\oplus$
2)	1	0	1	0	1	0	1	1	1	0	$\ominus$
3)	0	0	0	1	0	0	0	0	1	0	$\oplus$
4)	1	0	0	0	1	1	1	1	1	0	$\ominus$
5)	1	0	1	0	1	1	1	1	0	1	$\ominus$

Answer

First we show the weights following prediction of each example (this is *after* promotion or demotion once the correct classification is known):

Example	$\mathbf{w_1}$	$\mathbf{w_2}$	$\mathbf{w_3}$	$\mathbf{w_4}$	$\mathbf{w_5}$	$\mathbf{w_6}$	$\mathbf{w_7}$	$\mathbf{w_8}$	$\mathbf{w_9}$	$\mathbf{w_{10}}$
1)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00	1.00
2)	0.50	1.00	0.50	1.00	0.50	1.00	0.50	0.50	0.50	1.00
3)	0.50	1.00	0.50	2.00	0.50	1.00	0.50	0.50	1.00	1.00
4)	0.25	1.00	0.50	2.00	0.25	0.50	0.25	0.25	0.50	1.00
5)	0.125	1.00	0.25	2.00	0.125	0.25	0.125	0.125	0.50	0.50

Recall that the prediction $\hat{y} = \oplus$ if $\mathbf{w} \cdot \mathbf{x} > \theta$ otherwise $\ominus$.
Here is the trace of the predictions:

Example	Prediction	Mistake
1)	$\hat{y} = 6 > 2 = \oplus$	no mistake
2)	$\hat{y} = 6 > 2 = \oplus$	mistake
3)	$\hat{y} = 1.5 \leq 2 = \ominus$	mistake
4)	$\hat{y} = 4 > 2 = \oplus$	mistake
5)	$\hat{y} = 3.0 > 2 = \oplus$	mistake

The general result for WINNOW-type algorithms is that for r relevant features out of n total features, the worst-case mistake bound is $\mathcal{O}(r \log n)$. In this case $r \log n \approx 2 \times 2.3 = 4.6$, so we are within mistake bounds on this toy example.
A quick check reveals that after the last weight update the algorithm has converged to a correct solution for the training data. To generate this dataset the target concept $(x_2 \vee x_4)$ was used. Evidently we have not quite converged to this target concept (for this to be the case, weights $\mathbf{w_2}$ and $\mathbf{w_4}$ would both have to be greater than 2) so Winnow is still learning, but it is moving in the right direction.

Example	Prediction	Mistake
1)	$\hat{y} = 4.375 > 2 = \oplus$	no mistake
2)	$\hat{y} = 1.25 \leq 2 = \ominus$	no mistake
3)	$\hat{y} = 2.5 > 2 = \oplus$	no mistake
4)	$\hat{y} = 1.25 \leq 2 = \ominus$	no mistake
5)	$\hat{y} = 1.5 \leq 2 = \ominus$	no mistake

References

[Blum et al., 2018] Blum, A., Hopcroft, J., and Kannan, R. (2018). *Foundations of Data Science.*

[Mitchell, 1997] Mitchell, T. (1997). *Machine Learning.* McGraw-Hill, New York.