Answers Machine Learning Exercises 2

Tim van Erven

October 17, 2007

Exercises

1. Consider the LIST-THEN-ELIMINATE algorithm for the EnjoySport example with hypothesis space

$$\begin{aligned}\mathcal{H} = \{&\langle ?,?,?,?,?,? \rangle, \langle \text{Sunny},?,?,?,?,? \rangle,\\ &\langle \text{Cloudy},?,?,?,?,? \rangle, \ldots, \langle \emptyset,\emptyset,\emptyset,\emptyset,\emptyset,\emptyset \rangle\},\end{aligned}$$

as described on slide 19 of `mlslides3.pdf`. Please give an example of a hypothesis (i.e. a function from feature vectors to class labels) that is not contained in $\mathcal{H}$.

Answer: Recall from class that $\mathcal{H}$ contains only 973 hypotheses out of the 2^{96} hypotheses that are possible. There are therefore many possible answers. Here are two possibilities. The second answer is given literally in Section 2.7.1 of Mitchell.

Let $\mathbf{x}$ denote a 6-dimensional feature vector in the EnjoySport domain, where x_1 encodes the value of the attribute Sky, x_2 encodes AirTemp, etc. Then, for example, the following hypothesis h_1 is not a member of $\mathcal{H}$:

$$h_1(\mathbf{x}) = \begin{cases} \text{Yes} & \text{if } x_1 = \text{`Sunny' and } x_2 = \text{`Cold'},\\ \text{Yes} & \text{if } x_1 = \text{`Cloudy' and } x_2 = \text{`Warm'},\\ \text{No} & \text{otherwise.}\end{cases}$$

To see this, note that the first case implies that 'Sunny' and 'Cold' must be allowed values for the attributes Sky and AirTemp, respectively, and that the constraints for all the other attributes must be '?'. Likewise the second case implies that 'Cloudy' and 'Warm' should be allowed values for respectively Sky and AirTemp. The only constraint on Sky or AirTemp that is consistent with these requirements is '?'. Hence the only hypothesis in $\mathcal{H}$ that is consistent with the first two cases must be $\langle ?,?,?,?,?,? \rangle$. But that means that the combination x_1 = 'Cloudy' and x_2 = 'Cold' would also be allowed as part of the concept, which is inconsistent with the third case. Thus h_1 is not a member of $\mathcal{H}$. We may construct many similar hypotheses that are not contained in $\mathcal{H}$ by noting that $\mathcal{H}$ cannot represent dependencies between attributes, like for example: If x_1 = 'Sunny', then x_2 should be 'Cold'; If x_1 = 'Cloudy', then x_2 should be 'Warm'. Most of the hypotheses that are not in $\mathcal{H}$ are of this form.

Another example of a hypothesis, h_2, that is not a member of $\mathcal{H}$ is:

$$h_2(\mathbf{x}) = \begin{cases} \text{Yes} & \text{if } x_1 = \text{`Sunny' or } x_1 = \text{`Cloudy'}, \\ \text{No} & \text{otherwise.} \end{cases}$$

To see this, consider which constraint on x_1 would fit with h_2. Suppose this constraint would be '?'. Then the hypothesis would either classify some $\mathbf{x}$ with x_1 = 'Rainy' as 'Yes' or it would classify some $\mathbf{x}$ with x_1 = 'Sunny' as 'No' (if some other constraint were set to '$\emptyset$'). And if the constraint were set to 'Sunny', 'Cloudy', 'Rainy' or '$\emptyset$', then there would exist a feature vector with either x_1 = 'Sunny' or x_1 = 'Rainy' that would not be classified as 'Yes'. Hence h_2 is not a member of $\mathcal{H}$. In this case we have exploited the fact that Sky has three possible values, and the constraints cannot represent the fact that only two of them are allowed.

Grading:

- 2 points for a correct example.
- An explanation, although appreciated, is not required, because it is easy to verify whether or not a given example is a member of $\mathcal{H}$.

2. Suppose we are given the data in Table 1. Consider the LIST-THEN-ELIMINATE algorithm with the hypothesis space $\mathcal{H}$ consisting of all possible **decision trees** for the EnjoySport domain. Given the data in the top three rows of the table, how would this algorithm classify the last example in the table, where there is a '?' instead of the label?

Table 1: EnjoySport Data

		$\mathbf{x}$				y
Sky	**AirTemp**	**Humidity**	**Wind**	**Water**	**Forecast**	**EnjoySport**
Sunny	Warm	Normal	Strong	Warm	Same	Yes
Sunny	Warm	High	Strong	Warm	Same	Yes
Rainy	Cold	High	Strong	Warm	Change	No
Sunny	Warm	High	Strong	Cool	Change	?

Answer: The LIST-THEN-ELIMINATE algorithm would not be able to classify this new example, because $\mathcal{H}$ contains all possible hypotheses (it is an unbiased hypothesis space) and the features of the new example are not present in the previous training data.

Grading:

- 1 point for noting that the hypothesis space contains all possible hypotheses. This is the critical observation.
- 1 point for the observation that the new example cannot be classified by the LIST-THEN-ELIMINATE algorithm.

3. [This is Exercise 3.1 from Mitchell.] Consider classification of data in which the 4-dimensional feature vector $\mathbf{x}$ contains four Boolean features: A, B, C and D. (Boolean means that a feature can only take two possible values: 0 and 1, which we may think of as 'False' and 'True', respectively.) Furthermore, the class label y is also Boolean. Give decision trees to represent the following Boolean functions[1]:

   (a) $y = A \wedge \neg B$
   (b) $y = A \vee (B \wedge C)$
   (c) $y = A \otimes B$
   (d) $y = (A \wedge B) \vee (C \wedge D)$

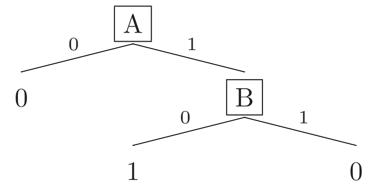
   The symbol '$\otimes$' denotes the **exclusive or** of its arguments. Its truth table is in Table 2.

Table 2: Truth table of exclusive or.

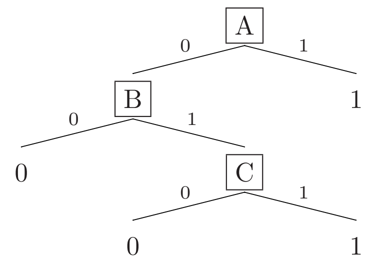
A	B	$A \otimes B$
0	0	0
0	1	1
1	0	1
1	1	0

Answer: Different answers are possible. For example:

(a)

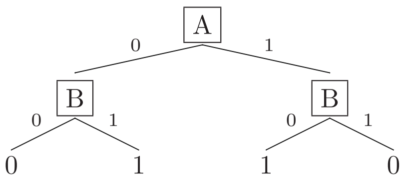


(b)

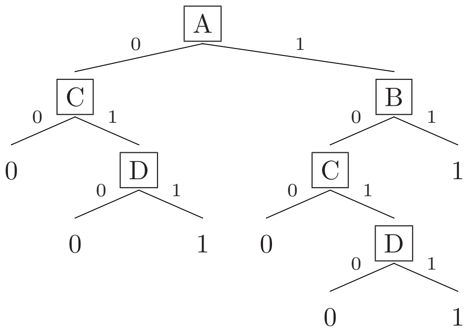


[1] I suggest that you draw these trees in some drawing program like, for example, Photoshop (on Windows) or the Gimp, Inkscape, or Xfig (on Linux). It is okay if your trees don't look pretty, but they should be readable without too much effort.

(c)



(d)



Grading:

- 0.75 points for each correct decision tree.

4. Suppose that $\Omega = \{\omega_1, \ldots, \omega_k\}$ is a sample space and that p and q are both probability mass functions on Ω. Let P and Q denote the probability distributions on Ω that are defined by p and q, respectively. Show that if $p \neq q$, then $P \neq Q$.

Answer: What it means when two functions are different: Note that p, q, P and Q are all functions. p and q take a single outcome $\omega \in \Omega$ as input. P and Q take an event $\mathcal{E} \subseteq \Omega$ as input. Two functions that have the same domain (set of possible inputs) and range (set of possible outputs) are different if and only if they assign a different value to at least one input.

Answer to the exercise: If $p \neq q$, then there exists at least one outcome $\omega \in \Omega$ such that $p(\omega) \neq q(\omega)$. It follows that, for this specific outcome ω,

$$P(\{\omega\}) = p(\omega) \neq q(\omega) = Q(\{\omega\})$$

and hence that P and Q are different, because they assign a different value to the event $\{\omega\}$.

Grading:

- 2 points for any correct proof.
- If the proof is not correct, then the each of the following observations still gives 0.5 points:
  - $p \neq q$ implies that there exists at least one $\omega \in \Omega$ such that $p(\omega) \neq q(\omega)$.
  - $p(\omega) \neq q(\omega)$ implies $P(\{\omega\}) \neq Q(\{\omega\})$.

Grading Policy

- Grades are between 1 and 10.
- You always start with 1 point.
- Partial points may be awarded for partially correct exercises.