Set 1

Claire Goeckner-Wald

October 3, 2016

The Learning Problem

1) D. (i) not learning, (ii) supervised learning, (iii) reinforcement learning

The first is not learning, because the statistical model is already generated. The developers are not using machine learning to create a statistical model. The second is supervised learning because the algorithm is given some (input, output) and uses this known data to infer decision boundaries to classify future coin inputs. The third is reinforcement learning because the algorithm receives a sort of grade (or, penalization) for the poor decisions it makes. It learns from these grades.

2) A. (ii) and (iv)

Classifying numbers into primes is not machine learning because we can do it mathematically. Determining the time it takes for an object to hit the ground is mathematically known, as well. Thus, (i) and (iii) are not applicable.

Detecting potential credit card fraud can use machine learning because there is not necessarily a 'best' mathematical model for weighting and accounting for suspicious activities. An algorithm here could use past data (the data of the card-holders as well as whether or not they were victims of fraud) to predict future fraud. Creating an optimal cycle for traffic lights at an intersection does not also necessarily have a best model- in this case, an algorithm may use time of day and number of cars currently waiting to develop its own model, using reinforcement learning.

Bins and Marbles

3) D. 2/3

All of the black balls are equally likely to be drawn at the beginning. Call the socks Black-black and Black-white. We draw a black ball. There are three black balls we could have drawn; two are in sock Black-black, and one is in sock Black-white. Since all of the balls were equally likely, the probability that we are holding the Black-black sock after drawing a black ball is $\frac{2}{3}$. The probability that we are holding the

Black-white sock is $\frac{1}{3}$. Thus:

$$\text{P(holding sock Black-black)} = \frac{2}{3}$$
$$\text{P(holding sock Black-white)} = \frac{1}{3}$$
$$\text{P(holding sock Black-black, that the next ball is black)} = 1$$
$$\text{P(holding sock Black-white, that the next ball is black)} = 0$$
$$\text{P(next ball is black)} = \text{P(holding sock Black-black)} \times 1 + \text{P(holding sock Black-white)} * 0 = \frac{2}{3} \times 1 + \frac{1}{3} * 0 = \frac{2}{3}$$

4) B. 3.405×10^{-4}

$$\text{P(draw one red)} = 0.55$$
$$\text{P(draw one green)} = 1 - \text{P(draw one red)} = 0.45$$
$$\text{P(v = 0 in one sample)} = \text{P(draw one green)}^{10} = (0.45)^{10} = 0.0003405$$

5) C. 0.289

$$\text{P(v = 0 in one sample)} = (0.45)^{10}$$
$$\text{P(v} \geq \text{1 in one sample)} = 1 - (0.45)^{10}$$
$$\text{P(v} \geq \text{1 in one thousand samples)} = (1 - (0.45)^{10})^{1000}$$
$$\text{P(v = 0 in at least one of one thousand samples)} = 1 - \left((1 - (0.45)^{10})^{1000}\right) \approx 0.289$$

Feasibility of Learning

6) The possible target function outputs are as follows:

(0,0,0)	(0,0,1)	(0,1,0)	(0,1,1)	(1,0,0)	(1,0,1)	(1,1,0)	(1,1,1)
0	1	1	0	1	0	0	0
0	1	1	0	1	0	0	1
0	1	1	0	1	0	1	0
0	1	1	0	1	0	1	1
0	1	1	0	1	1	0	0
0	1	1	0	1	1	0	1
0	1	1	0	1	1	1	0
0	1	1	0	1	1	1	1

[a] g returns 1 for all three points

$$\text{Score} = 3 * (1) + 2 * (3) + 1 * (3) + 0 * (1) = 12$$

[b] g returns 0 for all three points

$$\text{Score} = 3 * (1) + 2 * (3) + 1 * (3) + 0 * (1) = 12$$

[c] g is the XOR function applied to x, i.e., if the number of 1s in x is odd, g returns 1; if it is even, g returns 0.

$$\text{Score} = 3 * (1) + 2 * (3) + 1 * (3) + 0 * (1) = 12$$

[d] g returns the opposite of the XOR function: if the number of 1s is odd, it returns 0, otherwise returns 1.

$$\text{Score} = 3 * (1) + 2 * (3) + 1 * (3) + 0 * (1) = 12$$

The answer is [e] because they all have equivalent scores.

The Perceptron Learning Algorithm

7) B. 15

8) C. 0.1

9) B. 100

10) B. 0.01