Homework # I

Daniel dos Santos
Universidade Federal do Rio de Janeiro

April 24, 2023

The Learning Problem

Problem 01

What types of Machine Learning, if any, best describe the following three scenarios:

(i) A coin classification system is created for a vending machine. The developers obtain exact coin specifications from the U.S. Mint and derive a statistical model of the size, weight, and denomination, which the vending machine then uses to classify coins.

(ii) Instead of calling the U.S. Mint to obtain coin information, an algorithm is presented with a large set of labeled coins. The algorithm uses this data to infer decision boundaries which the vending machine then uses to classify its coins.

(iii) A computer develops a strategy for playing Tic-Tac-Toe by playing repeatedly and adjusting its strategy by penalizing moves that eventually lead to losing.

[a] (i) Supervised Learning, (ii) Unsupervised Learning, (iii) Reinforcement Learning

[b] (i) Supervised Learning, (ii) Not learning, (iii) Unsupervised Learning

[c] (i) Not learning, (ii) Reinforcement Learning, (iii) Supervised Learning

[d] (i) Not learning, (ii) Supervised Learning, (iii) Reinforcement Learning

[e] (i) Supervised Learning, (ii) Reinforcement Learning, (iii) Unsupervised Learning

Solution

(i) The developers can leverage the known characteristics of a coin to inform their decision-making process, without needing to learn from any data.
(ii) Labeled data can be used to train a model to approximate an unknown function in a supervised learning setting, where the model learns to make predictions based on pre-labeled inputs and outputs.
(iii) The Tic-Tac-Toe example demonstrates how an agent can improve its decision-making abilities through reinforcement learning, by receiving feedback in the form of rewards or penalties as it interacts with the environment.

Answer: **[d]**

Problem 02

Which of the following problems are best suited for Machine Learning?

(i) Classifying numbers into primes and non-primes.

(ii) Detecting potential fraud in credit card charges.

(iii) Determining the time it would take a falling object to hit the ground.

(iv) Determining the optimal cycle for traffic lights in a busy intersection.

[a] (ii) and (iv)

[b] (i) and (ii)

[c] (i), (ii), and (iii)

[d] (iii)

[e] (i) and (iii)

Solution

(i) There is a deterministic way of knowing if a number is prime or not, making this problem unsuited for Machine Learning;
(ii) By using data from past good and bad payors you can create a model that analyzes a set of features, this way you can effectively predict credit risk for a given client;
(iii) A mathematical formula can solve this problem without requiring the use of Machine Learning. (iv) The optimization of traffic lights in a busy intersection can be accomplished by using historical traffic data to train machine learning models, as well as by conducting simulations that can inform the development of reinforcement learning algorithms.

Answer: **[a]**

Bins and Marbles

Problem 03

We have 2 opaque bags, each containing 2 balls. One bag has 2 black balls and the other has a black ball and a white ball. You pick a bag at random and then pick one of the balls in that bag at random. When you look at the ball, it is black. You now pick the second ball from that same bag. What is the probability that this ball is also black?

[a] 1/4

[b] 1/3

[c] 1/2

[d] 2/3

[e] 3/4

Solution

Let $A =$ "First ball is BLACK" and $B =$ "Second ball is BLACK". We need to calculate the probability of the event B given that the event A has occurred, i.e., $P(B|A)$.
We can use the formula:

$$P(B|A) = \frac{P(A \cap B)}{P(A)}$$

Using the Law of Total Probability, we can calculate $P(A)$ as:

$$\begin{aligned} P(A) &= P(A|\text{"Bag 1"})P(\text{"Bag 1"}) + P(A|\text{"Bag 2"})P(\text{"Bag 2"}) \\ &= 1 * \frac{1}{2} + \frac{1}{2} * \frac{1}{2} \\ &= \frac{3}{4} \end{aligned}$$

Similarly, we can calculate $P(A \cap B)$ as:

$$\begin{aligned} P(A \cap B) &= P(A \cap B|\text{"Bag 1"})P(\text{"Bag 1"}) + P(A \cap B|\text{"Bag 2"})P(\text{"Bag 2"}) \\ &= 1 * \frac{1}{2} + 0 * \frac{1}{2} \\ &= \frac{1}{2} \end{aligned}$$

Substituting these values, we get:

$$\begin{aligned} P(B|A) &= \frac{P(A \cap B)}{P(A)} \\ &= \frac{1}{2} * \frac{4}{3} \\ &= \frac{2}{3} \end{aligned}$$

Answer: **[d]** $\frac{2}{3}$

Problem 04 and Problem 05

Consider a sample of 10 marbles drawn from a bin containing red and green marbles. The probability that any marble we draw is red is $\mu = 0.55$ (independently, with replacement). We address the probability of getting no red marbles ($\nu = 0$) in the following cases:

Problem 04

We draw only one such sample. Compute the probability that $\nu = 0$. The closest answer is ('closest answer' means: $\|$your answer $-$ given option$\|$ is closest to 0):

[a] 7.331×10^{-6}

[b] 3.405×10^{-4}

[c] 0.289

[d] 0.450

[e] 0.550

Solution

$P(\nu = 0) = (1 - \mu)^{10} = (1 - 0.55)^{10} = 3.405 \times 10^{-4}$

Answer: **[b]** 3.405×10^{-4}

Problem 05

We draw $1,000$ independent samples. Compute the probability that (at least) one of the samples has $\nu = 0$. The closest answer is:

[a] 7.331×10^{-6}

[b] 3.405×10^{-4}

[c] 0.289

[d] 0.450

[e] 0.550

Solution

From the previous problem, we know that the probability of one sample of size 10 not having any red marble is $P(\nu = 0) = 3.405 \times 10^{-4}$. Let X be a random variable that counts the number of samples of the same size that have no red marbles ($\nu = 0$) in 1000 repetitions:

$$X \sim Bin(1000; 3.405 \times 10^{-4})$$

The probability of any of the 1000 samples not having any red marble is:

$$\begin{aligned} P(X = 0) &= \binom{1000}{0}(3.405 \times 10^{-4})^0(1 - 3.405 \times 10^{-4})^{1000} \\ &= (1 - 3.405 \times 10^{-4})^{1000} \\ &= 0.7113\ldots \end{aligned}$$

Now we can calculate the complement of the probability to find the probability of at least one sample having at least one red marble in it:

$$1 - P(X = 0) = 1 - 0.7113... = 0.2886... \approx 0.289$$

Therefore, the probability of at least one sample from the 1000 having at least one red marble is approximately 0.289.

Answer: **[c]** 0.289

Feasibility of Learning

Consider a Boolean target function over a 3-dimensional input space $\mathcal{X} = \{0, 1\}^3$ (instead of our ± 1 binary convention, we use $0, 1$ here since it is standard for Boolean functions). We are given a data set $\mathcal{D}$ of five examples represented in the table below, where $y_n = f(x_n)$ for $n = 1, 2, 3, 4, 5$.

x_n			y_n
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1

Note that in this simple Boolean case, we can enumerate the entire input space (since there are only $2^3 = 8$ distinct input vectors), and we can enumerate the set of all possible target functions (there are only $2^{2^3} = 256$ distinct Boolean function on 3 Boolean inputs). Let us look at the problem of learning f. Since f is unknown except inside $\mathcal{D}$, any function that agrees with $\mathcal{D}$ could conceivably be f. Since there are only 3 points in $\mathcal{X}$ outside $\mathcal{D}$, there are only $2^3 = 8$ such functions. The remaining points in $\mathcal{X}$ which are not in $\mathcal{D}$ are: 101, 110, and 111. We want to determine the hypothesis that agrees the most with the possible target functions. In order to quantify this, count how many of the 8 possible target functions agree with each hypothesis on all 3 points, how many agree on just 2 of the points, on just 1 point, and how many do not agree on any points. The final score for each hypothesis is computed as follows:

Score *= (# of target functions agreeing with hypothesis on all* 3 *points)*$\times 3$ *+ (# of target functions agreeing with hypothesis on exactly* 2 *points)*$\times 2$ *+ (# of target functions agreeing with hypothesis on exactly* 1 *point)*$\times 1$ *+ (# of target functions agreeing with hypothesis on* 0 *points)*$\times 0$*.*

Problem 06

Which hypothesis g agrees the most with the possible target functions in terms of the above score?

- [**a**] g returns 1 for all three points.
- [**b**] g returns 0 for all three points.
- [**c**] g is the XOR function applied to x, i.e., if the number of 1s in $\mathbf{x}$ is odd, g returns 1; if it is even, g returns 0.
- [**d**] g returns the opposite of the XOR function: if the number of 1s is odd, it returns 0, otherwise returns 1.
- [**e**] They are all equivalent (equal scores for g in [**a**] through [**d**]).

Solution

Answer: [**F**]

The Perceptron Learning Algorithm