CISC 5800 – Machine Learning

Homework 0

Due January 27 and 30

Submit Parts A and B through Blackboard by 11:59pm January 27;
Submit Part C on your erdos account by 11:59pm January 30 (see Part C instructions below).

A. Probability:

1. Consider four multi-valued random variables A (age), O (oxygen level), S (salary), T (body temperature), and W (weight). **We know that S is independent of O and T; A is independent of T.** We are provided the probability tables/functions for the following six joint, marginal, and conditional probabilities.

The six probabilities:

P(S|W)	**P(A, T)**	**P(S, W)**
P(S|O,W)	**P(O|T)**	**P(O, T, W)**

For example, we could be told:
P(S=$25K | W=heavy) = 0.3, P(S=$50K | W=heavy)=0.3, P(S=$75K | W=heavy)=0.25,
P(S=$25K | W=light)=0.2, etc…

We are **not** provided any other probability tables; for example, we are not given values for:
P(T=hi) or P(A=old, O=low)

Explain how to combine the six five probabilities from above (and the knowledge that S is independent) to compute each probability below, or write "not possible" if it is not possible.
For example: P(A) = $\sum_t P(A, T = t)$

a) P(W=mid-weight|S=$50K)
b) P(T=low, W=high)
c) P(A=old)
d) P(S=$50K | O=low, T=low)
e) P(O=high, S=$75K)
f) P(A=mid-age, S=$100K, T=medium)

2. Consider the following joint probability table:

A	B	C	P(A,B,C)
0	0	0	0.21
0	0	1	0.07
0	1	0	0.11
0	1	1	0.05
1	0	0	0.15
1	0	1	0.27
1	1	0	0.02

a) What is P(A=1, B=1, C=0)?
b) What is P(A=1, C=1)?
c) What is P(A=0 or C=0)?
d) What is P(B=0)?
e) What is P(C=1|A=0)?
f) If C=0, is A independent of B? In other words, does P(A,B|C=0) = P(A|C=0) P(B|C=0) ? Show your reasoning.

B. Algebra/Calculus

Express x as a function of *a* and *b*, as simplified as possible.
Example question: 3a=6x+7b
Example answer: $x = \frac{3a-7b}{6}$

1. $\frac{11x^4}{3b^2} = 6(a^2 + 4a - x^4)$

2. $10 = \sum_{i=0}^{3}(a_i + 2^i x)$
simplify this as much as possible, remove the summation sign, note if j=3, x^j is "x cubed"

3. $a = \sum_{i=0}^{1} 5(b_i x)^{1-i}$

Consider the function $f(x) = \exp(2x^3 - 6x) = e^{2x^3-6x}$
4. What is the derivative of f(x) (the derivative with respect to x, $\frac{d}{dx}$)?

5. For what value(s) of x is f'(x)=0?

6. At the value you found above, will f(x) have its smallest possible value?

Consider the function g(x,y)= $\sum_{i=1}^{3}\left(x^i + y^{2i}\right)$ (Note, for example, $y^4 = y \times y \times y \times y$
7. What is the value of g(x,y) when x=2, y=1?

8. What is the derivative of g(x,y) with respect to y : $\frac{d}{dy}g(x,y)$?

C. Programming:

Use Python to solve the following tasks.

Submission instructions for Part C: Log into your erdos account (erdos.dsm.fordham.edu) – you can use Terminal on Mac or Putty or MobaXTerm on Windows (see Resources section on our course web site). Inside your folder called "private"

Linux command: `cd private`

create a folder called "CIS5800".

Linux command: `mkdir CIS5800`

Save the three programs, inside private/CIS5800/ in the file hw0.py . As course instructor, I will be able to access your files inside private/CIS5800/. You must have the necessary files in the proper directory by the deadline at the top of the homework.

You are welcome to write your programs on your local computer (or on erdos). To transfer files from your local computer to erdos, you may use a program such as FileZilla `https://filezilla-project.org/` . **Make sure you transfer your files into your private/CIS5800/ directory!** Connect to erdos using port 22.

If you have trouble accessing erdos for this assignment, you may e-mail me your programs by at the top of the homework – however, we will use erdos for code submission throughout the rest of the semester, so you must resolve your erdos troubles by the time the next homework is due!

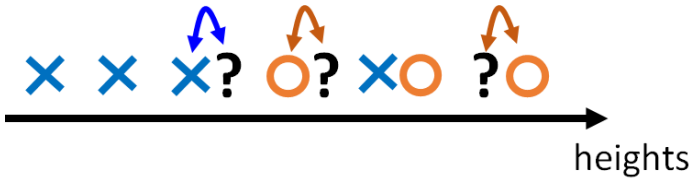
We will consider the world-famous problem of giraffe classification, discussed in the first lecture. We will make a very simple classifier based on animal height, and assign values for missing feature values.

1. Write a function called **`neighborClassify`** that takes in a 1D numpy array of numbers (the heights of unknown animals) and a 2D numpy array containing the heights of known animals. The function will return a list of 0s and 1s – a 0 for each non-giraffe input and a 1 for each giraffe input – using nearest neighbors classification (see below). Specifically, the function call **must look like this**:

```
neighborClassify(featureArray, trainArray)
```

`featureArray` will be a numpy array of shape (n) (where there are arbitrary number *n* animals to classify) and `trainArray` is a numpy array of shape (n,2) where each row contains first the height of a training animal and then its corresponding class (0 for non-giraffe, 1 for giraffe). Specifically, if `featureArray=np.array([6,3,9])` and `trainArray=np.array([[0.5,0], [1.5,0], [2.5,0], [4.5,1], [5,1], [7.5, 0], [8,1], [9.2,1]])`, the function will return the list `[1, 0, 1]`.

Classification is done by the nearest neighbors approach. Each test input is given the label of the nearest input in the training set. Below is a graphical example of this approach.



2. Write a function called **`findPrecision`** that takes in a list of approximated class labels output by the classifier (threshClassify) and a list of true labels provided in the training set, and calculates the precision of the classifier (for class 1) as a number between 0 and 1. Specifically, the function call **must look like this:**

```
findPrecision(classifierOutput, trueLabels)
```

If `classifierOutput=[0,1,1,0,0,1,0]` and `trueLabels=[1,1,0,0,0,1,1]`, the function will return the number `0.6667` (2 out of 3 of the values labeled as class 1 were actually in class 1).

3. Write a function called **`removeBlanks`** that takes in a `featureArray` (n,2) numpy array and returns a (m,2) numpy array where any row with a missing value is removed from the data set.~~all missing values are filled in with the number 2.~~ "Missing values" are any entry that is 0.

Specifically, the function call **must look like this:**

```
featArrayCorrected=removeBlanks(featArray)
```

If `featArray=np.array([[0,5],[0,3],[1,8],[10,0],[4,4]])` the function will return `np.array([[1,8],[4,4]])` into `featArrayCorrected`