ASSIGNMENT #2 – COMP 3106 ARTIFICIAL INTELLIGENCE

The assignment is an opportunity to demonstrate and solidify your knowledge on Bayes theorem and rule-based systems.

The assignment may be completed individually, or it may be completed in small groups of two or three students. The expectations will not depend on group size (i.e. same expectations for all group sizes).

Components

The assignment should contain two components: an implementation and a technical document.

Implementation

The implementation should provide a complete solution to the problem outlined below.

Technical Document

Your technical document should outline the algorithms you used and answer any questions posed below. After reading the technical document, the reader should understand exactly how your implementation works.

The technical document will be graded both on content and on presentation quality.

If the assignment was completed in a small group of students, the technical document must include a statement of contributions. This statement should identify: (1) whether each group member made significant contribution, (2) whether each group member made an approximately equal contribution, and (3) exactly which aspects of the assignment each group member contributed to.

Logistics

Assignment due date: Wednesday, November 17, 2021

Assignments are to be submitted electronically through cuLearn. If you work in a small group, it is sufficient for one group member to submit.

It is your responsibility to ensure that your assignment is submitted properly. Copying of assignments is NOT allowed. Discussion of assignment work with others is acceptable but each individual or small group are expected to do the work themselves.

Implementation

Programming language: Python 3

You may use any standard Python libraries. You may also use the NumPy, Pandas, and SciPy packages. Use of any additional packages requires approval of the instructor.

You must implement your code yourself. Do not copy-and-paste code from other sources, but you may use any pseudo-code we wrote in class as a basis for your implementation. Your implementation must follow the outlined specifications. Implementations which do not follow the specifications may receive a grade of zero. Please make sure your code is readable, as it will also be assessed for correctness. You do not need to prove correctness of your implementation.

You may be provided with a set of examples to test your implementation. Note that the provided examples do not necessarily represent a complete set of test cases. Your implementation may be evaluated on a different set of test cases.

Technical Document

Your technical document must answer all questions posed below. Ensure you answers are clear and concise. Submit the technical document as a single PDF file.

Implementation

Consider the problem of recognizing letters from black-and-white images. In this assignment, we will consider two different classifiers: a naïve Bayes classifier and a fuzzy classifier.

Assume that we have images of each of the following five letters from the English alphabet: A, B, C, D, E. For our purposes, we will ignore the other letters in the alphabet and assume that we always have majuscule (i.e. capital) letters. From the black-and-white images of the letters, we wish to classify which letter the images represents. To do so, we will extract the following features from the images.

1. PropBlack: The proportion of pixels in the image that are black.

$$PropBlack = \frac{Number\ of\ black\ pixels\ in\ the\ image}{Number\ of\ pixels\ in\ the\ image}$$

2. TopProp: The proportion of the black pixels that are in the top half of the image.

$$TopProp = \frac{Number\ of\ black\ pixels\ in\ the\ top\ of\ the\ image}{Number\ of\ black\ pixels\ in\ the\ entire\ image}$$

3. LeftProp: The proportion of the black pixels that are in the left half of the image.

$$LeftProp = \frac{Number\ of\ black\ pixels\ in\ the\ left\ side\ of\ the\ image}{Number\ of\ black\ pixels\ in\ the\ entire\ image}$$

Your implementation must contain a file named “assignment2.py” with two functions. One function must be named “naive_bayes_classifier”; one function must be named “fuzzy_classifier”. Both functions should take one input argument. The input argument should be the full file path to a comma separated value (CSV) file that contains a matrix representation of the image. The CSV file will have “1” where the image is black and “0” where the image is white.

The “naive_bayes_classifier” function should output two arguments. The first output argument should be a string indicating the most likely class for the input image. It should be either A, B, C, D, E. The second output argument should be a Python list indicating the probability the input belongs to each class: [A probability, B probability, C probability, D probability, E probability]. The distributions for each characteristic for each class are provided below.

Assume all features follow a normal distribution, and we can compute the probability a feature takes on a given value as proportional to the normal distribution:

$$P(feature = x|class) \propto \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2}$$

Where the particular μ and σ are given below for each class and feature.

A:
PropBlack: $\mu = 0.38, \sigma = 0.06$
TopProp: $\mu = 0.46, \sigma = 0.12$
LeftProp: $\mu = 0.50, \sigma = 0.09$

B:
PropBlack: $\mu = 0.51, \sigma = 0.06$
TopProp: $\mu = 0.49, \sigma = 0.12$
LeftProp: $\mu = 0.57, \sigma = 0.09$

C:
PropBlack: $\mu = 0.31, \sigma = 0.06$
TopProp: $\mu = 0.37, \sigma = 0.09$
LeftProp: $\mu = 0.64, \sigma = 0.06$

D:
PropBlack: $\mu = 0.39, \sigma = 0.06$
TopProp: $\mu = 0.47, \sigma = 0.09$
LeftProp: $\mu = 0.57, \sigma = 0.03$

E:
PropBlack: $\mu = 0.43, \sigma = 0.12$
TopProp: $\mu = 0.45, \sigma = 0.15$
LeftProp: $\mu = 0.65, \sigma = 0.09$

The prior probability of each letter is given below (following rough frequencies of use in the English language):

$$P(A) = 0.28$$
$$P(B) = 0.05$$
$$P(C) = 0.10$$
$$P(D) = 0.15$$
$$P(E) = 0.42$$

The "fuzzy_classifier" function should output two arguments. The first output argument should be a string indicating the output class with the highest membership for the input vector. It should be either A, B, C, D, E. The second output argument should be a Python list indicating the combined membership function value at each class: [A value, B value, C value, D value, E value]. The fuzzy membership function and fuzzy rules for the system are provided below. For the fuzzy rules use the Godel t-norm and the Godel s-norm.

Assume the fuzzy membership functions are trapezoidal on the range [0, 1] given by the following.

$$f(x) = \begin{cases} 0 \; if \; x \le a \\ \dfrac{x-a}{b-a} \; if \; a < x < b \\ 1 \; if \; b \le x \le c \\ \dfrac{d-x}{d-c} \; if \; c < x < d \\ 0 \; if \; d \le x \end{cases}$$

The values for a, b, c, d are given below for each fuzzy set and characteristic.

PropBlack:
Low: $a = 0, b = 0, c = 0.3, d = 0.4$
Medium: $a = 0.3, b = 0.4, c = 0.4, d = 0.5$
High: $a = 0.4, b = 0.5, c = 1, d = 1$

TopProp:
Low: $a = 0, b = 0, c = 0.3, d = 0.4$
Medium: $a = 0.3, b = 0.4, c = 0.5, d = 0.6$
High: $a = 0.5, b = 0.6, c = 1, d = 1$

LeftProp:

Low: $a = 0, b = 0, c = 0.3, d = 0.4$
Medium: $a = 0.3, b = 0.4, c = 0.6, d = 0.7$
High: $a = 0.6, b = 0.7, c = 1, d = 1$

The fuzzy rules associated with this system are:

IF PropBlack is Medium AND (TopProp is Medium OR LeftProp is Medium) THEN class A.
IF PropBlack is High AND TopProp is Medium AND LeftProp is Medium THEN class B.
IF (PropBlack is Low AND TopProp is Medium) OR LeftProp is High THEN class C.
IF PropBlack is Medium AND TopProp is Medium AND LeftProp is High THEN class D.
IF PropBlack is High AND TopProp is Medium AND LeftProp is High THEN class E.

Attached are example input images and the corresponding example outputs. Note that your functions should not write anything to file. These examples are provided in separate files for convenience. Also attached is skeleton code indicating the format your implementation should take. Note also that the classifiers will not always predict the "correct" letter as it appears in the image.

Grading

The implementation will be worth 60 marks.

20 marks will be allocated to correctness on a series of test cases for the naïve Bayes classifier, with consideration to each of the two outputs (i.e. most likely class, and class probabilities). 20 marks will be allocated to correctness on a series of test cases for the fuzzy classifier, with consideration to each of the two outputs (i.e. highest membership class, and class memberships).

10 marks will be allocated to human-based review of code of the naïve Bayes classifier. 10 marks will be allocated to human-based review of code of the fuzzy classifier.

Technical Document

Please answer the following questions in the technical document. Explain why your answers are correct.

1. Briefly describe how your implementation works. Include information on any important algorithms, design decisions, data structures, etc. used in your implementation. [10 marks]
2. What type of agent have you implemented (simple reflex agent, model-based reflex agent, goal-based agent, or utility-based agent)? [3 marks]
3. Suggest a particular feature vector for this problem for the naïve Bayes classifier where using equal prior probabilities for each class changes the most likely class (compared to using the above-specified prior probabilities). [4 marks]
4. Suggest a particular feature vector for this problem where the most likely class identified by the naïve Bayes classifier and the highest membership class identified by the fuzzy classifier are different. [4 marks]
5. Suppose we try to use our fuzzy classifier to classify a different letter which we have not seen before and do not have any rules for (i.e. not A nor B nor C nor D nor E). How might we identify such a case? [3 marks]
6. Suggest a particular instance of this problem for the fuzzy classifier where the class with the highest membership function would be different if we used the Goguen t-norm and Goguen s-norm (instead of using the Godel t-norm and Godel s-norm). [4 marks]
7. Suggest another feature we could use in these classifiers and explain how it could be computed on an input black-and-white image. [6 marks]
8. In this assignment, the probability densities for the naïve Bayes classifier have been provided. Suppose they were not provided. Suggest a method to determine these probability densities. Describe what data you would need to do so. [6 marks]

Grading

The technical document will be worth 40 marks, allocated as described above.