ASSIGNMENT #2 – COMP 3106 ARTIFICIAL INTELLIGENCE

The assignment is an opportunity to demonstrate your knowledge on informed Bayes' Theorem and fuzzy rule-based systems and to practice applying it to a problem.

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 in Python 3 to the problem outlined below.

The implementation will be graded both on content and use of good programming practices.

Technical Document

Your technical document should answer any questions posed below. Use these questions to elaborate on your implementation. After reading the technical document, the reader should understand 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: Friday, November 11, 2022

Assignments are to be submitted electronically through Brightspace. 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 the Python Standard Library (https://docs.python.org/3/library/). You may also use the Imageio, NumPy, Pandas, Pillow, 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 predicting the setting a photograph of nature comes from. In this assignment, we will consider two different classifiers for this task: naïve Bayes' classifier and a fuzzy classifier.

Assume that we only consider four different types of settings: tundra, forest, desert, and ocean (see examples attached). For this assignment, assume that all photographs come from one of these settings. From the photographs, we wish to classify which setting the image comes from. To do so, we will extract the following features based on the RGB pixels intensities. Recall that for an image, the RGB pixel values can be between 0 and 255 for each of Red, Green, and Blue.

1. Redness: The mean Red intensity value over the entire image.
2. Greenness: The mean Green intensity value over the entire image.
3. Blueness: The mean Blue intensity value over the entire image.

Your implementation must contain a single file named "assignment2.py" with two functions (you may have other variables/functions/classes in your file). 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 .jpg file containing an RGB image.

The "naive_bayes_classifier" function should return two values. The first returned value should be a string indicating the most likely class for the input image found using the naïve Bayes' classifier. It should be either "tundra", "forest", "desert", or "ocean". The second returned value should be a Python list indicating the probabilities found using the naïve Bayes' classifier that the input belongs to each class: [tundra probability, forest probability, desert probability, ocean probability].

For the naïve Bayes classifier, we will discretize our feature values. We will only consider whether each feature value is greater than 128. The distributions for each feature given each class are provided below.

$$P(redness > 128|tundra) = 0.85$$
$$P(redness > 128|forest) = 0.53$$
$$P(redness > 128|desert) = 0.94$$
$$P(redness > 128|ocean) = 0.18$$

$$P(greenness > 128|tundra) = 0.71$$
$$P(greenness > 128|forest) = 0.88$$
$$P(greeness > 128|desert) = 0.06$$
$$P(greenness > 128|ocean) = 0.27$$

$$P(blueness > 128|tundra) = 0.89$$
$$P(blueness > 128|forest) = 0.12$$
$$P(blueness > 128|desert) = 0.03$$
$$P(blueness > 128|ocean) = 0.98$$

The prior probability of each setting is given below (following the rough proportion of each setting covering the earth's surface):

$$P(tundra) = 0.03$$
$$P(forest) = 0.10$$
$$P(desert) = 0.11$$
$$P(ocean) = 0.76$$

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 found by the fuzzy classifier. It should

be either “tundra”, “forest”, “desert” or “ocean”. The second output argument should be a Python list indicating the combined membership function’s value found by the fuzzy classifier at each class: [tundra value, forest value, desert value, ocean value]. The fuzzy membership function and fuzzy rules for the system are provided below. For the fuzzy rules use the Goguen t-norm and the Goguen s-norm.

Assume the fuzzy membership functions are trapezoidal on the range [0, 255] given by the following.

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

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

Redness:
Low: $a = 0, b = 0, c = 85, d = 125$
Medium: $a = 85, b = 125, c = 130, d = 190$
High: $a = 130, b = 190, c = 255, d = 255$

Greenness:
Low: $a = 0, b = 0, c = 60, d = 120$
Medium: $a = 60, b = 120, c = 125, d = 185$
High: $a = 125, b = 185, c = 255, d = 255$

Blueness:
Low: $a = 0, b = 0, c = 55, d = 130$
Medium: $a = 55, b = 130, c = 140, d = 190$
High: $a = 140, b = 190, c = 255, d = 255$

The fuzzy rules associated with this system are:

IF Redness is High AND Greenness is High AND Blueness is High THEN tundra.
IF (Redness is Low OR Redness is Medium) AND Greenness is High AND (Blueness is Low OR Blueness is Medium) THEN forest.
IF Redness is High AND Greenness is Low AND Blueness is Low THEN desert.
IF Redness is Low AND Blueness is High THEN ocean.

Attached are example .jpg image files and corresponding example text files containing outputs. Note that your function 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.

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). These test cases will be run automatically by calling your

implementation from another Python script. To facilitate this, your implementation must adhere exactly to the specifications.

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. This human-based review will consider both correctness and use of good programming practices.

Technical Document

Please answer the following questions in the technical document. For all questions, 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. Is the task environment: [7 marks]
    a. Fully or partially observable?
    b. Single or multiple agent?
    c. Deterministic or stochastic?
    d. Episodic or sequential?
    e. Static or dynamic?
    f. Discrete or continuous?
    g. Known or unknown?
4. Suppose we wish to measure how well our methods work. Suggest what measure(s) of performance and/or what validation scheme should be used. Assume that we have a labelled set of images available for this task. [6 marks]
5. Suggest a particular feature vector [redness, greenness, blueness] where the most likely class found using the naïve Bayes classifier is different than the highest membership class found using the fuzzy rule-based system. [4 marks]
6. Suggest a particular feature vector [redness, greenness, blueness] where the class with the highest membership function would be different if we used the Godel t-norm and Godel s-norm (instead of using the Goguen t-norm and Goguen s-norm). [4 marks]
7. 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. Assume that we have a labelled set of images available for this task. [6 marks]

Grading

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