LAB Artificial Intelligence

Programming Assignment 4 – Fuzzy Logic

You are required to submit your solutions of this assignment on Moodle, at the end of the current lab time. Submit your own work. Cheating will not be tolerated and will be penalized.

Fuzzy logic: overview and terminology

Fuzzy Logic is a methodology predicated on the idea that the "truthiness" of something can be expressed over a continuum. This is to say that something isn't *true* or *false* but instead *partially true* or *partially false*.

A **fuzzy variable** has a **crisp value** which takes on some number over a pre-defined domain (in fuzzy logic terms, called a **universe**). The crisp value is how we think of the variable using normal mathematics. For example, if my fuzzy variable was how much to tip someone, it's universe would be 0 to 25% and it might take on a crisp value of 15%.

A fuzzy variable also has several **terms** that are used to describe the variable. The terms taken together are the **fuzzy set** which can be used to describe the "fuzzy value" of a fuzzy variable. These terms are usually adjectives like "poor," "mediocre," and "good." Each term has a **membership function** that defines how a crisp value maps to the term on a scale of 0 to 1. In essence, it describes "how good" something is.

So, back to the tip example, a "good tip" might have a membership function which has non-zero values between 15% and 25%, with 25% being a "completely good tip" (i.e., it's membership is 1.0) and 15% being a "barely good tip" (i.e., its membership is 0.1).

A **fuzzy control system** links fuzzy variables using a set of **rules**. These rules are simply mappings that describe how one or more fuzzy variables relates to another. These are expressed in terms of an IF-THEN statement; the IF part is called the **antecedent** and the THEN part is the **consequent**.

Problem statement: the tipping problem

The 'tipping problem' is commonly used to illustrate the power of fuzzy logic principles to generate complex behavior from a compact, intuitive set of expert rules.

We need to create a fuzzy control system which models how one might choose to tip at a restaurant. When tipping, you consider the service and the food quality, rated between 0 and 10 (10 being the best). You use this to leave a tip of between 0 and 25%.

We would formulate this problem as follows:

- **Antecedents (Inputs):** We have two antecedents/inputs for this problem.
  1. **Service**
     - Universe (i.e., crisp value range): How good was the service of the wait staff, on a scale of 0 to 10?
     - Fuzzy set (i.e., fuzzy value range): poor, acceptable, amazing

2. **food quality**
    - Universe: How tasty was the food, on a scale of 0 to 10?
    - Fuzzy set: bad, decent, great

- **Consequents (Outputs):** We only have one Consequents/Output, which is the **tip.**

    - Universe: How much should we tip, on a scale of 0% to 25%
    - Fuzzy set: low, medium, high

- **Rules**
    1. IF the *service* was good *or* the *food quality* was good, THEN the tip will be high.
    2. IF the *service* was average, THEN the tip will be medium.
    3. IF the *service* was poor *or* the *food quality* was poor THEN the tip will be low.

The Tipping problem using the skfuzzy library

For this problem, we are going to use **skfuzzy,** which is scikit-fuzzy library for SciPy.

Open the file tipping.ipynb. This file contains a fully working example of the tipping problem using skfuzzy. It contains all the code along with all the explanation needed.

The first part of this lab is to familiarize yourself with the skfuzzy library functions. Take your time to read every line in the code and understand it. Once done, you can move on to the second part of the lab where you are going to use fuzzy logic to solve the problem of water Irrigation.

Verification: If you run this code for a `quality = 6.5` and a `service = 9.8`, you should expect a `tip = 19.847607361963192.`

The fuzzy irrigation controller problem

In this second part, you are going to use the fuzzy irrigation controller exercise that you solved during your AI lectures.

Open a new Jupyter notebook document and name it irrigation.ipynb. In this file you are going to write the fuzzy code for the irrigation controller using the skfuzzy library.

Inspired by the tipping problem, write a code that takes as input the **humidity and the temperature** (as per your exercise), and gives as output the duration of irrigation needed in minutes.

- For the membership functions you can use:
  - `skfuzzy.trapmf (x,abcd)`, trapezoidal membership function generator.
  - `skfuzzy.trimf (x,abc)`, triangular membership function generator.
  - The x-coordinates values that you will need to use are:
    - for temperature: 0, 5, 10, 12.5, 15, 22.5, 27.5, 30, 32.5, 37.5, 45
    - for humidity: 0, 5, 12.5, 17.5, 20, 22.5, 30
    - for duration: 0, 10, 20, 30, 440, 50, 70

- Verification: The expected outcome for a `humidity = 10` and a `temperature = 35` is a `duration` of 39.68 minutes.

- After having verified your code, you need now to change the defuzzification method and see how it affects your output. Note that the skfuzzy library uses centroid defuzzification by default. To do so, you can use the built-in method:

  `Consequent.defuzzify_method = 'string'` #you need to put the name that you have chosen for the `Consequent`, and replace `'string'` with one of the following methods.

  For `'string'`, you can choose it from the following pool:
  'centroid': centroid of area
  'mom' : mean of maximum
  'som' : min of maximum
  'lom' : max of maximum

  PS: you can change the defuzzification method for the tipping problem as well and see how the output is affected.

Done! Now answer the questions on the next page.

Questions

Answer the following questions for the **irrigation problem**:

1. Fill the following table with the duration of irrigation for every case:

	centroid	**lom (**max of maximum)	**mom** (mean of maximum)	**som** (min of maximum)
humidity = 1 temperature = 1				
humidity = 1 temperature = 45				
humidity = 10 temperature = 35				
humidity = 25 temperature = 10				
humidity = 30 temperature = 1				
humidity = 30 temperature = 45				

2. Analyse vertically and horizontally the results obtained in the previous table, i.e. answer questions such as:
   - For humidity =1 and temperature = 45, do we always get the maximum duration of 70 minutes (which one might expect for this kind of input)? Explain.
   - Do some methods give same results for several inputs? Explain.
   - Is there a 'best' defuzzification method? Explain.
   - Etc.