Coursework on Fuzzy Inference System

Coffee Strength Inference System

In this case study, we present a Fuzzy Coffee Strength Inference Engine, a system designed to guide coffee drinkers in selecting the ideal coffee strength based on the type of coffee bean, brewing duration, and water temperature. The system uses Mamdani Fuzzy Logic principles to model the brewing process, providing intuitive and flexible recommendations tailored to the unique variables that affect coffee strength. The inputs are the coffee bean types, brewing duration and water temperature and the outputs is a range of strength based on those input categories.
Example:

```
If bean is Himalayan Medium and brew duration is medium and temperature is high, then coffee is strong.
```

Variables

For the inference system, we have the following input variables:

Inputs

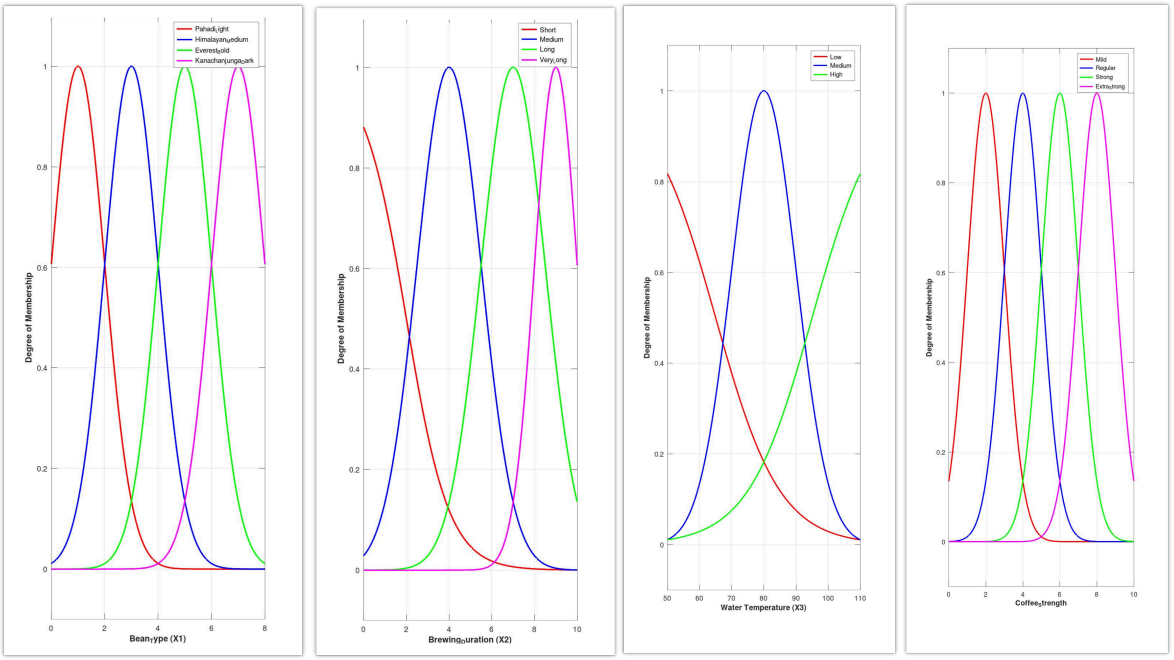
X1 = Bean Type(Pahadi Light, Himalayan Medium, Everest Bold, Kanchanjunga Dark)
X2 = Brewing Duration(Very Short, Short, Medium, Long, Very Long)
X3 = Water Temperature(Low, Medium, High)

- Bean Type(X1) is an input variable based on characteristics of a coffee bean like roast and flavours
  - Pahadi Light -> light roast with fruity and acidic in flavor
  - Himalayan Medium -> Medium Roast, well-balanced
  - Everest Bold -> Medium-Dark Roast, Slightly Caramel
  - Kanchanjunga Dark -> Dark Roast, Intense Smoky flavor
- Brewing Duration(X2) is the second input in the system, duration can vary for each type of coffee we want.
  - short -> espresso, quick pour
  - medium -> standard
  - long -> french press, slow brewing
  - very long -> cold brew, extended
- Water Temperature(X3) is our third input in the system with water temperature that can be low, medium and high.

Outputs

Y1 = Coffee Strength is our output of the system like Mild, Regular, Strong and Extra Strong.

Let us look at the membership state and the behaviour of our variables with the help of the following graphs generated from octave fuzzy toolkit.



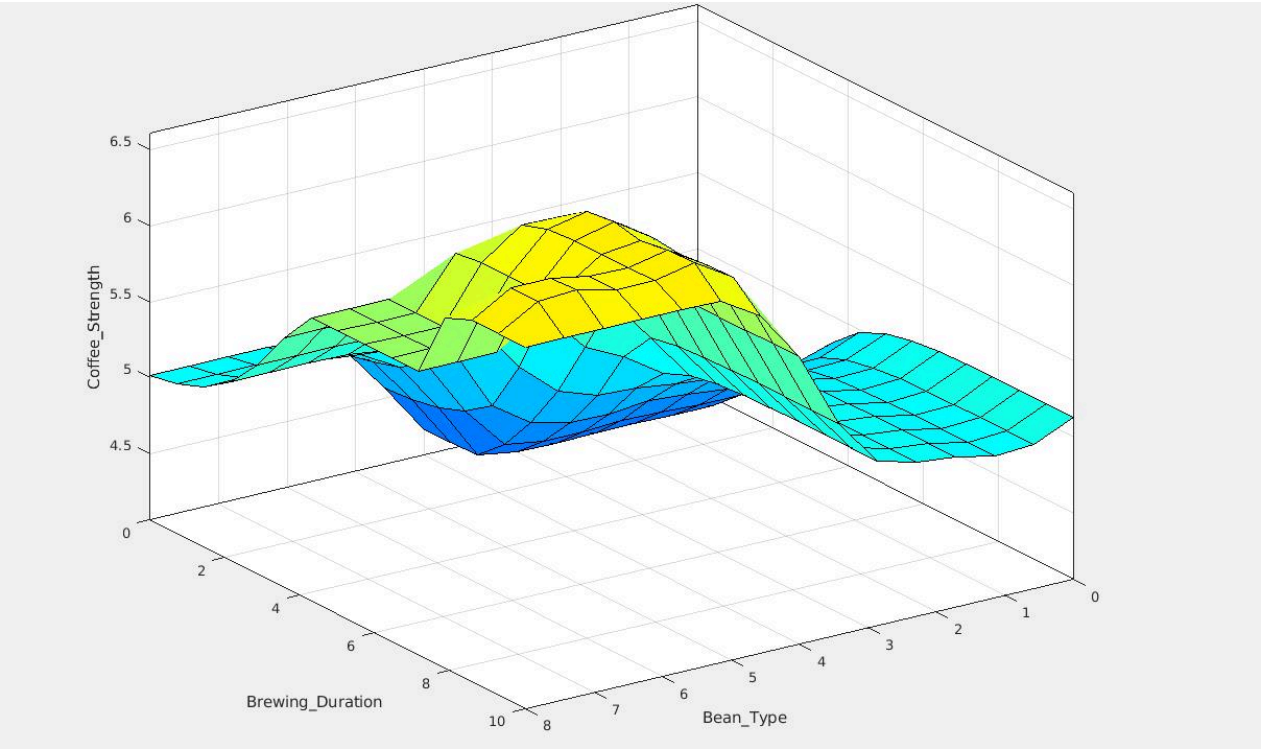
The above figure shows the type of memership function used either gaussian or sigmoid along with their interval values. Also mentioned in the code attached `q2.m` file.

Inference Output Y

Let us try and generate results testing three sample of inputs.

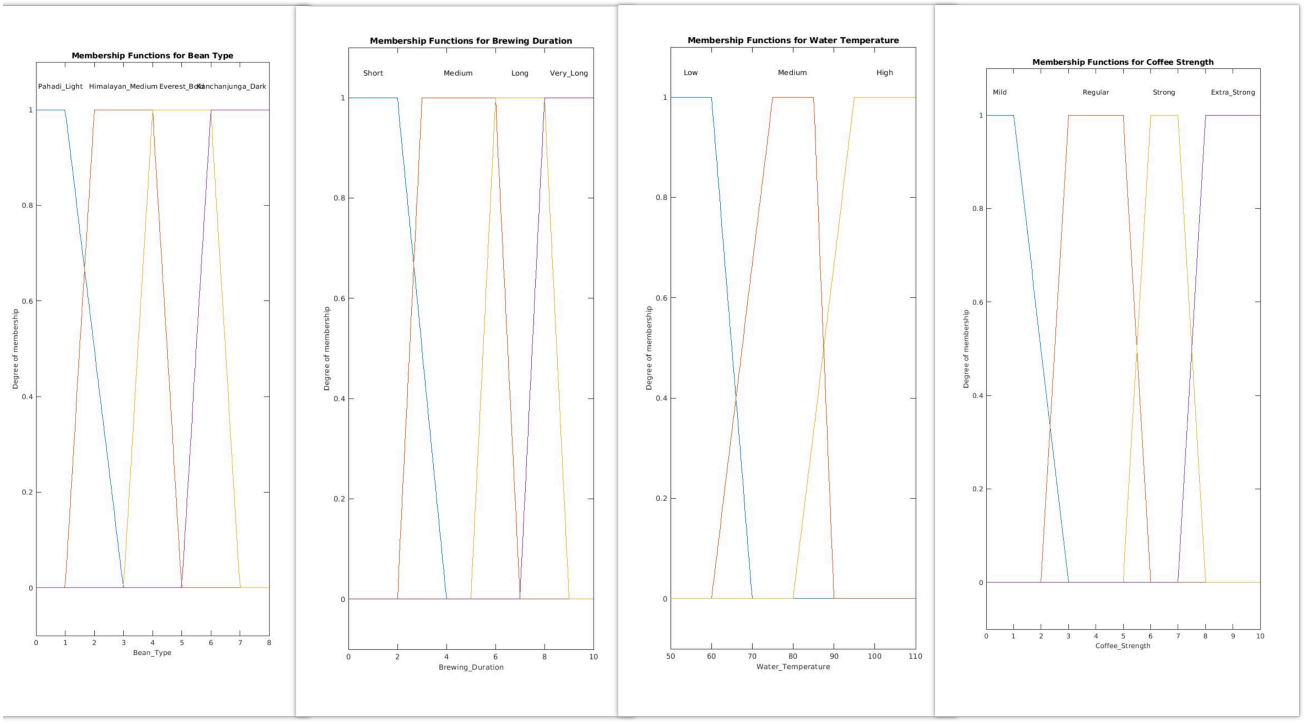
- For X1, X2 and X3 we use the values [3, 2, 70] i.e. the bean type input is 3, brew duration is 2 and the temperature is `70`. Then we expect the coffee strength to be around regular to strong. From the rule inference we see that the output is `4.15` which is in the range of regular to strong.
- For X1, X2 and X3 we use [7.5, 8.5, 100] i.e. bean type is 7.5, brew duration is 8.5 and temperature is 100, we expect the coffee to be strong. And the output for this is 6.11, which says that the coffee strength is strong.

Here is the rule surface generated using matlab.

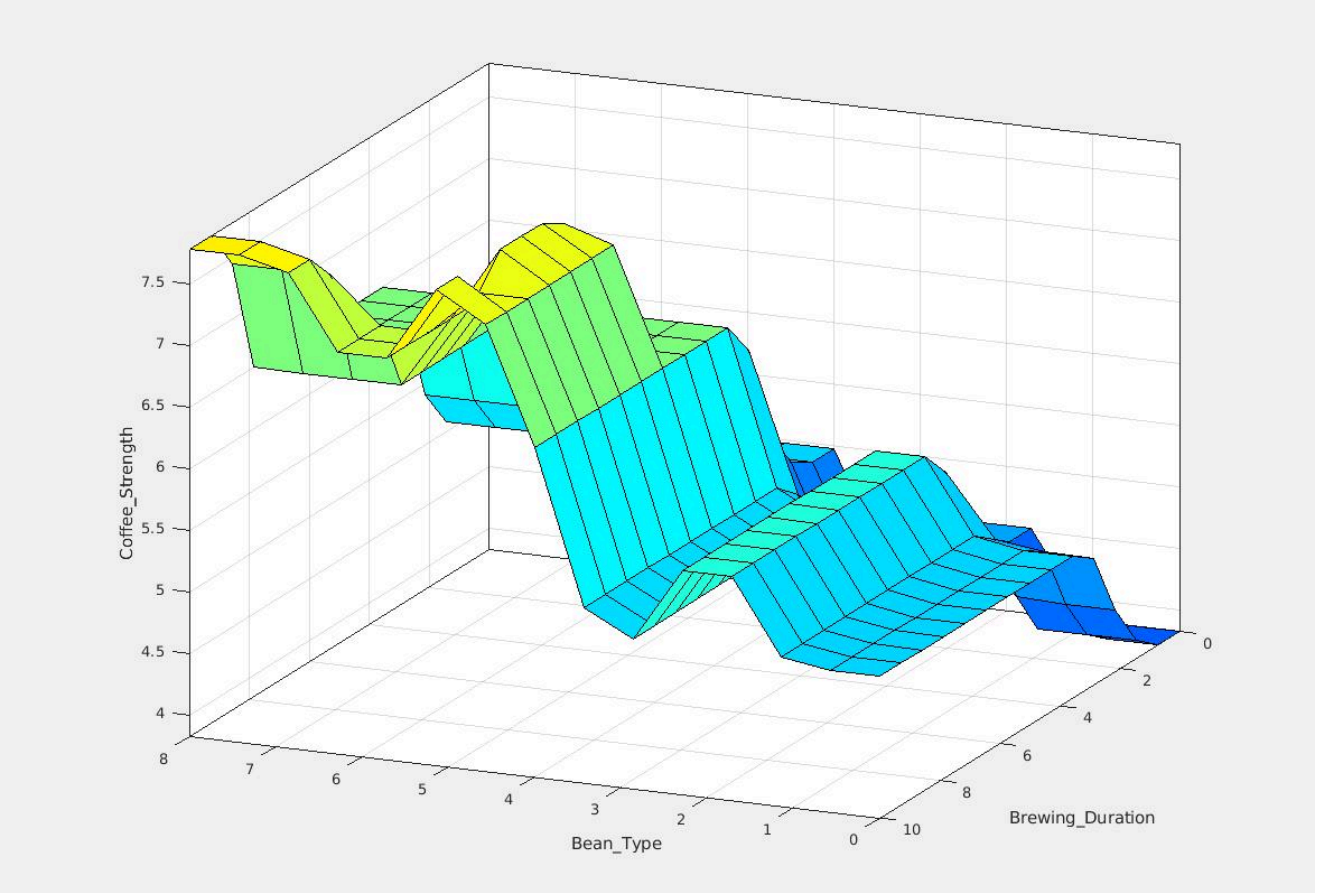


Using Trapezoidal Membership-functions

Let us define and use trapezoidal membership functions for all of the input/output variables in the system from above. The membership function plots will look like this:



And the rule surface looks like this:



Comparing with our original input variables the rule surface with trapezoidal membership functions tends to be higher than the original rule surface. The rule surface has flat regions at specific heights like 2, 4, 6 etc, corresponding to the centers of the output membership functions(mild, regular, strong and extra strong). The rule surface also looks like a stepped surface because the inputs move from one trapezoidal membership function to another causing a switch in rules. Also the surface is not smooth compared to our original rule surface because of transitions in our membership functions.

With the input values X1 = 7, X2 = 9, X3 = 80:
- Rule Viewer shows Rule 13 (Kanchanjunga_Dark, Very_Long → Extra_Strong) firing strongly, with output ~ 7.78, matching the top peak.
- Other rules like rules 26, 27 are also fired.

- Set X1 = 1, X2 = 1, X3 = 80:
  - Rule 1 (Pahadi_Light, Short → Mild) fires with other rules, with output ~3, matching the lowest peak.

Other Defuzzification techniques

There are various defuzzification techniques used like Centroid, Bisector, Mean of Maximum(mom), Smallest of Maximum(som) and Larges of Maximum(lom).

Method	Case 1 [3, 2, 70]	Case 2 [7.5, 8.5, 100]
centroid	4.145	6.106
bisector	4.100	6.200
mom	4.000	7.000
som	4.000	5.500
lom	4.000	8.500

Centroid is the default defuzzification method used in mamdani fuzzy inference system. But when we use other defuzzification methods for the same type of inputs only centroid and bisector are close to the actual output based on the inputs to the system. lom and som produces extreme values for case 2.