ECE449 Lab 4

October 18th, 2023

Fuzzy Logic

- **Foundation:** Simulates human reasoning and decision-making.
- **Difference with Classical Logic:** Works with degrees of truth, not just binary true or false (1 or 0).
- **Fuzzy Sets and Membership Degrees:** Variables can have degrees of membership in multiple sets, enabling nuanced decisions.
- **Rule-Based Systems:** Consists of a set of rules that map inputs to outputs, guiding decision-making.

Fuzzy Logic

Applications: Useful in control systems, decision-making, and pattern recognition, handling complexity and uncertainty.

Handling Uncertainty: Allows for more flexible and realistic decision-making in uncertain or vague scenarios.

Lab 4

Tipping Problem: A decision-making problem where fuzzy logic is used to determine an appropriate tipping amount in a restaurant based on service and food quality.

Use skfuzzy library:

```
import skfuzzy as fuzz
```

```
from skfuzzy import control as ctrl
```

Install if not installed:

```
!pip install scikit-fuzzy
```

Lab 4

Fuzzy Sets: Collections that allow objects to have degrees of membership.

Define input variables as Antecedent:

```
temperature = ctrl.Antecedent(np.linspace(0, 10, 11),
'temperature')
```

Define output variables as Consequent:

```
tip = ctrl.Consequent(np.linspace(0, 25, 26), 'tip')
```

Lab 4

Membership Functions: Mathematical functions that define how each point in the input space is mapped to a membership value (degree of membership) between 0 and 1.

```
temperature.automf(3)
```

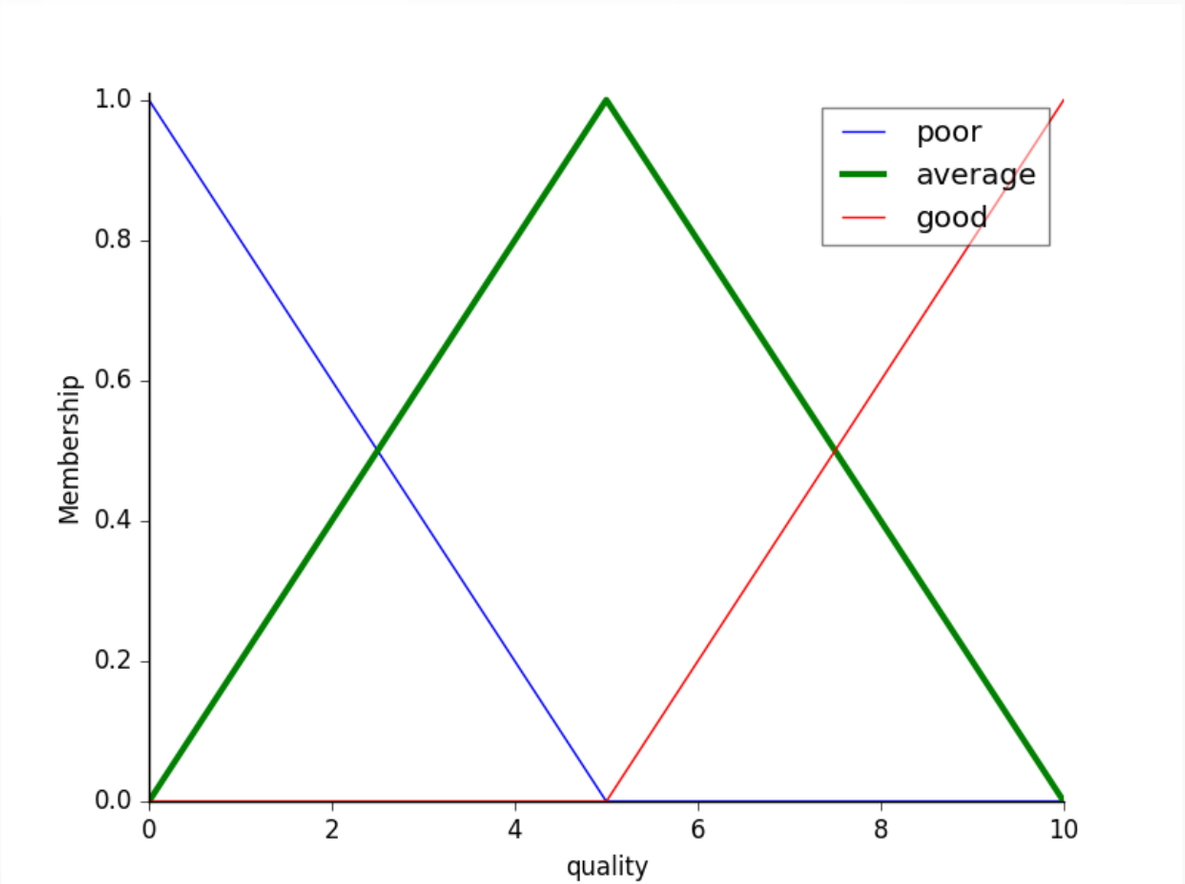
Automatically creates three fuzzy membership functions, typically labeled as 'poor', 'average', and 'good' associated with each fuzzy set.

Usually triangular or trapezoidal membership functions.

Lab 4

To help understand what the membership looks like, use the view methods.

quality['average'].view()



From https://pythonhosted.org/scikit-fuzzy/auto_examples/plot_tipping_problem_newapi.html#example-plot-tipping-problem-newapi-py

Lab 4

Fuzzy Rules: Logical statements that define the relationship between fuzzy sets in the input and output spaces, forming the basis of decision-making in fuzzy logic systems.

```
rule1_food = ctrl.Rule(temperature['poor'] | flavor['poor']
| portion_size['poor'], food_quality['poor’])


rule3_food = ctrl.Rule(temperature['good'] &
flavor['good'], food_quality['good'])
```

Lab 4

Create Controller: Create your control system and simulation using the defined rules.

```
food_ctrl = ctrl.ControlSystem([rule1_food, rule2_food, rule3_food])

food_sim = ctrl.ControlSystemSimulation(food_ctrl)
```

Fuzzy Inference: The process of mapping from a given input to an output using fuzzy logic, involving the application of fuzzy rules and aggregation methods.

Lab 4

Defuzzification: The conversion of a fuzzy output (having degrees of membership) into a single crisp value, typically representing a decision or recommendation.

For fuzzy inference and defuzzification use:

```
food_sim.compute()
```

The default defuzzification in this function is center-of-gravity defuzzification.

Lab 4

To access output values:

```
tip_output = tip_sim.output['tip']
```

To give user inputs to your control system:

```
food_sim.input['temperature'] = inputs['temperature']
```

Lab 4

Start by creating the input and output variable.

You have three control simulations in total:

Simulation 1: Determine food quality

Inputs➔ temperature, flavor, portion size

Output➔ food quality

Simulation 2: Determine service quality

Inputs➔ attentiveness, friendliness, speed

Output➔ service quality

Lab 4

Simulation 3: Determine tip amount

Inputs➔ food quality and service quality from previous simulations

Output➔ tip amount

This means:

```
food_quality_output = food_sim.output['food_quality']

service_quality_output = service_sim.output['service_quality']

tip_sim.input['food_quality_for_tip'] = food_quality_output

tip_sim.input['service_quality_for_tip'] = service_quality_output
```