Cairo University
Faculty of Computers & Artificial Intelligence
Department of Computer Science
Genetic Algorithms Course - 2020

Lab 3 – Fuzzy Logic

What is "Fuzzy Logic"?

It is a method of knowledge representation and reasoning that allows computers to think and make decisions using imprecise quantities like a human would.

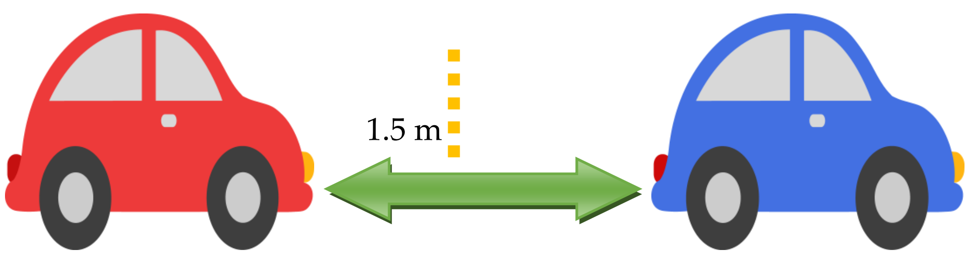
Traditional logic vs. fuzzy logic:

Let's say we would like to build an automatic braking system for a car. We will use the distance between our car and the car in front of it to determine whether our car should brake.

➢ **With traditional logic:**

We can say that the system rules are:

- If the distance is not long, brake.
- If the distance is long, don't brake.



So, here, if the distance between the 2 cars is greater than 1.5 meters (a crisp boundary), it is considered "long", otherwise, it is "not long".

- Our **knowledge representation** can be the **set** "Long" where:
  Long = {x: $x \in R^+$ & x > 1.5}.
- The system **input** is the **current distance** (assume D = 1 m).
- We will apply the **system rules** as follows:

  D = 1 & $1 \notin$ Long => long = false
  not long & (not long -> brake) => brake = true

In this representation, **"long"** can be either **true or false (0 or 1 only)** and **"brake"** can be either **true or false (0 or 1 only)** as well. This is **not suitable** for our system because 1 meter isn't too close; it may only require slowing down, but applying sudden pressure to brake would be dangerous. This means that we need a different representation that tolerates imprecision and uncertainty and would think and act like a human would in this situation (estimate the distance and slow down accordingly).

➢ With fuzzy logic:

When we use fuzzy logic and fuzzy sets to model our system, the distance variable won't have to be either long or not long! It will have **multiple sets**; "long", "medium" and "short". Each set will have its own range and the **truth value** of belonging to a set will **range between [0, 1]**. Similarly, "brake" will be represented by multiple sets; "move", "slow down" and "stop" and each set will have its own range and the truth value will range between [0, 1].

- Our **knowledge representation** can be the **fuzzy sets** described above.
- The system **input** is the **current distance** (assume D = 1 m).
- We will apply the **system rules** and predict the output/decision:

  D = 1, $\mu_{long}(1) = 0$, $\mu_{medium}(1) = 0.8$ & $\mu_{short}(1) = 0.2$ => brake $\mu_{slow\ down} = 0.4$ (hence, slow down)

This is a more suitable representation to the problem as it produces a smooth and flexible output.

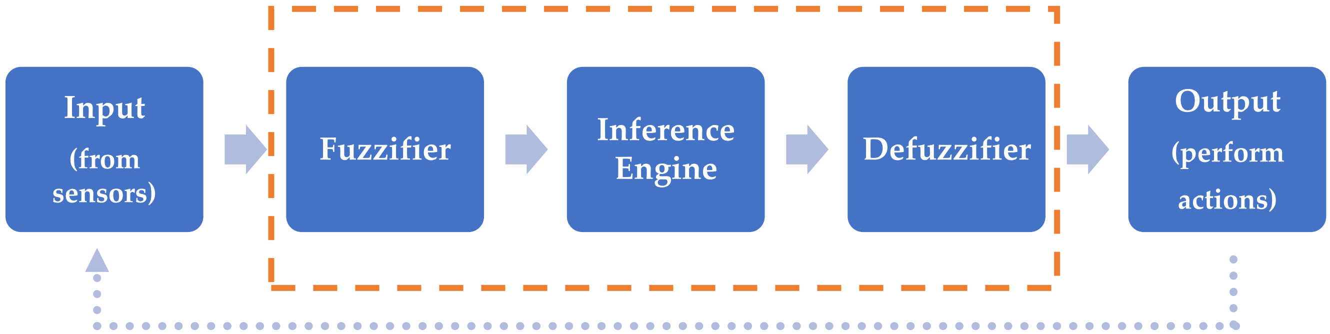
Note: The symbol **μ** represents the **degree of membership** to a certain fuzzy set. It is a measure of uncertainty and different from a probability.

To summarize:

Traditional (Boolean) Logic	*Fuzzy Logic*
A proposition in traditional logic is either **true or false (0 or 1)**.	The truth value / degree of membership has a **range [0, 1]**.
It uses **regular sets** where elements are either members or not members of each set.	It uses **fuzzy sets** where elements have a **degree of membership** to each set.
It is not suitable when the truth or falsehood can't be guaranteed.	It is very suitable when the truth or falsehood can't be guaranteed (allows imprecision). That's why it is used in **AI, decision making and control systems** (cars, appliances, etc.). However, it is not suitable for systems that require high precision.

The basic steps of solving a problem using fuzzy logic:

1. Fuzzification
2. Inference
3. Defuzzification



How to solve problems using fuzzy logic:

- ***Fuzzification:***

In this step, we assign the variables to fuzzy sets with degrees of membership using a membership function. Each variable (input or output) has multiple fuzzy sets; each with its own shape and range.

- ***Inference:***

In this step, we apply the "if-then" rules that exist in the system.

- ***Defuzzification:***

In this step, we get the predicted output, whether it's a continuous value or a decision, from the fuzzy truth values calculated.

Example:

Design a controller for an automatic washing machine to determine the wash time using fuzzy logic.

Given:

- The washing machine has a sensor that measures the **percentage** of dirt in water. The fuzzy sets of the amount of dirt are small (0,0,20,40), medium (20,40, 60, 80) and large (60, 80, 100, 100).
- The washing machine has a sensor that measures the **percentage** of softness of the fabric. The fuzzy sets of the fabric type are soft (0,0,20,40), ordinary (20,40, 60, 80) and stiff (60, 80, 100, 100).
- The wash time ranges from 0 to 60 minutes and its fuzzy sets are: (very short, short, standard, long, very long) and they are all triangular.

Objective:

- Predict the wash time to set when the dirt value = 60 and the fabric value = 25.

Rules:

1. If the amount of dirt is small and the fabric is soft then the wash time is very small.
2. If the amount of dirt is medium and the fabric is ordinary then the wash time is standard.
3. If the amount of dirt is small and the fabric is not soft or the amount of dirt is medium and the fabric is soft then the wash time is small.
4. If the amount of dirt is medium and the fabric is stiff then the wash time is large.
5. If the amount of dirt is large and the fabric is not soft then the wash time is very large.
6. If the amount of dirt is large and the fabric is soft then the wash time is standard.

Sol.:

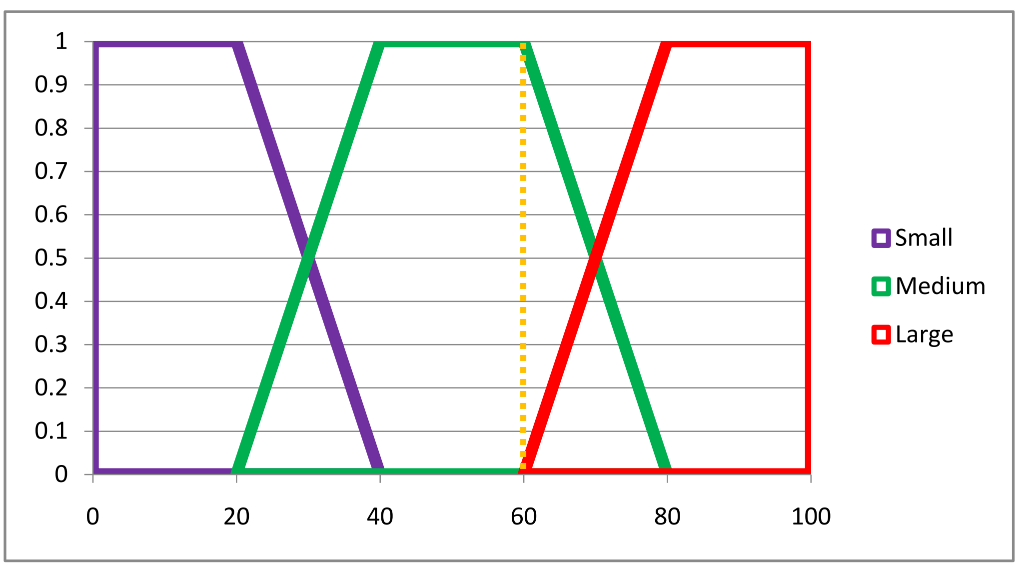
Step 1 – Fuzzification:

We need a membership function to map the crisp values to a membership degree.

Types of membership functions: triangular, rectangular, trapezoidal, left shoulder, right shoulder, singleton, …

In this problem both input variables have 3 fuzzy sets and all of them are trapezoidal while the output has 5 fuzzy sets and all of them are triangular. So, using the ranges given for each set:

a) ***Dirt:***



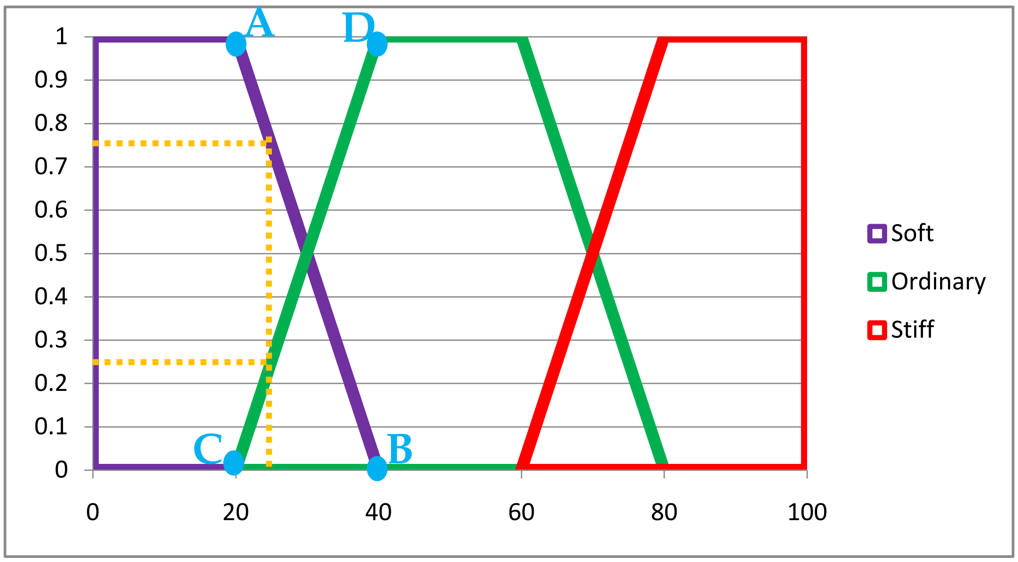
We can calculate the degree of membership of 60 to each fuzzy set directly from the graph or using the line equations. From the graph:

$\mu_{small}(60) = 0$

$\mu_{medium}(60) = 1$

$\mu_{large}(60) = 0$

b) Fabric:



We will calculate the degree of membership of 25 to each fuzzy set from the line equations. The crisp value 25 cuts both lines AB (in soft) and CD (in ordinary).

Equation of a line: y = mx + c where m is the **slope** and c is the **intercept**.

For AB => points (20, 1) and (40, 0)

- **Slope = $(y_2 - y_1) / (x_2 - x_1)$** = (0 – 1) / (40 – 20) = -1/20
- **Intercept** (substitute in the line equation):
  0 = (-1/20) * (40) + c, therefore c = 2
- **Equation:** y = (-1/20)x + 2

$\mu_{soft}(25) = (-1/20) * 25 + 2 = 0.75$

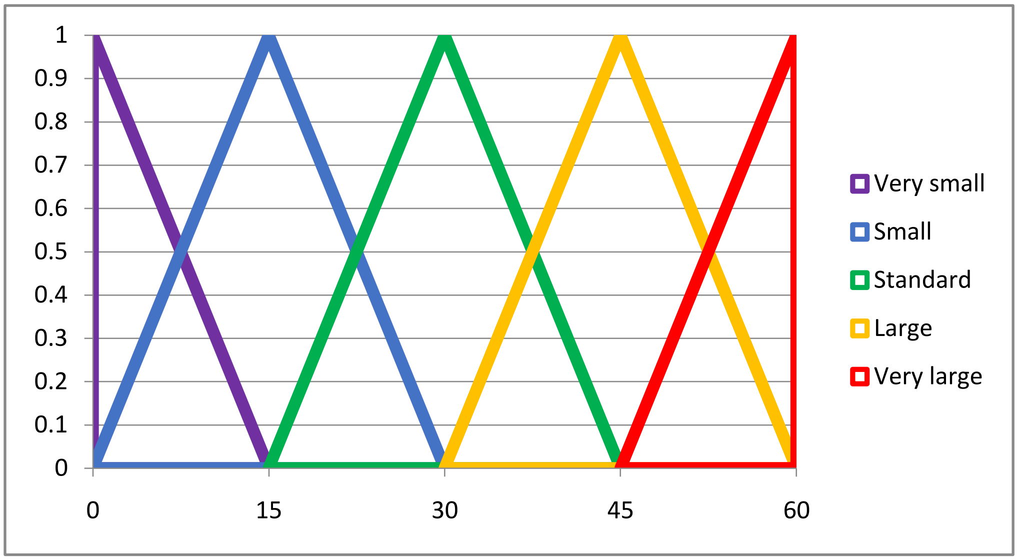
For CD => points (20, 0) and (40, 1)

- **Slope = $(y_2 - y_1) / (x_2 - x_1)$** = (1 – 0) / (40 – 20) = 1/20
- **Intercept** (substitute in the line equation):
  0 = (1/20) * (20) + c, therefore c = -1
- **Equation:** y = (1/20)x - 1

$\mu_{ordinary}(25) = (1/20) * 25 - 1 = 0.25$

From the graph, $\mu_{stiff}(25) = 0$

c) Wash time:



Step 2 – Inference:

Apply the rules as follows:

and	min
or	max
not	1 - x

1. **min(0, 0.75) = 0 ($\mu_{\text{very small}}$)**
2. **min(1, 0.25) = 0.25 (μ_{standard})**
3. **max (min(0, 1 - 0.75), min(1, 0.75)) = 0.75 (μ_{small})**
4. **min(1, 0) = 0 (μ_{large})**
5. **min(0, 1 - 0.75) = 0 ($\mu_{\text{very large}}$)**
6. **min(0, 0.75) = 0 (μ_{standard})**

   R2 or R6 => max(0.25, 0) = 0.25 (μ_{standard})

Step 3 – Defuzzification:

There are several defuzzification methods that could be used to predict the crisp value of the output.

1. **Weighted average method:**

   The crisp value of the output can be obtained by:

$$z^* = \frac{\sum \mu(\bar{z}).\bar{z}}{\sum \mu(\bar{z})}$$ where $\bar{z}$ is the centroid of the fuzzy set.

Since the output fuzzy sets are symmetrical triangles, we can use this method and the centroids are calculated as follows:

- $C_{\text{very small}} = (0 + 0 + 15) / 3 = 5$
- $C_{\text{small}} = (0 + 15 + 30) / 3 = 15$
- $C_{\text{standard}} = (15 + 30 + 45) / 3 = 30$
- $C_{\text{large}} = (30 + 45 + 60) / 3 = 45$
- $C_{\text{very large}} = (45 + 60 + 60) / 3 = 55$

Substituting in the weighted average equation:

$$z^* = \frac{\mathbf{0*5+0.75*15+0.25*30+0*45+0*55}}{\mathbf{0+0.75+0.25+0+0}} = 18.75 \text{ (=> small)}$$

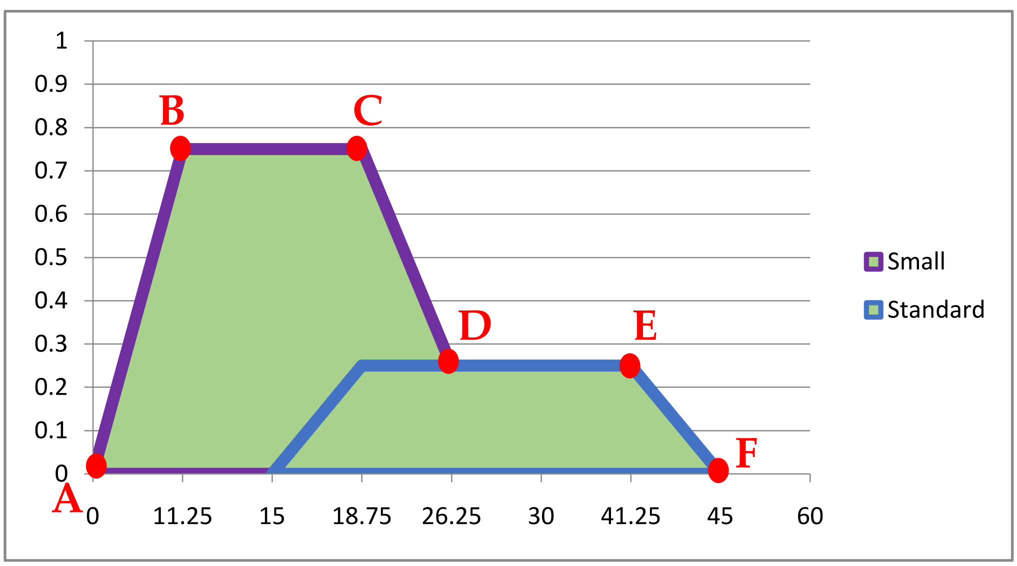
Note: The centroid of a non-self-intersecting closed polygon is calculated by:

$$C_x = \frac{1}{6A}\sum_{i=0}^{n-1}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1} y_i)$$

$$A \text{ (signed area)} = \frac{1}{2}\sum_{i=0}^{n-1}(x_i y_{i+1} - x_{i+1} y_i)$$

* The next methods would require performing an aggregation of the outputs of the rules.

2. **Centroid method:**



In this method, we want to calculate the centroid (center of gravity) of the new aggregated shape (shown above) and use this value as our crisp prediction.

$$z^* = \frac{\int \mu(z).z\, dz}{\int \mu(z) dz}$$

We need to calculate the area of the entire shape (integration), so we will divide it into parts:

AB:

- **Range: 0 to 11.25**
- **Eq.: y = (1/15)x**

BC:

- **Range: 11.25 to 18.75**
- **Eq.: y = 0.75**

CD:

- **Range: 18.75 to 26.25**
- **Eq.: y = (-1/15)x + 2**

DE:

- **Range: 26.25 to 41.25**
- **Eq.: y = 0.25**

EF:

- **Range: 41.25 to 45**
- **Eq.: y = (-1/15)x + 3**

Note: The upper limits of the ranges can be obtained by substituting **y** by **μ** in each line equation and calculating **x**. For example, 11.25 was obtained using: 0.75 = (1/15)x, therefore x = 11.25

Now, substituting in the centroid equation, we get:

$$z^* = \frac{\int_0^{11.25} \frac{1}{15} z.z\, dz + \int_{11.25}^{18.75} 0.75 z\, dz + \cdots}{\int_0^{11.25} \frac{1}{15} z\, dz + \int_{11.25}^{18.75} 0.75\, dz + \cdots} = 19.34 \text{ (=> small)}$$

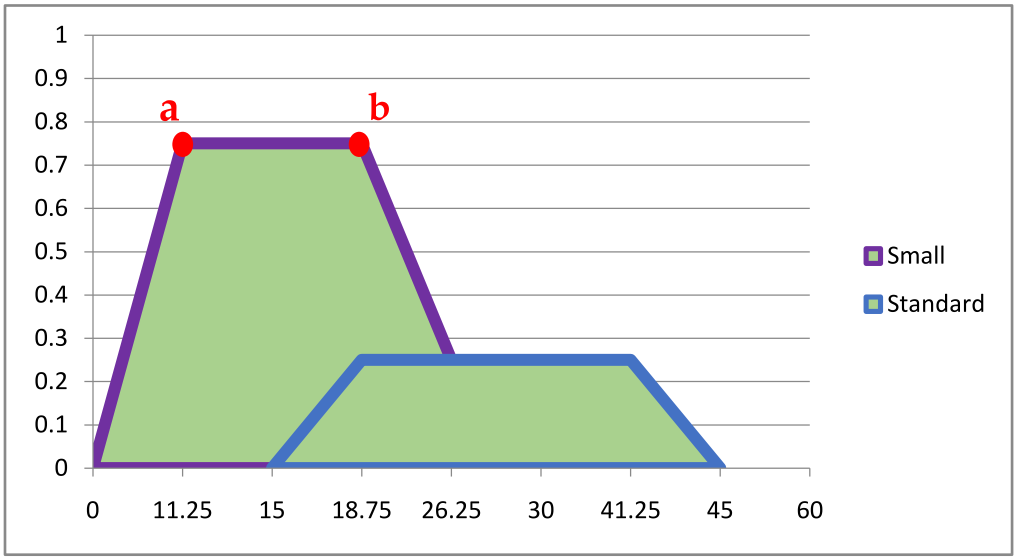
3. **Max. method:**

   Get the crisp value that has the highest membership in the aggregated graph. This method only works with peaked outputs, so it is not suitable to our problem.

4. **Mean of max. method:**

   Get the crisp value that has the highest membership in the aggregated graph when the highest membership is a plateau.

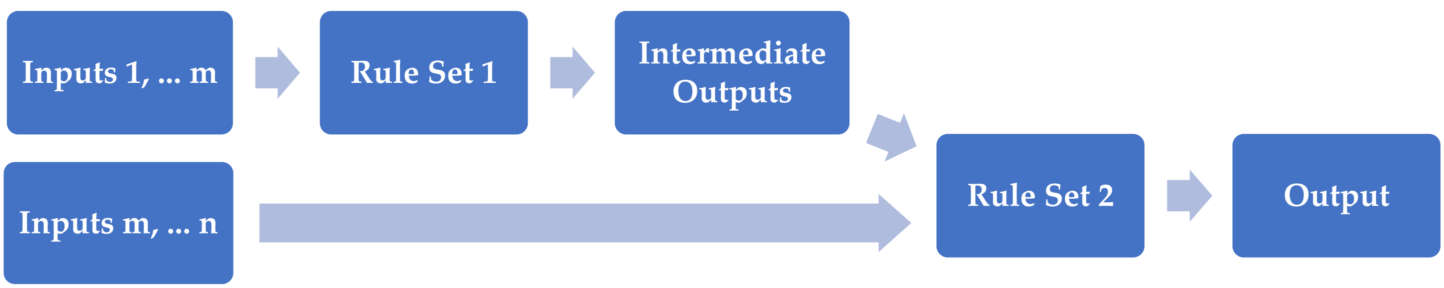
$$z^* = \frac{a+b}{2}$$



$$z^{*} = \frac{\mathbf{11.25+18.75}}{\mathbf{2}} = 15 \text{ (=> small)}$$

Note:

We can have multiple rule sets in which the output of a rule set can be used as an input to another.



Question:

In our washing machine example, let's say we added a new input variable that measures the amount of clothes (small, medium and large) and a new output variable that determines the amount of soap (small, medium and large) to be added.

Rule set 1: all the rules in the previous example +

7. If the amount of dirt is small or the fabric is soft then **I** (indicator) is low.
8. If the amount of dirt is not small and the fabric is not soft then **I** is high.

Rule set 2:

1. If the amount of clothes is large or **I** is high then the amount of soap is large.
2. If the amount of clothes is small and **I** is low then the amount of soap is small.
3. If the amount of clothes is medium and **I** is low, the amount of soap is medium.

Think about how you can solve the problem.