Lecture 16

Fuzzy Expert System - II: Fuzzy Inference
Fuzzy Inference

- Mamdani Fuzzy Inference
- Sugeno Fuzzy Inference
- Case Study
The most commonly used fuzzy inference technique is the so-called Mamdani method.

In 1975, Prof. Ebrahim Mamdani of London University built one of the first fuzzy systems to control a steam engine and boiler combination.

He applied a set of fuzzy rules supplied by experienced human operators.
Mamdani Fuzzy Inference

- The Mamdani-style fuzzy inference process is performed in four steps:
  - Fuzzification of the input variables,
  - Rule evaluation
  - Aggregation of the rule outputs, and finally
  - Defuzzification
We examine a simple *two-input one-output* problem that includes three rules:

**Rule: 1**

IF \( x \) is A3  
OR \( y \) is B1  
THEN \( z \) is C1

**Rule: 2**

IF \( x \) is A2  
AND \( y \) is B2  
THEN \( z \) is C2

**Rule: 3**

IF \( x \) is A1  
THEN \( z \) is C3

Mamdani Fuzzy Inference
The first step is to take the crisp inputs, $x_1$ and $y_1$ (project funding and staffing), and determine the degree to which these inputs belong to each of the appropriate fuzzy sets: degree of compatibility (match)

\[
\begin{align*}
\mu(x = A_1) &= 0.5 \\
\mu(x = A_2) &= 0.2 \\
\mu(y = B_1) &= 0.1 \\
\mu(y = B_2) &= 0.7
\end{align*}
\]
The second step is to take the fuzzified inputs, 
\[ \mu_{(x=A_1)} = 0.5, \mu_{(x=A_2)} = 0.2, \mu_{(y=B_1)} = 0.1 \text{ and } \mu_{(y=B_2)} = 0.7, \]
and apply them to the antecedents of the fuzzy rules.

If a given fuzzy rule has multiple antecedents, the fuzzy operator (AND or OR) is used to obtain a single number (firing strength) that represents the result of the antecedent evaluation.

This truth value (firing strength) is then applied to the consequent MF to get the qualified consequent MF.
To evaluate the disjunction of the rule antecedents, we use the OR fuzzy operation.

Typically, fuzzy expert systems make use of the classical fuzzy operation union:

\[ m_{A \cup B}(x) = \max[m_A(x), m_B(x)] \]

Similarly, to evaluate the conjunction of the rule antecedents, we apply the AND fuzzy operation as intersection:

\[ m_{A \cap B}(x) = \min[m_A(x), m_B(x)] \]
Mamdani-style rule evaluation

1. **Rule 1**: IF $x$ is $A_3 (0.0)$ OR $y$ is $B_1 (0.1)$ THEN $z$ is $C_1 (0.1)$

2. **Rule 2**: IF $x$ is $A_2 (0.2)$ AND $y$ is $B_2 (0.7)$ THEN $z$ is $C_2 (0.2)$

3. **Rule 3**: IF $x$ is $A_1 (0.5)$ THEN $z$ is $C_3 (0.5)$
Now the result of the antecedent evaluation, i.e. firing strength can be applied to the MF of the consequent to get the qualified consequent MF.

- The most common method of correlating the rule consequent with the truth value of the rule antecedent is to cut the consequent MF at the level of the antecedent truth. This method is called clipping.
  - Since the top of the MF is sliced, the clipped FS loses some information.
  - However, clipping is still often preferred because it involves less complex and faster mathematics, and generates an aggregated output surface that is easier to defuzzify.
Mamdani-style rule evaluation

- While clipping is a frequently used method, *scaling* offers a better approach for preserving the original shape of the FS.
  - The original MF of the rule consequent is adjusted by *multiplying* all its membership degrees by the truth value of the rule antecedent.
  - This method, which generally loses less information, can be very useful in fuzzy expert systems.

Fuzzy implication:
  i) Mamdani’s method (Min-method) - *clipping*
  ii) Larsen’s product method - *scaling*
Clipped and Scaled Membership Functions

Clipping

Scaling
Step 3: Aggregation of the Rule Outputs

Aggregation is the process of unification of the outputs of all rules.

We take the MFs of all rule consequents previously clipped or scaled and combine them into a single FS.

The input of the aggregation process is the list of qualified (clipped or scaled) consequent MFs, and the output is one FS for each output variable.
Aggregation of the Rule Outputs

\[ \text{z is } C_1 (0.1) \rightarrow \text{z is } C_2 (0.2) \rightarrow \text{z is } C_3 (0.5) \rightarrow \sum \]
The last step in the fuzzy inference process is defuzzification.

Fuzziness helps us evaluate the rules, but the final output of a fuzzy system has to be a crisp number.

The input for the defuzzification process is the aggregate output FS and the output is a single number.
There are several defuzzification methods, but probably the most popular one is the centroid technique.

- It finds the point where a vertical line would slice the aggregate set into two equal masses.

Mathematically, this *center of gravity* (COG) can be expressed as:

\[
COG = \frac{\int_a^b \mu_A(x) x \, dx}{\int_a^b \mu_A(x) \, dx}
\]
Step 4. Defuzzification

- Centroid defuzzification method finds a point representing the center of gravity of the FS $A$, on the interval, $ab$.

- A reasonable estimate can be obtained by calculating it over a sample of points.

Ex)

\[ \mu(x) \]

![Diagram showing defuzzification process with points $a$ and $b$ and the centroid $A$.]
Center of Gravity (COG)

\[ COG = \frac{(0 + 10 + 20) \times 0.1 + (30 + 40 + 50 + 60) \times 0.2 + (70 + 80 + 90 + 100) \times 0.5}{0.1 + 0.1 + 0.1 + 0.2 + 0.2 + 0.2 + 0.2 + 0.5 + 0.5 + 0.5 + 0.5} = 67.4 \]
A singleton, or more precisely a fuzzy singleton, is a FS with a MF that is unity at a single particular point on the universe of discourse and zero everywhere else.

Mamdani-style inference, as we have just seen, requires us to find the centroid of a two-dimensional shape by integrating across a continuously varying function. In general, this process is not computationally efficient.

Sugeno suggested to use a single spike, a singleton, as the MF of the rule consequent.

- A singleton, or more precisely a fuzzy singleton, is a FS with a MF that is unity at a single particular point on the universe of discourse and zero everywhere else.
Sugeno-style fuzzy inference is very similar to the Mamdani method. Sugeno changed only a rule consequent. Instead of a FS, he used a *mathematical function of the input variable*.

The format of the Sugeno-style fuzzy rule is

\[
\text{IF } x \text{ is } A \\
\text{AND } y \text{ is } B \\
\text{THEN } z \text{ is } f(x, y)
\]

where \( x, y \) and \( z \) are linguistic variables; \( A \) and \( B \) are FSs on universe of discourses \( X \) and \( Y \), respectively; and \( f(x, y) \) is a mathematical function.
The most commonly used zero-order Sugeno fuzzy model applies to fuzzy rules in the following form:

\[
\text{IF } x \text{ is } A \\
\text{AND } y \text{ is } B \\
\text{THEN } z \text{ is } k
\]

where \( k \) is a constant.

In this case, the output of each fuzzy rule is constant. All consequent MFs are represented by fuzzy singletons.
Sugeno-style rule evaluation

**Rule 1:** IF $x$ is $A_3$ (0.0) OR $y$ is $B_1$ (0.1) THEN $z$ is $k_1$ (0.1)

**Rule 2:** IF $x$ is $A_2$ (0.2) AND $y$ is $B_2$ (0.7) THEN $z$ is $k_2$ (0.2)

**Rule 3:** IF $x$ is $A_1$ (0.5) THEN $z$ is $k_3$ (0.5)
Sugeno-style aggregation of the rule outputs
Weighted average (\(WA\)):

\[
WA = \frac{\mu(k_1) \times k_1 + \mu(k_2) \times k_2 + \mu(k_3) \times k_3}{\mu(k_1) + \mu(k_2) + \mu(k_3)} = \frac{0.1 \times 20 + 0.2 \times 50 + 0.5 \times 80}{0.1 + 0.2 + 0.5} = 65
\]

Sugeno-style defuzzification

Crisp Output
How to make a decision on which method to apply?

- Mamdani method is widely accepted for capturing expert knowledge. It allows us to describe the expertise in more intuitive, more human-like manner.

  However, Mamdani-type fuzzy inference entails a substantial computational burden.

- On the other hand, Sugeno method is computationally effective and works well with optimization and adaptive techniques, which makes it very attractive in control problems, particularly for dynamic nonlinear systems.
Case Study: Building a Fuzzy Expert System

- A service center keeps spare parts and repairs failed ones.
- A customer brings a failed item and receives a spare of the same type.
- Failed parts are repaired, placed on the shelf, and thus become spares.
- The objective here is to advise a manager of the service center on certain decision policies to keep the customers satisfied.
Process of developing a fuzzy expert system

1. Specify the problem and define linguistic variables.
2. Determine FSs.
3. Elicit and construct fuzzy rules.
4. Encode the FSs, fuzzy rules and procedures to perform fuzzy inference into the expert system.
5. Evaluate and tune the system.
Step 1: Specify the problem and define linguistic variables

There are four main linguistic variables:

- average waiting time (mean delay), $m$,
- repair utilization factor of the service center, $\rho$,
- the number of servers, $s$, and
- the initial number of spare parts, $n$. 

Linguistic variables and their ranges

<table>
<thead>
<tr>
<th>Linguistic Variable: Mean Delay, $m$</th>
<th>Linguistic Value</th>
<th>Notation</th>
<th>Numerical Range (normalised)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Short</td>
<td>VS</td>
<td>[0, 0.3]</td>
<td></td>
</tr>
<tr>
<td>Short</td>
<td>S</td>
<td>[0.1, 0.5]</td>
<td></td>
</tr>
<tr>
<td>Medium</td>
<td>M</td>
<td>[0.4, 0.7]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Linguistic Variable: Number of Servers, $s$</th>
<th>Linguistic Value</th>
<th>Notation</th>
<th>Numerical Range (normalised)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Small</td>
<td>S</td>
<td>[0, 0.35]</td>
<td></td>
</tr>
<tr>
<td>Medium</td>
<td>M</td>
<td>[0.30, 0.70]</td>
<td></td>
</tr>
<tr>
<td>Large</td>
<td>L</td>
<td>[0.60, 1]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Linguistic Variable: Repair Utilisation Factor, $\rho$</th>
<th>Linguistic Value</th>
<th>Notation</th>
<th>Numerical Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Low</td>
<td>L</td>
<td>[0, 0.6]</td>
<td></td>
</tr>
<tr>
<td>Medium</td>
<td>M</td>
<td>[0.4, 0.8]</td>
<td></td>
</tr>
<tr>
<td>High</td>
<td>H</td>
<td>[0.6, 1]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Linguistic Variable: Number of Spares, $n$</th>
<th>Linguistic Value</th>
<th>Notation</th>
<th>Numerical Range (normalised)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Very Small</td>
<td>VS</td>
<td>[0, 0.30]</td>
<td></td>
</tr>
<tr>
<td>Small</td>
<td>S</td>
<td>[0, 0.40]</td>
<td></td>
</tr>
<tr>
<td>Rather Small</td>
<td>RS</td>
<td>[0.25, 0.45]</td>
<td></td>
</tr>
<tr>
<td>Medium</td>
<td>M</td>
<td>[0.30, 0.70]</td>
<td></td>
</tr>
<tr>
<td>Rather Large</td>
<td>RL</td>
<td>[0.55, 0.75]</td>
<td></td>
</tr>
<tr>
<td>Large</td>
<td>L</td>
<td>[0.60, 1]</td>
<td></td>
</tr>
<tr>
<td>Very Large</td>
<td>VL</td>
<td>[0.70, 1]</td>
<td></td>
</tr>
</tbody>
</table>
FSs can have a variety of shapes. However, a triangle or a trapezoid can often provide an adequate representation of the expert knowledge, and at the same time, significantly simplifies the process of computation.
Fuzzy Sets of Mean Delay $m$

Degree of Membership

Mean Delay (normalised)
Fuzzy Sets of the Number of Servers $s$

Degree of Membership

Number of Servers (normalised)
Fuzzy Sets of Repair Utilization Factor $\rho$
Fuzzy Sets of the Number of Spares $n$

Degree of Membership

VS  S  RS  M  RL  L  VL

Number of Spares (normalised)
Step 3: Elicit and construct fuzzy rules

To accomplish this task, we might ask the expert to describe how the problem can be solved using the fuzzy linguistic variables defined previously.

Required knowledge also can be collected from other sources such as books, computer databases, flow diagrams and observed human behavior.
The square FAM representation
## The Rule Table

<table>
<thead>
<tr>
<th>Rule</th>
<th>$m$</th>
<th>$s$</th>
<th>$\rho$</th>
<th>$n$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>VS</td>
<td>S</td>
<td>L</td>
<td>VS</td>
</tr>
<tr>
<td>2</td>
<td>S</td>
<td>S</td>
<td>L</td>
<td>VS</td>
</tr>
<tr>
<td>3</td>
<td>M</td>
<td>S</td>
<td>L</td>
<td>VS</td>
</tr>
<tr>
<td>4</td>
<td>VS</td>
<td>M</td>
<td>L</td>
<td>VS</td>
</tr>
<tr>
<td>5</td>
<td>S</td>
<td>M</td>
<td>L</td>
<td>VS</td>
</tr>
<tr>
<td>6</td>
<td>M</td>
<td>M</td>
<td>L</td>
<td>VS</td>
</tr>
<tr>
<td>7</td>
<td>VS</td>
<td>L</td>
<td>L</td>
<td>S</td>
</tr>
<tr>
<td>8</td>
<td>S</td>
<td>L</td>
<td>L</td>
<td>S</td>
</tr>
<tr>
<td>9</td>
<td>M</td>
<td>L</td>
<td>L</td>
<td>VS</td>
</tr>
<tr>
<td>10</td>
<td>VS</td>
<td>S</td>
<td>M</td>
<td>S</td>
</tr>
<tr>
<td>11</td>
<td>S</td>
<td>S</td>
<td>M</td>
<td>VS</td>
</tr>
<tr>
<td>12</td>
<td>M</td>
<td>S</td>
<td>M</td>
<td>VS</td>
</tr>
<tr>
<td>13</td>
<td>VS</td>
<td>M</td>
<td>M</td>
<td>RS</td>
</tr>
<tr>
<td>14</td>
<td>S</td>
<td>M</td>
<td>M</td>
<td>S</td>
</tr>
<tr>
<td>15</td>
<td>M</td>
<td>M</td>
<td>M</td>
<td>VS</td>
</tr>
<tr>
<td>16</td>
<td>VS</td>
<td>L</td>
<td>M</td>
<td>M</td>
</tr>
<tr>
<td>17</td>
<td>S</td>
<td>L</td>
<td>M</td>
<td>RS</td>
</tr>
<tr>
<td>18</td>
<td>M</td>
<td>L</td>
<td>M</td>
<td>S</td>
</tr>
<tr>
<td>19</td>
<td>VS</td>
<td>S</td>
<td>H</td>
<td>VL</td>
</tr>
<tr>
<td>20</td>
<td>S</td>
<td>S</td>
<td>H</td>
<td>L</td>
</tr>
<tr>
<td>21</td>
<td>M</td>
<td>S</td>
<td>H</td>
<td>M</td>
</tr>
<tr>
<td>22</td>
<td>VS</td>
<td>M</td>
<td>H</td>
<td>M</td>
</tr>
<tr>
<td>23</td>
<td>S</td>
<td>M</td>
<td>H</td>
<td>M</td>
</tr>
<tr>
<td>24</td>
<td>M</td>
<td>M</td>
<td>H</td>
<td>S</td>
</tr>
<tr>
<td>25</td>
<td>VS</td>
<td>L</td>
<td>H</td>
<td>RL</td>
</tr>
<tr>
<td>26</td>
<td>S</td>
<td>L</td>
<td>H</td>
<td>M</td>
</tr>
<tr>
<td>27</td>
<td>M</td>
<td>L</td>
<td>H</td>
<td>RS</td>
</tr>
</tbody>
</table>
1. If (utilisation_factor is L) then (number_of_spares is S)
2. If (utilisation_factor is M) then (number_of_spares is M)
3. If (utilisation_factor is H) then (number_of_spares is L)
4. If (mean_delay is VS) and (number_of_servers is S) then (number_of_spares is VL)
5. If (mean_delay is S) and (number_of_servers is S) then (number_of_spares is L)
6. If (mean_delay is M) and (number_of_servers is S) then (number_of_spares is M)
7. If (mean_delay is VS) and (number_of_servers is M) then (number_of_spares is RL)
8. If (mean_delay is S) and (number_of_servers is M) then (number_of_spares is RS)
9. If (mean_delay is M) and (number_of_servers is M) then (number_of_spares is S)
10. If (mean_delay is VS) and (number_of_servers is L) then (number_of_spares is M)
11. If (mean_delay is S) and (number_of_servers is L) then (number_of_spares is S)
12. If (mean_delay is M) and (number_of_servers is L) then (number_of_spares is VS)
Cube FAM of Rule Base 2
Step 4: Encoding and fuzzy inference into the expert system

Encode the FSs, fuzzy rules and procedures to perform fuzzy inference into the expert system.

To accomplish this task, we may choose one of two options: to build our system using a programming language such as C/C++ or Pascal, or to apply a fuzzy logic development tool such as MATLAB Fuzzy Logic Toolbox or Fuzzy Knowledge Builder.
The last, and the most laborious, task is to evaluate and tune the system. We want to see if our fuzzy system meets the requirements specified at the beginning.

Several test situations depend on the mean delay, the number of servers and repair utilization factor.

The Fuzzy Logic Toolbox can generate surface to help us analyze the system’s performance.
Three-dimensional plots for Rule Base 1
Three-dimensional plots for Rule Base 1

- Number of spares
- Mean delay
- Utilisation factor
Three-dimensional plots for Rule Base 2
Three-dimensional plots for Rule Base 2
However, even now, the expert might not be satisfied with the system performance.

To improve the system performance, we may use additional sets - *Rather Small* and *Rather Large* – on the universe of discourse of the *Number of Servers*, and then extend the rule base.
Modified Fuzzy Sets of the Number of Servers $s$

Degree of Membership

Number of Servers (normalised)
Cube FAM of Rule Base 3
Three-dimensional plots for Rule Base 3
Three-dimensional plots for Rule Base 3
Tuning Fuzzy Systems

1. Review model input and output variables, and if required, redefine their ranges.

2. Review the FSs, and if required, define additional sets on the universe of discourse.
   - The use of wide FSs may cause the fuzzy system to perform roughly.

3. Provide sufficient overlap between neighboring sets.
   - It is suggested that triangle-to-triangle and trapezoid-to-triangle FSs should overlap between 25% to 50% of their bases.
4. Review the existing rules, and if required, add new rules to the rule base.

5. Examine the rule base for opportunities to write hedge rules to capture the pathological behavior of the system.

6. Adjust the rule execution weights.
   - Most fuzzy logic tools allow control of *the importance* of rules by changing a weight multiplier.

7. Revise shapes of the FSs.
   - In most cases, fuzzy systems are highly tolerant of a shape approximation.