Fuzzy expert system

Small example

The problem

- we will consider a problem of operating a service center of spare parts (Turksen *et al*., 1992).
- 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. If the required spare is available on the shelf, the customer takes it and leaves the service center. However, if there is no spare on the shelf, the customer has to wait until the needed item becomes available. The objective here is to advise a manager of the service center on certain decision policies to keep the customers satisfied.

process in developing the fuzzy expert system

1. Specify the problem and define linguistic variables.
2. Determine fuzzy sets.
3. construct fuzzy rules.
4. Encode the fuzzy sets, 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

- determine problem input and output variables and their ranges.
- For our problem, there are four main linguistic variables:
  - average waiting time (mean delay) *m*
  - repair utilization factor of the service center
  - number of servers *s*
  - initial number of spare parts *n*.

- The customer's average waiting time, *m*, (the most important) should not exceed the limits acceptable to customers.
- The repair utilization factor of the service center, p, is the ratio of the customer arrival rate to the customer departure rate. Apparently, the repair rate is proportional to the number of servers, *s*.
- The number of servers, *s*, and the initial number of spares, *n*, directly affect the customer's average waiting time. By increasing *s* and *n*, we decrease mean delay, but, at the same time we increase the costs of employing new servers, building up the number of spares and expanding the inventory .

Linguistic variable: Mean delay, *m*		
Linguistic value	**Notation**	**Numerical range (normalised)**
Very Short	VS	[0, 0.3]
Short	S	[0.1, 0.5]
Medium	M	[0.4, 0.7]
Linguistic variable: Number of servers, *s*		
Linguistic value	**Notation**	**Numerical range (normalised)**
Small	S	[0, 0.35]
Medium	M	[0.30, 0.70]
Large	L	[0.60, 1]

Linguistic variable: Repair utilisation factor, ρ		
Linguistic value	**Notation**	**Numerical range**
Low	L	[0, 0.6]
Medium	M	[0.4, 0.8]
High	H	[0.6, 1]

Linguistic variable: Number of spares, n		
Linguistic value	**Notation**	**Numerical range (normalised)**
Very Small	VS	[0, 0.30]
Small	S	[0, 0.40]
Rather Small	RS	[0.25, 0.45]
Medium	M	[0.30, 0.70]
Rather Large	RL	[0.55, 0.75]
Large	L	[0.60, 1]
Very Large	VL	[0.70, 1]

process in developing the fuzzy expert system

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

Step 2: Determine fuzzy sets

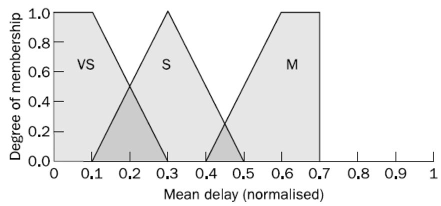

Figure 4.16 Fuzzy sets of mean delay *m*

Step 2: Determine fuzzy sets

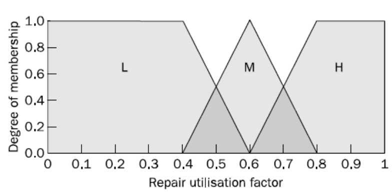

Figure 4.18 Fuzzy sets of repair utilisation factor ρ

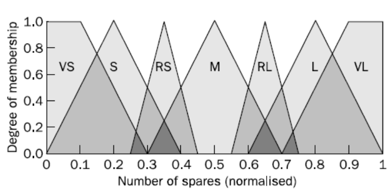

Figure 4.19 Fuzzy sets of number of spares *n*

process in developing the fuzzy expert system

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

Step 3: construct fuzzy rules

- ask the expert .
- Required knowledge also can be collected from other sources such as books, computer databases, flow diagrams and observed human behavior.
- In our case, we could apply rules provided in the research paper (Turksen *et al*., 1992).

Step 3: construct fuzzy rules

- Our problem is three input and one output variables
- represent fuzzy rules in a matrix form. A two-by one system (two inputs and one output) is depicted as an **M**x**N** matrix of input variables.
- For a three-by-one system (three inputs and one output), the representation takes the shape of an **M**x**N**x**K** cube. This form of representation is called a fuzzy associative memory (FAM).

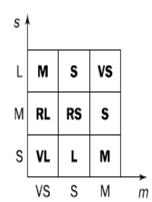

Figure 4.20 The square FAM representation

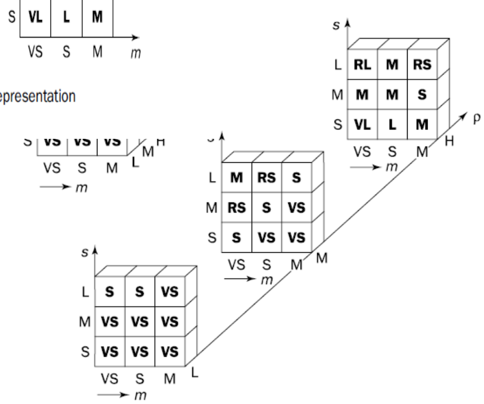

Figure 4.21 Cube FAM and sliced cube FAM representations

- The relation between the repair utilization factor , and the number of spares *n*, assuming that other input variables are fixed. This relation can be expressed in the following form:

  if increases, then *n* will not decrease. So

  we could write the following three rules:

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)

Table 4.4 The rule table

Rule	*m*	*s*	ρ	*n*	Rule	*m*	*s*	ρ	*n*	Rule	*m*	*s*	ρ	*n*
1	VS	S	L	VS	10	VS	S	M	S	19	VS	S	H	VL
2	S	S	L	VS	11	S	S	M	VS	20	S	S	H	L
3	M	S	L	VS	12	M	S	M	VS	21	M	S	H	M
4	VS	M	L	VS	13	VS	M	M	RS	22	VS	M	H	M
5	S	M	L	VS	14	S	M	M	S	23	S	M	H	M
6	M	M	L	VS	15	M	M	M	VS	24	M	M	H	S
7	VS	L	L	S	16	VS	L	M	M	25	VS	L	H	RL
8	S	L	L	S	17	S	L	M	RS	26	S	L	H	M
9	M	L	L	VS	18	M	L	M	S	27	M	L	H	RS

Rule Base 1

1. If (utilisation_factor is L) then (number_of_spares is S)
2. If (utilisation_factor is M) then (number_of_spares is M)
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)

Rule Base 2

1. If (mean_delay is VS) and (number_of_servers is S) and (utilisation_factor is L) then (number_of_spares is VS)
2. If (mean_delay is S) and (number_of_servers is S) and (utilisation_factor is L) then (number_of_spares is VS)
3. If (mean_delay is M) and (number_of_servers is S) and (utilisation_factor is L) then (number_of_spares is VS)

process in developing the fuzzy expert system

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

Step 4: Encode the fuzzy sets, fuzzy rules and procedures to perform fuzzy inference into the expert system

- we may choose one of two options:
  - to build our system using a programming language such as C or Pascal,
  - or to apply a fuzzy logic development tool such as MATLAB Fuzzy Logic Toolbox from the MathWorks or Fuzzy Knowledge Builder from Fuzzy Systems Engineering.

process in developing the fuzzy expert system

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

Step 5: Evaluate and tune the system

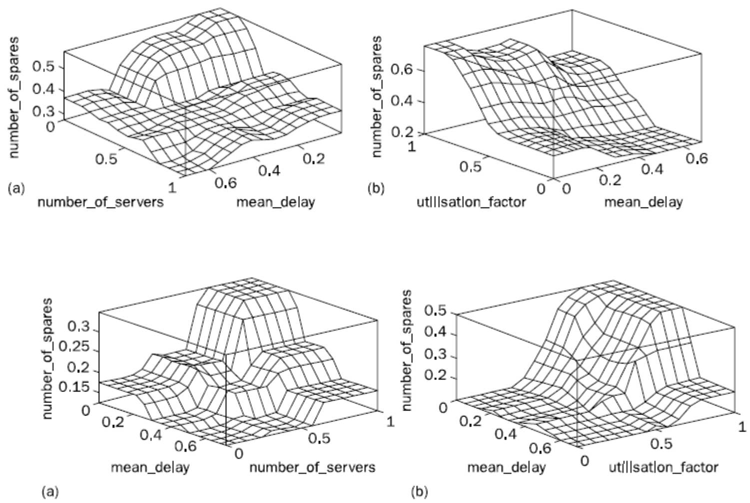

Figure 4.23 Three-dimensional plots for rule base 2

Tuning fuzzy systems

1) Review model input and output variables, and if required redefine their ranges. Pay particular attention to the variable units. Variables used in the same domain must be measured in the same units on the universe of discourse.

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

3) Provide sufficient overlap between neighboring sets. Although there is no precise method to determine the optimum amount of overlap, it is suggested that triangle-to-triangle and trapezoid-to-triangle fuzzy sets should overlap between 25 and 50 per cent of their bases (Cox, 1999).

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. In the Fuzzy Logic Toolbox, all rules have a default weight of (1.0), but the user can reduce the force of any rule by adjusting its weight. For example, if we specify If (utilisation_factor is H) then (number_of_spares is L) (0.6) then the rule's force will be reduced by 40 per cent.

7) Revise shapes of the fuzzy sets. In most cases, fuzzy systems are highly tolerant of a shape approximation, and thus a system can still behave well even when the shapes of the fuzzy sets are not precisely defined.

Increase the number of servers

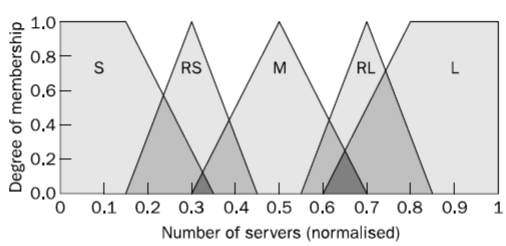

Figure 4.24 Modified fuzzy sets of number of servers *s*

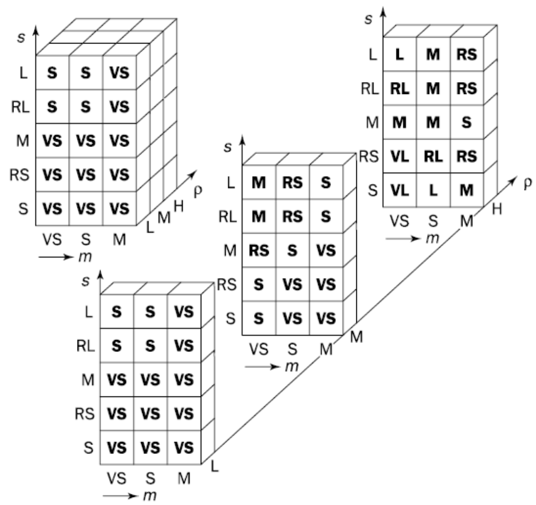

Figure 4.25 Cube FAM of rule base 3

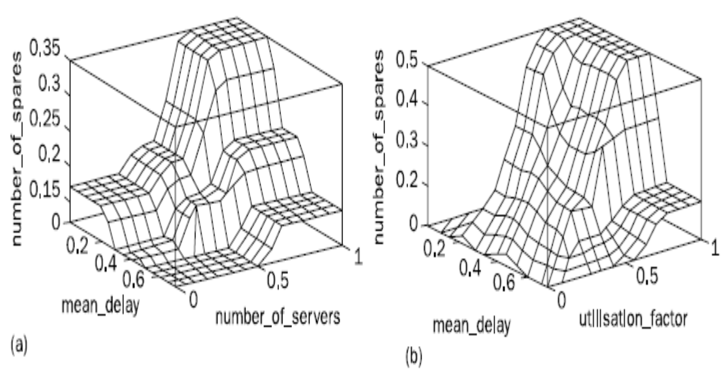

Figure 4.26 Three-dimensional plots for rule base 3