Computational Intelligence
Part II
Lecture 4: Fuzzy Systems

Farzaneh Abdollahi

Department of Electrical Engineering

Amirkabir University of Technology

Fall 2009

Fuzzifier

Defuzzifier

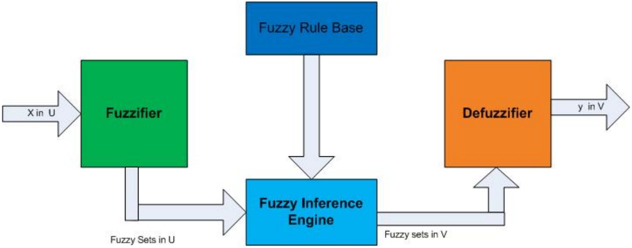
Fuzzy Rule Base

Inference Engine

1. Inference based on Combination Rules:
2. Inference based on Separation Rules:

Fuzzy Systems

- Fuzzy system scheme



Fuzzifier

- Fuzzzifier is a function from a point $X^* \in U \subset R^n$ to a fuzzy set A' in U.
- a proper fuzzuifier should
  - be able to attenuate or eliminate the effect of noise on corrupted input data
  - simplify the computations in inference engine.
    - One of the most complicated part of inference engine computations is finding $\sup_{X \in U}$

1. Singleton fuzzifier $$\mu_{A'}(X) = \begin{cases} 1, & X = X^* \\ 0, & \text{otherwise} \end{cases}$$
2. Gaussian fuzzifier $$\mu_{A'}(X) = e^{[\frac{x_1 - x_1^*}{a_1}]^2} * \ldots * e^{[\frac{x_n - x_n^*}{a_n}]^2}$$

   where a_i are pos. conts, and $*$ is t-norm which is selected by production or min
3. Triangular fuzzifier $$\mu_{A'}(X) = \begin{cases} [1 - \frac{|x_1 - x_1^*}{b_1}|] * \ldots * [1 - |\frac{x_n - x_n^*}{b_n}|], & |x_i - x_i^*| \leq b_i \\ 0, & \text{otherwise} \end{cases}$$

   where b_is are pos. conts, and $*$ is t-norm which is selected by production or min

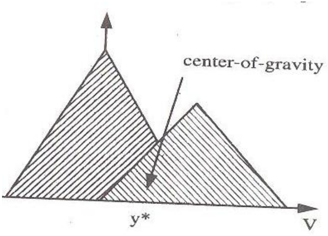
Fuzzifier

- Singleton fuzzifier simplifies the computations of inference engine for all types of function in IF-then rules.
- If the function If-then fuzzy rules are Gaussian/triangular the Gaussian/triangular fuzzifiers simplify the computations
- Gaussian and triangular fuzzifiers can eliminate the input noise effect, whereas, singleton fuzzifier cannot.
- **Defuzzifier:** is a function defined from a fuzzy set B' in $V \subset R$ (output of inference engine) to a point $y^* \in V$
  - Defuzzifier finds the best representation of fuzzy set B'.
  - Three issues should be considered in selecting defuzzifier
    1. Reasonability: y^* should be a proper representation of fuzzy set B', for example middle of B'.
    2. Simplify the computations It is important specially for fuzzy controls which are applied in realtime.
    3. Continuity: a small change in B' should not result in large change in y^*

Defuzzifier

1. Center of gravity:

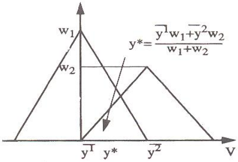
$$y^* = \frac{\int_V y \mu_{B'}(y) dy}{\int_V \mu_{B'}(y) dy}$$



2. Center Average: is the most popular diffuzifier in fuzzy control.

$$y^* = \frac{\sum_{l=1}^{M} \bar{y}^l w_l}{\sum_{l=1}^{M} w_l}$$

where $\bar{y}^l$ is the center of lth fuzzy set and w^l is its height



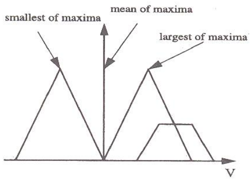
Defuzzifier

3. Maximum defuziifier: Consider

$$hgt(B') = \{y \in V | \mu_{B'}(y) - \sup_{y \in V} \mu_{B'}(y)\}$$

- $hgt(B')$ is a set of all point is V s.t. their $\mu_{B'}(y)$ is maximum
- Maximum defuzzifier defines y^* as a point in $y^* = hgt(B')$.
- If hgt has more than one member the defuzzifier can be defined as
  - Infimum of maxima: $y^* = \inf\{y \in hgt(B')\}$
  - Suprimum of maxima: $y^* = \sup\{y \in hgt(B')\}$
  - Mean of maxima:

$$y^* = \frac{\int_{hgt(B')} y dy}{\int_{hgt(B')} dy}$$



Defuzzifier

- The table below compares the performance of the three introduced defuzzifiers:

	Center of gravity	Center average	Max
Reasonability	yes	yes	yes
Simplifying the computations	no	yes	yes
Continuity	yes	yes	no

Fuzzy Rule Base

- Fuzzy rule base is a set of If-then rules:

$$Ru^l : \text{If} x_1, A_1^l, \ldots, x_n, A_n^l \text{then } y, B_l \tag{1}$$

  where A_i^l, B^l are fuzzy sets in $U_i \subset R$ and $V \subset R$, respectively; $x = (x_1, \ldots, x_n)$ and $y \in V$ are input and output of the fuzzy system, respectively; l is number of rules
- It can be considered as heart of a fuzzy system
- If-then rules can also be defined as
  - OR $$Ru^l : \text{If} x_1, A_1^l, \ldots, x_m, A_m^l, \text{or If} x_{m+1}, A_{m+1}^l, \ldots, x_n, A_n^l \text{then} y, B_l$$
  - Single $$y, B_l$$
- A good fuzzy rule based should be
  - Complete: for each $x \in U$ there is at least one rule.
  - Consistent: There is no rule with similar If but different Then
  - Continuous: Intersection of Then fuzzy sets of neighbor rules is not empty.

Inference Engine

- Inference engine maps the input given from fuzzifier to an output in defuzzifier using fuzzy logic.
- Inference engine is defined based on one of the following two methods
  1. Inference based on Combination Rules : All the rules in fuzzy rule base are combined and considered as a unified If-then rule. This combination could be based on
     - Union operator: This method is also called Mamadani Combination: Considering M rules defined as (1), the fuzzy relation Q_M in $U \times xV$ is

$$Q_M = \cup_{l=1}^{M} Ru^{(l)} \tag{2}$$

$$\mu_{Q_M}(x, y) = \mu_{ru^{(1)}}(x, y) \dot{+} \ldots \dot{+} \mu_{Ru^{(M)}}(x, y)$$

where $\dot{+}$ is s-norm

     - Intersection operatorThis method is also called Godal Combination

$$Q_G = \cap_{l=1}^{M} Ru^{(l)} \tag{3}$$

$$\mu_{Q_G}(x, y) = \mu_{ru^{(1)}}(x, y) * \ldots * \mu_{Ru^{(M)}}(x, y)$$

where $*$ is t-norm

Inference based on Rule Combination:

- The procedure of manipulation in inference engine based on combination rules is as follows
  1. For M fuzzy rules defined based on (1), find membership function as follows
  $$\mu_{A_1^l\times\ldots\times A_n^l}(x_1,\ldots,x_n)=\mu_{A_1^l}(x_1)*\ldots*\mu_{A_n^l}(x_n)$$
  for $l=1,\ldots,M$, $*$ is t-norm
  2. By considering $A_1^l\times\ldots\times A_n^l$ as premise (FP1) and B^l as conclusion (FP2) of one of implications such as Godal, mamdani, Zadeh, etc. (For example Mamdani implication is: $\mu_{Q_MM}(x,y)=\min[\mu_{FP1}(x),\mu_{FP2}(y)]$ or $\mu_{Q_MM}(x,y)=\mu_{FP1}(x),\mu_{FP2}(y))$, find $\mu_{ru^{(l)}}(x_1,\ldots,x_n)=\mu_{A_1^l\times\ldots\times A_n^l\to B}(x_1,\ldots,x_n)$
  3. Using (2) or (3) find $\mu_{Q_M}(x,y)$ or $\mu_{Q_G}(x,y)$
  4. For given input A' find output of inference engine, B' based on one of the following equations:
  For Mamdani combination $\mu_{B'}(y)=\sup_{x\in U}t[\mu_{A'}(x),\mu_{Q_M}(x,y)]$;
  for Godal combination $\mu_{B'}(y)=\sup_{x\in U}t[\mu_{A'}(x),\mu_{Q_G}(x,y)]$

2. Inference based on Separate Rules:

- Each rule in fuzzy rule base provides an output, the final output is combination of the M outputs
- The procedure of manipulation in inference engine based on separate rule is as follows
  1. Follow the first and second steps of the inference engine based on combination rules procedure.
  2. For each fuzzy set A' in U find output B'_l in V for each rule $Ru^{(l)}$, i.e.,

$$\mu_{B'_l}(y) = \sup t_{x \in U}[\mu_{A'}(x), \mu_{Ru^{(l)}}(x, y)], \quad l = 1, \ldots, M$$

  3. The output of the inference engin is a combination of M fuzzy output $\{B'_l, \ldots, B'_M\}$ in one of the following two methods:
     - Union: $\mu_{B'}(y) = \mu_{B'_1} \dot{+} \ldots \dot{+} \mu_{B'_M}(y)$
     - Intersection $\mu_{B'}(y) = \mu_{B'_1} * \ldots * \mu_{B'_M}(y)$ where $\dot{+}$ and $*$ are s-norm and t-norm respectively.

- The issues to select an inference engine
  - Select the inference engine based on the structure of of rules set. For example if the expert who defines the set of rules believes that the rules are independent of each other, the rules should be combined by union.
  - Computation: Based on defined fuzzy sets, an engine should be selected to have less complicated and easier computations
- Some of popular inference engines in fuzzy systems and control are:
  - Production: Inference is based on separate rules with union combination; Implication Mamdani product; t-norm: algebraic production; s-norm: max

$$\mu_{B'}(y) = max_{l=1}^{M}[\sup_{x\in U}(\mu_{A'}(x)\prod_{i=1}^{n}\mu_{A_i}(x_i)\mu_{B^l}(y))]$$

  - Minimum:Inference is based on separate rules with union combination; Implication: Mamdani minimum; t-norm: min; s-norm: max

$$\mu_{B'}(y) = max_{l=1}^{M}[\sup_{x\in U}\min(\mu_{A'}(x), \mu_{A_1^l}(x_1), \ldots, \mu_{A_n^l}(x_n), \mu_{B^l}(y)]$$

- Lemma: *Consider the fuzzy set B^l in (1) with center $\bar{y}^l$, The system fuzzy with the following specifications: fuzzy rule base (1); Inference engine: product; fuzzifier: singleton; defuzzifier: mean center, can be expressed as*

$$f(x) = \frac{\sum_{l=1}^{M} \bar{y}^l (\prod_{i=1}^{n} \mu_{A_i^l}(x_i))}{\sum_{l=1}^{M} (\prod_{i=1}^{n} \mu_{A_i^l}(x_i))}$$