Computational Intelligence
Lecture 6:Fuzzy Rule Base and Fuzzy Inference Engine

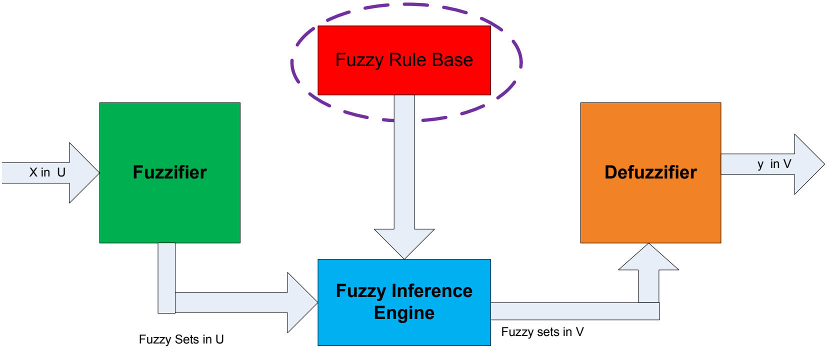
Farzaneh Abdollahi

Department of Electrical Engineering

Amirkabir University of Technology

Fall 2010

Fuzzy Rule Base



Fuzzy Rule Base

- We study only the multi-input-single-output case
- a multioutput system can be decomposed into a collection of single-output systems.
- A fuzzy rule base : a set of fuzzy IF-THEN rules.
- It is ♡ of the fuzzy system: all other components are used to implement these rules in a reasonable and efficient manner.
- The Canonical Fuzzy IF-THEN Rule
  $Ru^{(l)}$: IF x_1 is A_1^l and ... and x_n^l is A_n^l, THEN y is B^l
  - A_i^l: fuzzy set in $U_i \subset R$
  - B_l: Fuzzy set in $V \subset R$
  - $X = (x_1, ..., x_n)^T \in U$: Input linguistic variable
  - $y \in V$: Output linguistic variable
  - $l = 1, ..., M$, M: # of rules in the fuzzy rule base

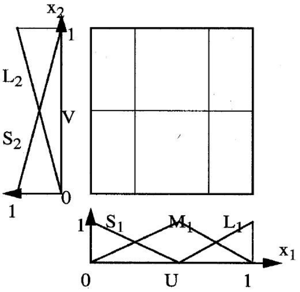
Most of Fuzzy Rules can be expressed by Canonical Fuzzy Rule

- Partial Rules:
  IF x_1 is A_1^l and ... and x_m is A_m^l, THEN y is B^l, $m < n$
  - It is equivalent to:
    IF x_1 is A_1^l and ... and x_m is A_m^l and x_{m+1} is I and ... and x_n is I THEN y is B^l
  - I is a fuzzy set in R, $\mu_I(x) = 1, \forall x \in R$
- Or Rules:
  IF x_1 is A_1^l and ... and x_m is A_m^l or x_{m+1} is A_{m+1}^l and ... and x_n is A_n^l THEN y is B^l,
  - It is equivalent to:
    IF x_1 is A_1^l and ... and x_m is A_m^l THEN y is B^l
    IF x_{m+1} is A_{m+1}^l and ... and x_n is A_n^l THEN y is B^l

- Single fuzzy statement:
  y **is** B^l,
  - It is equivalent to:
    IF x_1 is I and ... and x_n is I THEN y is B^l
- Gradual Rules:
  The smaller the x**, the bigger the** y,
  - Define S : a fuzzy set for "smaller": $\mu_S(x) = \frac{1}{1+exp(5(x+2))}$
  - Define B : a fuzzy set for "bigger": $\mu_B(y) = \frac{1}{1+exp(-5(y-2))}$
  - The rule is equivalent to:
    IF x is S THEN y is B
- Non-fuzzy Rules:
  - When $\mu_{A_i^l}$ and μ_{B^l} can only take values 1 or 0

Properties of Set of Rules

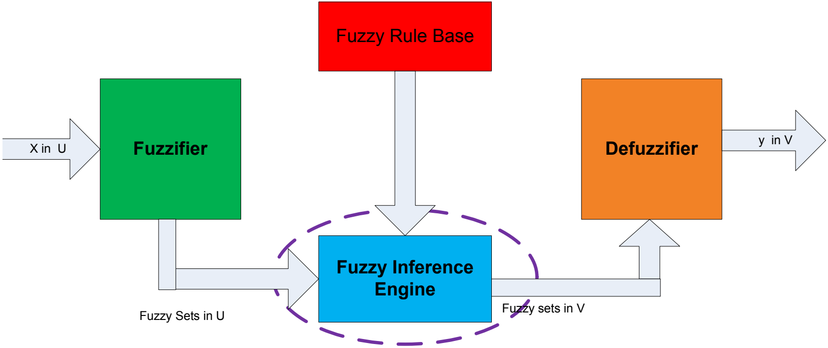
- **Complete Fuzzy Rules**
  - $\forall x \in U, \exists$ at least one rule (k) in fuzzy rule base s.t. $\mu_{A_i^k} \neq 0$, for all $i = 1, \ldots, n$
  - **Example:**
    - Consider a 2-input 1-output system
    - The completer Fuzzy rule set:
      IF x_1 **is** S_1 **and** x_2 **is** S_2**, THEN** y **is** B^1
      IF x_1 **is** S_1 **and** x_2 **is** L_2**, THEN** y **is** B^2
      IF x_1 **is** M_1 **and** x_2 **is** S_2**, THEN** y **is** B^3
      IF x_1 **is** M_1 **and** x_2 **is** L_2**, THEN** y **is** B^4
      IF x_1 **is** L_1 **and** x_2 **is** S_2**, THEN** y **is** B^5
      IF x_1 **is** L_1 **and** x_2 **is** L_2**, THEN** y **is** B^6
    - If each of them is missing is will loose the completeness



Properties of Set of Rules

- **Consistent Fuzzy Rules:**
  there are no rules with the same IF parts but different THEN parts.
  - Consistency is very important for nonfuzzy production rules, it causes conflicting rules
  - For fuzzy rules: the proper inference engine and defuzzifier use average solution to resolve the conflicting results
- **Continuous Fuzzy Rules:**
  there is no neighboring rules whose THEN part fuzzy sets have empty intersection.
  - $\therefore$ I/O behavior of fuzzy system should be smooth

Fuzzy Inference



Fuzzy Inference

- Using fuzzy logic principles, the fuzzy IF-THEN rules in the fuzzy rule base are combined to map a fuzzy set A' in U to a fuzzy set B' in V.
- It is the brain of fuzzy system
- There are two methods to infer the rules
  1. Composition based inference
  2. Individual-rule based inference
- **Composition Based Inference**
  - all rules in the fuzzy rule base are combined into a single fuzzy relation in $U \times V$
  - It is viewed as a single fuzzy IF-THEN rule.

Composition Based Inference

- What is appropriate logic to combine?
- There are two diff. views:
  1. Each rule is independent conditional statement $\rightsquigarrow$ operator: union
     - For M rules, $Ru^{(l)} = A_1^l \times \ldots \times A_n^l \rightarrow B^l$
     - Interpreted as a single fuzzy relation called Mamdani combination $Q_M = \bigcup_{l=1}^{M} Ru^{(l)}$
     - $\mu_{Q_M}(x, y) = \mu_{Ru^{(1)}}(x, y) \dot{+} \ldots \dot{+} \mu_{Ru^{(1)}}(x, y)$
       $\dot{+}$ is s-norm
  2. The rules are strongly coupled conditional statements; all the rules must be satisfied to have impact $\rightsquigarrow$ appropriate operator is intersection
     - Combined as a fuzzy relation called Godel combination $\mu_{Q_G}(x, y) = \bigcap_{l=1}^{M} Ru^{(l)}$
     - $\mu_{Q_G}(x, y) = \mu_{Ru^{(1)}}(x, y) \star \ldots \star \mu_{Ru^{(1)}}(x, y)$
       $\star$ is t-norm

Composition Based Inference

- Output of fuzzy inference engine is obtained using the generalized modus ponens:
  $\mu_{B'}(y) = sup_{x \in U} t[\mu_{A'}(x), \mu_{Q_{M/G}}(x, y)]$
- **The computational procedure of the composition based inference:**
  1. Consider M canonical IF-THEN rules ($Ru^{(l)}$: IF x_1 is A_1^l and ... and x_n^l is A_n^l, THEN y is B^l), determine membership fcn: $\mu_{A_1^l} \times \ldots \times \mu_{A_n^l}(x_1, \ldots, x_n) = \mu_{A_1^l}(x_1) \star \ldots \star \mu_{A_n^l}(x_1)$ for $l = 1, ..., M$

Composition Based Inference

- Output of fuzzy inference engine is obtained using the generalized modus ponens:
  $\mu_{B'}(y) = sup_{x \in U} t[\mu_{A'}(x), \mu_{Q_{M/G}}(x, y)]$
- **The computational procedure of the composition based inference:**
  1. Consider M canonical IF-THEN rules ($\text{Ru}^{(l)}$: IF x_1 is A_1^l and ... and x_n^l is A_n^l, THEN y is B^l), determine membership fcn: $\mu_{A_1^l} \times \ldots \times \mu_{A_n^l}(x_1, \ldots, x_n) = \mu_{A_1^l}(x_1) \star \ldots \star \mu_{A_n^l}(x_1)$ for $l = 1, ..., M$
  2. Considering $A_1^l \times \ldots \times A_n^l$ as $FP1$, and B^l as $FP2$, determine $\mu_{Ru^{(l)}}(x_1, \ldots, x_n, y) = \mu_{A_1^l \times \ldots \times A_n^l \to B^l}(x_1, \ldots, x_n, y)$ for $l = 1, ..., M$ using Dienes, Lukasiewicz, Zadeh, Godel, or Mamdani implications

Composition Based Inference

- Output of fuzzy inference engine is obtained using the generalized modus ponens:
  $\mu_{B'}(y) = sup_{x \in U} t[\mu_{A'}(x), \mu_{Q_{M/G}}(x, y)]$
- **The computational procedure of the composition based inference:**
  1. Consider M canonical IF-THEN rules ($Ru^{(l)}$: IF x_1 is A_1^l and ... and x_n^l is A_n^l, THEN y is B^l), determine membership fcn: $\mu_{A_1^l} \times \ldots \times \mu_{A_n^l}(x_1, \ldots, x_n) = \mu_{A_1^l}(x_1) \star \ldots \star \mu_{A_n^l}(x_1)$ for $l = 1, ..., M$
  2. Considering $A_1^l \times \ldots \times A_n^l$ as $FP1$, and B^l as $FP2$, determine $\mu_{Ru^{(l)}}(x_1, \ldots, x_n, y) = \mu_{A_1^l \times \ldots \times A_n^l \rightarrow B^l}(x_1, \ldots, x_n, y)$ for $l = 1, ..., M$ using Dienes, Lukasiewicz, Zadeh, Godel, or Mamdani implications
  3. Determine $\mu_{Q_G}(x, y) = \mu_{Ru^{(1)}}(x, y) \star \ldots \star \mu_{Ru^{(1)}}(x, y)$ or $\mu_{Q_M}(x, y) = \mu_{Ru^{(1)}}(x, y) \dot{+} \ldots \dot{+} \mu_{Ru^{(1)}}(x, y)$

Composition Based Inference

- Output of fuzzy inference engine is obtained using the generalized modus ponens:
  $\mu_{B'}(y) = sup_{x \in U} t[\mu_{A'}(x), \mu_{Q_{M/G}}(x, y)]$
- **The computational procedure of the composition based inference:**
  1. Consider M canonical IF-THEN rules ($\text{Ru}^{(l)}$: IF x_1 is A_1^l and ... and x_n^l is A_n^l, THEN y is B^l), determine membership fcn: $\mu_{A_1^l} \times \ldots \times \mu_{A_n^l}(x_1, \ldots, x_n) = \mu_{A_1^l}(x_1) \star \ldots \star \mu_{A_n^l}(x_1)$ for $l = 1, ..., M$
  2. Considering $A_1^l \times \ldots \times A_n^l$ as $FP1$, and B^l as $FP2$, determine $\mu_{Ru^{(l)}}(x_1, \ldots, x_n, y) = \mu_{A_1^l \times \ldots \times A_n^l \to B^l}(x_1, \ldots, x_n, y)$ for $l = 1, ..., M$ using Dienes, Lukasiewicz, Zadeh, Godel, or Mamdani implications
  3. Determine $\mu_{Q_G}(x, y) = \mu_{Ru^{(1)}}(x, y) \star \ldots \star \mu_{Ru^{(1)}}(x, y)$ or $\mu_{Q_M}(x, y) = \mu_{Ru^{(1)}}(x, y) \dot{+} \ldots \dot{+} \mu_{Ru^{(1)}}(x, y)$
  4. Determine output B', give input A' and $\mu_{B'}(y) = sup_{x \in U} t[\mu_{A'}(x), \mu_{Q_{M/G}}(x, y)]$

Individual-Rule Based Inference

- Each rule in the fuzzy rule base determines an output fuzzy set
- The output of the whole fuzzy inference engine is the combination of the M individual fuzzy sets using intersection or union for combination.
- **The computational procedure of the individual-rule based inference:**
  1. Follow Step 1 and 2 of composition based inference.

Individual-Rule Based Inference

- Each rule in the fuzzy rule base determines an output fuzzy set
- The output of the whole fuzzy inference engine is the combination of the M individual fuzzy sets using intersection or union for combination.
- **The computational procedure of the individual-rule based inference:**
  1. Follow Step 1 and 2 of composition based inference.
  2. For given input fuzzy set $A' \in U$, find output fuzzy set $B' \in V$ for each individual rule $RU^{(l)}$ according to the generalized modus ponens
     $\mu_{B'_l} = \sup_{x \in U} t[\mu_{A'}(x), \mu_{Ru^{(l)}}(x, y)]$ for $l = 1, ..., M$

Individual-Rule Based Inference

- Each rule in the fuzzy rule base determines an output fuzzy set
- The output of the whole fuzzy inference engine is the combination of the M individual fuzzy sets using intersection or union for combination.
- **The computational procedure of the individual-rule based inference:**
  1. Follow Step 1 and 2 of composition based inference.
  2. For given input fuzzy set $A' \in U$, find output fuzzy set $B' \in V$ for each individual rule $RU^{(l)}$ according to the generalized modus ponens
     $\mu_{B'_l} = \sup_{x \in U} t[\mu_{A'}(x), \mu_{Ru^{(l)}}(x, y)]$ for $l = 1, ..., M$
  3. The output of the fuzzy inference engine is the combination of the M fuzzy sets $B'_l, ..., B'_M$ either by union
     $\mu_{B'}(y) = \mu_{B'_1}(y) \dot{+} \ldots \dot{+} \mu_{B'_M}(y)$
     or by intersection
     $\mu_{B'}(y) = \mu_{B'_1}(y) \star \ldots \star \mu_{B'_M}(y)$

- There are a variety of choices in the fuzzy inference engine based on
  - Composition based inference or individual-rule based inference, and within the composition based inference, Mamdani inference or Godel inference
  - Type of implication: Dienes-Rescher implication, Lukasiewicz implication, Zadeh implication, Godel implication, or Mamdani implications
  - Different operations for the t-norms and s-norms
- **Three main criteria to choose these alternatives:**
  - Intuitive appeal: The choice should make sense intuitively. For example, if an expert who believes that the rules are independent of each other, use engine with combined by union.
  - Computational efficiency
  - Special properties: Some choice may result in an inference engine that has special properties.

A Number of Popular Fuzzy Inference Engine

- **Product Inference Engine:** Includes:
  - Individual rule based inference with union combination
  - Mamdani's product implication
  - Algebraic product for t-norms and max for all the s-norms

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

  for given fuzzy set $A' \in U$ az input
- **Minimum Inference Engine:** Includes:
  - Individual-rule based inference with union combination
  - Mamdani's minimum implication
  - min for t-norms and max for s-norms

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

  for given fuzzy set $A' \in U$ az input
- Product and Minimum inference engines are computationally simple; they are intuitively appealing for many practical problems, especially for fuzzy control.

A Number of Popular Fuzzy Inference Engine

- A disadvantage of the product and minimum inference engines: if at some $x \in U$ the $\mu_{A_i^l}(x_i)$ are very small, the obtained $\mu_{B'}(y)$ is very small. (due to local implication)
- **Lukasiewicz Inference Engine:** Includes:
  - Individual-rule based inference with intersection combination
  - Lukasiewicz implication
  - min for t-norms

$\mu_{B'}(y) = \min_{l=1}^{M}[\sup_{x \in U} \min(\mu_{A'}(x), \mu_{Ru^{(l)}}(x, y))]$

$\mu_{Ru^{(l)}}(x, y) = \min(1, 1 - \min_{i=1}^{n}(\mu_{A_i^l}(x_i)) + \mu_{B'}(y)))$

$\therefore \mu_{B'}(y) = \min_{l=1}^{M}[\sup_{x \in U} \min(\mu_{A'}(x), 1 - \min_{i=1}^{n}(\mu_{A_i^l}(x_i)) + \mu_{B'}(y))]$

- **Zadeh Inference Engine:** Includes:
  - Individual rule based inference with intersection combination
  - Zadeh implication
  - min for t-norms.

    $\mu_{B'}(y) = \min_{l=1}^{M}\{\sup_{x\in U}\min[\mu_{A'}(x), \max(\min(\mu_{A_1^l}(x_1), \ldots, \mu_{A_n^l}(x_n), \mu_B^l(y)), 1 - \min_{i=1}^{n}(\mu_{A_i^l}(x_i)))]\}$
- **Dienes-Rescher Inference Engine:** Includes:
  - Individual rule based inference with intersection combination
  - Dienes-Rescher implication
  - min for t-norms.

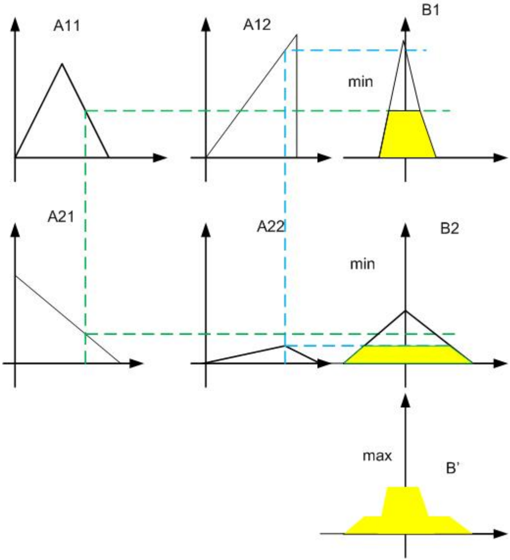
    $\mu_{B'}(y) = \min_{l=1}^{M}\{\sup_{x\in U}\min[\mu_{A'}(x), \max(1 - \min_{i=1}^{n}(\mu_{A_i^l}(x_i)), \mu_B^l(y))]\}$

- **Lemma:** The product inference engine is unchanged if "individual rule based inference with union combination" is replaced by "composition based inference with Mamdani combination"
- **Proof:**
  - composition based inference with Mamdani combination: $\mu_{B'}(y) = sup_{x \in U} t[\mu_{A'}(x), \mu_{Q_{M/G}}(x,y)]$
  - $\mu_{Q_M}(x,y)$ for max as s-norm, product as t-norm: $\mu_{Q_{M/G}}(x,y) = \max_{l=1}^{M}(\mu_{Ru^{(l)}}(x,y))$
  - $\therefore\ \mu_{B'}(y) = sup_{x \in U}[\mu_{A'}(x) \max_{l=1}^{M}(\mu_{Ru^{(l)}}(x,y))]$
  - Use Mamdani product and $\mu_{A_1^l} \times \ldots \times \mu_{A_n^l}(x_1, \ldots, x_n) = \mu_{A_1^l}(x_1) \star \ldots \star \mu_{A_n^l}(x_1)$
  - $\therefore \mu_{B'}(y) = sup_{x \in U} \max_{l=1}^{M}[\mu_{A'}(x) \prod_{l=1}^{n} \mu_{A_i^l}(x_i) \mu_{B^l}(y)]$
  - Since $\max_{l=1}^{M}$ and $sup_{x \in U}$ are interchangeable the above equation is equal to individual rule based inference with union combination

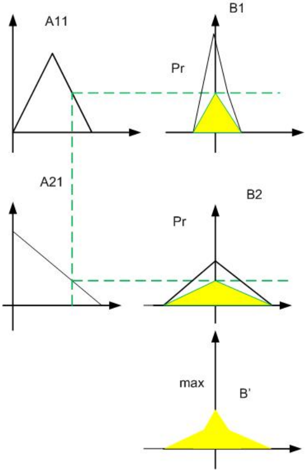
- If the fuzzy set A' is a fuzzy singleton: $\mu_{A'}(x) = \begin{cases} 1 & if \ x = x^* \\ 0 & otherwise \end{cases}$

  where x^* is some point in U
  - The product inference engine is $\mu_{B'}(y) = \max_{i=1}^{M}[\prod_{i=1}^{n} \mu_{A_i^l}(x_i^*)\mu_{B^l}(y)]$
  - The minimum inference engine is $\mu_{B'}(y) = \max_{i=1}^{M}[\min(\mu_{A_1^l}(x_1^*), \ldots, \mu_{A_n^l}(x_n^*), \mu_{B^l}(y))]$
  - The Lukasiewicz inference engine is $\mu_{B'}(y) = \min_{l=1}^{M}[1, 1 - \min_{i=1}^{n}(\mu_{A_i^l}(x_i^*)) + \mu_{B^l}(y))]$
  - The Zadeh inference engine is $\mu_{B'}(y) = \min_{l=1}^{M}\{\max[\min(\mu_{A_i^l}(x_i^*), \ldots, \mu_{A_n^l}(x_n^*), \mu_B^l(y)), 1 - \min_{i=1}^{n}(\mu_{A_i^l}(x_i^*))]\}$
  - The Dienes-Rescher inference engine is $\mu_{B'}(y) = \min_{l=1}^{M}\{\max(1 - \min_{i=1}^{n}(\mu_{A_i^l}(x_i^*)), \mu_B^l(y))\}$

- Graphical Minimum inference Engine



- Graphical Product inference Engine

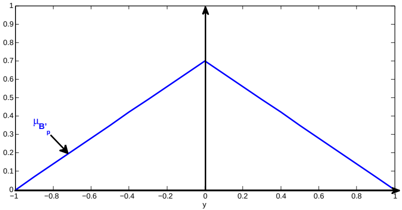


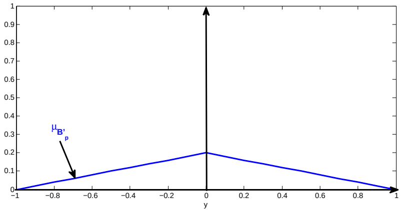
Example

- A fuzzy rule base: "IF x_1 is A_1 and ... and x_n is A_n, THEN y is B
- $\mu_B(y) = \begin{cases} 1-|y| & \textit{if} \quad -1 \le y \le 1 \\ 0 & \textit{otherwise} \end{cases}$
- A' is a fuzzy singleton: $\mu_{A'}(x) = \begin{cases} 1 & \textit{if} \;\; x = x^* \\ 0 & \textit{otherwise} \end{cases}$
- Find $\mu_{B'}(y)$ using $B'_P, B'_M, B'_L, B'_Z, B'_D$
- Let $\min[\mu_{A_1}(x_1^*), \ldots, \mu_{A_n}(x_n^*)] = \mu_{A_p}(x_p^*)$
- $\prod_{i=1}^{n} \mu_{A_i}(x_i^*) = \mu_A(x^*)$

- $\mu_{B'_p}(y) = \mu_A(x^*)\mu_{B'}(y)$

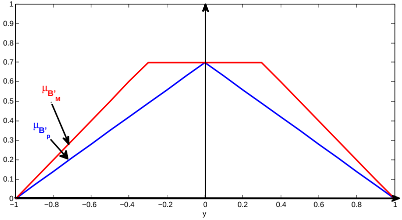
$\mu_{A_p}(x^*) > 0.5,$ $\mu_{A_p}(x^*) \leq 0.5$



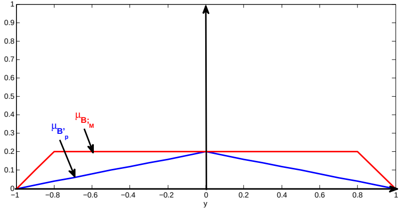


- $\mu_{B'_P}(y) = \mu_A(x^*)\mu_{B'}(y)$
- $\mu_{B'_M}(y) = \min(\mu_{A^*_p}(x^*_p), \mu_B(y))$

$\mu_{A_p}(x^*) > 0.5,$

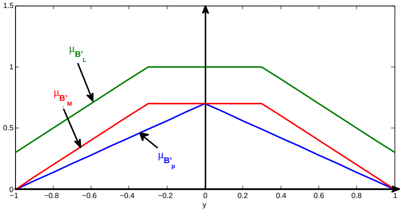


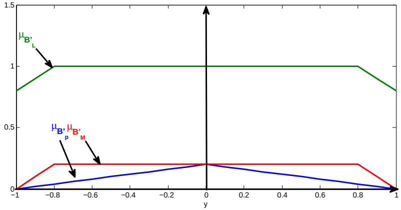
$\mu_{A_p}(x^*) \leq 0.5$



- $\mu_{B'_P}(y) = \mu_A(x^*)\mu_{B'}(y)$
- $\mu_{B'_M}(y) = \min(\mu_{A_p^*}(x_p^*), \mu_B(y))$
- $\mu_{B'L}(y) = \min[1, 1 - \mu_{A_p^*}(x_p^*) + \mu_B(y)]$

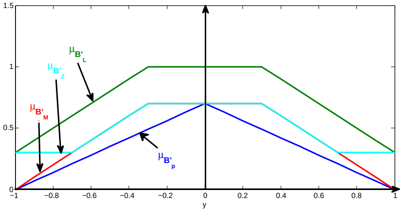
$\mu_{A_p}(x^*) > 0.5,$ $\mu_{A_p}(x^*) \leq 0.5$

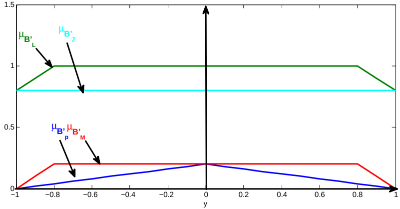




- $\mu_{B'_P}(y) = \mu_A(x^*)\mu_{B'}(y)$
- $\mu_{B'_M}(y) = \min(\mu_{A_p^*}(x_p^*), \mu_B(y))$
- $\mu_{B'L}(y) = \min[1, 1 - \mu_{A_p^*}(x_p^*) + \mu_B(y)]$
- $\mu_{B'_Z}(y) = \max\{\min[\mu_{A_p^*}(x_p^*), \mu_B(y), 1 - \mu_{A_p^*}(x_p^*)]\}$

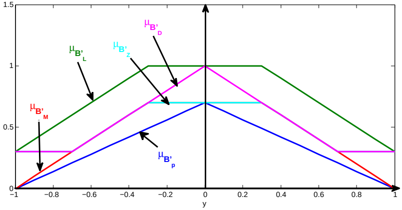
$\mu_{A_p}(x^*) > 0.5,$ $\mu_{A_p}(x^*) \leq 0.5$

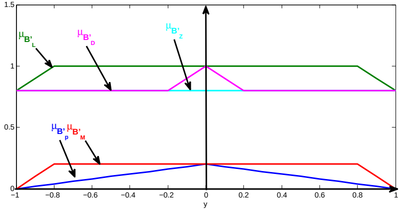




- $\mu_{B'_P}(y) = \mu_A(x^*)\mu_{B'}(y)$
- $\mu_{B'_M}(y) = \min(\mu_{A_p^*}(x_p^*), \mu_B(y))$
- $\mu_{B'L}(y) = \min[1, 1 - \mu_{A_p^*}(x_p^*) + \mu_B(y)]$
- $\mu_{B'_Z}(y) = \max\{\min[\mu_{A_p^*}(x_p^*), \mu_B(y), 1 - \mu_{A_p^*}(x_p^*)]\}$
- $\mu_{B'_D}(y) = \max(1 - \mu_{A_p^*}(x_p^*), \mu_B(y))$

$\mu_{A_p}(x^*) > 0.5,$ $\mu_{A_p}(x^*) \leq 0.5$





Example Cont'd

- $\mu_{A_p}(x^*) < 0.5 \rightsquigarrow \mu_{B'P}, \mu_{B'_M}$ is very small; $\mu_{B'L}(y)$,$\mu_{B'_Z}(y)$,$\mu_{B'_D}(y)$ is very large
- Product and Minimum inf. eng. are similar; Zadeh, Dienes, Lukasiewicz inf. eng. are similar
- Lukasiewicz inf. eng.: Largest output; Product inf. eng.: Smallest output