Fuzzy Inference Engines

Composition and Individual-Rule Based Composition, Non-Linear Mappings

Olaf Wolkenhauer

Control Systems Centre

UMIST

o.wolkenhauer@umist.ac.uk

www.csc.umist.ac.uk/people/wolkenhauer.htm

Contents

1. Approximate Reasoning

Let *proposition* take the form

$$\text{“}\mathbf{x} \text{ is } A\text{”}$$

with *fuzzy variable* $\mathbf{x}$ taking values in X and A modelled by a fuzzy set defined on the *universe of discourse* X by *membership function* $\mu \colon X \to [0, 1]$.

A *compound statement*,

$$\text{“}\mathbf{x} \text{ is } A \text{ AND } \mathbf{y} \text{ is } B\text{”}$$

is a *fuzzy set* $A \cap B$ in $X \times Y$ with

$$\mu_{A\cap B}(x, y) = T\big(\mu_A(x), \mu_B(y)\big)$$

For the sake of simplicity we consider a single rule of type

$$\text{IF } \mathbf{x} \text{ is } A, \ \text{THEN } \mathbf{y} \text{ is } B$$

which can be regarded as a *fuzzy relation*

$$\begin{aligned} R: \ X \times Y &\quad \longrightarrow \quad [0,1] \\ (x,y) &\quad \longmapsto \quad R(x,y) \end{aligned}$$

where $R(x,y)$ is interpreted as the strength of relation between x and y. Viewed as a fuzzy set, with

$$\mu_R(x,y) \doteq R(x,y)$$

denoting the degree of membership in the (fuzzy) subset R, $\mu_R(x,y)$ is computed by means of a *fuzzy implication*.

1.1. Modus Ponens

The (generalised) *modus ponens* provides a mechanism for inference :

Implication:	IF **x** is A, THEN **y** is B.
Premise:	**x** is A'.
Conclusion:	**y** is B'.

In terms of fuzzy relations the output fuzzy set B' is obtained as the relational sup-t composition, $B' = A' \circ R$.

The computation of the conclusion $\mu_{B'}(y)$ is realised on the basis of what is called the *compositional rule of inference*.

1.2. Compositional Rule of Inference

Given $\mu_{A'}(x)$, and $\mu_R(x,y)$, $\mu_{B'}(y)$ is found by generalising the 'crisp' rule (from functions to relations..)

$$\text{IF } \mathbf{x} = a \text{ AND } \mathbf{y} = f(\mathbf{x}), \text{ THEN } y = f(a)$$

The inference can be described in three steps :

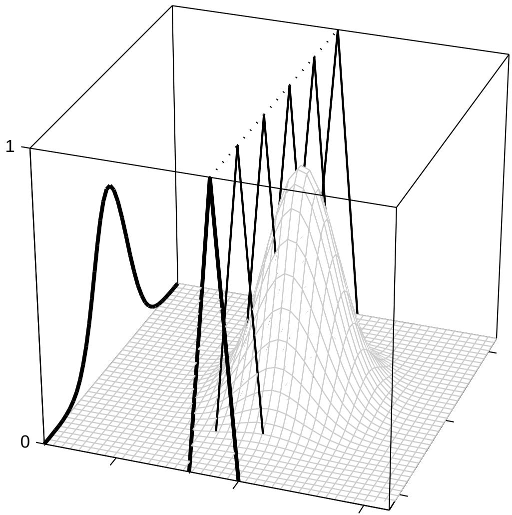
1. Extension of A' to $X \times Y$, i.e $\mu_{A'_{\text{ext}}}(x,y) = \mu_{A'}(x)$.

2. Intersection of A'_{ext} with R, i.e

$$\mu_{A'_{\text{ext}} \cap R}(x,y) = T\big(\mu_{A'_{\text{ext}}}(x,y), \mu_R(x,y)\big) \qquad \forall\ (x,y)$$

3. Projection of $A'_{\text{ext}} \cap R$ on Y, i.e

$$\begin{aligned} \mu_{B'}(y) &= \sup_{x \in X} \mu_{A'_{\text{ext}} \cap R}(x,y) \\ &= \sup_{x \in X} T\big(\mu_{A'_{\text{ext}}}(x,y),\ \mu_R(x,y)\big) \end{aligned} \tag{1}$$


1
0

1.3. Fuzzy Implication Operators

▷ *Dienes-Rescher implication:*

$$\mu_R(x,y) = \max\bigl(1 - \mu_A(x), \mu_B(y)\bigr)\ . \tag{2}$$

▷ *Zadeh implication:*

$$\mu_R(x,y) = \max\left(\min\bigl(\mu_A(x), \mu_B(y)\bigr), 1 - \mu_A(x)\right)\ . \tag{3}$$

▷ *Lukasiewicz implication:*

$$\mu_R(x,y) = \min\bigl(1, 1 - \mu_A(x) + \mu_B(y)\bigr) \tag{4}$$

▷ *Gödel implication:*

$$\mu_R(x,y) = \begin{cases} 1 & \text{if } \mu_A(x) \le \mu_B(y), \\ \mu_B(y) & \text{otherwise.} \end{cases} \tag{5}$$

▷ *Minimum implication:*

$$\mu_R(x, y) = \min\big(\mu_A(x), \mu_B(y)\big) \tag{6}$$

▷ *Product implication:*

$$\mu_R(x, y) = \mu_A(x) \cdot \mu_B(y) \tag{7}$$

2. Composition-Based Inference

The way rules are combined, depends on the interpretation for what a set of rules should mean. If rules are viewed as *independent conditional statements*, then a reasonable mechanism for aggregating n_R individual rules R_i (fuzzy relations) is the union :

$$\begin{aligned} R &\doteq \bigcup_{i=1}^{n_R} R_i \\ &= S\big(\mu_{R^1}(\mathbf{x}, y), \ldots, \mu_{R^n_R}(\mathbf{x}, y)\big) . \end{aligned} \tag{8}$$

On the other hand, if rules are seen as *strongly coupled conditional statements*, their combination should employ an intersection operator :

$$\begin{aligned} R &\doteq \bigcap_{i=1}^{n_R} R_i \\ &= T\big(\mu_{R^1}(\mathbf{x}, y), \ldots, \mu_{R^n_R}(\mathbf{x}, y)\big) . \end{aligned} \tag{9}$$

2.1. The Algorithm

For the n_R fuzzy if-then rules of the *conjunctive linguistic model*

R_i : IF x_1 is A_{i1} AND x_2 is $A_{i2} \ldots$ AND x_r is A_{ir}, THEN y is B_i

Step 1: Determine the fuzzy set membership functions

$$\mu_{A_{i1}\times\cdots\times A_{ir}}(x_1, \ldots, x_r) \doteq T\big(\mu_{A_{i1}}(x_1), \ldots, \mu_{A_{ir}}(x_r)\big) \ . \tag{10}$$

Step 2: $\mu_{R_i}(\mathbf{x}, y)$, $i = 1, \ldots, n_R$, is calculated according to any fuzzy implication (2)-(7).

Step 3: $\mu_R(\mathbf{x}, y)$ is determined according to (8) or (9).

Step 4: Finally, for an input A', the output B' is

$$\mu_{B'}(y) = \sup_{\mathbf{x}\in X} T\big(\mu_{A'}(\mathbf{x}), \mu_R(\mathbf{x}, y)\big) \ . \tag{11}$$

3. Individual-Rule Based Inference

Given input fuzzy set A' in X, the fuzzy set B_i' in Y is given by the generalised modus ponens (1), i.e

$$\mu_{B_i'}(y) = \sup_{\mathbf{x}\in X} T\big(\mu_{A'}(\mathbf{x}), \mu_{R_i}(\mathbf{x}, y)\big) \qquad i = 1, \ldots, n_R \tag{12}$$

The output of the fuzzy inference engine from the union

$$\mu_{B'}(y) = S\big(\mu_{B_1'}(y), \ldots, \mu_{B_r'}(y)\big) \tag{13}$$

or intersection

$$\mu_{B'}(y) = T\big(\mu_{B_1'}(y), \ldots, \mu_{B_r'}(y)\big) \tag{14}$$

of the individual output fuzzy sets $B_1', \ldots, B_r'$.

3.1. The Algorithm

Step 1: Determine the fuzzy set membership functions

$$\mu_{A_{i1}\times\cdots\times A_{ir}}(x_1,\ldots,x_r) \doteq T\big(\mu_{A_{i1}}(x_1),\ldots,\mu_{A_{ir}}(x_r)\big)\ . \tag{10}$$

Step 2: Equation (10) is viewed as the fuzzy set μ_A in the fuzzy implications (2)-(7) and $\mu_{R_i}(\mathbf{x}, y)$, $i = 1,\ldots,n_R$, is calculated according to any of the implications.

Step 3: For a given input fuzzy set A' in X, determine the output fuzzy set B_i' in Y for each rule R_i according to the generalised modus ponens (1), i.e

$$\mu_{B_i'}(y) = \sup_{\mathbf{x}\in X} T\big(\mu_{A'}(\mathbf{x}), \mu_{R_i}(\mathbf{x}, y)\big) \tag{12}$$

for $i = 1,\ldots,n_R$.

Step 4: The output of the fuzzy inference engine is obtained from either the union

$$\mu_{B'}(y) = S\big(\mu_{B'_1}(y), \ldots , \mu_{B'_r}(y)\big) \tag{13}$$

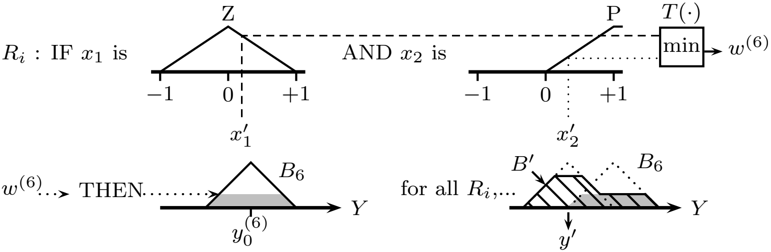
or intersection

$$\mu_{B'}(y) = T\big(\mu_{B'_1}(y), \ldots , \mu_{B'_r}(y)\big) \tag{14}$$

of the individual output fuzzy sets $B'_1, \ldots , B'_r$.

3.2. Example: Individual-Rule Based Inference

- ▷ Minimum inference.
- ▷ Singleton input.
- ▷ Union intersection.



3.3. Minimum Inference Engine

- ▷ Individual-rule based inference.
- ▷ **Union combination** (13).
- ▷ Fuzzy implication (6).
- ▷ Max. for all the t-conorm operators.
- ▷ Using (6) and the min for for all t-norm operators.

We obtain from (12) and (13) :

$$\mu_{B'}(y) = \max_{i=1,\dots,n_R} \left\{ \sup_{\mathbf{x}\in X} \min\Big(\mu_{A'}(\mathbf{x}), \mu_{A_{i1}}(x_1), \dots, \mu_{A_{ir}}(x_r), \mu_{B_i}(y)\Big) \right\} \tag{15}$$

3.4. Product Inference Engine

- Individual-rule based inference.
- **Union combination** (13).
- Fuzzy implication (7).
- Max. for all the t-conorm operators.
- Using (7) and the algebraic product for all t-norm operators

We obtain from (12) and (13) :

$$\mu_{B'}(y) = \max_{i=1,\dots,n_R} \left\{ \sup_{\mathbf{x}\in X} \left(\mu_{A'}(\mathbf{x}) \cdot \prod_{k=1}^{r} \mu_{A_{ik}}(x_k) \cdot \mu_{B_i}(y) \right) \right\} \tag{16}$$

3.5. Singleton Inputs

Let the fuzzy set A' is a *singleton*, that is, if we consider 'crisp' input data,

$$\mu_{A'}(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} = \mathbf{x}' \\ 0 & \text{otherwise,} \end{cases} \tag{17}$$

where $\mathbf{x}'$ is some point in X. Substituting (17) in

- (15) (*Minimum Inference Engine*) and
- (16) (*Product Inference Engines*),

we find that the maximum

$$\sup_{\mathbf{x} \in X}$$

is achieved at

$$\mathbf{x} = \mathbf{x}' \ .$$

Hence, the *Minimum Inference Engine* (15) reduces to,

$$\mu_{B'}(y) = \max_{i=1,\dots,n_R} \left\{ \min\left(\mu_{A_{i1}}(x'_1), \dots, \mu_{A_{ir}}(x'_r), \mu_{B_i}(y) \right) \right\} \tag{18}$$

and the *Product Inference Engine* (16) reduces to

$$\mu_{B'}(y) = \max_{i=1,\dots,n_R} \left\{ \prod_{k=1}^{r} \mu_{A_{ik}}(x'_k) \cdot \mu_{B_i}(y) \right\} \tag{19}$$

✘ A disadvantage of the minimum and product inference engines is that

- ▷ for some $\mathbf{x} \in X$, $\mu_{A_ik}(x_k)$ is very small,
- ▷ then $\mu_{B'}(y)$ obtained from (15) and (16) will be very small.

3.6. Dienes-Rescher Inference Engine

- Using individual-rule based inference.
- **Intersection combination** (14).
- Implication (2).
- Using the min t-norm in (14) and (10).

We obtain from (12) :

$$\mu_{B'}(y) = \min_{i=1,\ldots,n_R} \Big\{ \sup_{\mathbf{x}\in X} \min\big[\mu_{A'}(\mathbf{x}), \max\big(1 - \min_{k=1,\ldots,r}\big(\mu_{A_{ik}}(x_k)\big), \mu_{B_i}(y)\big)\big] \Big\} \quad (20)$$

3.7. Zadeh Inference Engine

- ▷ Using individual-rule based inference.
- ▷ **Intersection combination** (14).
- ▷ Implication (3).
- ▷ Using t-norm min in (14) and (10).

We obtain from (12) :

$$\mu_{B'}(y) = \min_{i=1,\dots,n_R} \Big\{ \sup_{\mathbf{x}\in X} \min\big[\mu_{A'}(\mathbf{x}), \max\big(\min\big[\mu_{A_{i1}}(x_1), \dots, \mu_{A_{ir}}(x_r), \mu_{B_i}(y)\big], 1 - \min_{k=1,\dots,r}\big(\mu_{A_{ik}}(x_k)\big)\big)\big] \Big\} \quad (21)$$

3.8. Lukasiewicz Inference Engine

- ▷ Using individual-rule based inference.
- ▷ **Intersection combination** (14).
- ▷ Implication (4).
- ▷ Using the min t-norm in (14) and (10).

We obtain from (12) :

$$\begin{aligned}
\mu_{B'}(y) &= \min_{i=1,\dots,n_R}\Big\{\sup_{\mathbf{x}\in X}\min\Big[\mu_{A'}(\mathbf{x}),\\
&\qquad \min\Big(1, 1-\min_{k=1,\dots,r}\big(\mu_{A_{ik}}(x_k)\big)+\mu_{B_i}(y)\Big)\Big]\Big\}\\
&= \min_{i=1,\dots,n_R}\Big\{\sup_{\mathbf{x}\in X}\min\Big[\mu_{A'}(\mathbf{x}),\\
&\qquad 1-\min_{k=1,\dots,r}\big(\mu_{A_{ik}}(x_k)\big)+\mu_{B_i}(y)\Big]\Big\}
\end{aligned} \tag{22}$$

3.9. Singleton Input

If the fuzzy set A' is a singleton, substituting (17) into the equations of the inference engines (20)-(22), the $\sup_{\mathbf{x}\in X}$ is obtained at $\mathbf{x} = \mathbf{x}'$, leading to the following singleton input inference engines :

▷ From the Dienes-Rescher Inference Engine (20) :

$$\mu_{B'}(y) = \min_{i=1,\ldots,n_R}\left\{\max\left[1 - \min_{k=1,\ldots,r}\left(\mu_{A_{ik}}(x'_k)\right), \mu_{B_i}(y)\right]\right\}$$

▷ From the Zadeh Inference Engine (21) :

$$\mu_{B'}(y) = \min_{i=1,\ldots,n_R}\left\{\max\left[\min\left(\mu_{A_{i1}}(x'_1), \ldots, \mu_{A_{ir}}(x'_r), \mu_{B_i}(y)\right), 1 - \min_{k=1,\ldots,r}\left(\mu_{A_{ik}}(x'_i)\right)\right]\right\}$$

▷ From the Lukasiewicz Inference Engine (22) :

$$\mu_{B'}(y) = \min_{i=1,\ldots,n_R}\left\{1, 1 - \min_{k=1,\ldots,r}\left(\mu_{A_{ik}}(x'_i)\right) + \mu_{B_i}(y)\right\}$$

4. Fuzzy Systems as Nonlinear Mappings

Linguistic model:

R_i : IF x_1 is A_{i1} AND x_2 is $A_{i2} \ldots$ AND x_r is A_{ir}, THEN y is B_i

Let the input data be crisp, i.e substituting (17) into the product inference engine (16), we have

$$\mu_{B'}(y) = \max_{i=1,\ldots,n_R} \left\{ \prod_{k=1}^{r} \mu_{A_{ik}}(x'_k) \cdot \mu_{B_i}(y) \right\} . \qquad (23)$$

4.1. Defuzzification

▷ A *defuzzifier* is a mapping from the fuzzy set B' in Y to a point y' in Y.

▷ To obtain a single-valued numerical output from the inference engines, one has to somehow capture the information given in $\mu_{B'}(y)$ by a single number.

▷ The *centre of gravity defuzzifier* determines y' as the centre of the area under the membership function $\mu_{B'}(y)$:

$$y' \doteq \frac{\int_Y \mu_{B'}(y) \cdot y \, \mathrm{d}y}{\int_Y \mu_{B'}(y) \, \mathrm{d}y} \tag{24}$$

✘ The main problem with this defuzzifier is the calculation of the integral for irregular shapes of $\mu_{B'}(y)$.

▷ Since the fuzzy set B' is the union or intersection of n_R fuzzy sets, the weighted average of the centres of the n_R fuzzy sets provides a reasonable approximation of (24).

▷ Let $y_0^{(i)}$ be the centre of the i^{th} fuzzy set and $w^{(i)}$ be its height, the *center average defuzzifier* calculates y' as

$$y' \doteq \frac{\sum_{i=1}^{n_R} y_0^{(i)} \cdot w^{(i)}}{\sum_{i=1}^{n_R} w^{(i)}} . \tag{25}$$

4.2. Product Inference Engine with Singleton Input Data

▷ Use the centre average defuzzifier (25).

▷ The centre of the fuzzy set $\mu_{A_{ik}}(x'_k) \cdot \mu_{B_i}(y)$ determines the centre of B_i, denoted $y_0^{(i)}$ in (25).

▷ The height of the i^{th} fuzzy set in (23) is

$$\prod_{k=1}^{r} \mu_{A_{ik}}(x'_k) \cdot \mu_{B_i}(y_0^{(i)}) = \prod_{k=1}^{r} \mu_{A_{ik}}(x'_k)$$

and equals $w^{(i)}$ in (25).

✘ This reduces the fuzzy system to

$$y' = \frac{\sum_{i=1}^{n_R} y_0^{(i)} \cdot \prod_{k=1}^{r} \mu_{A_{ik}}(x'_k)}{\sum_{i=1}^{n_R} \prod_{k=1}^{r} \mu_{A_{ik}}(x'_k)}$$

... or in general, we find that the fuzzy system is a nonlinear mapping

$$\begin{aligned} f : X &\rightarrow Y \\ \mathbf{x} &\mapsto f(\mathbf{x}) \end{aligned}$$

where $\mathbf{x} \in X \subset \mathbb{R}^r$ maps to $f(\mathbf{x}) \in Y \subset \mathbb{R}$, a weighted average of the consequent fuzzy sets :

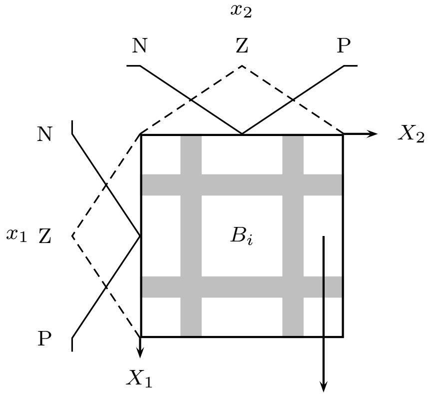
$$f(\mathbf{x}) = \frac{\sum_{i=1}^{n_R} y_0^{(i)} \cdot \prod_{k=1}^{r} \mu_{A_{ik}}(x_k)}{\sum_{i=1}^{n_R} \prod_{k=1}^{r} \mu_{A_{ik}}(x_k)} . \tag{26}$$

Similar to (26), we obtain for a fuzzy system, with

- minimum inference engine (15),
- singleton input (17) and
- centre average defuzzifier (25),

$$f(\mathbf{x}) = \frac{\sum_{i=1}^{n_R} y_0^{(i)} \cdot \min_{k=1}^{r} \mu_{A_{ik}}(x_k)}{\sum_{i=1}^{n_R} \min_{k=1}^{r} \mu_{A_{ik}}(x_k)} \; . \tag{27}$$

5. Comparison of Inference Engines



R_i : IF x_1 is A_{i1} AND x_2 is A_{i2}, THEN y is B_i

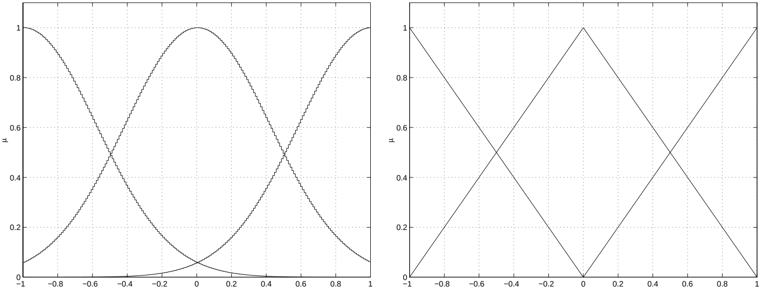


Figure 1: *Gaussian and trapecoidal input fuzzy sets.*

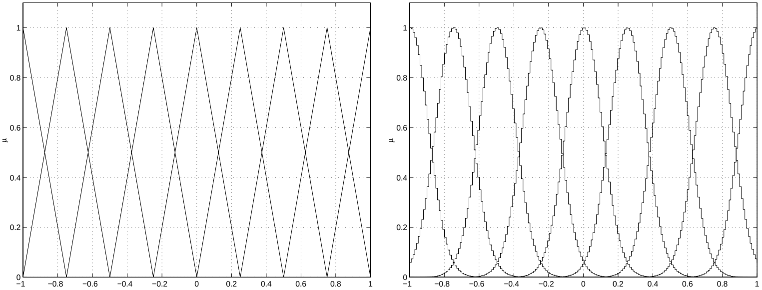


Figure 2: *Gaussian and trapecoidal outputs sets* B_i.

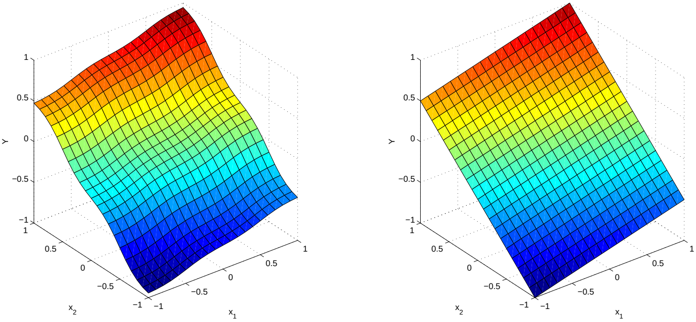


Figure 3: *Product inference with Gaussian and trapecoidal sets.*

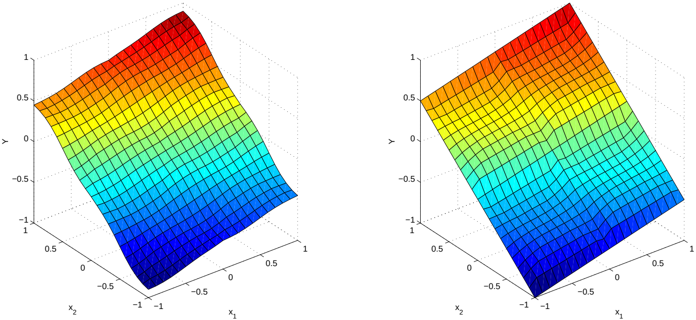


Figure 4: *Minimimum inference with Gaussian and trapecoidal sets.*

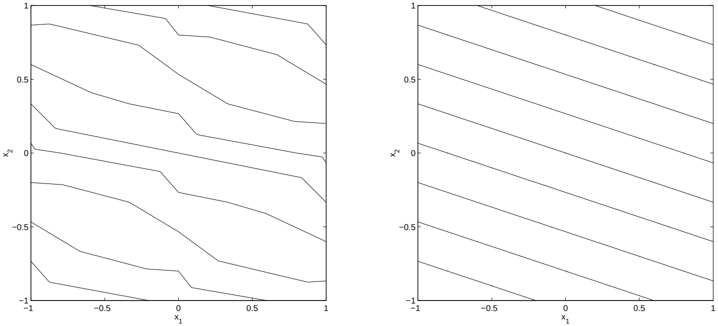


Figure 5: *Contourplots for minimum inference (left) vs product inference (right).*

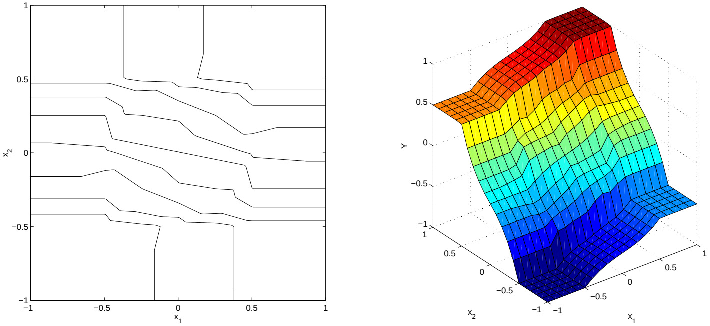


Figure 6: *Minimum inference with input fuzzy partition that does not have fully overlapping fuzzy sets.*

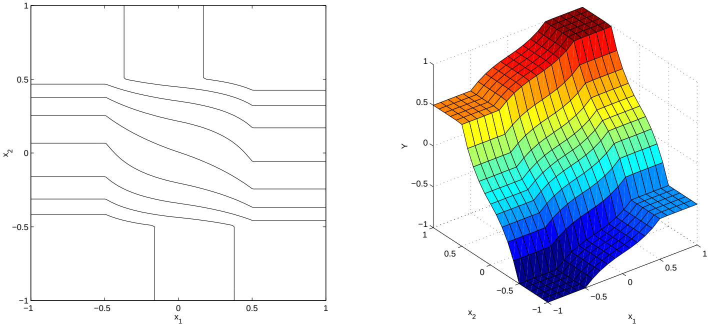


Figure 7: *Product inference with non-overlapping input fuzzy partition.*

References

[1] Kruse, R., Gebhardt, J. and Klawonn, F. : *Foundations of Fuzzy Systems.* Wiley, 1994.

[2] Wang, L.-X. : *A Course in Fuzzy Systems and Control.* Prentice Hall, 1997.