Intelligent Systems Engineering
FS-6 Fuzzy Rule-based Computing

Petr Musilek

University of Alberta

Outline

These notes are based on [Pedrycz, 2007] and [Keller et al. 2016]

Rule-based Systems

Rules provide a formal way of representing knowledge in form of directives and strategies. Rule-based systems (RBS) are appropriate when domain knowledge is available as empirical results or experience, e.g.

- IF the animal has stripes, THEN it is a zebra
- IF it is hot, THEN turn on the fan

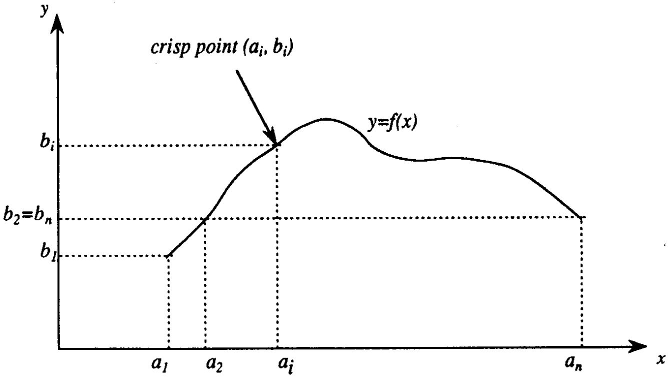
RBS also form basis for fuzzy rule-based systems and fuzzy control

- IF a region has color of skin, it is round, and located in the upper part of the filed of view, THEN confidence of face is high
- IF an object in the filed of view is moving, THEN use a short exposition time

As shown in previous section, fuzzy RBS are associated with fuzzy composition

From Function to Rules

To understand how an IF-THEN statement (and a set of IF-THEN-ELSE statements) can be represented as a relation, consider a crisp function.



Analytical representation

To say "$y = f(x)$" is an analytical form representing the function. Another way of representing this function would be to list all (or a sufficient number of) pairs that describe the function:

$$\begin{gathered}(a_1, b_1)\\(a_2, b_2)\\\dots\\(a_i, b_i)\\\dots\\(a_n, b_n)\end{gathered}$$

Linguistic representation

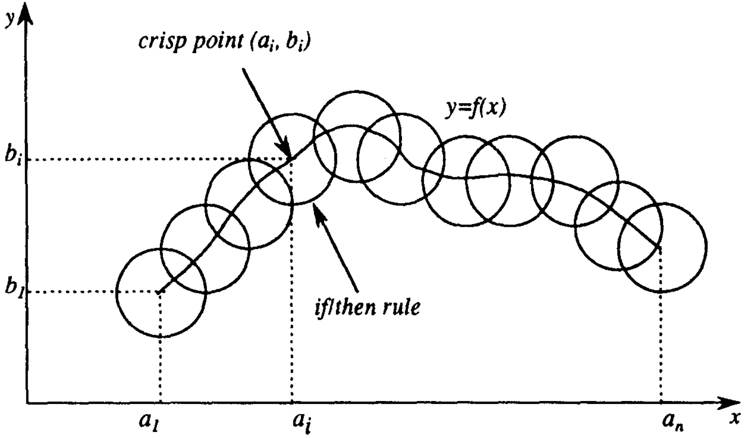
This form can be linguistically represented as

$$
\begin{array}{ll}
\text{IF } x \text{ is } a_1 & \text{THEN } y \text{ is } b_1 \\
\text{IF } x \text{ is } a_2 & \text{THEN } y \text{ is } b_2 \\
\dots & \\
\text{IF } x \text{ is } a_i & \text{THEN } y \text{ is } b_i \\
\dots & \\
\text{IF } x \text{ is } a_n & \text{THEN } y \text{ is } b_n
\end{array}
$$

which becomes more "crisply accurate" as n increases. But this is not always necessary (or even desired).

Fuzzy extension

The above concept can be extended to fuzzy input/output (using extension principle)



Fuzzy rules – linguistic representation

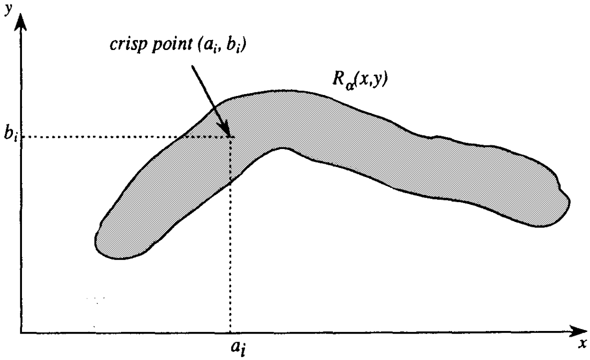
This new representation can be approximated using fuzzy rules as follows

$$
\begin{array}{ll}
\text{IF } x \text{ is } A_1 & \text{THEN } y \text{ is } B_1 \\
\text{IF } x \text{ is } A_2 & \text{THEN } y \text{ is } B_2 \\
\dots & \\
\text{IF } x \text{ is } A_i & \text{THEN } y \text{ is } B_i \\
\dots & \\
\text{IF } x \text{ is } A_n & \text{THEN } y \text{ is } B_n
\end{array}
$$

Where A_i is a fuzzy quantity defined on X, and B_i is a fuzzy quantity defined on Y. The analytical form of these rules is a fuzzy relation $R_i(x, y) \rightarrow$ (implication relation, covered shortly), and each rule used to approximate the function has its own implication relation.

Fuzzy algorithm

Merging the rules joined together by ELSE depends upon the manner by which the implication relations are created. Regardless the analytical form, this relation is called *fuzzy algorithm* (rather just a single rule).



Rules

Rule-based systems are based on IF-THEN statements in form

IF	*condition*	THEN	*action*, or alternatively
IF	*premise*	THEN	*conclusion*
IF	*antecedent*	THEN	*consequent*

Fuzzy Rules

Consider a rule in the form

$$\text{IF } x \text{ is } A \text{ AND } y \text{ is } B \text{ THEN } z \text{ is } C$$

with x, y and z being fuzzy quantities; e.g.

IF	*motor temperature* is *high*
AND	*motor speed* is *high*
THEN	*motor current* is *low*

Inference, Modus ponens

Rules are combined with facts (or collections of facts) to make inference. This leads to the sole rule of inference in propositional calculus, called *modus ponens* (from Latin: *mode that affirms*). Modus ponens can be formally expressed as.

$$\begin{array}{ccc} p & & \\ p & \rightarrow & q \\ \hline & & q \end{array}$$

Informally, this expression states that given an implication (rule, $p \rightarrow q$) and the presence of its premise p, the conclusion q can be taken as true.

Generalized modus ponens

In conventional logic, with modus ponens, the proposition x is p has to be observed to consider the proposition y is b . In other words, there has to be an exact match between the premise p and the antecedent of the rule $p \rightarrow q$.
In fuzzy logic, a proposition x is A', close to the premise x is A can be observed to provide a conclusion y is B', close to the conclusion y is B (A and B are fuzzy stes). This leads to *generalized modus ponens*.

Generalized modus ponens

$$\begin{array}{ccc} A' & & \\ A & \rightarrow & B \\ \hline & & B' \end{array}$$

First, A' is matched with A. To find B', the implication relation $R(x,y)$ is composed with A' (this process is sometimes called *forward chaining* or *data-driven inference*)

$$B' = R(x,y)A'$$

Generalized modus tolens

GMT is essentially the reverse of GMP

$$\begin{array}{ccc} & & B^{'} \\ A & \rightarrow & B \\ \hline A^{'} & & \end{array}$$

First, $B^{'}$ is matched to B. To find $A^{'}$, the implication relation $R(y,x)$ is composed with $B^{'}$

$$A^{'} = R(y,x)B^{'}$$

[Note that x and y are now swapped; this is analogous to finding and using $f^{-1}(y)$, the inverse of $f(x)$.]

Linguistic variables and values

Linguistic variables are described by fuzzy sets:

- Primary values of a fuzzy variable typically include terms like "small", "large", "high", etc.
- These can be further modified with *linguistic hedges* as "very", "more or less", "not", e.g.

$$\text{not}A(x) = 1 - A(x)$$
$$\text{very}A(x) = A^2(x)$$
$$\text{more or less}A(x) = \sqrt{A(x)}$$

Individual variables can also be connected with AND and OR, using appropriate triangular norms (**t** norms for AND, **s** norms for OR).

Accumulation and usage of knowledge

In fuzzy RBS, accumulation and usage of knowledge are facilitated by constructing relations and performing composition, respectively.
Accumulation of knowledge: It is evident that rules represent functional dependencies and relations between premises and conclusions. As such, they can be represented using Cartesian product. For example two fuzzy sets

$$A : \mathbf{X} \to [0,1] \text{ and } B : \mathbf{Y} \to [0,1]$$

can be combined using Cartesian product to form a relation $R : \mathbf{X} \times \mathbf{Y} \to [0,1]$. This can be modelled by a **t**-norm operation, such as min.

Multiple rules/relations

Because the relation between the sets is generally not uniform, multiple rules/relations are build for multiple segments, k, of the universe of discourse

$$R_k = A_k \times B_k$$

To describe the overall relation, the individual rules R_k are aggregated

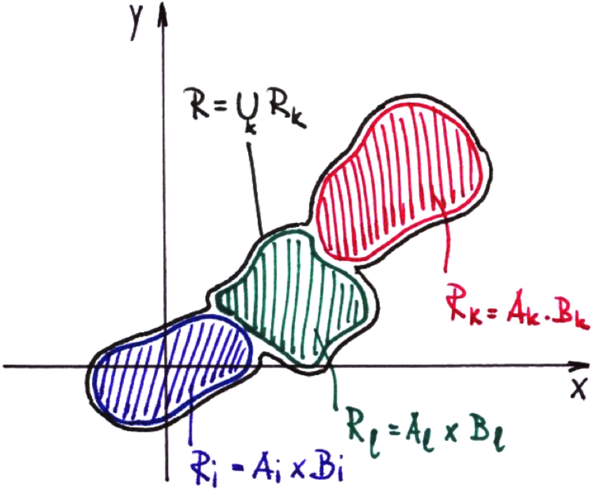
$$R = \cup_{k=1}^{N} R_k$$

This process corresponds to accumulation of knowledge and can be described (assuming min and max operation to model fuzzy intersection and union, respectively) as follows

$$R = \max_{k=1,2,\dots,N} R_k(x, y) = \max_{k=1,2,\dots,N} \min[A_k(x), B_k(y)]$$

Fuzzy patches

Graphically, this situation is depicted the following picture.



Accumulation and usage of knowledge

Usage of knowledge: Knowledge accumulated the way indicated above and codified using fuzzy rules can be used to make conclusions. Having described relation *R* between fuzzy sets *A* in $\mathbf{X}$ and *B* in $\mathbf{Y}$, the goal is to infer value of *B* given a particular value of *A*. Because the inference in fuzzy systems is driven by generalized modus ponens, exact match between *A* and one of the antecedentsA_k is not necessary. Using sup–$\mathbf{t}$ composition, fuzzy set *B* can determined as

$$B^{'} = A^{'} \circ R$$

Usage of knowledge (cont.)

Therefore, the resulting membership function is described as

$$B(y) = \sup_x [A(x) \,\mathbf{t}\, R(x, y)]$$

and because the relation $R(x, y)$ is a union of individual relations R_k, we can expand the above expression to (assuming min intersection and max union)

$$B(y) = \sup_x \min[A(x), R(x, y)] =$$

$$= \sup_x \left\{ \min \left[A(x), \max_k \left(\min \left(A_k(x), B_k(y) \right) \right) \right] \right\}$$

Rearranging this expression, we obtain

$$B(y) = \max_k \left\{ \min \left[\sup_x \left(\min \left(A(x), A_k(x) \right) \right), B_k(y) \right] \right\}$$

Usage of knowledge (cont.)

Term $\sup_x (\min (A(x), A_k(x)))$ is expressing the level of overlap between the antecedent A_k of rule R_k, and the actual value of fuzzy set A, or in other words the possibility of A with respect to A_k

$$\text{Poss}(A, A_k) = \sup_x (\min (A(x), A_k(x)))$$

By substituting $\lambda_k = \text{Poss}(A, A_k)$ we finally obtain the following expression to calculate fuzzy set B for given A

$$B(y) = \max_k \{\min [\lambda_k, B_k(y)]\}$$

Based on max-min composition, this expression is called *compositional rule of inference* (CRI, cf. the brief note on CRI in the previous section).

Implication operators

The results presented in the previous section represent only one specific case of fuzzy implication relation, based on max and min operations. There are, however, numerous other possibilities how implication can be modelled.

There are two general ways to define fuzzy implications:

(i) $A \wedge B$ which corresponds to matching operation; it provides a stronger (local) implication meaning A is coupled with B; examples are Larsen and Mamdani implications (see bellow);

(ii) $\overline{A} \vee B$ which corresponds (NOT A) OR B implication found in binary logic; it provides a weaker (global) implication, meaning "A entails B"; examples include Lukasziewicz and Boolean implications;

(iii) $(A \wedge B) \vee \overline{A}$ which is equivalent to (ii) and has similar properties; an example is Zadeh implication

Implication operators

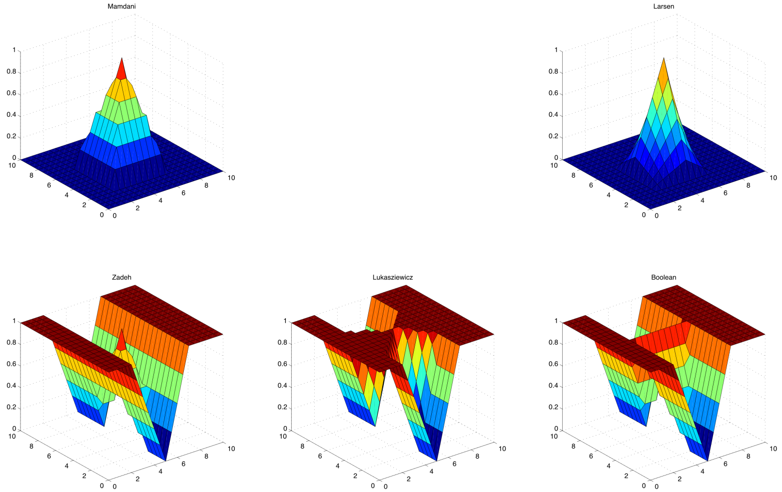
Rule IF x is A THEN y is B can be described in general analytical form of relation

$$R_{A\to B}(x,y) = \Phi[A(x), B(y)],$$

where Φ is the implication operator. A list of commonly used implication operators follows

Name	Implication operator $\Phi[A(x), B(y)]$
Mamdani (min)	$\min[A(x), B(y)]$
Larsen (product)	$A(x) \cdot B(y)$
Zadeh (max-min)	$\max\{\min[A(x), B(y)], 1 - A(x)\}$
Lukasziewicz (arithmetic)	$\min[1 - A(x) + B(y), 1]$
Boolean (Dienes-Rescher)	$\max[1 - A(x), B(y)]$

Fuzzy Implication Operators



Multivariate Implication

These elementary fuzzy implications can be extended to multivariate cases, in order to allow decision making or control with more than one independent variable. In such case, we consider a rule in the form

IF x_1 is A_1 AND x_2 is A_2 AND $\ldots$ AND x_m is A_m THEN y is B

The connective AND can be modeled either as min or product operations leading to implication operators in one of the following forms

$$\Phi\{\min[A_1(x_1), A_2(x_2), \ldots, A_m(x_m)], B(y)\}, \text{ or}$$

$$\Phi\{A_1(x_1)A_2(x_2)\ldots A_m(x_m), B(y)\}$$

.

Fuzzy Algorithm

A fuzzy algorithm is a procedure of collecting fuzzy IF-THEN rules. The rules are defined over the same universes (entering the Cartesian product of corresponding fuzzy sets), and are connected by connectives ELSE:

$$
\begin{array}{ll}
\text{IF } x \text{ is } A_1 & \text{THEN } y \text{ is } B_1 \text{ ELSE} \\
\text{IF } x \text{ is } A_2 & \text{THEN } y \text{ is } B_2 \text{ ELSE} \\
\ldots & \\
\text{IF } x \text{ is } A_i & \text{THEN } y \text{ is } B_i \text{ ELSE} \\
\ldots & \\
\text{IF } x \text{ is } A_n & \text{THEN } y \text{ is } B_n
\end{array}
$$

Each rule is represented by a relation whose form depends on particular implication operator, Φ, used. The form of ELSE connective, in turn, depends on the form of the implication, Φ.

Implication operator and ELSE connectives

Name	$\Phi[A(x), B(y)]$	ELSE connectives
Mamdani	$\min[A(x), B(y)]$	OR (**s**-norm)
Larsen	$A(x) \cdot B(y)$	OR (**s**-norm)
Zadeh	$\max\{\min[A(x), B(y)], 1 - A(x)\}$	AND (**t**-norm)
Lukasziewicz	$\min[1 - A(x) + B(y), 1]$	AND (**t**-norm)
Boolean	$\max[1 - A(x), B(y)]$	AND (**t**-norm)