Fuzzy Reasoning with a Rete-OO Rule Engine

University of Applied Sciences Münster, Germany ¹
DEIS, Facolta di Ingegneria, Universita di Bologna, Italy ²

Nikolaus Wulff ¹ (Speaker)
Davide Sottara ²
Why Fuzzy Reasoning?

• *Fuzzy Control* is a widely known technology well established with a rich set of industrial applications.

• *Fuzzy Reasoning* itself had gained much attention in academia but despite many beautiful math it had missed a long time the same commercial success compared to Fuzzy Control.

• With the advent of the Semantic Web and e.g. the Fuzzy RuleML flavor of rule expressions there is hope that Fuzzy Reasoning, i.e. Fuzzy Logic, will gain the required attraction and momentum to find entrance into commercial rule engines.
Why Drools?

• We were interested in the answer which effort is required to turn a classical RE into a fuzzy one.
• We had made some experience with the classical Drools rule engine during some third party projects.
• D. Sottara et. al had expressed interest into an uncertain extension of Drools at RuleML 2008.
• Therefore Drools was a natural choice for or experiment...
Why JEFIS?

• The Java Expert Fuzzy Inference System (JEFIS) had been developed during two of our one year master projects and had proven its applicability within a third party project.

• JEFIS includes basic fuzzy sets, fuzzy operations and customizable algorithms for (de)fuzzyfication.
  – Fuzzy Trapez/Triangle/Polygons and FuzzyP/S/Z shapes
  – Join/Intersection via different t-/s-norms
  – Implication operators like MinMax, Gödel, Luca etc...
  – Input-Output via XML interface and GUI

• JEFIS was an environment we were familiar with.
JEFIS Architecture

- JEFIS uses an object-oriented approach to model fuzzy sets and operations.
JEFIS uses an abstraction layer for the Drools integration and is executable with or without Drools as a separate library, available as a Maven2 project: http://www.lab4inf.fh-muenster.de/lab4inf/Lab4Jefis
Drools Rule Language

rule "R11"
  when
    $c: Cooler($t: temperature is "cold"
    && $c: fanCurrent is "low" )
  then
    $c.setStatus("green",
        drools.getConsequenceDegree());
end

- Example for a fuzzy Drools rule.
- The Drools Chance extensions allow an integration of JEFIS enhanced fuzzy rules into the rule engine.
Fuzzy Inference

- Given a fuzzy rule
  \[ R: \text{if } x \text{ is } A \text{ then } y \text{ is } B \]
- for two fuzzy sets \( A \) and \( B \), the generalized fuzzy modus ponens for an input \( \tilde{A}(x) \) is written as
  \[
  \tilde{B} = \tilde{A} \circ (A \rightarrow B)
  \]
- to be calculated with help of an AND-ing T-norm and an implication operator \((p \rightarrow q) =: I(p, q)\)
  \[
  \tilde{B}(y) = \sup_{x \in \Omega} T(\tilde{A}(x), I(A(x), B(y)))
  \]
Fuzzy Rule Base

A fuzzy rule base of \( n \) rules

\[ R_j: \quad A_j \rightarrow B_j \quad j = 1, \ldots, n \]

can be solved as a relation equation

\[ B_j = A_j \circ R \quad j = 1, \ldots, n \]

and a solution is given if the conjunction

\[ C = \bigcap_{j=1}^{n} (A_j \rightarrow_{\text{Gö}} B_j) \]

is not empty and \( C \) is a solution for every rule \( R_j \).
Fuzzy Approximations

• For a given (fuzzy) input \( \tilde{A} \) the solution is then

\[
\hat{B} = \tilde{A} \circ \left( \bigcap_{j=1}^{n} (A_j \rightarrow_{\text{Gö}} B_j) \right)
\]

• A rule engine executes this rules sequentially and an approximate superset of the solution is:

\[
\tilde{B} = \bigcap_{j=1}^{n} \tilde{A} \circ (A_j \rightarrow B_j) \supseteq \hat{B}
\]

• Within fuzzy control we mostly broaden(!) this approximation to a solution \( \tilde{C} \) via a disjunction:

\[
\tilde{C} = \bigcup_{j=1}^{n} \tilde{A} \circ (A_j \rightarrow_{\text{XY}} B_j) \supseteq \tilde{B}
\]

Zadeh's compositional rule of inference
Emerging Questions

• This OR aggregation of the results had been proven successful for fuzzy control in many applications, mostly used for Mamdani type of inference...

• The choice to take AND or OR aggregation depends on the problem space: Are the rules *independent* conditional statements or are they *strongly coupled*? The later is presumably the case for Fuzzy Logic where the AND operation is required.

• To illustrate the problem we did a *fuzzy computer experiment*...
Fuzzy Experiment

- We want to calculate $y = 1 - x$ with a set of fuzzy rules over three fuzzy numbers partitioning the unit interval.

Rule Base for $y = 1 - x$

$R_1$: if $x$ is „low“ than $y$ is „high“

$R_2$: if $x$ is „med“ than $y$ is „med“

$R_3$: if $x$ is „high“ than $y$ is „low“
Fuzzy Partition

• We call a *natural fuzzy partition* of a linguistic variable \( n \) unique overlapping fuzzy sets \( A_j \), with

\[
0 \equiv (A_j \cap A_k)(x) \quad \forall x \in \Omega, \left| j - k \right| > 1
\]

\[
1 \equiv \sum_{j=1}^{n} A_j(x) \quad \forall x \in \Omega
\]

• In our example the \( n=3 \) rules are defined as

\[
R_k: \text{ if } x \text{ is } A_k \text{ then } y \text{ is } A_{n-k+1}
\]

– Correlar: At least \( n-2 \) rules have no match for any given \( x \).
Mamdani sup-Min $\cup$ - Inference

I(p,q)=min(p,q)

OR-aggregation:

output matches nicely to $y = 1 - x$

final result with COG
Larsen sup-Prod $\cup$ - Inference

$I(p,q) = p \cdot q$

OR-aggregation:

Also the sup-* inference does well...
Łukasiewicz ∪ - Inference

\[
\min(1, 1-p+q)
\]

OR-aggregation:

If one rule does not match the discriminating power is lost via a flooded „one“...
Łukasiewicz $\cup$ - Inference

OR-aggregation:

Even discarding unmatched rules within a RE will not help...

$$\min(1,1-p+q)$$
What had happened...?

- The Mamdani Min- and Larsen Product implications are not "real logical inference" operators, as they both give the (wrong) implication:
  \[ I(0,q) = 0 \text{ if the premise } p \text{ is zero.} \]
- Exactly therefore they are adequate to OR the resulting rule sets as a none matched rule will not show up.
- Instead logical implication operators with \( I(0,q)=1 \) will spoil the OR-ed result set almost everywhere.
- Therefore AND-aggregation is the right choice for a "correct" implication operator.
Łukasiewicz $\cap$ - Inference

\[
\min(1, 1-p+q)
\]

**AND-aggregation:**

For a „logic“ implication the $\cap$ operation is the right thing to do...
Gougen $\cap$ - Inference

AND-aggregation:

The shape of the solution set changes drastically...

$\min(1, q/p)$
Gödel ∩ - Inference

\[ I = \begin{cases} 
  1 & p \leq q \\
  q & \text{else} 
\end{cases} \]

AND-aggregation:

Gödel offers the smallest solution set, but in our toy experiment the less accurate COG
Results so far

Obviously not a complicated and ample experiment it nevertheless shows evidence that:

- For OR-aggregation of rules Mamdani- and Larsen inference are appropriate.
- AND aggregation is best with $I(0,q)=1$ operators, even if rule engines might have no match for $p=0$.
- The integration of fuzzy implications into a rule engine, e. g. into Drools, is possible, but still it is a challenge to gain speed and execution power.
- Still open: to develop a realistic test bed for logical inference to compare different approaches...