Towards Efficient Algorithms for Approximating a Fuzzy Relation by Fuzzy Rules: Case When “And”- and “Or”-Operation are Distributive

Christian Servin\textsuperscript{1,2} and Vladik Kreinovich\textsuperscript{2}

\textsuperscript{1}Information Technology Dept., El Paso Community College
El Paso, Texas, USA, cservin@gmail.com

\textsuperscript{2}TRACS Center, Department of Computer Science
University of Texas@El Paso, El Paso, Texas, USA, vladik@utep.edu
1. Relations Are Ubiquitous

- Many real-life quantities $x_1, \ldots, x_n$ are related:
  - once we know the value of one or more of the quantities,
  - this knowledge restricts possible values of other quantities.

- In some cases, we have a functional relation – the values of the quantities $x_1, \ldots, x_{n-1}$ uniquely determine $x_n$.

- Example: Ohm’s law $V = I \cdot R$.

- In many other cases, however, we have relations which are not functional.

- In mathematical terms, a relation between real-valued quantities $x_i$ is defined as a mapping $R : \mathbb{R}^n \to \{0, 1\}$:
  $$ R(x_1, \ldots, x_n) = 1 \iff (x_1, \ldots, x_n) \text{ is possible.} $$
2. Real-Life Relations Are Often Fuzzy

- In practice, about some combinations \((x_1, \ldots, x_n)\), we are not 100% sure whether they are possible.
- We must describe, for each combination \(x = (x_1, \ldots, x_n)\), our degree of certainty that \(x\) is possible.
- In the computer, “true” is usually represented as 1, and “false” as 0.
- It is therefore natural to represent intermediate degrees of certainty as numbers from the interval \([0, 1]\):
  - the larger the number,
  - the larger our degree of confidence.
- The resulting mapping \(R : \mathbb{IR}^n \to [0, 1]\) is known as a fuzzy relation.
3. Need For a Concise Representation of a Fuzzy Relation

- Theoretically, each of the quantities $x_i$ can have infinitely many different values.
- Due to measurement uncertainty, for each variable $x_i$, we only have finitely many distinguishable values $x_{i1}, \ldots, x_{ij}, \ldots, x_{iN_i}$.
- In principle, we can store the degrees of certainty corresponding to all $N_1 \cdot \ldots \cdot N_n$ combinations.
- However, when $n$ is large, the resulting number of values becomes astronomically high.
- We therefore need to come up with a more concise representation of fuzzy relations.
4. Fuzzy Rules As a Natural Concise Representation of Fuzzy Relations

• Many fuzzy relations come from *fuzzy rules*, i.e., from a combination of rules of the type

  “if $A_{r,1}(x_1)$ and ... and $A_{r,n-1}(x_{n-1})$ then $A_{r,n}(x_n)$”.

• The ubiquity of such rules comes from the fact that this is how experts often describe their decisions.

• E.g.: “if a car in front is close, and it starts breaking seriously, one needs to hit the brakes hard right away”.

• Such rules use imprecise (fuzzy) words like “close”, “seriously”, “hard”, thus we need fuzzy techniques.

• One of the most common ways to formalize the fuzzy rules is the *Mamdani approach*. 
5. **Mamdani Approach: Reminder**

- A tuple \((x_1, \ldots, x_n)\) is reasonable if for one of the rules, conditions and conclusions are satisfied:
  \[
  (A_{1,1}(x_1) \& \ldots \& A_{1,n-1}(x_{n-1}) \& A_{1,n}(x_n)) \lor \ldots \lor \\
  (A_{n_r,1}(x_1) \& \ldots \& A_{n_r,n-1}(x_{n-1}) \& A_{n_r,n}(x_n)).
  \]

- We use an “and”-operation (t-norm) \(f_\&(a, b)\) and an “or”-operation (t-conorm) \(f_\lor(a, b)\) to represent \& and \lor:
  \[
  d(x_1, \ldots, x_n) = f_\lor(d_1(x_1, \ldots, x_n), \ldots, d_{n_r}(x_1, \ldots, x_n)),
  \]
  where \(d_r(x_1, \ldots, x_n) = f_\&(A_{r,1}(x_1), \ldots, A_{r,n-1}(x_{n-1}), A_{r,n}(x_n))\).

- Fuzzy rules are a natural concise way of representing a relation.

- Thus, it is reasonable to try to approximate a given fuzzy relation by an appropriate family of rules.
6. Why t-Norms and t-Conorms: Reminder

- Often, we know the expert’s degrees of confidence \( a = d(A) \) and \( b = d(B) \) in two statements \( A \) and \( B \).
- We want to estimate the expert’s degree of confidence in \( A \& B \) or \( A \lor B \) based on \( a \) and \( b \).
- The resulting estimates \( f_\& (a, b) \) and \( f_\lor (a, b) \) are known as “and”- and “or”-operations.
- The composite statements \( A \& B \) and \( B \& A \) are equivalent to each other.
- Thus, we require that the estimates \( f_\& (a, b) \) and \( f_\& (b, a) \) be equal, i.e., that \( f_\& (a, b) \) be commutative.
- Also, \( (A \& B) \& C \) and \( A \& (B \& C) \) are equivalent, hence \( f_\& \) should be associative: \( f_\& (f_\& (a, b), c) = f_\& (a, f_\& (b, c)) \).
- Similarly, \( f_\lor \) should be commutative and associative.
7. What About Distributivity?

• Statements $A \& (B \lor C)$ and $(A \& B) \lor (A \& C)$ are also equivalent to each other.

• So why not require distributivity

$$f_\& (a, f_\lor (b, c)) = f_\lor (f_\& (a, b), f_\& (a, c)).$$

• Distributivity does hold when $f_\lor (a, b) = \max (a, b)$.

• *Problem*: distributivity only holds for $f_\lor = \max$, while other t-conorms are also useful.

• *Proof*: for $b = c = 1$, we get $f_\& (a, 1) = f_\lor (f_\& (a, 1), f_\& (a, 1))$, hence $a = f_\lor (a, a)$, which implies max.

• *Solution*: require $f_\& (a, f_\lor (b, c)) = f_\lor (f_\& (a, b), f_\& (a, c))$ only when $f_\lor (b, c) < 1$.

• *Example*: $f_\& (a, b) = a \cdot b$, $f_\lor (a, b) = \min (a + b, 1)$. 
8. Need to Describe Distributive “And”- and “Or”-Operations

- The main objective of this talk is to approximate a general fuzzy relation by fuzzy rules.

- We explained why it is reasonable to require that the “and”- and “or”-operations are distributive.

- Our goal is thus to approximate a general fuzzy relation by fuzzy rules that use distributive operations.

- We would like to produce an algorithm which is applicable for each distributive pairs of operations.

- So, to approach this approximate problem, let us see how we can describe such a generic pair.

- To come up with such a description, let us first recall how we can describe a generic “or”-operation.
9. Different Types of “Or”-Operations: Reminder

- Some “or”-operations are Archimedean: for all $a, b \in (0, 1)$ there is an $n$ s.t. $f_\lor(a, a, \ldots, a)$ ($n$ times) $> b$.
- Example: “algebraic sum” $f_\lor(a, b) = a + b - a \cdot b$.
- All such operations are isomorphic to addition: $f_\lor(a, b) = \psi^{-1}(\psi(a) + \psi(b))$ for some $\psi(a)$.
- We also have $\max(a, b)$ and operations isomorphic to $\min(a + b, 1)$: $f_\lor(a, b) = \psi^{-1}(\min(\psi(a) + \psi(b), 1))$.
- Every “or”-operation is isomorphic to a lexicographic combination of:
  - Archimedean operations,
  - max, and
  - operations isomorphic to $f_\lor(a, b) = \min(a + b, 1)$.
10. We Can Consider Approximate Descriptions

- The main purpose of “or”-operation $f_\vee(a, b)$ is to estimate the expert’s degree of belief $d(A \lor B)$.
- We can thus replace an “or”-operation with a close one without changing its estimation accuracy.
- **Known:** for every “or”-operation $f_\vee(a, b)$ and for every $\varepsilon > 0$, there is an $\varepsilon$-close Archemedean $f'_\vee(a, b)$:
  \[ |f'_\vee(a, b) - f_\vee(a, b)| \leq \varepsilon. \]
- Alas, for Archemedean operations, $b, c < 1$ imply $f_\vee(b, c) < 1$, so distributivity implies $f_\vee = \max$.
- We thus need to approximate by a non-Archimedean operation.
- **New Result:** for every $\varepsilon > 0$, each “or”-operation can be $\varepsilon$-approximated by an operation isomorphic to $\min(a + b, 1)$.
11. A New Universal Approximation Result: Idea of the Proof

- *Known:* each Archimedean “or”-operation has the form $f'_\lor(a, b) = \psi^{-1}(\psi(a) + \psi(b))$ for some function $\psi$.

- For every $\delta > 0$, consider a new function $\psi'(a)$ s.t.:
  - $\psi'(a) = \psi(a)$ for all $a \leq 1 - \delta$ and
  - $\psi'(a) = \psi(1 - \delta) + (a - (1 - \delta))$ for all $a \in (1 - \delta, 1]$.

- For this function $\psi'(a)$, we form an “or”-operation
  $$f''_\lor(a, b) \overset{\text{def}}{=} (\psi')^{-1}(\min(\psi'(a) + \psi'(b), \psi'(1))).$$

- When $\delta \to 0$, we have $f''_\lor(a, b) \to f'_\lor(a, b)$.

- Thus, for sufficiently small $\delta > 0$, $f''_\lor(a, b)$ is close to $f'_\lor(a, b)$ and thus, to $f_\lor(a, b)$.

- In terms of $\psi''(a) \overset{\text{def}}{=} \frac{\psi'(a)}{\psi'(1)}$, we get
  $$f''_\lor(a, b) = (\psi'')^{-1}(\min(\psi''(a) + \psi''(b), 1)).$$
12. Let Us Use the New Approximation Result to Describe All Distributive Pairs

- Each “or”-operation can be, with arbitrary accuracy, approximated by an operation isom. to \( \min(a + b, 1) \).
- Thus, for all practical purposes, we can assume that the actual “or”-operation is isomorphic to \( \min(a + b, 1) \).
- In the corresponding scale, \( c = f_\lor(a, b) \) takes the form \( c' = \min(a' + b', 1) \), and distributivity means
  \[
  b' + c' < 1 \Rightarrow f_\land(a', b' + c') = f_\land(a', b') + f_\land(a', c').
  \]
- In other words, for each \( a' \), the function \( b' \to f_\land(a', b') \) is a monotonic additive function of \( b' \).
- It is known that all monotonic additive functions have the form \( f(x) = k \cdot x \).
- Thus, we have \( f_\land(a', b') = k(a') \cdot b' \) for some \( k(a') \).
13. Let’s Describe All Distributive Pairs (cont-d)

- For \( f_\& (a', b') = k(a') \cdot b' \), commutativity means
  \[
  k(a') \cdot b' = k(b') \cdot a'.
  \]

- Dividing both sides of this equality by \( a' \cdot b' \), we conclude that
  \[
  \frac{k(a')}{a'} = \frac{k(b')}{b'}.
  \]

- In other words, we conclude that the ratio \( \frac{k(a')}{a'} \) has the same value for all possible values \( a' \in [0, 1] \).

- In other words, we conclude that this ratio is a constant; let us denote this constant by \( r \).

- Then, from \( \frac{k(a')}{a'} = r \), we conclude that \( k(a') = r \cdot a' \).

- Therefore, \( f_\& (a', b') = k(a') \cdot b' = r \cdot a' \cdot b' \).

- From the requirement that \( f_\& (1, 1) = 1 \), we conclude that \( r = 1 \) and thus, \( f_\& (a', b') = a' \cdot b' \).
14. General Description of Distributive Pairs: Result

- Each “or”-operation can be, with arbitrary accuracy, approximated by an operation isom. to \( \min(a + b, 1) \).
- Thus, for all practical purposes, we can assume that the actual “or”-operation is isomorphic to \( \min(a + b, 1) \).
- Under this assumption, each distributive pair is isomorphic to the pair consisting of:
  
  - an “and”-operation \( f_\lor(a, b) = \min(a + b, 1) \) and
  
  - the algebraic-product “and”-operation \( f_\&(a, b) = a \cdot b \).
15. Approximating a Fuzzy Relation by Fuzzy Rules: What We Propose

- **Reminder:**
  - we have a fuzzy relation $R(x_1, \ldots, x_n)$,
  - we have a distributive pair of “and”- and “or”- operations, and
  - we want to represent $R$ as
    \[
    R(x_1, \ldots, x_n) = f_\vee(d_1(x_1, \ldots, x_n), \ldots, d_{nr}(x_1, \ldots, x_n)),
    \]
    where $d_r(x_1, \ldots, x_n) = f_\&(A_{r,1}(x_1), \ldots, A_{r,n}(x_n))$.

- Since operations are distributive, after rescaling $\psi''(a)$, we get $\min(a + b, 1)$ and product.

- Thus, the desired representation takes the form
  \[
  R'(x_1, \ldots, x_n) = \min\left(\sum_{r=1}^{nr} \prod_{i=1}^{n} A'_{ri}(x_i), 1\right).
  \]
16. What We Propose (cont-d)

- Approximating a function \( R'(x_1, \ldots, x_n) \) by expressions
  \[ \sum_{r=1}^{n_r} \prod_{i=1}^{n} A_{r_i}(x_i) \]
  is known as tensor decomposition.

- Many efficient algorithms have been developed for solving this problem.

- We therefore propose to solve the original problem of approximating a relation \( R(x_1, \ldots, x_n) \) as follows:
  - first, we form \( R'(x_1, \ldots, x_n) = \psi''(R(x_1, \ldots, x_n)) \);
  - then, we use a tensor decomposition algorithm to find the \( A_{r_i}(x_i) \) approximating \( R'(x_1, \ldots, x_n) \);
  - finally, we “re-scale” the resulting functions \( A_{r_i}(x_i) \) back to the original scale, i.e., form functions
    \[ A_{r_i}(x_i) \overset{\text{def}}{=} (\psi'')^{-1}(A_{r_i}(x_i)) \].

- Thus, we approximate the relation \( R \) by fuzzy rules.
17. Discussion

- We are interested in representations with non-negative values $A_{ri}(x_i)$.
- Most tensor decomposition algorithms allow representations with functions of arbitrary sign.
- So we may end up with negative values of $A_{ri}(x_i)$.
- This is OK if all we are interested in is approximation.
- However, we may want an approximation s.t. $A_{ri}(x_i)$ are membership functions, i.e., $A_{ri}(x_i) \geq 0$.
- To achieve this, we replace each negative value by the closest non-negative one, i.e., by 0.
- It should be mentioned, however, that this replacement may somewhat decrease the approximation accuracy.
18. Acknowledgment

This work was supported in part by the National Science Foundation grants:

- HRD-0734825 and HRD-1242122 (Cyber-ShARE Center of Excellence), and
- DUE-0926721.