Most Robust Fuzzy Extensions of Binary Logical Operations

Irvin L. Bosquez and Vladik Kreinovich

Department of Computer Science
University of Texas at El Paso
El Paso, TX 79968, USA,
ilbosquez@miners.utep.edu, vladik@utep.edu
1. Need for Fuzzy Extensions

- In many practical situations when we solicit expert opinions, we are interested in a property:
  - which experts cannot easily directly estimate,
  - but we know that this property is a boolean combination of easier-to-estimate properties.

- Example: a medicine is efficient when the blood pressure is high ($H$) but not very high ($\neg V$):
  $$H \& \neg V.$$ 

- Only a few experts can estimate the medicine’s efficiency.

- However, doctors usually have a good understanding of when the blood pressure is high or very high.
2. Need for Fuzzy Extensions (cont-d)

- The difficulty is that in many situations, an expert is not 100% confident in his/her statement.
- At best, the expert can mark, on a scale from 0 to 1, to what extent he/she believes in this statement.
- We can thus get expert’s degrees of belief $a$ and $b$ in statements $A$ and $B$.
- We need to estimate the resulting degree of belief in the corr. boolean combination $A \ast B$ (such as $A \& \neg B$).
- When experts are absolutely sure, i.e., when each $a$ and $b$ is 0 or 1, we should get the usual boolean results.
- Thus, what we need is to extend the usual boolean operations $a \ast b$:
  - from the 2-valued set $\{0, 1\}$
  - to the function $f_\ast(a, b)$ defined for all $a, b \in [0, 1]$. 
3. Need for Robustness, and Resulting Extension Procedure

- The values $a$ and $b$ are not very accurate.
- If we ask the same expert twice, he/she may give slightly different values.
- It is desirable to make sure that the resulting estimate $f_*(a, b)$ be minimally affected by this difference.
- For example, if we fix $a$ and for some $n$, take values $b_i = i/n$, then for $c_i = f_*(a, b_i)$ we should have
  \[ c_i \approx c_{i+1}, \text{i.e., } c_i - c_{i+1} \approx 0. \]
- In other words, a multi-D point $(c_1 - c_2, c_2 - c_3, \ldots)$ should be close to $(0, 0, \ldots)$.
- The distance between these points is the smallest when its square $\sum (c_i - c_{i+1})^2$ is the smallest.
4. Need for Robustness (cont-d)

- **Reminder:** we minimize the expression $\sum_{i}(c_i - c_{i+1})^2$.
- Equating the derivative of this expression to 0, we conclude that $c_i - c_{i+1} = c_{i-1} - c_i$ for all $i$.
- Thus, the expression $f_*(a, b)$ is linear in $b$.
- So, if we know $f_*(a, 0)$ and $f_*(a, 1)$, we can get the values $f_*(a, b)$ for all $b$ by linear interpolation.
- Similarly, $f_*(a, b)$ should be linear in $a$.
- Thus, based on the values $f_*(0, 0)$ and $f_*(0, 1)$, we can use linear interpolation to find the values $f_*(0, b)$.
- Similarly, we get $f_*(1, b)$.
- For each $b$, based on the values $f_*(0, b)$ and $f_*(1, b)$, we can similarly find the values $f_*(a, b)$. 
5. What Is Known and What We Do

• The results of using this procedure are known for $\&$ and $\lor$.

• We are extending it to all possible binary boolean operations.
6. Results

- There are four pairs of boolean values:
  
  00, 01, 10, and 11.

- For each such pair, the operation can give 0 or 1.

- Thus, there are $2^4 = 16$ possible binary boolean operations.

- We can describe each such operation by listing the 0-1 values corresponding to inputs 00, 01, 10, and 11.

- The sequence 0000 corresponds to $a \ast b = 0$ which interpolates to $f_*(a, b) = 0$.

- For 0001, we get $a \ast b = a \& b$, and $f_*(a, b) = a \cdot b$.

- For 0010, we get $a \ast b = a \& \neg b$ and $f_*(a, b) = a \cdot (1 - b)$.

- For 0011, we get $a \ast b = a$ and $f_*(a, b) = a$.

- For 0100, we get $a \ast b = \neg a \& b$ and $f_*(a, b) = (1 - a) \cdot b$. 
7. Results (cont-d)

- For 0101, we get $a \ast b = b$ and $f_\ast(a, b) = b$.

- For 0110, we get exclusive “or”, with
  \[ f_\ast(a, b) = a + b - 2a \cdot b. \]

- For 0111, we get $a \ast b = a \lor b$ and $f_\ast(a, b) = a + b - a \cdot b$.

- For $\ast = 1\varepsilon_1\varepsilon_2\varepsilon_3$, we get $a \ast b = \neg(a \ast' b)$, where $\ast' = 0(1 - \varepsilon_1)(1 - \varepsilon_2)(1 - \varepsilon_3)$, and $f_\ast(a, b) = 1 - f_\ast'(a, b)$.

- Comment:
  - all boolean operations can be described in terms of $&$, $\lor$, and $\neg$;
  - however, fuzzy exclusive “or” cannot be described in terms of fuzzy $&$, $\lor$, and $\neg$;
  - so, we need to add exclusive “or” to basic operations.