Belief Logic Programming with
Cyclic Dependencies

3rd International Conference on Web Reasoning and Rule Systems
Hui Wan
Stony Brook University, New York
Uncertainty in Rule-based KBs

• Formalisms for representing uncertainty
  – Probabilities
  – Fuzzy sets
  – Belief functions

• Our approach: based on belief functions
Belief Concept in Everyday Life

- “Based on the tests, the doctor is 60% sure that Tim has cancer.”

- “Look, the ground is wet, I can say with 0.7 certainty that it rained last night.”

- “Based on the evidence, we believe with the confidence of 0.99 that O.J. Simpson committed the murder.”

*These are not random events.*

*Degrees of belief, not probabilities.*
Motivation:
Evidence Combination in Rule-based KBs

- What is the belief in \( a \)?

\[
\begin{align*}
  a & : - b. \\
  a & : - c \land e. \\
  b & : - c \land d.
\end{align*}
\]

\[
\begin{align*}
  0.5 & : c. \\
  0.8 & : d. \\
  0.6 & : e.
\end{align*}
\]

\[\Phi(x,y) = x+y-xy \quad \text{combination function}\]

\[\Phi(0.3,0.4) = 0.58\]

- But inferences for \( a \) are **correlated** so should **not** be combined outright.
  - Use combination function “max”?
    - No entailment between the two pieces of evidence.
  - The belief in \( a \) should be < 0.58, and > 0.4. - How much?

- Things become even more complicated when rules are uncertain.
• **Goal**
  – Theory for combining evidence from sources that might be *correlated*

• **Prior work**
  – Does not account for structural correlation through rules (as in the example)

• **Our prior work**: Belief LP (LPNMR09, SUM09)
  – solves this problem, but no ground recursion is allowed
Syntax

\[ [v, w] \quad \text{Head} : \neg \text{Body} \]

- **Belief factor**, \(0 \leq v \leq w \leq 1\)
  - e.g., \([0.2, 0.4]\)

- **An atom**
  - e.g., \(p(X)\)

- **Conjunction of literals**
  - e.g., \(p(X) \land b \land \neg c\)

Our prior work:

no cyclic dependencies among atoms

\[
\begin{align*}
[0.6, 0.8] & \quad a : - b \land c \\
[0.3, 0.9] & \quad b : - d \land \neg e \\
[0.8, 1] & \quad d : - a \land f
\end{align*}
\]
Meaning of Rule

\[ [v, w] \quad X : – \quad Body \]

• If \textit{Body} holds, then this rule supports
  – \( X \) to the degree of \( v \)
  – \( \overline{X} \) to the degree of \( 1 - w \)

• The semantics ensures that
  if only one rule supports \( X \)
  \[
  \frac{Bel(X \land Body)}{Bel(Body)} = v \quad \frac{Bel(\overline{X} \land Body)}{Bel(Body)} = 1 - w
  \]

Similar intuition holds for multiple rules
(with appropriate modifications)
Combination Functions

• **Combination function:**

\[
\Phi(\{ [v_1, w_1], \ldots, [v_n, w_n] \}) = [v, w]
\]

- \(\Phi(\{ [v, w] \}) = [v, w]\), \(\Phi(\{\}) = [0, 1]\)

• **Each atom** \(X\) **has an associated combo function** \(\Phi_X\)
  - \(\Phi_X\) **decides how to combine beliefs in** \(X\)
  - **Different** \(\Phi_X\) **for different application domains**
  - **Examples:** Dempster’s rule, \(\text{max}\), ...
## Semantics: Basics

<table>
<thead>
<tr>
<th>Notion</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Herbrand base</strong> $B_P$</td>
<td>$B_P = {a, b, c}$</td>
</tr>
<tr>
<td><strong>Truth valuation</strong> $I: B_P \to {t, f, u}$</td>
<td>$I_1(a) = t, I_1(b) = t, I_1(c) = t$</td>
</tr>
<tr>
<td>$I_2(a) = t, I_2(b) = t, I_2(c) = f$</td>
<td></td>
</tr>
<tr>
<td>$I_3(a) = t, I_3(b) = t, I_3(c) = u$</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
<tr>
<td>$I_{27}(a) = u, I_{27}(b) = u, I_{27}(c) = u$</td>
<td></td>
</tr>
<tr>
<td><strong>Support function</strong> $m: tval(P) \to [0,1]$</td>
<td>$m(I_1) = 0.5$</td>
</tr>
<tr>
<td>$m(I_3) = 0.5$</td>
<td></td>
</tr>
<tr>
<td>$m(I_k) = 0$, if $k \neq 1, k \neq 3$</td>
<td></td>
</tr>
<tr>
<td><strong>Belief function</strong> $bel$</td>
<td>Plays the role of interpretation in classical LP.</td>
</tr>
<tr>
<td>$bel(F) = \sum_{I \models F} m(I)$</td>
<td></td>
</tr>
</tbody>
</table>

$F$: a boolean formula  
$\models$: entailment in Lucasiewicz 3-valued logic
Semantics

• Model of a belief logic program:
  – a belief function

• Properties
  – Non-monotonic
  – Strong relation to paraconsistent reasoning and defeasible reasoning
Declarative Semantics

- The set of rules that support $X$ and fire in $I$
  $$P_I(X) = \{ R \mid R \in P, X \in \text{Head}(R), \ I \models \text{Body}(R) \}$$

- $s_p(I,X)$: the support in $I$ for $X$
  - If $P_I(X) = \emptyset$
    $$s_p(I,X) = \begin{cases} 1 & \text{if } I(X) = u \\ 0 & \text{if } I(X) = t \text{ or } I(X) = f \end{cases}$$
  - If $P_I(X) = \{ R_1, \ldots, R_k \}$
    let $[v, w]$ be the result of applying $\Phi_X$ to the belief factors of $R_1, \ldots, R_k$
    $$s_p(I,X) = \begin{cases} v & \text{if } I(X) = t \\ 1 - w & \text{if } I(X) = f \\ w - v & \text{if } I(X) = u \end{cases}$$

- The support in $I$ for $P$ as a whole
  $$m_P(I) = \prod_{X \in B_P} s_p(I, X)$$

- Model for $P$:
  A belief function $\text{bel}$:
  $$F \rightarrow \sum_{I \in \text{Itval}(B_P), I \models F} m_P(I)$$
  a boolean formula
This Paper

• Method to deal with arbitrary belief logic programs, including the cyclic ones
  – transformation-based
  – has good properties
  – circular feedback is not problematic
Cyclic Program

\[ [0.8, 1] \quad disease(X) :\neg test\_pos(X). \]

// Positive test result supports the diagnosis of a person having the disease.

\[ [0.6, 1] \quad disease(X) :\neg contact(X, Y) \land disease(Y). \]

// The disease is contagious.

\[ [1, 1] \quad test\_pos(p3). \quad [1, 1] \quad test\_pos(p4). \]
\[ [1, 1] \quad contact(p3, p4). \quad [1, 1] \quad contact(p4, p3). \]

r1 \quad [0.8, 1] \quad disease(p3) :\neg test\_pos(p3).

r2 \quad [0.8, 1] \quad disease(p4) :\neg test\_pos(p4).

r3 \quad [0.6, 1] \quad disease(p3) :\neg contact(p3, p4) \land disease(p4).

r4 \quad [0.6, 1] \quad disease(p4) :\neg contact(p4, p3) \land disease(p3).

r5 \quad [1, 1] \quad test\_pos(p3).

r6 \quad [1, 1] \quad test\_pos(p4).

r7 \quad [1, 1] \quad contact(p3, p4).

r8 \quad [1, 1] \quad contact(p4, p3).
Dependency Graph

- `disease(p3)`
  - `test_pos(p3)`
  - `r1` [0.8,1]
  - `r5` [1,1]
- `contact(p3,p4)`
  - `r3` [0.6,1]
  - `r7` [1,1]
- `contact(p4,p3)`
  - `r4` [0.6,1]
  - `[1,1]`
- `test_pos(p4)`
  - `r2` [0.8,1]
  - `[1,1]`
- `r6` [1,1]
- `r8` [1,1]
Atom Cliques

Two atoms are in the same clique iff they depend on each other.
Partial Proof DAG (pp-DAG)

- Corresponds to an SLD derivation of the query.
- No cycles allowed.
- Within the clique.
Child pp-DAG

G3 is a child pp-DAG of G2.

G1 is a child pp-DAG of G4.
Decyclification

- For every rule whose head atom is in a cycle

\[ R \begin{array}{|c|c|} \hline [v,w] & A_0 : - A_1 \wedge \ldots \wedge A_k \wedge \overline{A_{k+1}} \wedge \ldots \wedge \overline{A_n} \wedge Conj. \end{array} \]

where \( A_0, \ldots, A_n \) are in the same clique, and no atom in \( Conj \) is in this clique.
- Replace it with

\[ [v,w] \quad A_0 : - R. \]

- For every list of pp-DAGs, \( G_0, \ldots, G_n \), such that
  - \( A_i \) is \( G_i \)'s root, and \( R \) is in \( G_0 \)
  - \( G_1, \ldots, G_n \) are children of \( G_0 \)

add rules

\[ [v,w] \quad A_0^{G_0} : - A_1^{G_1} \wedge \ldots \wedge A_k^{G_k} \wedge \overline{A_{k+1}^{G_{k+1}}} \wedge \ldots \wedge \overline{A_n^{G_n}} \wedge Conj. \]
\[ [1,1] \quad R : - A_1^{G_1} \wedge \ldots \wedge A_k^{G_k} \wedge \overline{A_{k+1}^{G_{k+1}}} \wedge \ldots \wedge \overline{A_n^{G_n}} \wedge Conj. \]

- Intuition:
  - The belief in \( A^G \) is precisely that part of the belief in \( A \), which is justified by the derivation that corresponds to \( G \).
  - The belief in \( R \) is the belief in the rule body being derived without any loop influence.
  - The belief in \( A_0 \) is the combination of the support to \( A_0 \) from all the rules with \( A_0 \) in head.
\[\text{acyclic}(P)\]

\[
\begin{align*}
[0.8, 1] & \quad \text{disease}(p3) : - r1. \\
[1, 1] & \quad r1 : - \text{test}_\text{pos}(p3). \\
[0.8, 1] & \quad \text{disease}(p4) : - r2. \\
[1, 1] & \quad r2 : - \text{test}_\text{pos}(p4). \\
[0.6, 1] & \quad \text{disease}(p3) : - r3. \\
[1, 1] & \quad r3 : - \text{contact}(p3, p4) \land \text{disease}^{G2}(p4). \\
[0.6, 1] & \quad \text{disease}(p4) : - r4. \\
[1, 1] & \quad r4 : - \text{contact}(p4, p3) \land \text{disease}^{G1}(p3). \\
[0.8, 1] & \quad \text{disease}^{G1}(p3) : - \text{test}_\text{pos}(p3). \\
[0.8, 1] & \quad \text{disease}^{G2}(p4) : - \text{test}_\text{pos}(p4). \\
[0.6, 1] & \quad \text{disease}^{G3}(p3) : - \text{contact}(p3, p4) \land \text{disease}^{G2}(p4). \\
[0.6, 1] & \quad \text{disease}^{G4}(p4) : - \text{contact}(p4, p3) \land \text{disease}^{G1}(p3). \\
\end{align*}
\]

\[
\begin{align*}
\text{bel(\text{disease}(p3))} & = 0.896 \\
\text{bel(\text{disease}(p4))} & = 0.896
\end{align*}
\]

G1, G2 contribute to disease(p3).
G3, G4 contribute to disease(p4).
G3 is a child pp-DAG of G2.
G1 is a child pp-DAG of G4.
• The model of a general program $P$ is the model of $\text{acyclic}(P)$.

• Properties
  – Backward compatible with acyclic semantics
  – Circular feedback is not problematic
• Transformational semantics
  – based on a global decyclification

• Fixpoint semantics
  – based on a sequence of local decyclifications

• Equivalent results
A step in the fixpoint computation:

– Consider: the rules with head in an atom clique $C$

– Already known: the support to truth valuation $I$ for the atoms which $C$ depends on

\[
\begin{array}{c|c}
0.8, 1 & a \leftarrow d \\
0.8, 1 & b \leftarrow e \\
0.6, 1 & a \leftarrow b \land f_1 \\
0.6, 1 & b \leftarrow a \land f_2 \\
1, 1 & d \\
1, 1 & e \\
0.5, 0.5 & f_1 \\
0.5, 1 & f_2 \leftarrow f_1 \\
\end{array}
\]

$[0.8, 1] \ a \leftarrow a$.  
$[0.8, 1] \ b \leftarrow e$.  
$[0.6, 1] \ a \leftarrow b \land f_1$.  
$[0.6, 1] \ b \leftarrow a \land f_2$.  
$[1, 1] \ d$.  
$[1, 1] \ e$.  
$[0.5, 0.5] \ f_1$.  
$[0.5, 1] \ f_2 \leftarrow f_1$.  

$m(I) = 0.25$

$m_c: \ tval(\{a,b\}) \rightarrow [0,1]$

\[
\{ m'(I+I'_k) = m(I) \ast m_c(I'_k) \mid I'_k \text{ is in } tval(\{a,b\}) \}
\]
$B_P$ can be divided into cliques $C_1, \ldots, C_l$ such that $C_k$ only depend on $C_1 \ldots C_{k-1}, \quad 1 < k \leq l$

$S_k = C_1 \cup \ldots \cup C_k$

$m_{S_k}: tval(S_k) \to [0,1]$

$S_0 = \{\}$

$S_n = B_P$

Model -- the belief function based on $m_{S_n}$

Fixpoint semantics coincides with the transformational semantics
Modular Acyclicity

• In every step, under every $I$ being supported, there is no cycle formed by the rules involved.
  – No decyclification needed!

\[
\begin{array}{c|cc|c|c}
\hline
I1 & f1 & f2 & d & e \\
\hline
\text{m(I1)=0.5} & \text{t} & \text{f} & \text{t} & \text{t} \\
\hline
\end{array}
\]

\[
\begin{array}{c|cc|c|c}
\hline
I2 & f1 & f2 & d & e \\
\hline
\text{m(I2)=0.5} & \text{f} & \text{u} & \text{t} & \text{t} \\
\hline
\end{array}
\]
Conclusion

• Belief LP: a novel framework for reasoning with uncertainty in the presence of correlated evidence.
  – Declarative semantics
  – Operational semantics
  – Query answering algorithm

• This paper
  – extends the semantics to arbitrary belief logic programs

• Future work:
  – extend the query answering algorithm to arbitrary belief logic programs
Thank you!

Questions?