WOLED: A Tool for Online Learning Weighted Answer Set Rules for Temporal Reasoning Under Uncertainty

Nikos Katzouris\textsuperscript{1}, Alexander Artikis\textsuperscript{2,1}

http://cer.iit.demokritos.gr

\textsuperscript{1}National Center for Scientific Research Demokritos, Athens, Greece
\textsuperscript{2}University of Piraeus, Piraeus, Greece

KR 2020
Application: Complex Event Recognition

Simple Events

Event Recognition System

Complex Event Definitions

Input ▶ Recognition ▶ Output ■

Complex Events

Learn this From These

... ■ ■ ■ ■ ...
Application: Complex Event Recognition

Simple Events

Complex Event Definitions

Event Recognition System

Input ▶ Recognition ▶ Output ■

Complex Events

happensAt(\textit{active}(id_0), 10)
holdsAt(\textit{coord}(id_0, 20.88, 11.90), 10)
happensAt(\textit{active}(id_1), 10)
holdsAt(\textit{coord}(id_1, 22.34, 15.23), 10)

holdsAt(\textit{meet}(id_0, id_1), 11)
holdsAt(\textit{meet}(id_0, id_1), 12)
holdsAt(\textit{meet}(id_0, id_1), 13)
Application: Complex Event Recognition

Event Calculus as a Reasoning Engine

holdsAt(F, T + 1) ←
initiatedAt(F, T)

holdsAtAt(F, T + 1) ←
holdsAt(F, T),
not terminatedAt(F, T).


Simple Events

happensAt(active(id_0), 10)
holdsAt(coord(id_0, 20.88, 11.90), 10)
happensAt(active(id_1), 10)
holdsAt(coord(id_1, 22.34, 15.23), 10)
...

Complex Events

happensAt(meet(id_0, id_1), 11)
holdsAt(meet(id_0, id_1), 12)
holdsAt(meet(id_0, id_1), 13)
...

Learn this From These
Application: Complex Event Recognition

Event Calculus as a Reasoning Engine

\[
\text{holdsAt}(F, T + 1) \leftarrow \\
\text{initiatedAt}(F, T) \\
\text{holdsAt}(F, T + 1) \leftarrow \\
\text{holdsAt}(F, T), \\
\text{not terminatedAt}(F, T).
\]

Learning for Complex Event Recognition

Event Calculus as a Reasoning Engine

holdsAt(F, T + 1) ← initiatedAt(F, T)

holdsAtAt(F, T + 1) ← holdsAt(F, T), not terminatedAt(F, T).


Input ▶ Recognition ▶ Output ■

Event Recognition System

Complex Event Definitions

Simple Events
... ■ ■ ■ ■ ...

Complex Events
... ■ ■ ■ ■ ...

Learn this

From These
Learning Requirements

- Event recognition applications deal with noisy data streams.
  - Resilience to noise & uncertainty:
    - Statistical Relational Learning.
    - Logical representations + probability.
  - Big data, data streams.
    - Online, single-pass learning.
Learning Requirements

- Event recognition applications deal with noisy data streams.
  - Resilience to noise & uncertainty:
    - Statistical Relational Learning.
    - Logical representations + probability.
  - Big data, data streams.
    - Online, single-pass learning.

- Statistical Relational Learning:
  - Rules’ structure learning.
    - Inductive Logic Programming
  - Weight learning.
    - Gradient-based techniques.
Statistical Relational Learning in Answer Set Programming

▶ Why?
  ▶ Non-monotonic semantics.
  ▶ Sophisticated off-the-self ASP solvers.
  ▶ Structure & weight learning tasks easily encoded as optimization problems in ASP.

▶ How?
  ▶ Setting very similar to Markov Logic Networks.
  ▶ Real-valued weights attached to rules in an ASP program \( \Pi \).
  ▶ Larger weights, larger confidence to rules.
  ▶ Weights define a probability distribution over answer sets of \( \Pi \).
  ▶ Lee & Young, Weighted rules under the stable model semantics, KR 2016.
Why?
- Non-monotonic semantics.
- Sophisticated off-the-self ASP solvers.
- Structure & weight learning tasks easily encoded as optimization problems in ASP.

How?
- Setting very similar to Markov Logic Networks.
- Real-valued weights attached to rules in an ASP program $\Pi$.
- Larger weights, larger confidence to rules
- Weights define a prob. distribution over answer sets of $\Pi$.

Probabilistic MAP Inference

- Task: find most probable answer set.
- Turns out to be a weighted MaxSat problem:
  - Find an answer set that maximizes the sum of weights of satisfied rules.
Task: find most probable answer set.

Turns out to be a weighted MaxSat problem:

- Find an answer set that maximizes the sum of weights of satisfied rules.

Handled directly by an ASP solver:

\[
\text{head}_i \leftarrow \text{satisfied}(i) \\
\{\text{satisfied}(i)\} \leftarrow \text{body}_i \\
: \sim \text{satisfied}(i). [\neg w_i]
\]

\[
\text{head}_i \leftarrow \text{body}_i \textit{ is the i-th rule with weight } w_i.
\]
Weight Learning

- Compare results in MAP-inferred state with true state.
- Update weights according to mistakes.

\[ w_{t+1}^i = \text{sign}(w_t^i - \eta C_t^i \Delta g_t^i) \max\{0, |w_t^i - \eta C_t^i \Delta g_t^i| - \lambda \eta C_t^i \} \]

- Current weight of the \( i \)-th rule
- Previous weight of the \( i \)-th rule
- Learning rate
- Rule's current mistakes
- Term proportional to the rule's accumulated past mistakes
- Regularization rate

\( \Delta g_t^i \) (\( i \)-th rule's mistakes at time \( t \)): difference in rule's true groundings in the true state and the MAP-inferred state.
Weight Learning

▶ Compare results in **MAP-inferred state** with **true state**.
▶ Update weights according to mistakes.

**AdaGrad-based weight update rule:**

\[
    w_{t+1}^i = \text{sign}(w_t^i - \frac{\eta}{C_t^i} \Delta g_t^i) \max\{0, |w_t^i - \frac{\eta}{C_t^i} \Delta g_t^i| - \lambda \frac{\eta}{C_t^i}\}
\]

- Current weight of the \(i\)-th rule
- Previous weight of the \(i\)-th rule
- Learning rate
- Rule's current mistakes
- Regularization rate
- Term proportional to the rule's accumulated past mistakes

\(\Delta g_t^i\) (\(i\)-th rule's mistakes at time \(t\)): difference in rule's true groundings in the true state and the MAP-inferred state.
Structure Learning I: Learning New Rules from Mistakes

- Techniques from non-monotonic Inductive Logic Programming.
- Reasoning with existing weighted rules and generalizing new bottom rules part of the same optimization process in ASP.
Structure Learning I: Revising Existing rules

- As new data arrive the structure of rules often need to be revised.
  - Specialize rules.
  - Online Hill-Climbing via Hoeffding tests.
  - Using a small part of the input stream at each specialization decision point.
Putting it All Together

Background Knowledge

\[ \text{holdsAt}(F, T + 1) \leftarrow \text{initiatedAt}(F, T). \]

\[ \text{holdsAt}(F, T + 1) \leftarrow \text{holdsAt}(F, T), \not\text{terminatedAt}(F, T). \]

Mode Declarations

head(\text{initiatedAt}(\text{move}(+id, +id), +time))

head(\text{terminatedAt}(\text{move}(+id, +id), +time))

body(\text{happensAt}(\text{walking}(+id, +id), +time))

body(\not\text{happensAt}(\text{walking}(+id, +id), +time))

body(\text{distLessThan}(+id, +id, #dist, +time))

body(\text{dirLessThan}(+id, +id, #dist, +time))

Current Weighted ASP Theory \( H_t \):

1.345 \text{initiatedAt}(\text{move}(X, Y), T) \leftarrow \text{happensAt}(\text{walking}(X), T), \text{happensAt}(\text{walking}(Y), T), \text{distLessThan}(X, Y, 34, T)

0.865 \text{terminatedAt}(\text{move}(X, Y), T) \leftarrow \text{happensAt}(\text{inactive}(X), T), \not\text{distLessThan}(X, Y, 34, T)

WOLED-ASP

MAP Inference

Theory Expansion

Hoeffding Tests/Rule Expansion

Weight Updates

Training Stream

Training Interpretation \( I_t \)

\[ \text{holdsAt}(\text{move}(id_1, id_2), 10) \]

\[ \text{happensAt}(\text{walking}(id_1), 9) \]

\[ \text{happensAt}(\text{walking}(id_2), 9) \]

\[ \text{coords}(id_1, 23, 104, 9) \]

\[ \text{coords}(id_2, 42, 84, 9) \]

\[ \text{direction}(id_1, 212, 9) \]

\[ \text{direction}(id_2, 78, 9) \]

Training Interpretation \( I_t' \)

\[ \not\text{holdsAt}(\text{move}(id_1, id_2), 100) \]

\[ \text{happensAt}(\text{walking}(id_1), 99) \]

\[ \text{happensAt}(\text{walking}(id_2), 99) \]

\[ \text{coords}(id_1, 205, 23, 99) \]

\[ \text{coords}(id_2, 462, 24, 99) \]

\[ \text{direction}(id_1, 212, 9) \]

\[ \text{direction}(id_2, 798, 99) \]
Experimental Evaluation

▶ Applications & datasets:
  ▶ Activity recognition
    ▶ 28 videos transcribed in logical form.
    ▶ Target events: Two persons moving together, or meeting each other.
  ▶ Maritime Surveillance
    ▶ AIS signals of vessels sailing around the area of Brest, France.
    ▶ 6 months worth of data.
    ▶ Target event: Suspicious vessel Rendezvous.
  ▶ Vehicle fleet management
    ▶ Signals from on-vehicle sensors.
    ▶ 1 month worth of data.
    ▶ Target event: Dangerous driving.
Scalability of MAP Inference (MLN vs. ASP)

- Fixed event pattern set.
- Map inference for weight learning.
- Varying batch sizes.

![Graphs showing scalability comparison between WOLED-ASP and WOLED-MLN for different problem domains: Moving, Vessel Rendezvous, Dangerous Driving.]

- **WOLED-ASP**
  - Clingo
- **WOLED-MLN**
  - LoMRF\(^1\) lib for Markov Logic.
  - lpsolve\(^2\) lib for Integer Linear Programming.

---

\(^1\)https://github.com/anskarl/LoMRF

\(^2\)https://sourceforge.net/projects/lpsolve
## Learning Performance

<table>
<thead>
<tr>
<th>Method</th>
<th>Prequential Loss</th>
<th>$F_1$-score (test set)</th>
<th>Theory size</th>
<th>Inference Time (sec)</th>
<th>Pred. Compl. Time (sec)</th>
<th>Total Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Moving</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WOLED-ASP</td>
<td>1.723</td>
<td>0.821</td>
<td>26</td>
<td>15</td>
<td>–</td>
<td>112</td>
</tr>
<tr>
<td>WOLED-MLN</td>
<td>2.817</td>
<td>0.801</td>
<td>47</td>
<td>187</td>
<td>28</td>
<td>478</td>
</tr>
<tr>
<td>OLED</td>
<td>3.755</td>
<td>0.730</td>
<td>24</td>
<td>13</td>
<td>–</td>
<td>74</td>
</tr>
<tr>
<td>HandCrafted</td>
<td>6.342</td>
<td>0.637</td>
<td>28</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>HandCrafted-WL</td>
<td>4.343</td>
<td>0.702</td>
<td>28</td>
<td>16</td>
<td>–</td>
<td>52</td>
</tr>
<tr>
<td><strong>Meeting</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WOLED-ASP</td>
<td>1.212</td>
<td>0.887</td>
<td>34</td>
<td>12</td>
<td>–</td>
<td>82</td>
</tr>
<tr>
<td>WOLED-MLN</td>
<td>2.554</td>
<td>0.841</td>
<td>56</td>
<td>134</td>
<td>12</td>
<td>145</td>
</tr>
<tr>
<td>OLED</td>
<td>3.224</td>
<td>0.782</td>
<td>42</td>
<td>10</td>
<td>–</td>
<td>36</td>
</tr>
<tr>
<td>HandCrafted</td>
<td>5.734</td>
<td>0.735</td>
<td>23</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>HandCrafted-WL</td>
<td>4.024</td>
<td>0.753</td>
<td>23</td>
<td>13</td>
<td>–</td>
<td>31</td>
</tr>
<tr>
<td><strong>Rendezvous</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WOLED-ASP</td>
<td>0.023</td>
<td>0.98</td>
<td>18</td>
<td>647</td>
<td>–</td>
<td>4,856</td>
</tr>
<tr>
<td>WOLED-MLN</td>
<td>0.088</td>
<td>0.98</td>
<td>18</td>
<td>2,923</td>
<td>434</td>
<td>6,218</td>
</tr>
<tr>
<td>OLED</td>
<td>0.092</td>
<td>0.98</td>
<td>18</td>
<td>623</td>
<td>–</td>
<td>4,688</td>
</tr>
<tr>
<td><strong>Dang.Drive</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WOLED-ASP</td>
<td>0.045</td>
<td>0.99</td>
<td>21</td>
<td>341</td>
<td>–</td>
<td>2,465</td>
</tr>
<tr>
<td>WOLED-MLN</td>
<td>1.234</td>
<td>0.99</td>
<td>28</td>
<td>926</td>
<td>287</td>
<td>3,882</td>
</tr>
<tr>
<td>OLED</td>
<td>1.756</td>
<td>0.99</td>
<td>21</td>
<td>312</td>
<td>–</td>
<td>2,435</td>
</tr>
</tbody>
</table>
Summary

▶ An online structure & weight learner entirely implemented in ASP.
▶ Significantly more efficient & simpler to use than MLN.
  ▶ Single back-end tool – Clingo.
▶ Structure & weight learning tightly coupled.
▶ https://github.com/nkatzzz/ORL

Future work:
▶ Concept drift.
▶ Distributed learning.