Online Learning for Stream Reasoning

Vagelis Michelioudakis, Alexander Artikis and George Paliouras

Institute of Informatics & Telecommunications,
NCSR Demokritos, Athens, Greece

http://cer.iit.demokritos.gr
Complex Event Recognition

INPUT ▶ RECOGNITION ▶ OUTPUT ■

Event Recognition System

CE Definitions

Streams of SDEs

Recognised CEs
Machine Learning for Complex Event Recognition

---

**INPUT** → **CE Definition Construction** ← **INPUT**

- Streams of SDEs
- Annotated CEs

**Machine Learning System**

**CE Definitions**
Online Structure Learning in Markov Logic Networks (OSLα)

Learnt Hypothesis $H_t$:

\[0.51 \text{HoldsAt}(\text{move}(id_1, id_2), t+1) \iff \text{HappensAt}(\text{walking}(id_1), t) \land \text{HappensAt}(\text{walking}(id_2), t)\]

MLN–EC Axioms:

- \(\text{HoldsAt}(f, t+1) \iff \text{InitiatedAt}(f, t)\)
- \(\text{HoldsAt}(f, t+1) \iff \text{HoldsAt}(f, t) \land \neg \text{TerminatedAt}(f, t)\)
- \(\neg \text{HoldsAt}(f, t+1) \iff \text{TerminatedAt}(f, t)\)
- \(\neg \text{HoldsAt}(f, t+1) \iff \neg \text{HoldsAt}(f, t) \land \neg \text{InitiatedAt}(f, t)\)

Data Stream/Training Examples

Micro-Batch $D_t$

- \(\text{HappensAt}(\text{walking}(ID_1), 99)\)
- \(\text{HappensAt}(\text{walking}(ID_2), 99)\)
- \(\text{OrientationMove}(ID_1, ID_2, 99)\)
- \(\text{Close}(ID_1, ID_2, 34, 99)\)
- \(\text{Next}(99, 100)\)
- \(\text{HoldsAt}(\text{move}(ID_1, ID_2), 100)\)

Micro-Batch $D_{t+1}$

- \(\text{HappensAt}(\text{exit}(ID_1), 200)\)
- \(\text{HappensAt}(\text{walking}(ID_2), 200)\)
- \(\neg \text{OrientationMove}(ID_1, ID_2, 200)\)
- \(\neg \text{Close}(ID_1, ID_2, 34, 200)\)
- \(\text{Next}(200, 201)\)
- \(\neg \text{HoldsAt}(\text{move}(ID_1, ID_2), 201)\)

...
# Training Example (Micro-Batch)

<table>
<thead>
<tr>
<th>Simple Derived Events</th>
<th>Complex Event Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>HappensAt(walking(ID₁), 99)</td>
<td></td>
</tr>
<tr>
<td>HappensAt(walking(ID₂), 99)</td>
<td></td>
</tr>
<tr>
<td>OrientationMove(ID₁, ID₂, 99)</td>
<td>HoldsAt(move(ID₁, ID₂), 100)</td>
</tr>
<tr>
<td>Close(ID₁, ID₂, 34, 99)</td>
<td></td>
</tr>
<tr>
<td>Next(99, 100)</td>
<td></td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Hypergraph
Hypergraph and Relational Pathfinding

\[ y_t^P = \neg \text{HoldsAt}(\text{MoveID}_1 \text{ID}_2, 100) \]

\{ \text{HoldsAt}(\text{MoveID}_1 \text{ID}_2, 100), \]
Hypergraph and Relational Pathfinding

\[ y_t^P = (\neg \text{HoldsAt}(\text{MoveID}_1 \text{ID}_2, 100)) \]

\{ \text{HoldsAt}(\text{MoveID}_1 \text{ID}_2, 100), \text{Next}(99, 100), \}
Hypergraph and Relational Pathfinding

\[ y_t^P = \neg \text{HoldsAt}(\text{MoveID}_1 \text{ID}_2, 100) \]

\{ \text{HoldsAt}(\text{MoveID}_1 \text{ID}_2, 100), \text{Next}(99, 100), \text{HappensAt}(\text{WalkingID}_1, 99), \}
Hypergraph and Relational Pathfinding

\[ y_t^P = \neg \text{HoldsAt}(\text{MoveID}_1 \text{ID}_2, 100) \]

\{ \text{HoldsAt}(\text{MoveID}_1 \text{ID}_2, 100), \text{Next}(99, 100), \text{HappensAt}(\text{WalkingID}_1, 99), \text{HappensAt}(\text{WalkingID}_2, 99) \}
Clause Creation, Clause Evaluation and Weight Learning

Clause creation:

▶ Generalize each path into a definite clause

\[
\text{InitiatedAt}(\text{move}(id_1, id_2), t) \iff \\
\text{HappensAt}(\text{walking}(id_1), t) \land \\
\text{HappensAt}(\text{walking}(id_2), t)
\]
Clause Creation, Clause Evaluation and Weight Learning

Clause creation:
- Generalize each path into a definite clause

\[
\text{InitiatedAt}(\text{move}(id_1, id_2), t) \iff \\
\quad \text{HappensAt}(\text{walking}(id_1), t) \land \\
\quad \text{HappensAt}(\text{walking}(id_2), t)
\]

Clause evaluation:
- Keep clauses whose coverage of the annotation is significantly greater than that of the clauses already learnt.
Clause Creation, Clause Evaluation and Weight Learning

Clause creation:

- Generalize each path into a definite clause

\[ \text{InitiatedAt}(\text{move}(id_1, id_2), t) \leftarrow \text{HappensAt}(\text{walking}(id_1), t) \land \text{HappensAt}(\text{walking}(id_2), t) \]

Clause evaluation:

- Keep clauses whose coverage of the annotation is significantly greater than that of the clauses already learnt.

Weight learning:

- Extended clauses inherit initially the weights of their ancestors.
- Optimize the weights of all clauses.
Empirical Evaluation: Activity Recognition

<table>
<thead>
<tr>
<th>Method</th>
<th>F₁ score</th>
<th>Moving together</th>
<th>Meeting</th>
</tr>
</thead>
<tbody>
<tr>
<td>EC&lt;sub&gt;crisp&lt;/sub&gt;</td>
<td>0.7506</td>
<td>0.7620</td>
<td></td>
</tr>
<tr>
<td>MaxMargin</td>
<td>0.8901</td>
<td>0.8629</td>
<td></td>
</tr>
<tr>
<td>CDA</td>
<td>0.7697</td>
<td>0.6342</td>
<td></td>
</tr>
<tr>
<td>AdaGrad</td>
<td>0.7726</td>
<td>0.7833</td>
<td></td>
</tr>
<tr>
<td>OSL</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>OSL&lt;sub&gt;α&lt;/sub&gt;</td>
<td>0.7780</td>
<td>0.8347</td>
<td></td>
</tr>
</tbody>
</table>
## Empirical Evaluation: Activity Recognition

<table>
<thead>
<tr>
<th>Method</th>
<th>$F_1$ score</th>
<th>Moving together</th>
<th>Meeting</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{EC}_{\text{crisp}}$</td>
<td>0.7506</td>
<td>0.7620</td>
<td></td>
</tr>
<tr>
<td>MaxMargin</td>
<td><strong>0.8901</strong></td>
<td><strong>0.8629</strong></td>
<td></td>
</tr>
<tr>
<td>CDA</td>
<td>0.7697</td>
<td>0.6342</td>
<td></td>
</tr>
<tr>
<td>AdaGrad</td>
<td>0.7726</td>
<td>0.7833</td>
<td></td>
</tr>
<tr>
<td>OSL</td>
<td>—</td>
<td>—</td>
<td>—</td>
</tr>
<tr>
<td>OSL$\alpha$</td>
<td>0.7780</td>
<td>0.8347</td>
<td></td>
</tr>
</tbody>
</table>
Traffic Management: Real Data

- **F1 score**
  - Batch size (minutes)
  - #batches
  - Avg. batch processing (seconds)

- **F1 score**
  - Batch size (#sensor events)
  - Avg. batch processing (seconds)
Traffic Management: Synthetic Data

![Graph 1](image1)

- Batch size (minutes)
- #batches
- $F_1$ score

![Graph 2](image2)

- Batch size (minutes)
- #batches
- $F_1$ score
Summary

\(\text{OSL}^\alpha\):  
- Learns CE definitions orders of magnitude faster than OSL.
- Is at least as accurate as event recognition based on weighted manual rules.

Resources:
- https://github.com/anskarl/LoMRF