OSLα: Online Structure Learning using Background Knowledge Axiomatization

Evangelos Michelioudakis[1], Anastasios Skarlatidis[1], Georgios Paliouras[1] and Alexander Artikis[2,1]

[1]Institute of Informatics and Telecommunications, NCSR "Demokritos"
[2]Department of Maritime Studies, University of Piraeus

September 20, 2016

Motivation

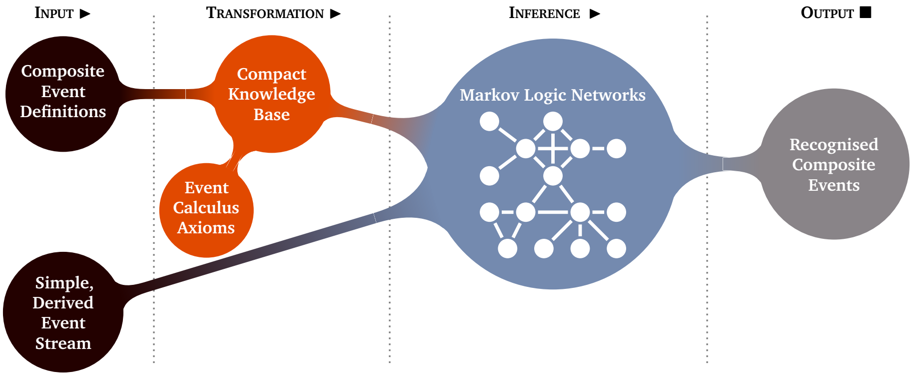
- Targets:
  - Learning in temporal domains
  - Handle uncertainty and complex relational structure
  - Handle large (streaming) training data
- Approach:
  - Markov Logic Networks, Event Calculus (`MLN–EC`)
  - Online structure learning ($\texttt{OSL}\alpha$)
- Starting point: Online Structure Learning (OSL) algorithm
  - ✓ Online strategy for updating the model
  - × Cannot handle a search space having large domain of constants
  - × Does not exploit background knowledge
  - × Does not support first-order logic functions

Running Example: Activity Recognition

- Recognize human activities in multimedia content
- Video frames are annotated by humans

Individual Activities		Complex Activities	
340	enter(id_0)		
340	walking(id_0)		
340	coord(id_0)=(20.88, −11.90)		
340	walking(id_1)		
340	coord(id_1)=(22.88, −14.80)	340	moving(id_0, id_1)
...	...		
345	inactive(id_0)		
345	coord(id_1)=(32.74, −5.24)		
345	exit(id_1)		

MLN–EC: Probabilistic Event Calculus based on MLNs



$\texttt{OSL}\alpha$: Online Structure Learning using Background Knowledge Axiomatizaiton

Learnt Hypothesis $\mathbf{H_t}$:
0.51 $\texttt{HoldsAt}(\texttt{move}(id_1, id_2),\ t{+}1) \Leftarrow$
$\texttt{HappensAt}(\texttt{walking}(id_1),\ t) \wedge$
$\texttt{HappensAt}(\texttt{walking}(id_2),\ t)$

$\texttt{MLN–EC}$ **Axioms**:

$\texttt{HoldsAt}(f,\ t{+}1) \Leftarrow$
$\texttt{InitiatedAt}(f,\ t)$

$\texttt{HoldsAt}(f,\ t{+}1) \Leftarrow$
$\texttt{HoldsAt}(f,\ t) \wedge$
$\neg\texttt{TerminatedAt}(f,\ t)$

$\neg\texttt{HoldsAt}(f,\ t{+}1) \Leftarrow$
$\texttt{TerminatedAt}(f,\ t)$

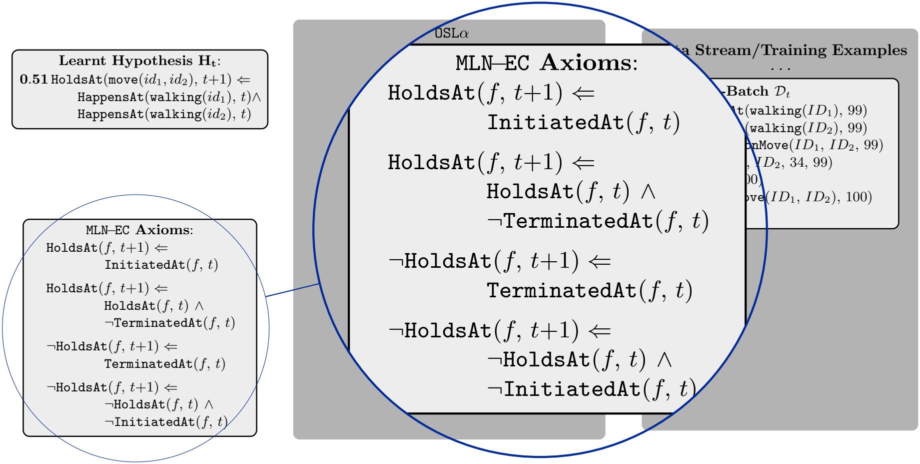
$\neg\texttt{HoldsAt}(f,\ t{+}1) \Leftarrow$
$\neg\texttt{HoldsAt}(f,\ t) \wedge$
$\neg\texttt{InitiatedAt}(f,\ t)$

$\texttt{OSL}\alpha$

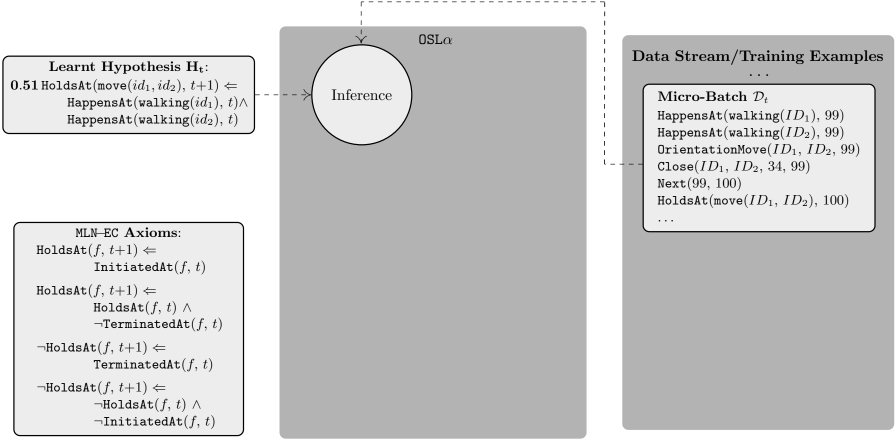
Data Stream/Training Examples
...

Micro-Batch $\mathcal{D}_t$
$\texttt{HappensAt}(\texttt{walking}(ID_1),\ 99)$
$\texttt{HappensAt}(\texttt{walking}(ID_2),\ 99)$
$\texttt{OrientationMove}(ID_1,\ ID_2,\ 99)$
$\texttt{Close}(ID_1,\ ID_2,\ 34,\ 99)$
$\texttt{Next}(99,\ 100)$
$\texttt{HoldsAt}(\texttt{move}(ID_1,\ ID_2),\ 100)$
...

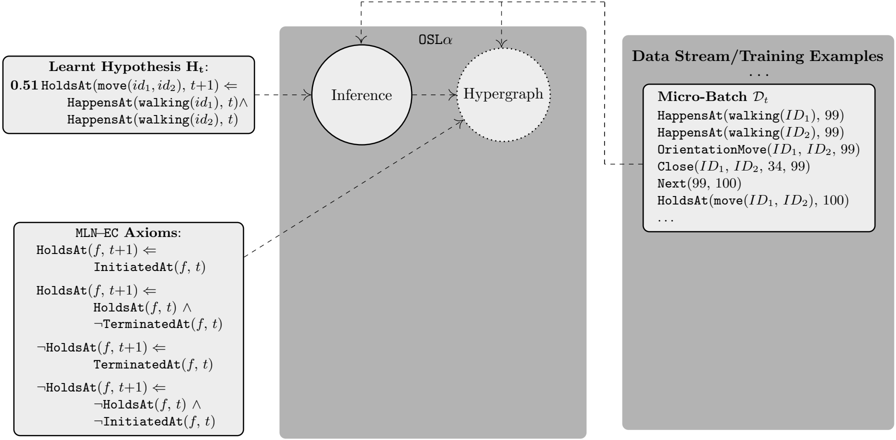
OSLα: Online Structure Learning using Background Knowledge Axiomatizaiton



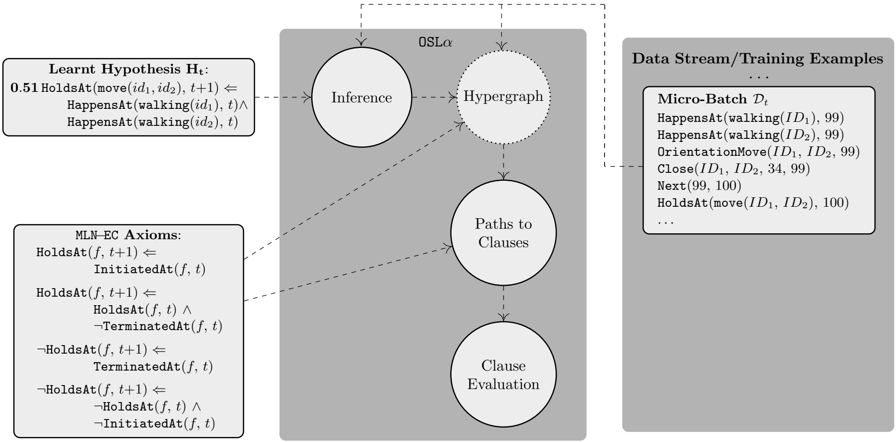
OSLα: Online Structure Learning using Background Knowledge Axiomatizaiton



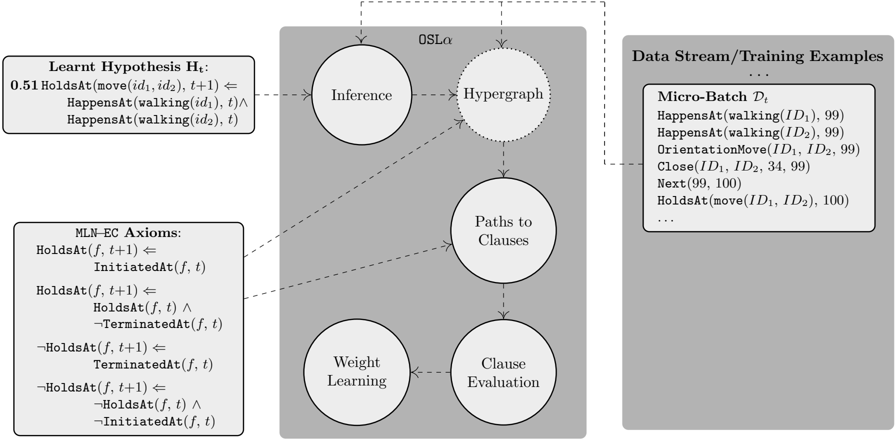
OSLα: Online Structure Learning using Background Knowledge Axiomatizaiton



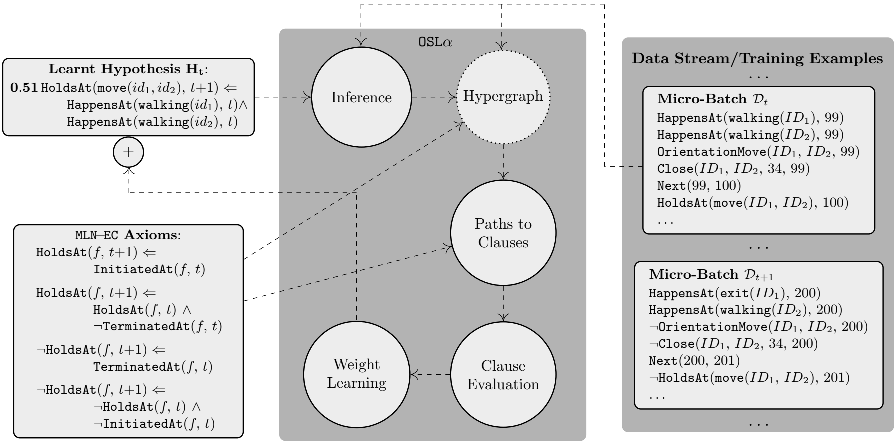
OSLα: Online Structure Learning using Background Knowledge Axiomatizaiton



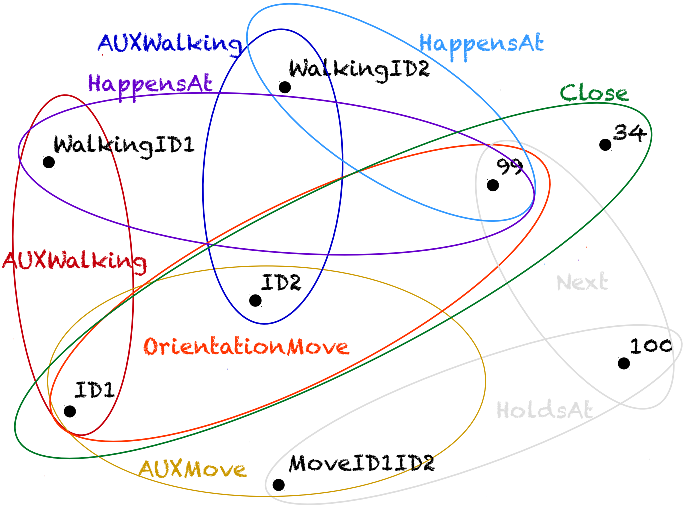
OSLα: Online Structure Learning using Background Knowledge Axiomatizaiton



OSLα: Online Structure Learning using Background Knowledge Axiomatizaiton



Reduced Hypergraph and Relational Pathfinding



Template Guided Search

Wrongly predicted atom:
$\neg$`HoldsAt(MoveID`$_1$`ID`$_2$`, 100)`

+

$\texttt{HoldsAt}(f, t_1) \Leftarrow$
$\quad \texttt{InitiatedAt}(f, t_0) \wedge$
$\quad \texttt{Next}(t_0, t_1)$

Template Guided Search

Wrongly predicted atom:
$\neg\texttt{HoldsAt}(\texttt{MoveID}_1\texttt{ID}_2,\ 100)$

+

$$\texttt{HoldsAt}(f, t_1) \Leftarrow \texttt{InitiatedAt}(f, t_0) \wedge \texttt{Next}(t_0, t_1)$$

=

$$\texttt{HoldsAt}(\texttt{MoveID}_1\texttt{ID}_2, 100) \Leftarrow \texttt{InitiatedAt}(\texttt{MoveID}_1\texttt{ID}_2, t_0) \wedge \texttt{Next}(t_0, 100)$$

Template Guided Search

Wrongly predicted atom:
$\neg\texttt{HoldsAt}(\texttt{MoveID}_1\texttt{ID}_2,\ 100)$

+

$\texttt{HoldsAt}(f, t_1) \Leftarrow$
$\quad \texttt{InitiatedAt}(f, t_0) \wedge$
$\quad \texttt{Next}(t_0, t_1)$

=

$\texttt{HoldsAt}(\texttt{MoveID}_1\texttt{ID}_2, 100) \Leftarrow$
$\quad \texttt{InitiatedAt}(\texttt{MoveID}_1\texttt{ID}_2, t_0) \wedge$
$\quad \texttt{Next}(t_0, 100)$

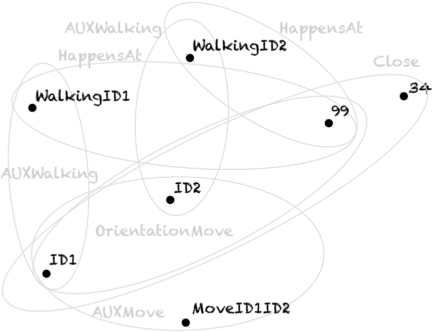
$\texttt{InitiatedAt}(\texttt{MoveID}_1\texttt{ID}_2, 99)$

Hypergraph and Relational Pathfinding

$$\mathbf{y}_t^P = \Big(\neg\mathtt{HoldsAt}(\mathtt{MoveID_1ID_2},\, 100)\Big)$$

$$= \Big(\mathtt{InitiatedAt}(\mathtt{MoveID_1ID_2},\, 99)\Big)$$

$\{\mathtt{InitiatedAt}(\mathtt{MoveID_1ID_2},\, 99),$

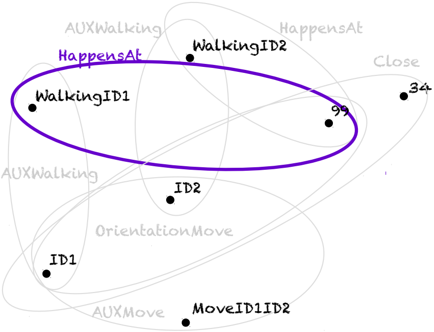


Hypergraph and Relational Pathfinding

$$\mathbf{y}_t^P = \Big(\neg\texttt{HoldsAt}(\texttt{MoveID}_1\texttt{ID}_2, 100)\Big)$$

$$= \Big(\texttt{InitiatedAt}(\texttt{MoveID}_1\texttt{ID}_2, 99)\Big)$$

$\{\texttt{InitiatedAt}(\texttt{MoveID}_1\texttt{ID}_2, 99),$
$\texttt{HappensAt}(\texttt{WalkingID}_1, 99),$



Hypergraph and Relational Pathfinding

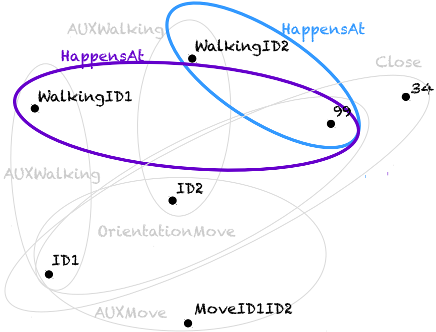
$$\mathbf{y}_t^P = \Big(\neg\texttt{HoldsAt}(\texttt{MoveID}_1\texttt{ID}_2,\ 100)\Big)$$

$$= \Big(\texttt{InitiatedAt}(\texttt{MoveID}_1\texttt{ID}_2,\ 99)\Big)$$

$\{\texttt{InitiatedAt}(\texttt{MoveID}_1\texttt{ID}_2,\ 99),$

$\texttt{HappensAt}(\texttt{WalkingID}_1,\ 99),$

$\texttt{HappensAt}(\texttt{WalkingID}_2,\ 99),$

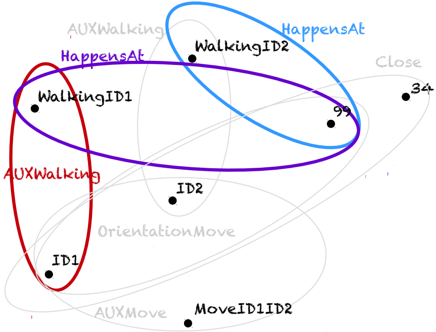


Hypergraph and Relational Pathfinding

$$\mathbf{y}_t^P = \Big(\neg \texttt{HoldsAt}(\texttt{MoveID}_1\texttt{ID}_2, 100)\Big)$$

$$= \Big(\texttt{InitiatedAt}(\texttt{MoveID}_1\texttt{ID}_2, 99)\Big)$$

$\{\texttt{InitiatedAt}(\texttt{MoveID}_1\texttt{ID}_2, 99),$
$\texttt{HappensAt}(\texttt{WalkingID}_1, 99),$
$\texttt{HappensAt}(\texttt{WalkingID}_2, 99),$
$\texttt{AUXwalking}(\texttt{WalkingID}_1, \texttt{ID}_1),$

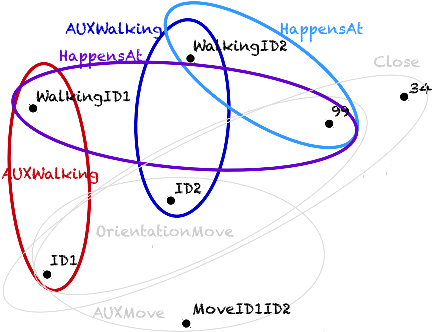


Hypergraph and Relational Pathfinding

$$\mathbf{y}_t^P = \Big(\neg\texttt{HoldsAt}(\texttt{MoveID}_1\texttt{ID}_2,\ 100)\Big)$$

$$= \Big(\texttt{InitiatedAt}(\texttt{MoveID}_1\texttt{ID}_2,\ 99)\Big)$$

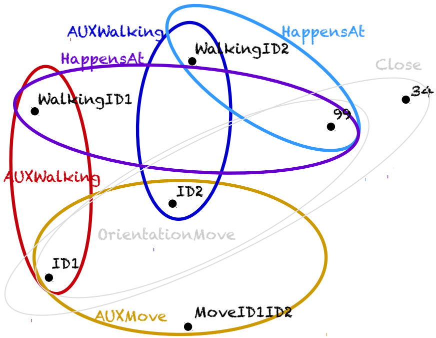
$\{\texttt{InitiatedAt}(\texttt{MoveID}_1\texttt{ID}_2,\ 99),$
$\texttt{HappensAt}(\texttt{WalkingID}_1,\ 99),$
$\texttt{HappensAt}(\texttt{WalkingID}_2,\ 99),$
$\texttt{AUXwalking}(\texttt{WalkingID}_1,\ \texttt{ID}_1),$
$\texttt{AUXwalking}(\texttt{WalkingID}_2,\ \texttt{ID}_2)$



Hypergraph and Relational Pathfinding

$$\mathbf{y}_t^P = \Big(\neg\mathtt{HoldsAt}(\mathtt{MoveID_1ID_2},\ 100)\Big)$$
$$= \Big(\mathtt{InitiatedAt}(\mathtt{MoveID_1ID_2},\ 99)\Big)$$

$\{\mathtt{InitiatedAt}(\mathtt{MoveID_1ID_2},\ 99),$
$\mathtt{HappensAt}(\mathtt{WalkingID_1},\ 99),$
$\mathtt{HappensAt}(\mathtt{WalkingID_2},\ 99),$
$\mathtt{AUXwalking}(\mathtt{WalkingID_1},\ \mathtt{ID_1}),$
$\mathtt{AUXwalking}(\mathtt{WalkingID_2},\ \mathtt{ID_2}),$
$\mathtt{AUXmove}(\mathtt{MoveID_1ID_2},\ \mathtt{ID_1},\ \mathtt{ID_2})\}$



Clause Creation, Clause Evaluation and Weight Learning

- Generalize each path into a definite clause:

$$\begin{aligned}\texttt{InitiatedAt}(\texttt{move}(id_1, id_2), t) \Leftarrow& \\ &\texttt{HappensAt}(\texttt{walking}(id_1), t) \wedge \\ &\texttt{HappensAt}(\texttt{walking}(id_2), t)\end{aligned}$$

- Clause Evaluation:
  - Keep clauses whose coverage of the annotation is significantly greater than that of the clauses already learnt.
- Weight Learning using AdaGrad:
  - Extended clauses inherit initially the weights of their ancestors
  - Optimize the weights of all clauses

Experimental Setup

- Activity recognition using the CAVIAR dataset
  - 28 surveillance videos
- Learn target concepts for `meet` and `move` CEs
- 19 sequences of SDEs and CE annotations
  - The total length of the extracted sequences is 12869 frames
- 10-fold cross-validation
- Implemented on LoMRF[1]
  - Open-source implementation of Markov Logic Networks

[1] http://github.com/anskarl/LoMRF

OSLα Accuracy

Method	meet			move		
	Precision	Recall	F_1-score	Precision	Recall	F_1score
$\mathtt{EC}_{\mathtt{crisp}}$	0.6868	0.8556	0.7620	0.9093	0.6390	0.7506
AdaGrad	0.7228	0.8547	0.7833	0.9172	0.6674	0.7726
MaxMargin	0.9189	0.8133	**0.8629**	0.8443	0.9410	**0.8901**
OSLα	0.8192	0.8509	0.8347	0.8056	0.7522	0.7780

OSLα Runtimes

Method	meet		move	
	training	testing	training	testing
OSLα	**22m 49s**	54s	**1h 56m 2s**	1m 6s
OSL	$>$ 25h	-	-	-

Summary

Conclusions

- Probabilistic online structure learning (`OSL`α) based on MLNs
  - Exploits background knowledge axiomatization (`MLN–EC`)
  - Considers both types of wrongly predicted atoms
  - Supports a subset of first-order logic functions

Future Work

1. Faster hypergraph search
   - Heuristic or randomized (parallel) graph search (e.g., random walks)
2. Learn hierarchical definitions and support negated literals
3. Structure learning in the presence of unobserved data

Any Questions?

Appendix

MAP Inference using Linear Programming

- State-of-the-art method for MAP inference in MLNs

Express Markov network as an optimization problem:

Markov Network		**Linear Programming**
$3\ \texttt{HoldsAt}(CE_A, 5)$	max:	$3y_1+0.5z_1+z_2$
$0.5\ \texttt{HoldsAt}(CE_A, 5) \vee \texttt{HoldsAt}(CE_B, 5)$	st.	$y_1+y_2\geq z_1$
$-1\ \texttt{HoldsAt}(CE_B, 5) \vee \texttt{HoldsAt}(CE_C, 5)$	$\Rightarrow$	$1-y_2\geq z_2,\ 1-y_3\geq z_2$
$\infty\ \neg\texttt{HoldsAt}(CE_A, 5) \vee \neg\texttt{HoldsAt}(CE_B, 5)$		$(1-y_1)+(1-y_2)\geq 1$
$\infty\ \neg\texttt{HoldsAt}(CE_A, 5) \vee \neg\texttt{HoldsAt}(CE_C, 5)$		$(1-y_1)+(1-y_3)\geq 1$
$\infty\ \neg\texttt{HoldsAt}(CE_B, 5) \vee \neg\texttt{HoldsAt}(CE_C, 5)$		$(1-y_2)+(1-y_3)\geq 1$

- Relax the variables domain to be the interval $[0, 1]$
- The solutions are usually non-integral
  - Due to the NP-hardness of the problem
  - Approximate integral solution using a rounding scheme

Max-Margin Weight Learning

- Maximize the ratio of the probability of the **correct world $\mathbf{y}$** to the **closest incorrect world $\widehat{\mathbf{y}}$**:

$$\frac{P(Y{=}\mathbf{y}|X{=}\mathbf{x})}{P(Y{=}\widehat{\mathbf{y}}|X{=}\mathbf{x})}$$

$$\widehat{\mathbf{y}} = \underset{\bar{\mathbf{y}} \in \mathbf{Y} \setminus \mathbf{y}}{\operatorname{argmax}} P(\bar{\mathbf{y}}, \mathbf{x})$$

- By applying the conditional likelihood and taking the log:

$$\log\left(\frac{1/Z(\mathbf{x})\exp(\mathbf{w}^T\mathbf{n}(\mathbf{x},\mathbf{y}))}{1/Z(\mathbf{x})\exp(\mathbf{w}^T\mathbf{n}(\mathbf{x},\hat{\mathbf{y}}))}\right) \Rightarrow$$

$$\log(\exp(\mathbf{w}^T\mathbf{n}(\mathbf{x},\mathbf{y}))) - \log(\exp(\mathbf{w}^T\mathbf{n}(\mathbf{x},\hat{\mathbf{y}}))) \Rightarrow$$

$$\mathbf{w}^T\mathbf{n}(\mathbf{x},\mathbf{y}) - \mathbf{w}^T\mathbf{n}(\mathbf{x},\hat{\mathbf{y}}) = \gamma(\mathbf{x},\mathbf{y};\mathbf{w}) \Rightarrow$$

$$\gamma(\mathbf{x},\mathbf{y};\mathbf{w}) = \mathbf{w}^T\mathbf{n}(\mathbf{x},\,\mathbf{y}) - \max_{\bar{\mathbf{y}} \in \mathcal{Y} \setminus \mathbf{y}} \mathbf{w}^T\mathbf{n}(\mathbf{x},\,\bar{\mathbf{y}})$$

Online Weight Learning

Coordinate Dual Ascent update (CDA)

- Update the weight vector according to the prediction-based loss (difference of the **correct world $\mathbf{y}_t$** and the **predicted world $\mathbf{y}_t^P$**):

$$\Delta\mathbf{n}_t^{PL} = \mathbf{n}_t(\mathbf{x}_t, \mathbf{y}_t) - \mathbf{n}_t(\mathbf{x}_t, \mathbf{y}_t^P)$$

$$\mathbf{w}_{t+1} = \frac{t-1}{t}\mathbf{w}_t + \min\Big\{\frac{1}{\sigma t}, \frac{\ell_{PL}}{||\Delta\mathbf{n}_t^{PL}||_2^2}\Big\}\Delta\mathbf{n}_t^{PL}$$

- CDA learning rate is controlled by the predictive quality

Adaptive Subgradient update (AdaGrad)

- Update the weight vector according to the subgradient of the prediction-based loss ℓ_{PL}:

$$\mathbf{g}_t^{PL} = \mathbf{n}_t(\mathbf{x}_t, \mathbf{y}_t^P) - \mathbf{n}_t(\mathbf{x}_t, \mathbf{y}_t) = -\Delta\mathbf{n}_t \text{ and } H_{t,i} = \delta + ||\mathbf{g}_{1:t,i}||_2$$

$$w_{t+1,i} = \operatorname{sign}\Big(w_{t,i} - \frac{\eta}{H_{t,i}}g_{t,i}\Big)\Big[\Big|w_{t,i} - \frac{\eta}{H_{t,i}}g_{t,i}\Big| - \frac{\lambda\eta}{H_{t,i}}\Big]_+$$

AdaGrad Online Learner

- Prediction-based loss:

$$\ell_{PL} = \left[\rho(\mathbf{y}_t, \mathbf{y}_t^P) - \langle \mathbf{w}_t, \Delta\mathbf{n}_t \rangle\right]_+ =$$
$$\left[\rho(\mathbf{y}_t, \mathbf{y}_t^P) - \langle \mathbf{w}_t, \mathbf{n}_t(\mathbf{x}_t, \mathbf{y}_t) - \mathbf{n}_t(\mathbf{x}_t, \mathbf{y}_t^P) \rangle\right]_+$$
$$\vartheta_{\mathbf{w}_t} \ell_{PL} = \mathbf{n}_t(\mathbf{x}_t, \mathbf{y}_t^P) - \mathbf{n}_t(\mathbf{x}_t, \mathbf{y}_t) = \Delta\mathbf{n}_t$$

- Update rule:

$$H_{t,i} = \delta + ||\mathbf{g}_{1:t,i}||_2 = \delta + \sqrt{\sum_{j=1}^{t} (g_{j,i})^2}$$
$$w_{t+1,i} = \mathrm{sign}\left(w_{t,i} - \frac{\eta}{H_{t,i}} g_{t,i}\right)\left[\left|w_{t,i} - \frac{\eta}{H_{t,i}} g_{t,i}\right| - \frac{\lambda\eta}{H_{t,i}}\right]_+$$

Template Predicate Elimination

$$\begin{aligned}&\texttt{InitiatedAt}(\texttt{move}(id_1, id_2), t) \Leftrightarrow \\ &\qquad \big(\texttt{HappensAt}(\texttt{walking}(id_1), t) \wedge \\ &\qquad \texttt{HappensAt}(\texttt{walking}(id_2), t)\big) \vee \\ &\qquad \big(\texttt{Close}(id_1, id_2, 34, t)\big)\end{aligned}$$

Template Predicate Elimination

$$\texttt{InitiatedAt}(\texttt{move}(id_1, id_2), t) \Leftrightarrow$$
$$\big(\texttt{HappensAt}(\texttt{walking}(id_1), t) \land$$
$$\texttt{HappensAt}(\texttt{walking}(id_2), t)\big) \lor$$
$$\big(\texttt{Close}(id_1, id_2, 34, t)\big)$$

+

$$\texttt{HoldsAt}(f, t{+}1) \Leftarrow$$
$$\texttt{InitiatedAt}(f, t)$$

$$\neg\texttt{HoldsAt}(f, t{+}1) \Leftarrow$$
$$\neg\texttt{HoldsAt}(f, t) \land$$
$$\neg\texttt{InitiatedAt}(f, t)$$

Template Predicate Elimination

$$\mathtt{HoldsAt}(\mathtt{move}(id_1, id_2), t{+}1) \Leftarrow \Big(\mathtt{HappensAt}(\mathtt{walking}(id_1), t) \land \mathtt{HappensAt}(\mathtt{walking}(id_2), t)\Big) \lor \Big(\mathtt{Close}(id_1, id_2, 34, t)\Big)$$

$=$

$$\neg\mathtt{HoldsAt}(\mathtt{move}(id_1, id_2), t{+}1) \Leftarrow \neg\mathtt{HoldsAt}(\mathtt{move}(id_1, id_2), t) \land \neg\Big[\Big(\mathtt{HappensAt}(\mathtt{walking}(id_1), t) \land \mathtt{HappensAt}(\mathtt{walking}(id_2), t)\Big) \lor \Big(\mathtt{Close}(id_1, id_2, 34, t)\Big)\Big]$$

CNF Example

Assume that we have the following formulas:

$$
\begin{aligned}
&\texttt{HoldsAt}(\texttt{move}(id_1, id_2), t{+}1) \Leftarrow \\
&\quad \Big(\texttt{HappensAt}(\texttt{walking}(id_1), t) \wedge \texttt{HappensAt}(\texttt{walking}(id_2), t)\Big) \vee \\
&\quad \Big(\texttt{Close}(id_1, id_2, 34, t)\Big)
\end{aligned}
$$

$$
\begin{aligned}
&\neg\texttt{HoldsAt}(\texttt{move}(id_1, id_2), t{+}1) \Leftarrow \\
&\quad \neg\texttt{HoldsAt}(\texttt{move}(id_1, id_2), t) \wedge \\
&\qquad \neg\Big[\Big(\texttt{HappensAt}(\texttt{walking}(id_1), t) \wedge \texttt{HappensAt}(\texttt{walking}(id_2), t)\Big) \vee \\
&\qquad\quad \Big(\texttt{Close}(id_1, id_2, 34, t)\Big)\Big]
\end{aligned}
$$

CNF Example (1/2)

1. Eliminate all implications and equivalences:

$$\begin{aligned}&\texttt{HoldsAt}(\texttt{move}(id_1, id_2), t{+}1) \lor \\ &\quad \neg\Big[\Big(\texttt{HappensAt}(\texttt{walking}(id_1), t) \land \texttt{HappensAt}(\texttt{walking}(id_2), t)\Big) \lor \\ &\qquad \Big(\texttt{Close}(id_1, id_2, 34, t)\Big)\Big]\end{aligned}$$

$$\begin{aligned}&\neg\texttt{HoldsAt}(\texttt{move}(id_1, id_2), t{+}1) \lor \\ &\quad \neg\Big[\neg\texttt{HoldsAt}(\texttt{move}(id_1, id_2), t) \land \\ &\qquad \neg\Big[\Big(\texttt{HappensAt}(\texttt{walking}(id_1), t) \land \texttt{HappensAt}(\texttt{walking}(id_2), t)\Big) \lor \\ &\qquad\quad \Big(\texttt{Close}(id_1, id_2, 34, t)\Big)\Big]\Big]\end{aligned}$$

2. Move inwards all negations:

$$\begin{aligned}&\texttt{HoldsAt}(\texttt{move}(id_1, id_2), t{+}1) \lor \\ &\quad \Big[\Big(\neg\texttt{HappensAt}(\texttt{walking}(id_1), t) \lor \neg\texttt{HappensAt}(\texttt{walking}(id_2), t)\Big) \land \\ &\qquad \neg\texttt{Close}(id_1, id_2, 34, t)\Big]\end{aligned}$$

$$\begin{aligned}&\neg\texttt{HoldsAt}(\texttt{move}(id_1, id_2), t{+}1) \lor \\ &\quad \texttt{HoldsAt}(\texttt{move}(id_1, id_2), t) \lor \\ &\qquad \Big[\Big(\texttt{HappensAt}(\texttt{walking}(id_1), t) \land \texttt{HappensAt}(\texttt{walking}(id_2), t)\Big) \lor \\ &\qquad\quad \texttt{Close}(id_1, id_2, 34, t)\Big]\end{aligned}$$

CNF Example (2/2)

3. Distribute operators (disjunctions and conjunctions):

$$\texttt{HoldsAt}(\texttt{move}(id_1, id_2), t{+}1) \vee \neg\texttt{HappensAt}(\texttt{walking}(id_1), t) \vee \neg\texttt{HappensAt}(\texttt{walking}(id_2), t) \wedge \texttt{HoldsAt}(\texttt{move}(id_1, id_2), t{+}1) \vee \neg\texttt{Close}(id_1, id_2, 34, t)$$

$$\neg\texttt{HoldsAt}(\texttt{move}(id_1, id_2), t{+}1) \vee \texttt{HoldsAt}(\texttt{move}(id_1, id_2), t) \vee \texttt{HappensAt}(\texttt{walking}(id_1), t) \vee \texttt{Close}(id_1, id_2, 34, t) \wedge \neg\texttt{HoldsAt}(\texttt{move}(id_1, id_2), t{+}1) \vee \texttt{HoldsAt}(\texttt{move}(id_1, id_2), t) \vee \texttt{HappensAt}(\texttt{walking}(id_2), t) \vee \texttt{Close}(id_1, id_2, 34, t)$$

4. Extract clauses:

$$\{\texttt{HoldsAt}(\texttt{move}(id_1, id_2), t{+}1) \vee \neg\texttt{HappensAt}(\texttt{walking}(id_1), t) \vee \neg\texttt{HappensAt}(\texttt{walking}(id_2), t)\}$$

$$\{\texttt{HoldsAt}(\texttt{move}(id_1, id_2), t{+}1) \vee \neg\texttt{Close}(id_1, id_2, 34, t)\}$$

$$\{\neg\texttt{HoldsAt}(\texttt{move}(id_1, id_2), t{+}1) \vee \texttt{HoldsAt}(\texttt{move}(id_1, id_2), t) \vee \texttt{HappensAt}(\texttt{walking}(id_1), t) \vee \texttt{Close}(id_1, id_2, 34, t)\}$$

$$\{\neg\texttt{HoldsAt}(\texttt{move}(id_1, id_2), t{+}1) \vee \texttt{HoldsAt}(\texttt{move}(id_1, id_2), t) \vee \texttt{HappensAt}(\texttt{walking}(id_2), t) \vee \texttt{Close}(id_1, id_2, 34, t)\}$$

Dataset Statistics

total SDEs	63147
average SDEs per fold	56832
total `meet` positive CEs	3722
total `move` positive CEs	6272
average `meet` positive CEs	3350
average `move` positive CEs	5600

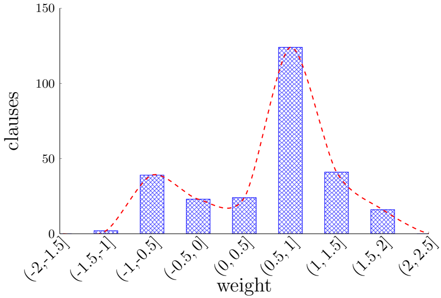
Weight Learning Accuracy

Method	meet			move		
	Precision	Recall	F_1-score	Precision	Recall	F_1score
$\texttt{EC}_{\texttt{crisp}}$	0.6868	**0.8556**	0.7620	0.9093	0.6390	0.7506
CDA	0.9061	0.4878	0.6342	0.9032	0.6706	0.7697
AdaGrad	0.7228	0.8547	0.7833	**0.9172**	0.6674	0.7726
MaxMargin	**0.9189**	0.8133	**0.8629**	0.8443	**0.9410**	**0.8901**

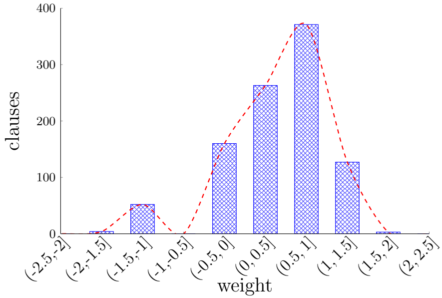
Weight Learning Runtimes

Method	meet		move	
	training	testing	training	testing
`CDA`	**1m 24s**	11s	1m 44s	13s
`AdaGrad`	1m 26s	11s	**1m 35s**	15s
`MaxMargin`	18m 53s	11s	28m 10s	12s

Learned Weight Distribution

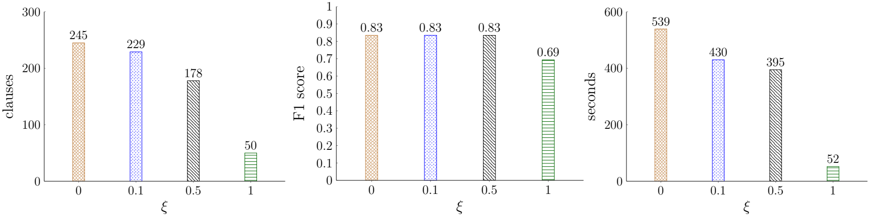


(a) `meet`

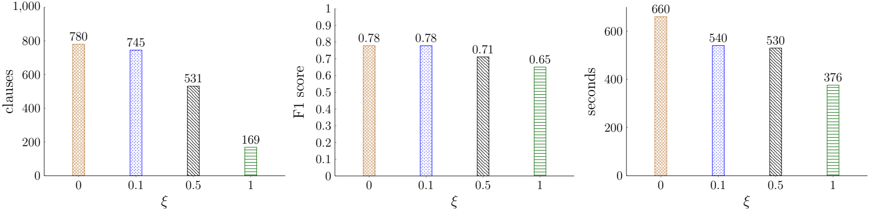


(b) `move`

Model Pruning using ξ Threshold



(c) `meet`



(d) `move`