Imperial College London

Iterative Learning of Answer Set Programs from Context Dependent Examples

Mark Law, Alessandra Russo and Krysia Broda

September 4, 2016

Context-dependent examples

- In standard ILP, we search for hypotheses H such that:
  - $\forall e \in E^{+}\ B \cup H \models e$
  - $\forall e \in E^{-}\ B \cup H \not\models e$
- Given *context-dependent examples*, it must be the case that:
  - $\forall \langle e, C \rangle \in E^{+}\ B \cup H \cup C \models e$
  - $\forall \langle e, C \rangle \in E^{-}\ B \cup H \cup C \not\models e$.

Context-dependent examples

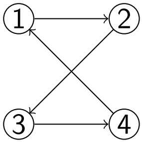
- In standard ILP, we search for hypotheses H such that:
  - $\forall e \in E^{+} \; B \cup H \models e$
  - $\forall e \in E^{-} \; B \cup H \not\models e$
- Given *context-dependent examples*, it must be the case that:
  - $\forall \langle e, C \rangle \in E^{+} \; B \cup H \cup C \models e$
  - $\forall \langle e, C \rangle \in E^{-} \; B \cup H \cup C \not\models e$.

For example, we may wish to learn that when it is raining a user prefers to take the bus; otherwise, they prefer to walk.

$$E^{+} = \left\{ \begin{array}{l} \langle \text{“take bus”}, \{\texttt{rain.}\} \rangle, \\ \langle \text{“walk”}, \{\} \rangle \end{array} \right. \quad E^{-} = \left\{ \begin{array}{l} \langle \text{“walk”}, \{\texttt{rain.}\} \rangle, \\ \langle \text{“take bus”}, \{\} \rangle \end{array} \right.$$

Learning from Answer Sets (ILP_{LAS})

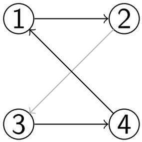
- In ILP_{LAS} (Law et al. 2014), examples are *partial interpretations*.
- A partial interpretation e is a set of pairs of atoms $\langle e^{inc}, e^{exc} \rangle$.



$$\left\langle \left\{ \begin{array}{l} \texttt{size}(4) \\ \texttt{edge}(1,2) \\ \texttt{edge}(2,3) \\ \texttt{edge}(3,4) \\ \texttt{edge}(4,1) \end{array} \right\}, \left\{ \begin{array}{c} \texttt{edge}(1,1) \\ \texttt{edge}(1,3) \\ \texttt{edge}(1,4) \\ \ldots \end{array} \right\} \right\rangle$$

Learning from Answer Sets (ILP_{LAS})

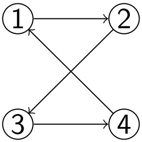
- In ILP_{LAS} (Law et al. 2014), examples are *partial interpretations*.
- A partial interpretation e is a set of pairs of atoms $\langle e^{inc}, e^{exc} \rangle$.



$$\left\langle \left\{ \begin{array}{l} \texttt{size}(4) \\ \texttt{edge}(1,2) \\ \texttt{edge}(2,3) \\ \texttt{edge}(3,4) \\ \texttt{edge}(4,1) \end{array} \right\}, \left\{ \begin{array}{c} \texttt{edge}(1,1) \\ \texttt{edge}(1,3) \\ \texttt{edge}(1,4) \\ \ldots \end{array} \right\} \right\rangle$$

Learning from Answer Sets (ILP_{LAS})

- In ILP_{LAS} (Law et al. 2014), examples are *partial interpretations*.
- A partial interpretation e is a set of pairs of atoms $\langle e^{inc}, e^{exc} \rangle$.

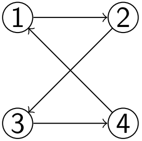


$$\left\langle \left\{ \begin{array}{l} \texttt{size}(4) \\ \texttt{edge}(1,2) \\ \texttt{edge}(2,3) \\ \texttt{edge}(3,4) \\ \texttt{edge}(4,1) \end{array} \right\}, \left\{ \begin{array}{c} \texttt{edge}(1,1) \\ \texttt{edge}(1,3) \\ \texttt{edge}(1,4) \\ \ldots \end{array} \right\} \right\rangle$$

- An answer set A extends e iff $e^{inc} \subseteq A$ and $e^{exc} \cap A = \emptyset$.
- A positive (resp. negative) example e is covered if at least one (resp. no) answer set of $B \cup H$ extends e.

Imperial College London

ILP_{LAS} Encoding of the Hamiltonian Example



$$\left\langle \left\{ \begin{array}{l} \texttt{size}(4) \\ \texttt{edge}(1,2) \\ \texttt{edge}(2,3) \\ \texttt{edge}(3,4) \\ \texttt{edge}(4,1) \end{array} \right\}, \left\{ \begin{array}{c} \texttt{edge}(1,1) \\ \texttt{edge}(1,3) \\ \texttt{edge}(1,4) \\ \dots \end{array} \right\} \right\rangle$$

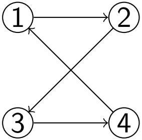
B :

$\texttt{1\{size(1..4)\}1.}$
$\texttt{node(1..N):-size(N).}$
$\texttt{0\{edge(V0,V1)\}1:-node(V0),}$
$\texttt{node(V1).}$

H :

$\texttt{reach(V0):-in(1,V0).}$
$\texttt{reach(V1):-in(V0,V1),reach(V0).}$
$\texttt{0\{in(V0,V1)\}1:-edge(V0,V1).}$
$\texttt{:-node(V0),not reach(V0).}$
$\texttt{:-in(V0,V1),in(V0,V2),V1} \neq \texttt{V2.}$

Context-dependent Hamiltonian Example



$\left\langle \langle \emptyset, \emptyset \rangle, \left\{ \begin{array}{l} \texttt{node(1..4).} \\ \texttt{edge(1,2).} \\ \texttt{edge(2,3).} \\ \texttt{edge(3,4).} \\ \texttt{edge(4,1).} \end{array} \right\} \right\rangle$

B :

None!

H :

```
reach(V0):-in(1,V0).
reach(V1):-in(V0,V1),reach(V0).
0{in(V0,V1)}1:-edge(V0,V1).
:-node(V0),not reach(V0).
:-in(V0,V1),in(V0,V2),V1 ≠ V2.
```

Journey Preferences in ASP

$$H = \left\{ \begin{array}{l} :\sim \texttt{mode(L, walk), crime\_rating(L, R), R} > \texttt{3.[1@3, L, R]} \\ :\sim \texttt{mode(L, bus).[1@2, L]} \\ :\sim \texttt{mode(L, walk), distance(L, D).[D@1, L, D]} \end{array} \right.$$

Journey Preferences in ASP

$$H = \left\{ \begin{array}{l} :\sim \texttt{mode(L, walk), crime\_rating(L, R), R} > \texttt{3.[1@3, L, R]} \\ :\sim \texttt{mode(L, bus).[1@2, L]} \\ :\sim \texttt{mode(L, walk), distance(L, D).[D@1, L, D]} \end{array} \right.$$

ILP_{LOAS} (Law et al. 2015) is an extension of ILP_{LAS} with *ordering examples* of the form $\langle e_1, e_2 \rangle$.

Journey Preferences in ASP

$$H = \left\{ \begin{array}{l} \texttt{:}\sim \texttt{mode(L, walk), crime\_rating(L, R), R} > \texttt{3.[1@3, L, R]} \\ \texttt{:}\sim \texttt{mode(L, bus).[1@2, L]} \\ \texttt{:}\sim \texttt{mode(L, walk), distance(L, D).[D@1, L, D]} \end{array} \right.$$

$$B = \left\{ \begin{array}{l} \texttt{1\{choose(j}_1\texttt{)}, \ldots, \texttt{choose(j}_n\texttt{)\}1.} \\ \texttt{mode(leg1, walk):-choose(j}_1\texttt{).} \\ \texttt{crime\_rating(leg1, 2):-choose(j}_1\texttt{).} \\ \texttt{distance(leg1, 1000):-choose(j}_1\texttt{).} \\ \qquad\qquad \ldots \end{array} \right.$$

$$e_1 = \langle \{\texttt{choose(j}_1\texttt{)}\}, \emptyset \rangle, \qquad e_2 = \langle \{\texttt{choose(j}_2\texttt{)}\}, \emptyset \rangle, \qquad \ldots$$

$$O^b = \left\{ \begin{array}{c} \langle e_1, e_2 \rangle \\ \ldots \end{array} \right\}$$

Journey Preferences in ASP

$$H = \left\{ \begin{array}{l} :\sim \texttt{mode(L, walk), crime\_rating(L, R), R} > \texttt{3.[1@3, L, R]} \\ :\sim \texttt{mode(L, bus).[1@2, L]} \\ :\sim \texttt{mode(L, walk), distance(L, D).[D@1, L, D]} \end{array} \right.$$

$$B = \{ \quad \textit{None}!$$

$$e_1 = \langle \langle \emptyset, \emptyset \rangle, \left\{ \begin{array}{l} \texttt{mode(leg1, walk).} \\ \texttt{crime\_rating(leg1, 2).} \\ \texttt{distance(leg1, 1000).} \end{array} \right\} \rangle \qquad \ldots$$

$$O^b = \left\{ \begin{array}{c} \langle e_1, e_2 \rangle \\ \ldots \end{array} \right\}$$

Complexity

- In the paper, we present a mapping $\mathcal{T}_{LOAS}$ from any $ILP^{context}_{LOAS}$ task to an ILP_{LOAS} task.

Theorem 1

For any $ILP^{context}_{LOAS}$ task T, $ILP_{LOAS}(\mathcal{T}_{LOAS}(T)) = ILP^{context}_{LOAS}(T)$.

Theorem 2

The complexity of deciding whether an $ILP^{context}_{LOAS}$ task is satisfiable is Σ^P_2-complete.

ILASP2i

- The mapping $\mathcal{T}_{LOAS}$ means that we can use ILASP2 to compute solutions for any context dependent task:
  - This would be by calling $ILASP2(\mathcal{T}_{LOAS}(\langle B, S_M, E \rangle))$.
  - However, ILASP2 is known to scale poorly wrt the number of examples.
- Our new algorithm, ILASP2i, iteratively computes a subset of the examples *Rel*, called *relevant examples*.
  - In each iteration, we call $ILASP2(\mathcal{T}_{LOAS}(\langle B, S_M, Rel \rangle))$.

Theorem 4

ILASP2i is sound for any well defined $ILP_{LOAS}^{context}$ task, and returns an optimal solution if one exists.

Benchmarks

Learning task	#examples				time/s		Memory/kB	
	E^+	E^-	O^b	O^c	2	2i	2	2i
Hamilton A (no context)	100	100	0	0	10.3	**4.3**	9.7×10^4	$\mathbf{1.2\times10^4}$
Hamilton B (context dep.)	100	100	0	0	32.0	**3.6**	3.6×10^5	$\mathbf{1.4\times10^4}$
Journeys (context dep.)	386	0	200	0	1031.4	**5.0**	1.4×10^7	$\mathbf{3.4\times10^4}$

- ILASP2 runs the automatic translation ($\mathcal{T}_{LOAS}$) of context dependent tasks.
- $\mathcal{T}_{LOAS}$(`Hamilton B`) is less efficient than `Hamilton A`.
- $\mathcal{T}_{LOAS}$(`Journeys`) is the same as the non-context dependent Journey task.

Related work under the answer set semantics

Learning Task	Normal Rules	Choice Rules	Constraints	Classical Negation	Brave	Cautious	Weak Constraints	Context	Algorithm for optimal solutions
Brave Induction [Sakama, Inoue 2009]	✔	✔	✖	✔	✔	✖	✖	✖	✖
Cautious Induction [Sakama, Inoue 2009]	✔	✔	✖	✔	✖	✔	✖	✖	✖
XHAIL [Ray 2009] & ASPAL [Corapi et al 2011]	✔	✖	✖	✖	✔	✖	✖	✖	✔
Induction of Stable Models [Otero 2001]	✔	✖	✖	✖	✔	✖	✖	✖	✖
Induction from Answer Sets [Sakama 2005]	✔	✖	✔	✔	✔	✔	✖	✖	✖
LAS [Law et al 2014]	✔	✔	✔	✖	✔	✔	✖	✖	✔
LOAS [Law et al 2015]	✔	✔	✔	✖	✔	✔	✔	✖	✔
Context Dependent LOAS	✔	✔	✔	✖	✔	✔	✔	✔	✔

Current Work

- Improve the scalability of ILASP for tasks with:
  - Noisy examples
  - Large hypothesis spaces

Current Work

- Improve the scalability of ILASP for tasks with:
  - Noisy examples
  - Large hypothesis spaces

- ILASP2 and ILASP2i are available to download from `https://www.doc.ic.ac.uk/~ml1909/ILASP`

Context-dependent examples

- In standard ILP, we search for hypotheses H such that:
  - $\forall e \in E^{+} \; B \cup H \models e$
  - $\forall e \in E^{-} \; B \cup H \not\models e$
- Given *context-dependent examples*, it must be the case that:
  - $\forall \langle e, C \rangle \in E^{+} \; B \cup H \cup C \models e$
  - $\forall \langle e, C \rangle \in E^{-} \; B \cup H \cup C \not\models e$.

For example, we may wish to learn that when it is raining a user prefers to take the bus; otherwise, they prefer to walk.

$$E^{+} = \left\{ \begin{array}{l} \langle \text{“take bus”}, \{\mathtt{1\{rain, snow\}1.}\} \rangle, \\ \langle \text{“walk”}, \{\} \rangle \end{array} \right.$$

$$E^{-} = \left\{ \begin{array}{l} \langle \text{“walk”}, \{\mathtt{rain.}\} \rangle, \\ \langle \text{“take bus”}, \{\} \rangle \end{array} \right.$$

ILASP2i

procedure ILASP2I($\langle B, S_M, E\rangle$)
    $Relevant = \langle \emptyset, \emptyset, \emptyset, \emptyset \rangle$; $H = \emptyset$;
    $re = findRelevantExample(\langle B, S_M, E\rangle, H)$;
    while $re \neq$ `nil` **do**
        $Relevant << re$;
        $H = ILASP2(\mathcal{T}_{LOAS}(\langle B, S_M, Relevant\rangle))$;
        if($H ==$ `nil`) **return** `UNSATISFIABLE`;
        else $re = findRelevantExample(\langle B, S_M, E\rangle, H)$;
    end while
    return H;

Theorem 4

ILASP2i is sound for any well defined $ILP_{LOAS}^{context}$ task, and returns an optimal solution if one exists.

Journey Preference Experiments

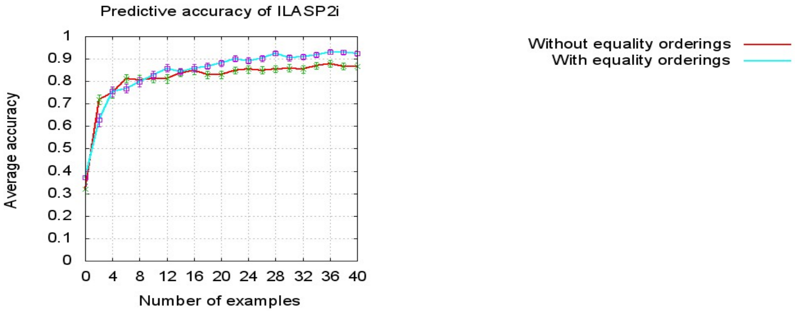


Figure : average accuracy of ILASP2i

Journey Preference Experiments

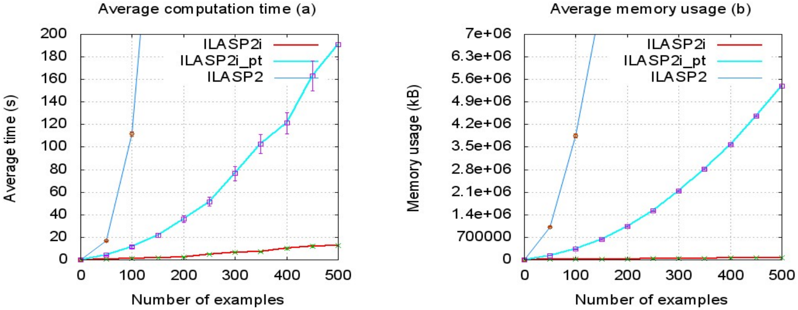


Figure : (a) the average computation time and (b) the memory usage of ILASP2, ILASP2i and ILASP2i_pt for learning journey preferences.

Hamilton Experiment

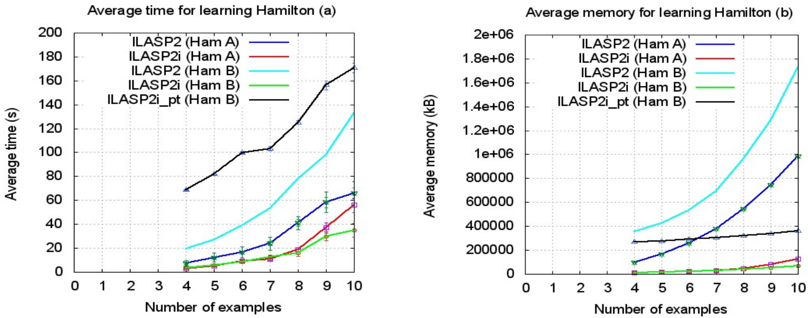


Figure : (a) the average computation time and (b) the memory usage of ILASP2, ILASP2i and ILASP2i_pt for Hamilton A and B.

Imperial College London

Corapi, D., Russo, A., and Lupu, E. 2012.
Inductive logic programming in answer set programming.
In *Inductive Logic Programming*. Springer, 91–97.

Law, M., Russo, A., and Broda, K. 2014.
Inductive learning of answer set programs.
In *Logics in Artificial Intelligence (JELIA 2014)*. Springer.

Law, M., Russo, A., and Broda, K. 2015.
Learning weak constraints in answer set programming.
Theory and Practice of Logic Programming 15, 4-5, 511–525.

Otero, R. P. 2001.
Induction of stable models.
In *Inductive Logic Programming*. Springer, 193–205.

Ray, O. 2009.

Imperial College London

Nonmonotonic abductive inductive learning.
Journal of Applied Logic 7, 3, 329–340.

SAKAMA, C. AND INOUE, K. 2009.
Brave induction: a logical framework for learning from incomplete information.
Machine Learning 76, 1, 3–35.