Iterative Learning of Answer Set Programs from Context Dependent Examples

Mark Law, Alessandra Russo and Krysia Broda

October 21, 2016

Imperial College London

Inductive Logic Programming

- Given a set of positive examples E^+, negative examples E^- and a background knowledge B, the goal is to find a hypothesis H such that:
  - $\forall e \in E^+ : B \cup H \models e$
  - $\forall e \in E^- : B \cup H \not\models e$

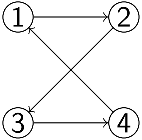
Inductive Logic Programming

- Given a set of positive examples E^+, negative examples E^- and a background knowledge B, the goal is to find a hypothesis H such that:
  - $\forall e \in E^+ : B \cup H \models e$
  - $\forall e \in E^- : B \cup H \not\models e$
- The key advantages are that:
  - The hypotheses are human readable.
  - Can define useful concepts in the background knowledge.
  - Can give a very structured language bias to guide the search.

Imperial College London

Learning from Answer Sets (ILP_{LAS})

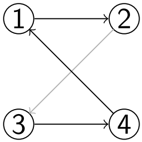
- In ILP_{LAS} (Law et al. 2014), examples are *partial interpretations*.
- A partial interpretation e is a set of pairs of atoms $\langle e^{inc}, e^{exc} \rangle$.



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

Learning from Answer Sets (ILP_{LAS})

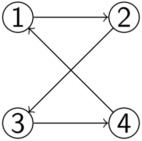
- In ILP_{LAS} (Law et al. 2014), examples are *partial interpretations*.
- A partial interpretation e is a set of pairs of atoms $\langle e^{inc}, e^{exc} \rangle$.



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

Learning from Answer Sets (ILP_{LAS})

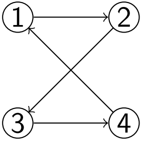
- In ILP_{LAS} (Law et al. 2014), examples are *partial interpretations*.
- A partial interpretation e is a set of pairs of atoms $\langle e^{inc}, e^{exc} \rangle$.



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

- 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.

ILP_{LAS} Encoding of the Hamiltonian Example



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

B :

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

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.
```

ILP_{LOAS}

ILP_{LOAS} (Law et al. 2015) is a generalisation of ILP_{LAS} which enables the learning of weak constraints.

Definition

An ordering example o is a pair $\langle e_1, e_2 \rangle$. *A program P is said to bravely (resp. cautiously) respect o if for at least one (resp. every) pair* $\langle A_1, A_2 \rangle$ *such that* $A_1, A_2 \in AS(P)$, A_1 *extends* e_1 *and* A_2 *extends* e_2, *it is the case that* $A_1 \prec_P A_2$.

ILP_{LOAS}

Definition

An ILP_{LOAS} task is a tuple $T = \langle B, S_M, E^+, E^-, O^b, O^c \rangle$.
A hypothesis $H \subseteq S_M$ is in $ILP_{LOAS}(T)$, the set of all inductive solutions of T, if and only if:

- $\forall e \in E^+ \; \exists A \in AS(B \cup H)$ *such that* A *extends* e
- $\forall e \in E^- \; \nexists A \in AS(B \cup H)$ *such that* A *extends* e
- $\forall o \in O^b \;\; B \cup H$ *bravely respects* o
- $\forall o \in O^c \;\; B \cup H$ *cautiously respects* o

Journey Preferences

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

Journey A	Journey B	Journey C	Journey D
• Walk 400m through an area with crime rating of 2. • Take the bus 3km through an area with crime rating 4.	• Take the bus 4km through an area with crime rating of 2 • Walk 1km through an area with crime rating 5.	• Take the bus 400m through an area with crime rating of 2. • Take a second bus 3km through an area with crime rating 4	• Take a bus 2km through an area with crime rating 5. • Walk 2km through an area with crime rating 1.

Journey A > Journey D > Journey C > Journey B

Learning Journey Preferences

Journey A

- Walk 400m through an area with crime rating of 2.
- Take the bus 3km through an area with crime rating 4.

>

Journey B

- Take the bus 4km through an area with crime rating of 2
- Walk 1km through an area with crime rating 5.

Learning Journey Preferences

Journey A

- Walk 400m through an area with crime rating of 2.
- Take the bus 3km through an area with crime rating 4.

>

Journey B

- Take the bus 4km through an area with crime rating of 2
- Walk 1km through an area with crime rating 5.

- Given examples of this form, we can learn:

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

Journey Preferences in ILP_{LOAS}

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

$$B = \left\{ \begin{array}{l} \mathtt{1\{choose(j_1), \ldots, choose(j_n)\}1.} \\ \mathtt{mode(leg1, walk) :\!- choose(j_1).} \\ \mathtt{crime\_rating(leg1, 2) :\!- choose(j_1).} \\ \mathtt{distance(leg1, 1000) :\!- choose(j_1).} \\ \ldots \end{array} \right.$$

$e_1 = \langle\{\mathtt{choose(j_1)}\}, \emptyset\rangle, \quad e_2 = \langle\{\mathtt{choose(j_2)}\}, \emptyset\rangle, \quad \ldots$

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

Imperial College
London

Journey Preference Experiments

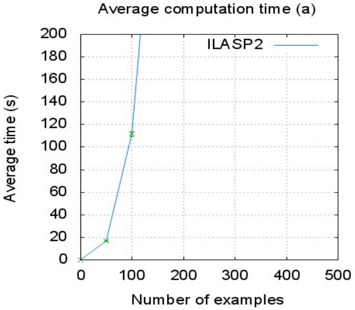


Figure: (a) the average computation time and (b) the memory usage of ILASP2 for learning journey preferences.

Reason for Scalability Issues

- The background knowledge contains all the attributes of each journey
- Can we divide this background knowledge into pieces that only apply for particular examples?

Imperial College London

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:
  - $\forall \langle e, C\rangle \in E^{+}\ B \cup H \cup C \models e$
  - $\forall \langle e, C\rangle \in E^{-}\ B \cup H \cup C \not\models e$.

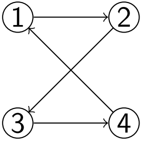
Context-dependent examples

- 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.

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

ILP_{LAS} Encoding of the Hamiltonian Example



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

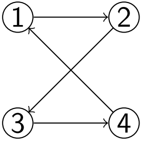
B :

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

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.
```


Context-dependent Hamiltonian Example



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

B :

None!

H :

$\texttt{reach(V0):-in(1,V0).}$

$\texttt{reach(V1):-in(V0,V1),reach(V0).}$

$\texttt{0\{in(V0,V1)\}1:-edge(V0,V1).}$

$\texttt{:-node(V0),not reach(V0).}$

$\texttt{:-in(V0,V1),in(V0,V2),V1} \neq \texttt{V2.}$

Journey Preferences in ILP_{LOAS}

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

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

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

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

Imperial College London

Journey Preferences in $ILP_{LOAS}^{context}$

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

$B = \{$ *None*!

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

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

Imperial College London

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 $ILASP2(\mathcal{T}_{LOAS}(\langle B, S_M, Rel\rangle))$.

ILASP2i_pt

procedure $\text{ILASP2I\_PT}(\langle B, S_M, E\rangle)$

$\quad\langle B', S'_M, E'\rangle = \mathcal{T}_{LOAS}(\langle B, S_M, E\rangle)$;

$\quad Relevant = \langle\emptyset, \emptyset, \emptyset, \emptyset\rangle$; $\quad H = \emptyset$;

$\quad re = findRelevantExample(\langle B', S'_M, E'\rangle, H)$;

$\quad$**while** $re \neq \texttt{nil}$ **do**

$\quad\quad Relevant << re$;

$\quad\quad H = ILASP2(\langle B', S'_M, Relevant\rangle)$;

$\quad\quad$**if**$(H == \texttt{nil})$ **return** $\texttt{UNSATISFIABLE}$;

$\quad\quad$**else** $re = findRelevantExample(\langle B', S'_M, E\rangle, H)$;

$\quad$**end while**

$\quad$**return** H;

ILASP2i_pt

procedure ILASP2I_PT($\langle B, S_M, E \rangle$)

    $\langle B', S'_M, E' \rangle = \mathcal{T}_{LOAS}(\langle B, S_M, E \rangle)$;

    $Relevant = \langle \emptyset, \emptyset, \emptyset, \emptyset \rangle$; $H = \emptyset$;

    $re = findRelevantExample(\langle B', S'_M, E' \rangle, H)$;

    while $re \neq$ `nil` **do**

        $Relevant << re$;

        $H = ILASP2(\langle B', S'_M, Relevant \rangle)$;

        if($H ==$ `nil`) **return** `UNSATISFIABLE`;

        else $re = findRelevantExample(\langle B', S'_M, E \rangle, H)$;

    end while

    return H;

Translation occurs once, at the start of the algorithm.

Journey Preference Experiments

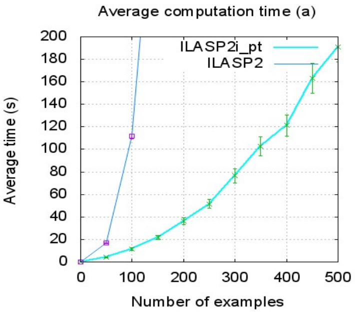


Figure: (a) the average computation time and (b) the memory usage of ILASP2 and ILASP2i_pt for learning journey preferences.

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;

Translation occurs in each iteration, using only the *relevant* contexts.

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 $ILP^{context}_{LOAS}$ *task, and returns an optimal solution if one exists.*

Imperial College
London

Journey Preference Experiments

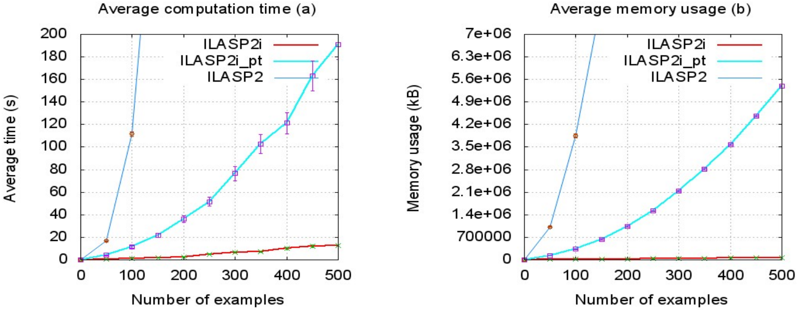


Figure: (a) the average computation time and (b) the memory usage of ILASP2, ILASP2i and ILASP2i_pt for learning journey preferences.

Journey Preference Experiments

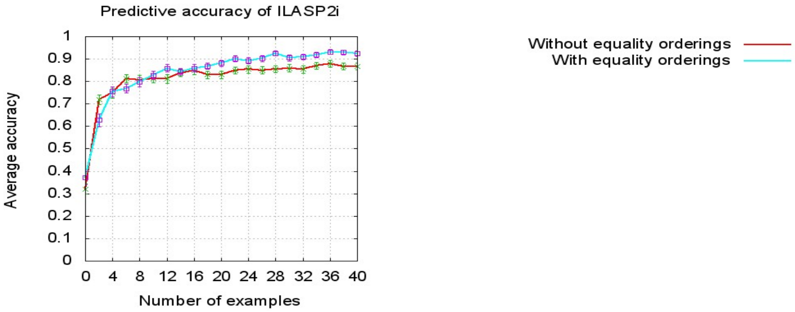


Figure: average accuracy of ILASP2i

Experiments

Learning task	#examples				time/s			Memory/kB		
	E^+	E^-	O^b	O^c	2	2i_pt	2i	2	2i_pt	2i
Hamilton A (no context)	100	100	0	0	10.3	4.2	**4.3**	9.7×10^4	1.2×10^4	$\mathbf{1.2 \times 10^4}$
Hamilton B (context dep.)	100	100	0	0	32.0	84.9	**3.6**	3.6×10^5	2.7×10^5	$\mathbf{1.4 \times 10^4}$
Journeys (context dep.)	386	0	200	0	1031.4	45.2	**5.0**	1.4×10^7	1.1×10^6	$\mathbf{3.4 \times 10^4}$

- 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	✔	✔	✔	✖	✔	✔	✔	✔	✔

Related Incremental Learner

- ILASP2i incrementally constructs the set of relevant examples, learning a new hypothesis each time.
  - ILASP2i's relevant example set could become very large.
  - ILASP2i is guaranteed to find an optimal solution.
- ILED (Katzouris et al. 2015) is an incremental extension of XHAIL, which is targeted at learning event definitions.
  - ILED incrementally learns a hypothesis through theory revision.
  - ILED is not guaranteed to find an optimal solution.

Imperial College London

Current Work

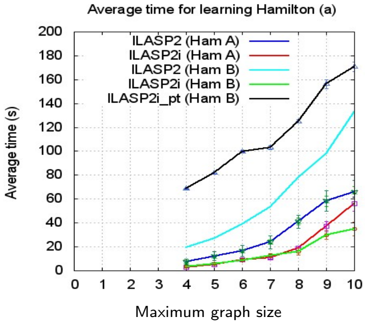
- Improve the scalability of ILASP for tasks with:
  - Noisy examples
  - Large hypothesis spaces

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`

Hamilton Experiment



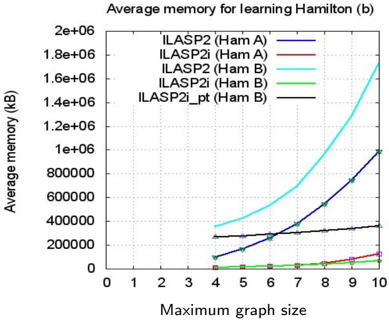


Figure: (a) the average computation time and (b) the memory usage of ILASP2, ILASP2i and ILASP2i_pt for Hamilton A and B.

ILP_{LAS}

Definition

An ILP_{LAS} task is a tuple $T = \langle B, S_M, E^+, E^- \rangle$.
A hypothesis $H \subseteq S_M$ is in $ILP_{LAS}(T)$, the set of all inductive solutions of T, if and only if:

- $\forall e \in E^+ \; \exists A \in AS(B \cup H)$ *such that A extends e*
- $\forall e \in E^- \; \nexists A \in AS(B \cup H)$ *such that A extends e.*

Context-dependent ILP_{LAS}

Definition

An $ILP_{LAS}^{context}$ *task is a tuple* $T = \langle B, S_M, E^+, E^- \rangle$.
A hypothesis $H \subseteq S_M$ *is in* $ILP_{LAS}^{context}(T)$, *the set of all inductive solutions of* T, *if and only if:*

- $\forall \langle e, C \rangle \in E^+ \; \exists A \in AS(B \cup C \cup H)$ *such that* A *extends* e
- $\forall \langle e, C \rangle \in E^- \; \nexists A \in AS(B \cup C \cup H)$ *such that* A *extends* e.

$ILP_{LOAS}^{context}$

Definition

A context-dependent ordering example o is a pair $\langle\langle e_1, C_1\rangle, \langle e_2, C_2\rangle\rangle$. A program P is said to bravely (resp. cautiously) respect o if for at least one (resp. every) pair $\langle A_1, A_2\rangle$ such that $A_1 \in AS(P \cup C_1)$, $A_2 \in AS(P \cup C_2)$, A_1 extends e_1 and A_2 extends e_2, it is the case that $A_1 \prec_P A_2$.

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.

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

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

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.

Katzouris, N., Artikis, A., and Paliouras, G. 2015.
Incremental learning of event definitions with inductive logic programming.
Machine Learning 100, 2-3, 555–585.

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.
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.