Imperial College
London

Inductive Learning of Answer Set Programs from Noisy Examples

Mark Law, Alessandra Russo and Krysia Broda

August 18, 2018

Inductive Reasoning for Cognitive Systems

Humans are capable of performing cognitive activities, such as:

- Learning from past experience.
- Making predictions and reasoning using learned knowledge.
- Revising and extending knowledge, based on new information.
- Communicating learned knowledge to others.
- Learning in the presence of incomplete and inaccurate (or noisy) information.

Inductive Reasoning for Cognitive Systems

Humans are capable of performing cognitive activities, such as:

- Learning from past experience.
- Making predictions and reasoning using learned knowledge.
- Revising and extending knowledge, based on new information.
- Communicating learned knowledge to others.
- Learning in the presence of incomplete and inaccurate (or noisy) information.

To realise human-like levels of cognition, Machine Learning solutions have to achieve the above points.

Inductive Logic Programming

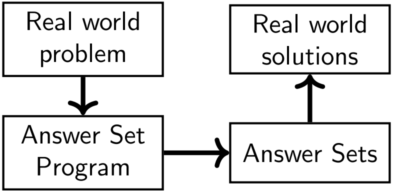
- Given E^+, E^- and B, the goal is to find a hypothesis H such that:
  - $\forall e \in E^+ : B \cup H \models e$
  - $\forall e \in E^- : B \cup H \not\models e$

Inductive Logic Programming

- Given E^+, E^- and 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 of ILP for Cognitive Systems are that:
  - The hypotheses are human readable, and can therefore be communicated to humans or other machines.
  - Can define useful concepts in the background knowledge; it is thus possible to extend an existing knowledge base, rather than starting from scratch.

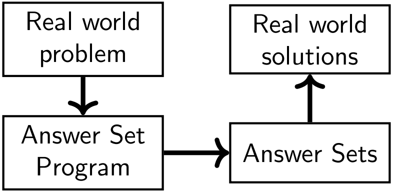
Answer Set Programming (ASP)

- Expressive declarative environment for logical reasoning.



Answer Set Programming (ASP)

- Expressive declarative environment for logical reasoning.



Desirable features for representing knowledge for Cognitive Systems:

- Negation as failure can be used to model defaults and exceptions.
- Choice rules can be used to model non-determinism and choice.
- Preferences can be modelled as weak constraints.

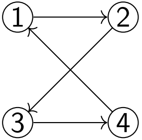
Initial Approaches to learning ASP

- The early approaches to learning ASP were either *brave* or *cautious*, meaning that examples had to be covered in either *at least one* or *every* answer set (respectively).

- We showed in (Law et al. 2014) that to learn some programs a combination of *both* brave and cautious semantics is required, and presented the ILP_{LAS} framework which combines brave and cautious semantics.

ILP_{LAS} Encoding of the Hamiltonian Example

- An answer set A *extends* a partial interpretation $\langle e^{inc}, e^{exc}\rangle$ iff $e^{inc} \subseteq A$ and $e^{exc} \cap A = \emptyset$.



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

B :

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

H :

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

Context-dependent Examples

- In standard ILP, for each example e, $B \cup H \models e$.

- In our framework we can make use of *context-dependent* examples. Each example has its own context C_e, and the coverage condition is that $B \cup H \cup C_e \models e$.

Imperial College London

Weak Constraints

Avoid walking through areas with a high crime rating	Minimise the number of buses	Minimise the distance walked

```
:~ mode(Leg, walk), crime_rating(Leg, C), C > 4 . [1@3]
:~ mode(Leg, bus) . [1@2]
:~ mode(Leg, walk), distance(Leg, Distance) . [Distance@1]
```

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.

Weak Constraints

Avoid walking through areas with a high crime rating	Minimise the number of buses	Minimise the distance walked

$$:\sim \mathrm{mode(Leg, walk), crime\_rating(Leg, C), C > 4\ .\ [1@3]}$$
$$:\sim \mathrm{mode(Leg, bus)\ .\ [1@2]}$$
$$:\sim \mathrm{mode(Leg, walk), distance(Leg, Distance)\ .\ [Distance@1]}$$

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

Context-dependent Ordering Examples

- Using examples such as:

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.

- We can learn:

```
:~ mode(Leg, walk), crime_rating(Leg, C), C > 4 . [1@3]
:~ mode(Leg, bus) . [1@2]
:~ mode(Leg, walk), distance(Leg, Distance) . [Distance@1]
```

▶ An *ordering example* is a pair of positive examples expressing that the first example is preferred to the second.

Learning from Noisy Examples

- The tasks up to this point have been non-noisy. For non-noisy tasks ILASP searches for a hypothesis that covers all the examples and minimises $|H|$.
- In noisy tasks, some (usually all) examples are given a positive integer penalty, and ILASP searches for a hypothesis H that minimizes $\mathcal{S}(H, T)$ (and for which $\mathcal{S}(H, T)$ is finite), where:

$$\mathcal{S}(H, T) = |H| + \sum_{e \in uncov(H,T)} e_{pen}$$

ILASP

Algorithm	Scales with			
	negative examples	large numbers of examples	noise	large hypothesis spaces
ILASP1	No	No	No	No
ILASP2	Yes	No	No	No
ILASP2i	Yes	Yes	No	No

- ILASP is a collection of algorithms for solving $ILP_{LOAS}^{context}$ and, more recently, ILP_{LOAS}^{noise} tasks.
- Each ILASP algorithm is sound and complete wrt the optimal solutions of a LOAS tasks.

Imperial College London

ILASP3

Algorithm	Scales with negative examples	large numbers of examples	noise	large hypothesis spaces
ILASP1	No	No	No	No
ILASP2	Yes	No	No	No
ILASP2i	Yes	Yes	No	No
ILASP3	Yes	Yes	Yes	No

- ILASP is a collection of algorithms for solving $ILP_{LOAS}^{context}$ and, more recently, ILP_{LOAS}^{noise} tasks.
- Each ILASP algorithm is sound and complete wrt the optimal solutions of a LOAS tasks.
- Our most recent algorithm, ILASP3, specifically targets learning in the presence of noise.

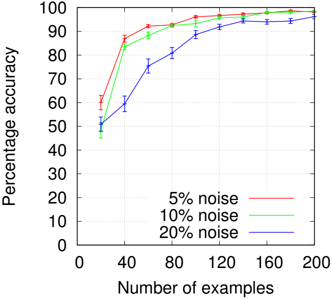
Synthetic Evaluation

Noisy Hamilton Evaluation

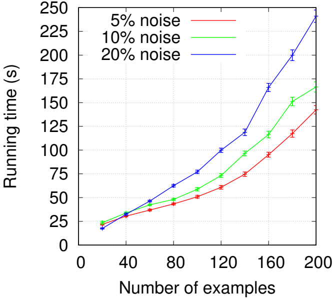
For $n = 0, 20, 40, \ldots, 200$:

- n graphs of up to size 4 were generated, half of which were Hamiltonian.
- For $p = 5, 10$ and 20, $p\%$ of the graphs were incorrectly labelled.
- ILASP3 was then used to learn a hypothesis, which was then tested on a further 1000 graphs.
- Each experiment was repeated 50 times.

Noisy Hamilton Results



(a)



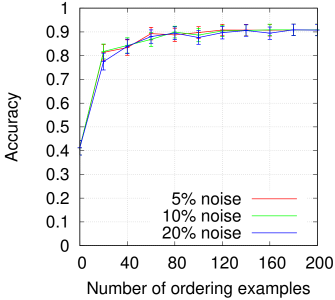
(b)

Noisy Journey Preference Evaluation

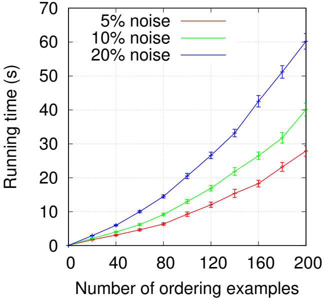
We randomly generated 50 “target hypotheses”, each consisting of between 1-3 weak constraints. For each target hypothesis:

- For $n = 0, 20, 40, \ldots, 200$, we generated 200 ordering examples (each consisting of a pair of journeys and a comparison operator – either $<$ or $=$).
- For $p = 5, 10$ and 20 we changed the comparison operator of a random p% of the examples.
- ILASP3 was then used to learn a hypothesis, which was then tested on a further set of journey pairs.

Noisy Journey Preference Results



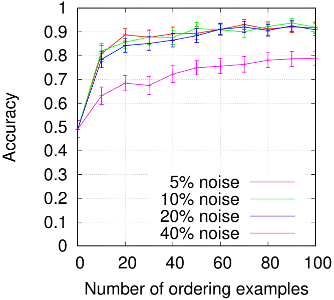
(a)



(b)

Imperial College London

Noisy Journey Preference Results



(c)

Comparison to Approximate Algorithms

- ILASP is guaranteed to return an optimal solution of any ILP^{noise}_{LOAS} task (resources permitting).
- Several other algorithms for learning under the answer set semantics exist, but often, they make no such guarantee.
- We have compared the accuracy of the optimal hypotheses returned by ILASP3 with the accuracy of these other *approximate algorithms* on (mostly) real data.

Sentence Chunking Dataset

- In (Kazmi et al. 2017), the Inspire system was evaluated on a sentence chunking dataset (Agirre et al. 2016), which contains examples of how sentences should be *chunked*.
- For instance, according to the dataset, the sentence "Thai opposition party to boycott general election." should be split into the three chunks "Thai opposition party", "to boycott" and "general election".
- Inspire claims that such a dataset requires apprioximate algorithms, which are not guaranteed to find optimal solutions.

Imperial College London

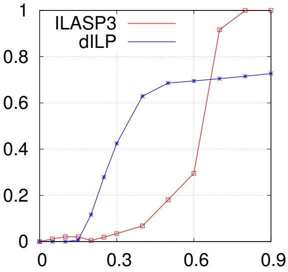
Sentence Chunking Dataset

		Inspire F_1 score	ILASP F_1 score	ILASP computation time (s)
100 examples	Headlines S1	73.1	74.2	351.2
	Headlines S2	70.7	73.0	388.3
	Images S1	81.8	83.0	144.9
	Images S2	73.9	75.2	187.2
	Students S1/S2	67.0	72.5	264.5
500 examples	Headlines S1	69.7	75.3	1616.6
	Headlines S2	73.4	77.2	1563.6
	Images S1	75.3	80.8	929.8
	Images S2	71.3	78.9	935.8
	Students S1/S2	66.3	75.6	1451.3

Comparison to δILP

- In (Evans and Grefenstette 2018), it was claimed that ILP approaches are unable “to handle noisy, erroneous, or ambiguous data” and that “If the positive or negative examples contain any mislabelled data, [ILP approaches] will not be able to learn the intended rule”.
- To learn from noisy data, (Evans and Grefenstette 2018) introduces the δILP algorithm, based on artificial neural networks, which is able to achieve a high accuracy even with a large proportion of noise in the examples.

Comparison to δILP: predecessor

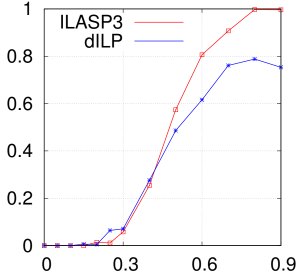


(a)

`predecessor(V1, V0):-succ(V0, V1).`

Imperial College London

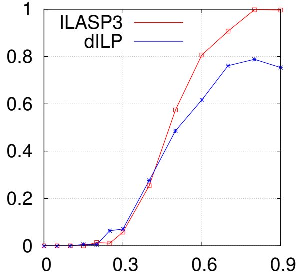
Comparison to δILP: less than



(b)

```
less_than(V0, V1):-succ(V0, V1).
less_than(V0, V2):-succ(V0, V1), less_than(V1, V2).
```

Comparison to δILP: less than



(b)

```
geq(V0,V0):-succ(V0,V1).
geq(V1,V1):-succ(V0,V1).
geq(V0,V2):-succ(V2,V1),geq(V0,V1).
```

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	Noise
Brave Induction [Sakama, Inoue 2009], XHAIL [Ray 2009], ASPAL [Corapi et al 2011], RASPAL [Athakravi et al 2013], ILED [Katzouris 2015], Inspire [Schüller 2016]	✔	✔	✖	✔	✔	✖	✖	✖	✔	✔
Cautious Induction [Sakama, Inoue 2009]	✔	✔	✖	✔	✖	✔	✖	✖	✖	✖
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 [Law et al 2016]	✔	✔	✔	✖	✔	✔	✔	✔	✔	✖
Learning from Noisy Answer Sets [Law et al 2018]	✔	✔	✔	✖	✔	✔	✔	✔	✔	✔

Related Work

- Early approaches to relational learning (e.g. (Mooney and Ourston 1991) and (Cohen 1995)) were able to learn definite rules from noisy data.
- Many early ILP approaches, such as (Cohen 1995) and (Muggleton 1995), give algorithms which learning one clause at a time.
- ILP systems which iteratively learn single clauses are common when the target hypotheses are definite logic programs (with no negation), as the programs being learned are *monotonic*.
- Learning *non-monotonic* ASP programs with negation requires a different approach, due to the non-monotonicity.

Conclusion

- Learning interpretable knowledge is a key requirement for cognitive systems.
- ASP programs are capable of representing complex knowledge, such as defaults, exceptions and preferences.
- ILASP3 can learn even in the presence of high proportions noisy examples.
- Our experiments show that in most cases ILASP3 is able to learn with a higher accuracy than existing approximate systems, which are not guaranteed to find optimal solutions of the tasks.
- In current work, we are developing ILASP systems which are more scalable wrt the size of the hypothesis space.

Imperial College London

Backup Slides

Learning from Noisy Examples

Definition 1

An ILP^{noise}_{LOAS} task T is the form $\langle B, S_M, E \rangle$. Given a hypothesis $H \subseteq S_M$,

1. $uncov(H, T)$ is the set consisting of all examples in E that H does not cover.
2. $\mathcal{S}(H, T)$, is the sum $|H| + \sum_{e \in uncov(H,T)} e_{pen}$.
3. H is an inductive solution of T if and only if $\mathcal{S}(H, T)$ is finite.
4. H is an *optimal inductive solution* of T if and only if $\mathcal{S}(H, T)$ is finite and $\nexists H' \subseteq S_M$ such that $\mathcal{S}(H, T) > \mathcal{S}(H', T)$.

Complexity

Framework	Verification	Satisfiablity
ILP_b	NP-complete	NP-complete
ILP_{sm}	NP-complete	NP-complete
ILP_c	DP-complete	Σ_2^P-complete
ILP_{LAS}	DP-complete	Σ_2^P-complete
ILP_{LOAS}	DP-complete	Σ_2^P-complete
$ILP_{LOAS}^{context}$	DP-complete	Σ_2^P-complete
ILP_{LOAS}^{noise}	DP-complete	Σ_2^P-complete

Learning from Answer Sets

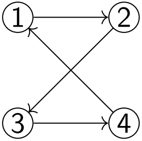
Definition 2

An ILP_{LAS} task is a tuple $T = \langle B, S_M, \langle E^+, E^- \rangle \rangle$. A hypothesis $H \subseteq S_M$ is an *inductive solution* of T if and only if:

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

- An answer set A *extends* a partial interpretation $\langle e^{inc}, e^{exc} \rangle$ iff $e^{inc} \subseteq A$ and $e^{exc} \cap A = \emptyset$.

ILP_{LAS} Encoding of the Hamiltonian Example



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

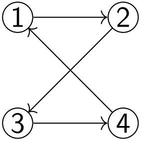
B :

```
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



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

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.
```

Brave and Cautious Ordering Examples

Definition

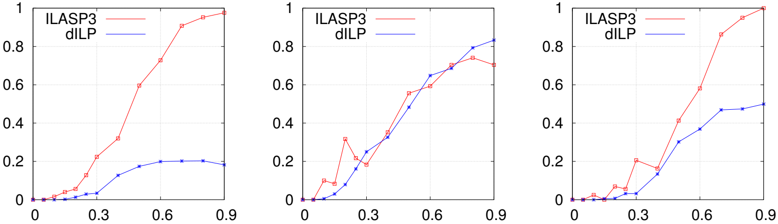
An ordering example is a tuple $o = \langle e_1, e_2 \rangle$ *where* e_1 *and* e_2 *are partial interpretations.*

- *An ASP program* P bravely respects o *iff* $\exists A_1, A_2 \in AS(P)$ *such that* A_1 *extends* e_1, A_2 *extends* e_2 *and* $A_1 \succ_P A_2$.
- P cautiously respects o *iff* $\forall A_1, A_2 \in AS(P)$ *such that* A_1 *extends* e_1 *and* A_2 *extends* e_2*, it is the case that* $A_1 \succ_P A_2$.

- In the tasks in this paper, the notion of brave and cautious orderings coincided (as in the preference learning tasks all positive examples were extended by exactly one answer set).

Comparison to δILP: member

Member | Connected | Undirected Edge



(c) (d) (e)

Related Work: noise thresholds

- ILP systems often use a cost function, e.g. (Srinivasan 2001, Muggleton 1995, Bragaglia and Ray 2014).
- When examples are noisy, this cost function is sometimes combined with a notion of maximum threshold, e.g. (Srinivasan 2001, Oblak and Bratko 2010, Athakravi et al. 2013).
- Our ILP_{LOAS}^{noise} framework addresses the problem of computing optimal solutions (with respect to the cost function) and in doing so does not require knowledge a priori of the level of noise in the data.

Agirre, E., Gonzalez Agirre, A., Lopez-Gazpio, I., Maritxalar, M., Rigau Claramunt, G., and Uria, L. 2016.
Semeval-2016 task 2: Interpretable semantic textual similarity.
In *Proceedings of the Tenth International Workshop on Semantic Evaluation*. Association for Computational Linguistics, 512–524.

Athakravi, D., Corapi, D., Broda, K., and Russo, A. 2013.
Learning through hypothesis refinement using answer set programming.
In *Proceedings of the Twenty-third International Conference on Inductive Logic Programming, Rio de Janeiro, Brazil, August 28-30, 2013*, G. Zaverucha, V. S. Costa, and A. Paes, Eds. Lecture Notes in Computer Science, vol. 8812. Springer, 31–46.

Bragaglia, S. and Ray, O. 2014.
Nonmonotonic learning in large biological networks.
In *Proceedings of the Twenty-fourth International Conference on Inductive Logic Programming, Nancy, France, September 14-16, 2014*, J. Davis and J. Ramon, Eds. Lecture Notes in Computer Science, vol. 9046. Springer, 33–48.

Cohen, W. W. 1995.
Fast effective rule induction.
In *Machine Learning, Proceedings of the Twelfth International Conference on Machine Learning, Tahoe City, California, USA, July 9-12, 1995*, A. Prieditis and S. J. Russell, Eds. Morgan Kaufmann, 115–123.

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

Evans, R. and Grefenstette, E. 2018.
Learning explanatory rules from noisy data.
Journal of Artificial Intelligence Research 61, 1–64.

Kazmi, M., Schüller, P., and Saygin, Y. 2017.
Improving scalability of inductive logic programming via pruning and best-effort optimisation.
Expert Systems with Applications 87, 291–303.

Law, M., Russo, A., and Broda, K. 2014.
Inductive learning of answer set programs.
In *Proceedings of the Fourteenth European Conference on Logics in Artificial Intelligence, 2014, Funchal, Madeira, Portugal, September 24-26, 2014.*, E. Fermé and J. Leite, Eds. Lecture Notes in Computer Science, vol. 8761. Springer, 311–325.

Mooney, R. J. and Ourston, D. 1991.
Theory refinement with noisy data (Technical Report AI 91-153).
Artificial Intelligence Laboratory, University of Texas at Austin.

Muggleton, S. 1995.
Inverse entailment and Progol.
New Generation Computing 13, 3-4, 245–286.

Oblak, A. and Bratko, I. 2010.
Learning from noisy data using a non-covering ILP algorithm.
In *Proceedings of the Twentieth International Conference on Inductive Logic Programming, 2010, Florence, Italy, June 27-30*, P. Frasconi and F. A. Lisi, Eds. Lecture Notes in Computer Science, vol. 6489. Springer, 190–197.

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.

Srinivasan, A. 2001.
The Aleph manual.
Machine Learning at the Computing Laboratory, Oxford University.