Using Abduction for Induction of Normal Logic Programs

(XHAIL system presentation)

Oliver Ray

Department of Computing
Imperial College London

AIAI'06 (August 29th, 2006)

Motivation: Nonmonotonic ILP

- Inductive Logic Programming (ILP) [MD94]
  - expressive, understandable
  - $H \leftarrow B_1, \ldots, B_n$
- Utility of Negation as Failure (NAF) [Cla78]
  - compact, non-monotonic
  - $not\ flight(london, riva\_del\_garda)$
- Nonmonotonic ILP (NM-ILP) [Sak05]
  - lack of effective tool support
  - In practice use Horn systems like Progol5 and Alecto

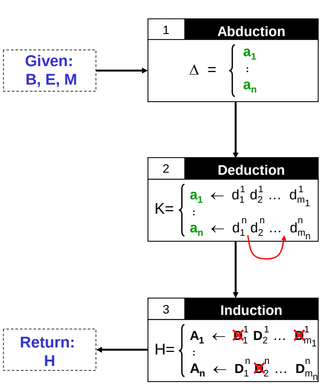
Application: Learning Action theories

- Recent focus on inducing domain axioms in temporal formalisms like the Event Calculus [MM99, Moy02] and Situation Calculus [Otr05]
  - 2 sessions on learning action descriptions at ILP06
  - previously used to learn robot navigation programs
- Exposes limitations of existing ILP systems such as Progol [Mug95] and Alecto [Moy04]
  - limited ability to reason with negation when computing the head atoms of a hypothesis; unsound
  - hence cannot be applied to emerging problems like the extraction of requirements from scenarios [ARRU06]

Approach: Abductive-Inductive Learning

- Abductive Logic Programming (ALP) [KKT92]
  - hypothetical reasoning under incomplete information
  - given T,G,IC,A find θ, $\Delta \subseteq A$ such that $T \cup \Delta \models \exists G\theta$ & $\forall IC$
  - formal correspondence between NAF and ALP [EK'88]
- Extend the mode-directed framework of Hybrid Abductive Inductive Learning (HAIL) [Ray05]
  - uses ALP to overcome several limitations of Progol5 and Alecto in the Horn clause case
  - already includes a full ALP interpreter with support for negation
  - core techniques of constructing and generalising Kernel Sets lifts to the nonmonotonic case

(eXtended) Hybrid Abductive Inductive Learning



Find a set of (head) atoms that abductively explain the positive and negative examples:
i.e. $B \wedge a_1 \wedge \ldots \wedge a_n \models E$

Find a set of (body) literals that deductively follow from the prior knowledge:
i.e. $B \models d_1^1 \wedge \ldots \wedge d_{m_n}^n$

Find a compressive theory that theta-subsumes the Kernel Set and explains the examples
i.e. $H \geq K$ and $B \wedge H \models E$

Example: Requirements Engineering

- Background Knowledge (B):

holdsAt(F,T) ← happens(A,T'), T' < T, initiates(A,F,T'), not clipped(T',F,T).
clipped(T',F,T) ← happens(A,T''), T' ≤ T'' < T, terminates(A,F,T'').
happens(A,T) ← attempt(A,T), not impossible(A,T).

fluent(pumpOn).	**fluent(methane)**
initiates(turnOn,pumpOn,T).	**terminates(turnOff,pumpOn,T).**
initiates(leakGas,methane,T).	**terminates(ventGas,methane,T).**
attempt(leakGas,0).	**attempt(turnOn,1).**
attempt(ventGas,1).	**attempt(turnOff,3).**

- Examples (E):

not holdsAt(pumpOn,1).	**holdsAt(methane,1).**
not holdsAt(pumpOn,2).	**holdsAt(methane,2).**
not holdsAt(pumpOn,3).	**not holdsAt(methane,3).**
holdsAt(pumpOn,3).	**not holdsAt(methane,3).**

- Mode Declarations (M):

modeh(*, impossible(turnOn,+int)).
modeb(*, holdsAt(#fluent,+int)). **modeb(*, not holdsAt(#fluent,+int)).**

Abductive Phase

- Abducibles (A)
  - ground instances of a head declaration
- Theory (T)
  - definite background clauses
- Integrity Constraints (IC)
  - negative background clauses
- Goals (G)
  - positive and negative examples

$$\Delta = \{impossible(turnOn, 1)\}$$

Deductive Phase

- Abducibles (A)
  - none
- Theory (T)
  - definite background clauses
- Integrity Constraints (IC)
  - negative background clauses
- Goals (G)
  - each instance of a body declaration with input variables substituted by input terms from each head atom in Δ

$$K = \{impossible(turnOn, 1) \leftarrow holdsAt(methane, 1), not\ holdsAt(pumpOn, 1)\}$$

Inductive Phase

- Abducibles (A)
  - use/2
- Theory (T)
  - definite background clauses plus theory K' (see next slide) that encodes the search as an ALP problem
- Integrity Constraints (IC)
  - negative background clauses
- Goals (G)
  - positive and negative examples

$$H = \{impossible(turnOn, X) \leftarrow holdsAt(methane, X)\}$$

Inductive Phase Translation

$$K' = \left\{ \begin{array}{l} impossible(turnOn, X) \leftarrow try(1, 1, [X]), try(1, 2, [X]). \\ \\ try(1, 1, [X]) \leftarrow not\ use(1, 1). \\ try(1, 1, [X]) \leftarrow use(1, 1), holdsAt(methane, X). \\ \\ try(1, 2, [X]) \leftarrow not\ use(1, 2). \\ try(1, 2, [X]) \leftarrow use(1, 2), not\ holdsAt(pumpOn, X). \end{array} \right\}$$

Conclusion

- XHAIL provides a (sable model) semantics and proof procedure for NM-ILP
- It uses mode declarations in the construction of a Kernel Set to reduce generalisation search space
- It has been applied in a requirements engineering example where existing systems are inapplicable
- It supports the hypothesis that abduction and induction can be usefully integrated