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
  - \( H \models B_1, \ldots, B_n \)

- Utility of Negation as Failure (NAF) [Cla78]
  - compact, non-monotonic
  - not \( \text{flight}(\text{London}, \text{Riva del Garda}) \)

- 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 [Ord95]
  - 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 \( \theta, \Delta \subseteq A \) such that \( T \cup \Delta \models \exists G \theta \land \forall IC \)

- 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

Example: Requirements Engineering

- Background Knowledge (B):

- Examples (E):

- Mode Declarations (M):

(eXtended) Hybrid Abductive Inductive Learning

1 Abduction

\[ \Delta \models \gamma_1, \ldots, \gamma_m \]

Find a set of (head) atoms that abductively explain the positive and negative examples:

\[ i.e. \ B \land \gamma_1 \land \ldots \land \gamma_m \models E \]

2 Deduction

\[ K = \gamma_1 \land \ldots \land \gamma_m \]

Find a set of (body) literals that deductively follow from the prior knowledge:

\[ i.e. \ B \models \gamma_1 \land \ldots \land \gamma_m \]

3 Induction

\[ H = \gamma_1 \land K \land \gamma_1 \land K \land \gamma_1 \land K \land \gamma_1 \land K \]

Find a compressive theory that theta-subsumes the Kernel Set and explains the examples

\[ i.e. \ H \models K \land B \land H \models E \]
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

\[ \Delta = \{\text{impossible(turnOn,1)}\} \]

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 \( \Delta \)

\[ K = \{\text{impossible(turnOn,1)}, \text{holdsAt(methane,1)}, \text{not holdsAt(pumpOn,1)}\} \]

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

\[ I = \{\text{impossible(turnOn,X)} \leftarrow \text{holdsAt(methane,X)}\} \]

Inductive Phase Translation

\[ K' = \begin{cases} 
\text{impossible(turnOn,X)} & \leftarrow \text{try(1,1,[X]),try(1,2,[X])}. \\
\text{try(1,1,[X])} & \leftarrow \text{not use(1,1)}. \\
\text{try(1,1,[X])} & \leftarrow \text{use(1,1),holdsAt(methane,X)}. \\
\text{try(1,2,[X])} & \leftarrow \text{not use(1,2)}. \\
\text{try(1,2,[X])} & \leftarrow \text{use(1,2),not holdsAt(pumpOn,X)}. 
\end{cases} \]

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