Equivalence Issues in Abduction and Induction

Chiaki Sakama
Wakayama University
Katsumi Inoue
National Institute of Informatics

AIAI'07, Aix-en-Provence, 15th Sept. 2007

Equivalence issues in AI and Program Development

- Identification: identifying different knowledge bases developed by different experts
- Verification: correct implementation of a given declarative specification
- Optimization: transforming a program to an efficient coding

Program Equivalence in LP

- P1 and P2 are weakly equivalent if they have the same declarative meaning.
- P1 and P2 are strongly equivalent if $P1 \cup R$ and $P2 \cup R$ have the same declarative meaning for any program R.
- These equivalence relations compare capabilities of deductive reasoning between programs.

Comparing non-deductive capabilities between programs

- Intelligent agents perform non-deductive commonsense reasoning as well as deductive reasoning.
- Comparing capabilities of non-deductive reasoning such as abduction and induction is meaningful to measure intelligence of agents.

Abduction and Induction

- Abduction and induction both produce hypotheses to explain observations using background theories.
- There are at least 3 parameters in this task: background theories, explanations, and observations.

Equivalence issues in abduction and induction

- **Equivalence of background theories**:
  Two background theories are equivalent if they produce the same explanations for any observation.
- **Equivalence of explanations**:
  Two explanations are equivalent if they account for the same observation under a given background theory.
- **Equivalence of observations**:
  Two observations are equivalent if they produce the same explanations under a given background theory.

Abductive Equivelence
[Inoue and Sakama, IJCAI-05, MBR-06]

- Explainable equivalence considers whether two theories have the same explainability for any observation.
- Explanatory equivalence considers whether two theories have the same explanations for any observation.

Necessary and sufficient conditions are provided for abductive equivalence in FOL and abductive logic programming.

Example

- B1={ *rained* → *wet-grass* } and
  B2={ *sprinkler-on* → *wet-grass* } with
  H={ *rained, sprinkler-on* }
  are explainably equivalent, because
  wet-grass, *rained*, *sprinkler-on*
  are all explainable in both (B1,H) and (B2,H).
- Two theories are not explanatorily equivalent, because *wet-grass* is explained by *rained* in B1, but it is not explained by *rained* in B2.

Inductive Equivalence [Sakama & Inoue, ILP-05]

- A background theory is inductively equivalent to another background theory if they induce the same hypothesis in face of any example.
- Conditions for inductive equivalence are compared in different ILP systems.

Remaining Issues

- Conditions for equivalence of explanations.
- Conditions for equivalence of observations.
- Conditions for explainable equivalence in skeptical abduction.
- Computational complexities of testing those equivalences.

We investigate these issues in both FOL and (nonmonotonic) logic programming.

Abduction in FOL

- Abductive theory (B, H) where B and H are sets of first-order formulas, respectively representing a background theory and a candidate hypothesis.
- Given an observation O as any formula, $E (\subseteq H)$ is an explanation of O if
  - $B \cup E \models O$
  - $B \cup E$ is consistent.

Remark

- The above definition also characterizes (explanatory) induction.
- Given a finite set G of examples, induction finds a hypothesis E satisfying

  $B \cup E \models G$ where $B \cup E$ is consistent.
- Put $O = \bigwedge_{g \in G} g$ and $H = \mathcal{F}$, where $\mathcal{F}$ is the set of all formulas.
- Then, induction is characterized as abduction, and we do not distinguish them hereafter.

General Extended Disjunctive Program (GEDP)

- Rules

  $L_1 ;\ldots; L_k ; \textbf{not } L_{k+1} ;\ldots; \textbf{not } L_l$

  $\leftarrow L_{l+1} ,\ldots, L_m , \textbf{not } L_{m+1} ,\ldots, \textbf{not } L_n$

  where L_i are literals.

- Meaning

  If all $L_{l+1} ,\ldots, L_m$ hold and all $L_{m+1} ,\ldots, L_n$ do not hold, then some of $L_1 ,\ldots, L_k$ hold or some of $L_{k+1} ,\ldots, L_l$ do not hold.

Answer Set Semantics

- A declarative meaning of a GEDP is given by the answer set semantics (Gelfond and Lifschitz).
- A program is consistent if it has a consistent answer set. The set of answer sets of P is denoted by AS(P).
- A literal L is a consequence of skeptical/credulous reasoning in a program P if L is included in every/some answer set of P.
- For a consistent program P, define

$skp(P) = \bigcap_{S \in AS(B \cup E)} S$ and $crd(P) = \bigcup_{S \in AS(B \cup E)} S$

Abductive Logic Program

- Abductive program $\langle P, A \rangle$ where P and A are GEDPs, respectively representing a background theory and a candidate hypothesis.
- Every element in A is called an abducible.
- Given an observation O as a ground literal, $E(\subseteq H)$ is a credulous/skeptical explanation of O in $\langle P, A \rangle$ if O is included in some/every consistent answer set of $B \cup E$.

Example

- P: $watchTV\,;\ sleeping \leftarrow holiday,\ \textbf{not}\ busy,$
  $working \leftarrow holiday,\ busy,$
  $holiday \leftarrow .$
  A: $busy$.
- P has two answer sets: $\{holiday,\ watchTV\}$ and $\{holiday,\ sleeping\}$. So, $O1 = watchTV$ has the credulous explanation $E1 = \{\}$.
- P U { $busy$ } has the single answer set: $\{holiday,\ busy,\ working\}$. So, $O2 = working$ has the skeptical explanation $E2 = \{busy\}$.

Remark

- The above definition also characterizes inductive logic programming.
- Given a finite set G of ground literals as examples, build a rule $O \leftarrow G$ and put $B' = B \cup \{ O \leftarrow G \}$. Then,

  $$B \cup E \models G \text{ iff } B' \cup E \models O .$$

- Again, induction is characterized as abduction in the context of logic programming.

Equivalence of Explanations in First-Order Abduction

- **Definition** Given an abductive theory (B,H), two explanations E1 and E2 are equivalent if, for any observation O, E1 is an explanation of O in (B,H) iff E2 is an explanation of O in (B,H).

Result

- **Theorem** Let (B,H) be an abductive theory. Then, two explanations E1 and E2 are equivalent iff $B \cup E1 \equiv B \cup E2$.

- Given

  B: $p \supset q$, $q \supset p$, $p \wedge q \supset r$

  H: p , q ,

  E1={ p }, E2={ q }, E3={ p, q }

  are all equivalent explanations.

Equivalence of Explanations in Abductive LP

- **<u>Theorem</u>** Let $\langle P, A\rangle$ be an abductive program. Then, two explanations E1 and E2 are equivalent if $AS(P \cup E1)=AS(P \cup E2)$ where $P \cup E1$ and $P \cup E2$ are consistent.
- Given

  P : $p \leftarrow \textbf{\textit{not}}\ q$, $q \leftarrow \textbf{\textit{not}}\ p$, $r \leftarrow \textbf{\textit{not}}\ r$

  A : $r \leftarrow p$, $r \leftarrow \textbf{\textit{not}}\ q$,

E1={ $r \leftarrow p$ }, E2={ $r \leftarrow \textbf{\textit{not}}\ q$ }, E3={ $r \leftarrow p$, $r \leftarrow \textbf{\textit{not}}\ q$ } are all equivalent explanations.

Equivalence of Observtions in First-Order Abduction

- **<u>Definition</u>** Given an abductive theory (B,H), two observations O1 and O2 are equivalent if, for any E(⊆H),
  E is an explanation of O1 in (B,H)
  iff E is an explanation of O2 in (B,H).

Result

- **Theorem** Let (B,H) be an abductive theory. Then, two observations O1 and O2 are equivalent iff $B \cup E \models O1 \equiv O2$ for any $E \subseteq H$ such that $B \cup E$ is consistent.
- Given
  B1: $p \supset q$,   H1: p , q ,
  O1= p and O2= $p \wedge q$ are equivalent.
  But O1 and O2 are not equivalent in
  B2: $p \vee q$,   H2: p , q .

Equivalence of Observations in Abductive LP

- **Theorem** Let $\langle P, A\rangle$ be an abductive program, O1 and O2 be observations.
  - O1 and O2 are equivalent in credulous abduction iff $O1 \in crd(P \cup E) \Leftrightarrow O1 \in crd(P \cup E)$
  - O1 and O2 are equivalent in skeptical abduction iff $O1 \in skp(P \cup E) \Leftrightarrow O2 \in skp(P \cup E)$

for any $E \subseteq A$ s.t. $P \cup E$ is consistent.

Example

- Given

  P : $wet \leftarrow rain, \textbf{not}\ \neg wet$,

  $\neg wet \leftarrow rain, \textbf{not}\ wet$

  A : *rain*
- Putting E={}, P ∪ E has the answer set {}.
- Putting E={*rain* }, P ∪ E has two answer sets {*rain, wet* } and {*rain*, ¬ *wet* }.

Then, O1=*wet* and O2= ¬*wet* are equivalent in both credulous/skeptical abduction.

Summary of Results
necessary and sufficient conditions

Logic	Background Theory		Explanation	Observation
	explainable	explanatory		
FOL (B1,H1) v.s. (B2,H2)	Ext(B1,H1) =Ext(B2,H2)*	B1≡B2	$B \cup E1 \equiv B \cup E2$	$B \cup E \models O1 \equiv O2$ for any $E \subseteq H$
ALP (credulous) ⟨P1, A1⟩ v.s. ⟨P2, A2⟩	$\bigcup_{E \in A1}$ crd(B1 $\cup$E) $=\bigcup_{F \in A2}$ crd(B2 $\cup$ F)	B1 and B2 are strongly equivalent wrt A1=A2.	AS(P $\cup$ E1) =AS(P $\cup$ E2)	O1$\in$crd(P $\cup$ E) $\Leftrightarrow$ O1$\in$crd(P $\cup$ E)
(skeptical)	$\exists E \in A1$, $\exists F \in A2$ s.t. skp(P1$\cup$ E) =skp(P2 $\cup$F)	B1 and B2 are strongly equivalent wrt A1=A2.	AS(P $\cup$ E1) =AS(P $\cup$ E2)	O1$\in$skp(P $\cup$ E) $\Leftrightarrow$ O2$\in$skp(P $\cup$ E)

* Ext(B,H)=Th(B $\cup$ S) where S is a maximal subset of H s.t. B $\cup$S is consistent.

Summary of Results
Computational Complexities (Propositional Case)

Logic	Background Theory		Explanation	Observation
	explainable	explanatory		
FOL (B1,H1) v.s. (B2,H2)	Π^P_2-complete*	coNP-complete	coNP-complete	coNP-complete
ALP (credulous) ⟨P1, A1⟩ v.s. ⟨P2, A2⟩	Π^P_2-hard (in Δ^P_3)	Π^P_2-complete	Π^P_2-complete	Σ^P_2-complete
(skeptical)	Π^P_2-hard (in Δ^P_4)	Π^P_2-complete	Π^P_2-complete	Π^P_2-complete

*Testing explainable equivalence of Horn programs is tractable.

Conclusion

- Logical equivalence characterizes equivalence problems in first-order abduction. In abductive LP, strong equivalence, weak equivalence, and other equivalence notions characterize different problems.
- What makes comparison of abductive programs more complicated is nonmonotonicity in ALP, which also makes computational task of equivalence testing harder than first-order abduction in general.
- From the viewpoint of program development, program transformations such as unfold/fold do not preserve strong equivalence of programs. Hence, they are not used for optimizing background theories without changing the results of abduction/induction in general.