Inductive Equivalence of Logic Programs

Chiaki Sakama
Wakayama University
Katsumi Inoue
National Institute of Informatics

ILP 2005. 8. 12

Equivalence issues in Logic Programming

- Identification: identifying different KBs developed by different experts.
- Verification: correct implementation of a given declarative specification.
- Optimization: efficient coding of a program.
- Simplification: Simplifying a part of program without affecting the whole meaning.

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 results of non-deductive reasoning between programs is meaningful and important to know relative intelligence between agents.

Abductive Equivelence [Inoue and Sakama, IJCAI-05]

- 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.

† They provide necessary and sufficient conditions for abductive equivalence in FOL and abductive logic programming.

Example

- $P1=\{ a \rightarrow p \}$, $H1=\{a, b\}$ and $P2=\{ b \rightarrow p \}$, $H2=\{a, b\}$ are explainably equivalent, but not explanatorily equivalent; because p, a, b are all explainable in (P1,H1) and (P2,H2), but p is explained by a in P1, but is not explained by a in P2.

Equivalence Issues in ILP

- When can we say that induction with a background theory is equivalent to induction with another background theory?
- When can we say that induction from a set of examples is equivalent to induction from another set of examples?
- When can we say that induced hypotheses are equivalent to another induced hypotheses?
- Do conditions for these equivalence depend on underlying logics?

Inductive Equivalence

- A background theory B1 is said inductively equivalent to another background theory B2 if B1 and B2 induce the same hypothesis H in face of any set E of examples.

Inductive Equivalence
When useful?

- If an agent has a program B1 that is inductively equivalent to a program B2 of another agent, these two agents are considered equivalent wrt inductive capability.
- If a program B1 is optimized to another syntactically different program B2, inductive equivalence of two theories guarantees identification of results of induction from each theory.

Inductive Equivalence
Underlying Logics

- Conditions for inductive equivalence are argued in different logics of background theories.
- In this study, we consider 3 different logics:
  - (Full) Clausal Theories
  - Horn Logic Programs
  - Nonmonotonic Extended Logic Programs

Problem Setting

- **Logical Setting:**
  - a logical theory B in the Herbrand universe
  - Mod(B) : the set of all Herbrand models
  - SEM(B) : the set of canonical models selected from Mod(B)
  - $B \models_L F$ if F is true in any $I \in SEM(B)$ (under a logic L)

- **Induction Problem**:

  Given a background theory B and a set E of examples;

  find a hypothesis H such that

  - $B \cup H \models_L E$
  - B U H is consistent.

Equivalence Relations

- Let B1 and B2 be two theories which have the common underlying language.
  - B1 and B2 are logically equivalent (B1 $\equiv$ B2) if Mod(B1)=Mod(B2).
  - B1 and B2 are weakly equivalent (B1 $\equiv_w$ B2) if SEM(B1)=SEM(B2).
  - B1 and B2 are strongly equivalent (B1 $\equiv_s$ B2) if B1 U Q $\equiv_w$ B2 U Q for any theory Q.

† B1 $\equiv_s$ B2 implies B1 $\equiv_w$ B2

Example

B1 : $a \vee b$, $c \vee \neg a$, $c \vee \neg b$;

B2 : $a \vee b$, c ;

B3 : $a \vee b$, $\neg a \vee \neg b$, c.

- Mod(B1)=Mod(B2)={{a,c},{b,c},{a,b,c}}, and

  Mod(B3)={{a,c},{b,c}}.

- B1 $\equiv$ B2, B1 $\not\equiv$ B3, B2 $\not\equiv$ B3 .

- If we set SEM(Bi)=MM(Bi), MM(B1)=MM(B2)=MM(B3).

  Then, B1 $\equiv_w$ B3, B2 $\equiv_w$ B3, but B1 $\not\equiv_s$ B3, B2 $\not\equiv_s$ B3,

  because the addition of $\{a \wedge b\}$ makes B3 inconsistent.

Inductive equivalence Definition

- Two theories B1 and B2 are inductively equivalent under a logic L if it holds that

  $$B1 \cup H \models_L E \quad \text{iff} \quad B2 \cup H \models_L E$$

  for any set E of examples and for any hypothesis H such that $B1 \cup H$ and $B2 \cup H$ are consistent.

Results in Full Clausal Logic

- **Theorem**: Two clausal theories B1 and B2 are inductively equivalent under clausal logic iff B1 ≡ B2.
- **Theorem**: Two clausal theories B1 and B2 are inductively equivalent under the minimal model semantics iff B1 ≡ B2 .
- **Corollary**: Two clausal theories are inductively equivalent under clausal logic iff they are inductively equivalent under the minimal model semantics.
- **Corollary**: Deciding inductive equivalence of two propositional clausal theories is coNP-complete.

Results in Horn Logic Programs

- Two Horn LPs B1 and B2 are inductively equivalent (under the least model semantics) if $B1 \cup H \models_{LM} E$ iff $B2 \cup H \models_{LM} E$ for any set E of examples and for any hypothesis H such that $B1 \cup H$ and $B2 \cup H$ are consistent.
- **Theorem**: Two Horn logic programs B1 and B2 are inductively equivalent iff $B1 \equiv B2$.
- **Corollary**: Deciding inductive equivalence of two propositional Horn LPs is done in polynomial-time.

Induction in Nonmonotonic LPs

- An Extended Logic Program (ELP) is a set of rules: $L_0 \leftarrow L_1, \ldots, L_m, \textbf{\textit{not}}\ L_{m+1}, \ldots, \textbf{\textit{not}}\ L_n$ where each L_i is a literal and ***not*** represents negation as failure.
- A declarative semantics of an ELP is defined as the collection of answer sets (Gelfond and Lifschitz, 90).

Results in Extended LPs

- Two ELPs B1 and B2 are inductively equivalent (under the answer set semantics) if $B1 \cup H \models_{AS} E$ iff $B2 \cup H \models_{AS} E$ for any set E of examples and for any hypothesis H such that $B1 \cup H$ and $B2 \cup H$ are consistent.
- **Theorem**: Two function-free ELPs B1 and B2 are inductively equivalent iff $B1 \equiv_s B2$.
- **Corollary**: Deciding inductive equivalence of two propositional ELPs is coNP-complete.

Comparison of Results

Full Clausal Theory	general	logical equiv.
	CF-induction [Inoue, 04]	logical equiv.
	confirmatory induction	logical equiv.
Horn LPs	general	logical equiv.
	RLGG(GOLEM)	weak equiv.
	IE(Progol)	logical equiv.
Extended LPs (f-f case)	general	strong equiv.
	categorical [Sakama, 05]	weak equiv.
	E: ground atoms [Otero,02]	Mod(B1)=Mod(B2)

Discussion

Connection to Abductive Equivalence

- In FOL, logical equivalence of two theories is necessary and sufficient for explanatory equivalence.
- In nonmonotonic LPs, strong equivalence of two programs is necessary and sufficient for explanatory equivalence.
- In clausal logic, inductive equivalence coincides with explanatory equivalence if one permits arbitrary clauses as abductive hypotheses.

Discussion

Difference from Abductive Equivalence

- In abduction, a hypothesis space H is prespecified as abducibles.
- This leads to the characterization by relative strong equivalence, i.e., strong equivalence with respect to H.

  ($B1 \cup Q \equiv_w B2 \cup Q$ for any theory $Q \subseteq H$).
- In ALP, abducibles and observations are restricted to ground literals.

Discussion

Abduction, Induction, Strong Equivalence

- Abduction and induction are both ampliative reasoning and extend theories.
- Strong equivalence takes the influence of addition of rules to a program into account.
- So it succeeds in characterizing the effect of abduction/induction that are not captured by weak equivalence of programs.

Discussion

Computational Viewpoint

- Testing inductive equivalence of propositional clausal theories is converted to UNSAT testing. In propositional Horn LPs, it is tractable.
- Testing strong equivalence of propositional nonmonotonic LPs is done using SAT solvers.
- Inductive equivalence is computed when background KB is given as a function-free Datalog or a database of propositional sentences.

Discussion

Application

- Testing correct/complete-ness of an algorithm:
  If two strongly equivalent programs produce different hypotheses in face of some common examples, it indicates that the algorithm is incomplete or incorrect.
- Comparison of different induction algorithms:
  If one algorithm can distinguish two theories and another one cannot, the former is inductively more sensitive than the latter.

Discussion
Program Development

- Some basic program transformations (e.g., unfolding/folding) do not preserve strong equivalence of logic programs.
- Such basic transformations are not applicable to optimize background theories. If applied, the result of induction may change in general.
- Those transformations are still effective if one use induction algorithms that require the condition of weak equivalence.

Conclusion

- Inductive equivalence compares inductive capabilities between different background theories.
- Strong equivalence is useful to characterize equivalence in non-deductive reasoning.
- Other equivalence issues (regarding examples, hypotheses, etc) are left open.