Heuristic Inverse Subsumption in Full-clausal Theories

Yoshitaka Yamamoto[1], Katsumi Inoue[2], Koji Iwanuma[1]

[1] University of Yamanashi
4-3-11 Takeda, Kofu-shi, Yamanashi 400-8511, Japan.
[2] National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku, Tokyo 101-8430, Japan.

1 Introduction

Given a background theory B and examples E, the task of *explanatory induction* [1–3, 13] is to seek for a consistent hypothesis H such that $B \wedge H \models E$. Based on its equivalent relation $B \wedge \neg E \models \neg H$, H is derivable from the input theories B and E. This approach is called *inverse entailment* (IE) [3, 5].

IE consists of two procedures: construction of a *bridge theory* F and generalization of F into H. Note that F is an intermediate theory satisfying the condition $B \wedge \neg E \models F$ and $F \models \neg H$. Once a bridge theory F is constructed, H is generated via the theory F based on the *generalization relation* $\neg F \dashv H$.

Every IE-based method [3–5, 7–9, 12] can be divided into two kinds in accordance with the generalization relation used in it. On the one hand, systems like CF-induction [3] use the inverse relation of entailment, called *anti-entailment*. They can find any hypothesis H such that $\neg F \dashv H$, but need several nondeterministic operators like *inverse resolution* [6]. On the other hand, systems like Progol [5, 12], HAIL [7, 8], Imparo [4] and Residure procedure [9] generate H with the inverse relation of subsumption $\neg F \preceq H$, called *anti-subsumption*. In the following, we call this restricted approach *inverse subsumption* (IS) [11] to distinguish it with the general IE. IS methods focus the search space on the subsumption lattice bounded by $\neg F$ due to the computational efficiency, but may fail to generate relevant hypotheses, unlike complete IE methods.

For this trade-off between IE and IS, it has been shown that anti-entailment $\neg F \dashv H$ can be logically reduced to anti-subsumption $F^* \preceq H$ [11]. Note that F^* is an alternative theory logically equivalent to the original $\neg F$. This logical reduction means that IS can become a complete explanatory procedure in full-clausal theories only by replacing $\neg F$ with F^*.

This paper aims at investigating how this complete IS works well in practice. Unfortunately, empirical evaluations have not been done yet, because the previous work [11] does not mention how to systematically search the subsumption lattice in full-clausal theories. In this paper, we provide a heuristic IS algorithm based on the notion of *bottom generalization* (BG) used in Progol-like ILP systems [5, 10, 12]. The key techniques of BG lie in the restriction of the search space and the heuristic lattice search. First, BG uses a language bias, called a *mode declaration*, which syntactically describe the target hypotheses. Next, BG defines a heuristic function to evaluate hypotheses in accordance with their coverage of examples and description length. Last, BG performs so-called A^**-like* algorithm that searches the subsumption lattice in the best-first search manner.

Those BG techniques are originally used only for finding hypotheses in Horn clauses. In turn, they are extended so as to be applicable for finding hypotheses in full-clausal theories. We have implemented them embedded into the complete IS method. In this paper, we first sketch our IS system in brief, and then show its experimental result suggesting that the complete IS method indeed generates better hypotheses which are beyond reach for the previous IS methods.

2 Background

2.1 Notion and terminologies

Here, we review the notion and terminology in ILP [13]. A *clause* is a finite disjunction of literals which is often identified with the set of its disjuncts. A clause $\{A_1, \ldots, A_n, \neg B_1, \ldots, \neg B_m\}$, where each A_i, B_j is an atom, is also written as $B_1 \wedge \cdots \wedge B_m \supset A_1 \vee \cdots \vee A_n$. A *Horn clause* is a clause which contains at most one positive literal; otherwise it is a *non-Horn clause*. It is known that a clause is *tautology* if it has two complementary literals. A *clausal theory* is a finite set of clauses. A clausal theory is *full* if it contains at least one non-Horn clause. A clausal theory S is often identified with the conjunction of its clauses and is said to be in *Conjunctive Normal Form* (CNF).

Let C and D be two clauses. C *subsumes* D, denoted $C \succeq D$, if there is a substitution θ such that $C\theta \subseteq D$. C *properly* subsumes D if $C \succeq D$ but $D \not\succeq C$. For a clausal theory S, μS denotes the set of clauses in S not properly subsumed by any clause in S. Let S and T be clausal theories. S (*theory-*) *subsumes* T, denoted by $S \succeq T$, if for every $D \in T$, there is a clause $C \in S$ such that $C \succeq D$.

When S is a clausal theory, the complement of S, denoted by $\overline{S}$, is defined as a clausal theory obtained by translating $\neg S$ into CNF using a standard translation procedure [13]. (In brief, $\overline{S}$ is obtained by converting $\neg S$ into prenex conjunctive normal form with standard equivalence-preserving operations and Skolemizing it.) Note that the complement $\overline{S}$ may contain redundant clauses like tautologies or subsumed ones. Especially, we call the clausal theory consisting of the non-tautological clauses in $\mu\overline{S}$ the *minimal complement* of S, denoted by $M(S)$.

We give the definition of *hypotheses* in the logical setting of ILP as follows:

Definition 1 (Hypotheses). Let B and E be clausal theories, representing a background theory and (positive) examples, respectively. Then H is a *hypothesis wrt* B *and* E if and only if H is a clausal theory such that $B \wedge H \models E$ and $B \wedge H$ is consistent. We simply call it a "hypothesis" if no confusion arises.

Inverse entailment (IE) is a fundamental approach to find hypotheses in Definition 1. It generates a hypothesis H via some bridge theory F, defined as follows:

Definition 2 (Bridge theories). Let B and E be a background theory and examples. A ground clausal theory F is a *bridge theory* wrt B and E if $B \wedge \neg E \models F$ holds. If no confusion arises, we simply call it a "bridge theory".

2.2 Inverse subsumption with minimal complements

After constructing F, IE methods generate H with anti-entailment $\neg F \Dashv H$. Here, we logically reduce IE into IS using the notion of *induction fields* as follows.

Definition 3 (Induction fields). An *induction field*, denoted by $\mathcal{I_H} = \langle \mathbf{L} \rangle$, where $\mathbf{L}$ is a finite set of literals to appear in ground hypotheses. A ground hypothesis H_g *belongs to* $\mathcal{I_H}$ if every literal in H_g is included in $\mathbf{L}$. Given an induction field $\mathcal{I_H} = \langle \mathbf{L} \rangle$, $Taut(\mathcal{I_H})$ is defined as the set of tautologies $\{\neg A \vee A \mid A \in \mathbf{L}\ and\ \neg A \in \mathbf{L}\}$.

Definition 4 (Hypotheses wrt $\mathcal{I_H}$ and F). Let H be a hypothesis. H is a *hypothesis* wrt $\mathcal{I_H}$ and F if there is a ground hypothesis H_g such that H_g consists of instances from H, $F \models \neg H_g$ and H_g belongs to $\mathcal{I_H}$.

Theorem 1. [11] Let H be any hypothesis wrt $\mathcal{I_H}$ and F. Then, it holds that

$$H \succeq M(F \cup Taut(\mathcal{I_H})).$$

Theorem 1 shows that inverse subsumption (IS) can derive any hypothesis H only by adding tautologies associated with $\mathcal{I_H}$ to the original bridge theory F.

Example 1. Let B, E, and $\mathcal{I_H}$ be as follows:

$$\begin{aligned} B &= \{buy(y, diaper) \vee buy(y, beer)\},\ E = \{shopping(y, at\_night)\}, \\ \mathcal{I_H} &= \langle \{\neg buy(y, diaper),\ buy(y, beer),\ \neg buy(y, beer),\ shopping(y, at\_night)\} \rangle. \end{aligned}$$

Note here that $Taut(\mathcal{I_H})$ contains one tautology: $buy(y, beer) \vee \neg buy(y, beer)$.
The following is a hypothesis wrt $\mathcal{I_H}$ and the bridge theory $F = B \wedge \neg E$.

$$H = \{buy(X, diaper) \supset buy(X, beer),\ buy(Y, beer) \supset shopping(Y, at\_night)\}.$$

Though H does not subsume $M(F)$, it subsumes $M(F \cup Taut(\mathcal{I_H}))$ as follows:

$$\begin{aligned} \{ & \boxed{\neg buy(y, diaper) \vee buy(y, beer)} \vee shopping(y, at\_night), \\ & \boxed{\neg buy(y, beer) \vee shopping(y, at\_night)}\ \}. \end{aligned}$$

Like Example 1, the complete IS method *can* generate better hypotheses which are beyond reach for the previous IS methods. On the other hand, it has not been clarified yet how this IS actually works well in the practical point of view.

3 Heuristic inverse subsumption

For the empirical analysis, we provide a heuristic IS algorithm embedded into the complete IS method. It is based on three key techniques of bottom generalization: *mode declarations*, an *evaluation function* and a *heuristic lattice search*, which are used in several state-of-the-art ILP systems like Progol and Aleph [5, 10, 12].

Definition 5 (Full-clausal mode language $\mathcal{L}(M)$). Let M be a set of Progol's mode declarations[3]. A clause $\{A_1, \ldots, A_n, \neg B_1, \ldots, \neg B_m\}$ is *in* the full-clausal mode language $\mathcal{L}(M)$ if each A_i (*resp.* B_j) is an atom belonging to some modeh (*resp.* modeb) declaration in M. A variable of the place-marker +type in some B_j is *complete* if it corresponds to the variable either of +type in some A_i or of -type in some B_k ($k \neq j$); otherwise *incomplete*. A clause in $\mathcal{L}(M)$ is *complete* if there is no incomplete variable in it.

Example 2. Let a set of mode declarations M and types T be as follows:

$$\begin{aligned} M = \ & \{modeh(1, buy(+man, \#item)),\ modeh(1, shopping(+man, \#date)), \\ & \ modeb(1, buy(+man, \#item))\}, \\ T = \ & \{man(y),\ item(diaper),\ item(beer),\ date(at\_night)\}. \end{aligned}$$

We recall H in Example 1. Then, both clauses of H are in $\mathcal{L}(M)$ and complete. In contrast, the clause $buy(X, diaper) \supset buy(Y, beer)$ is in $\mathcal{L}(M)$ but incomplete.

Definition 6 (Evaluation function). Let B, F, $\mathcal{I_H}$ and M be a background theory, a bridge theory, an induction field and a set of mode declarations. Let C be a clause in $\mathcal{L}(M)$. We denote by $cover(C)$, $length(C)$, $incvar(C)$ and $const(C)$ the number of clauses in $M(F \cup Taut(\mathcal{I_H}))$ subsumed by C, the number of literals in C, the number of incomplete variables in C and the consistency status of C wrt B (if C is consistent with B, $const(C) = 1$; otherwise $const(C) = 0$), respectively. Then, the evaluation function $f(C)$ is defined as follows:

$$f(C) = p_1 * cover(C) - (p_2 * length(C) + p_3 * incvar(C) + p_4 * const(C)),$$

where each p_i ($1 \leq i \leq 4$) is a parameter with a non-negative value (1 in default).

Usiing the notions of mode declarations and the evaluation function, we generate each hypothesis clause one by one in the best-first search manner as follows:

Input: B, F, $\mathcal{I_H}$ and M
Output: A consistent hypothesis H wrt $\mathcal{I_H}$ and F in $\mathcal{L}(M)$
Step 1. $\bot := C$; //C is any clause in $M(F \cup Taut(\mathcal{I_H}))$.
Step 2. $S := \{\Box\}$; //S is the set of candidate clauses.
Step 3. $T := \Box$; //T is the best candidate clause in S.
Step 4. $H := \{\Box\}$; //H is a hypothesis.
Step 5. $while(!terminate(T, H, B))$ do
//$B \cup H \cup \{T\}$ is inconsistent or T is incomplete.
$S := S \cup refine(T, \bot, M)$; $\cdots (\star)$
$T := best(S)$; //$f(T)$ has the maximal value among S
Step 6. Remove the clauses from $M(F \cup Taut(\mathcal{I_H}))$ subsumed by T;
Step 7. If $M(F \cup Taut(\mathcal{I_H}))$ is empty return H; otherwise go to Step 1;

Fig. 1. A heuristic IS algorithm

($\star$) $refine(T, \bot, M)$ compute all the refinements each of which is of form $T \cup \{l\}$ subsuming $\bot$, where l is the atom belonging to some mode declaration in M.

[3] Due to space limitations, we refer to [5, 12] for their concrete definition.

4 Empirical evaluation

We have implemented the heuristic IS algorithm of Fig. 1. Using our IS system, we empirically investigate how the complete IS works well in practical examples. Here, we show the experimental result obtained by using one material[4] to learn the concept of "addition of numbers". We give 10 positive examples E on the sum of two numbers, like $plus(1,1,2)$ meaning that $1+1=2$, as well as 16 facts in the background theory B that contains the information on the successor relation between numbers and the relation that $0+X=X$ for each number X. Given B and E, we can consider the recursive rule that $X+Y=Z$ if $X+(Y-1)=(Z-1)$ as one collect hypothesis wrt B and E. Though the problem size is very small, it is difficult for the previous IS methods to generate the target recursive rule only with the prior background theory B.

Using this material, we have evaluated the hypotheses obtained by our IS system in the viewpoint of their predictive accuracies. We apply a leave-one-out test strategy, that is, randomly leave out one example and use the rest 9 examples as training data. The predictive accuracy is evaluated by varying the size of randomly chosen training data (9 points ranged from 20% to 100% of the 9 examples). For each training data size, we randomly generate 10 training sets and then compute their success ratio as the predictive accuracy.

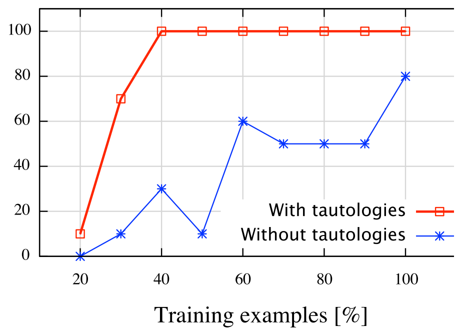

Fig. 2. Predictive accuracy [%]

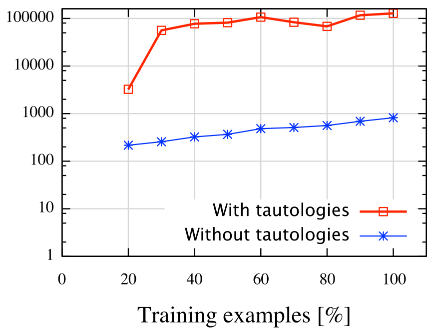

Fig. 3. Generalization time [msec]

Fig. 2 describes the performance of two cases: the one (thick line pointed with squares) generating H from $M(F \cup Taut(\mathcal{I}_{\mathcal{H}}))$ and the other (thin line pointed with stars) from $M(F)$. In other words, the former (*resp.* latter) case is for the complete (*resp.* incomplete) IS method. We then notice that the complete IS succeeds in generating better hypotheses with high (almost 100%) predictive accuracies. In fact, it generated the target hypothesis in more than 40% training data size. On the other hand, its computational cost was much expensive than the incomplete IS without adding tautologies, as shown in Fig. 3. Note that $Taut(\mathcal{I}_{\mathcal{H}})$ contains 10 tautologies. $M(F \cup Taut(\mathcal{I}_{\mathcal{H}}))$ can be blow-up in size as increasing the size of those tautologies, which make the performance inefficient.

[4] Available from: `http://www.iwlab.org/our-lab/our-staff/yy/SampleData`

5 Concluding remarks and future work

There are two generalization approaches for hypothesis finding: inverse entailment (IE) and inverse subsumption (IS). Recently, it has been shown that IE can be logically reduced into a new form of IS, provided that it ensures the completeness of IE. This paper aims at investigating how this complete IS method works well in the practical point of view. For the analysis, we have provided and implemented a heuristic IS algorithm based on the techniques of bottom generalization used in the state-of-the-art ILP systems. We also show one experimental result suggesting that the complete IS method actually finds better hypotheses than the incomplete IS one that does not add tautologies to the original F.

It is an important future work to compare our IS system with the other state-of-the-art ILP systems like Progol using more practical data sets. However, it may not work in case that the number of tautologies increases due to the blow-up of $M(F \cup Taut(\mathcal{I}_{\mathcal{H}}))$ in size. It will be then fruitful to consider if $M(F \cup Taut(\mathcal{I}_{\mathcal{H}}))$ can be treated in some compact representation formalization like BDD.

References

1. De Raedt, L. (1997). Logical setting for concept-learning. *Artificial Intelligence*, 95, 187–201.
2. Flach, P. A. (1996). Rationality postulates for induction. *Proceedings of the 6th international conference on theoretical aspects of rationality and knowledge*, 267–281.
3. Inoue, K. (2004). Induction as consequence finding. *Machine Learning*, 55(2), 109–135.
4. Kimber, T., Broda, K., & Russo, A. (2009). Induction on failure: learning connected Horn theories. *Proc. of the 10th int. conf. on logic programming and nonmonotonic reasoning* (pp. 169–181). LNCS 5753, Springer.
5. Muggleton, S. H. (1995). Inverse entailment and Progol. *New Generation Computing*, 13, 245–286.
6. Muggleton, S. H., & Buntine, W. L. (1988). Machine invention of first order predicates by inverting resolution. *Proc. of the 5th Int. Conf. on ML.* pages 339–352.
7. Ray, O., Broda, K., & Russo, A. M. (2003). Hybrid abductive inductive learning. *Proc. of the 13th int. conf. on inductive logic programming* (pp. 311–328). LNCS 2835, Springer.
8. Ray, O. & Inoue, K. (2008). Mode directed inverse entailment for full clausal theories. *Proc. of the 17th int. conf. on inductive logic programming* (pp. 225–238). LNCS 4894, Springer.
9. Yamamoto, A. (2003). Hypothesis finding based on upward refinement of residue hypotheses. *Theoretical Computer Science*, 298, 5–19.
10. Srinivasan, A. (2007). The Aleph Manual. University of Oxford, Oxford.
11. Yamamoto, Y., Inoue, K., & Iwanuma, K. (2012). Inverse subsumption for complete explanatory induction. *Machine Learning*, 86, 115–139.
12. Tamaddoni-Nezhad, A., & Muggleton, S. H. (2009). The lattice structure and refinement operators for the hypothesis space bounded by a bottom clause. *Machine Learning*, 76, 37–72.
13. Nienhuys-Cheng, S., & De Wolf, R. (1997). *Foundations of inductive logic programming.* LNCS 1228, Springer.