Inductive Extension of Abduction

Chiaki Sakama*

1 Introduction

Abduction and induction are two forms of commonsense reasoning in artificial intelligence. Generally, abduction infers explanatory facts for an observation using known rules in a background knowledge base, while induction infers new rules for an observation using known facts in a knowledge base.

In the context of logic programming, Muggleton and Raedt [1994] distinguish two inferences as

> in inductive logic programming (ILP), the facts (examples) are usually assumed to be stable and the clauses are to be learned; in contrast, in abductive logic programming (ALP), the clauses are stable and the facts are to be learned.

This observation comes from the fact that in standard ALP [Kakas, *et al.*, 1992], hypotheses are restricted to atoms which are pre-specified as *abducibles*. However, distinction on the sentential form of hypotheses is not necessarily the point. In fact, the ALP framework is easily extended to include *abducible rules* [Inoue and Sakama, 1994]. Namely, a set of hypothetical clauses are specified in advance, which are used in the process of abduction. Technically, such an extension is realized in the framework of standard ALP. Given a knowledge base K, if one wants to consider a hypothetical clause

$$C: \ H \leftarrow B,$$

the same effect is achieved by introducing the clause

$$C': \ H \leftarrow B, A$$

to K and considering the new atom A as an abducible. Then, C is abduced in K iff A is abduced in $K \cup \{C'\}$.

Thus, the point is not in the form of hypotheses but in the process of hypotheses generation. In abduction, sources of hypotheses are included in a knowledge base and the task is finding *best* explanations from those candidates. In contrast, in induction no such candidate is given in advance and the task is forming hypotheses from the beginning. Borrowing Dimopoulos and Kakas's [1996] words,

> abduction *extracts* hypotheses from a theory, while induction *constructs* hypotheses using information from the theory.

Thus abduction and induction are generally different on the one hand, while they are not independent of each other on the other hand. According to Peirce [1932],

> if we are ever to learn anything or to understand phenomena at all, it must be by abduction that this is to be brought about.

In other words,

> abduction is the process of conjecturing inductive hypotheses, constrained by the requirement that they should comply with the available observation [Flach, 1996].

In this respect, abduction is considered as a step to induction. Then a question is:

> *How do we use abduction in the process of induction?*

or

> *How to compute induction through abduction?*

Answering these questions will contribute to

- enhancing reasoning ability of abduction,
- making induction more feasible, and
- integrating abduction and induction in AI.

As a step to the goal, this paper studies a method of inductive extension of abduction. This position paper sketches the outline of idea, and formal discussion, including proofs of theorems, are presented in the full paper.

2 Abduction vs. Induction

Given a background knowledge base K and a (positive) observation O^+, abduction infers a hypothetical explanation E satisfying

$$K \cup E \models O^+ \qquad (1)$$

*Department of Computer and Communication Sciences, Wakayama University, Sakaedani, Wakayama 640, Japan.
Email: sakama@sys.wakayama-u.ac.jp
URL: http://www.wakayama-u.ac.jp/~sakama

where $K \cup E$ is consistent.

On the other hand, induction is defined as inference satisfying the conditions [Muggleton and Raedt, 1994]:

(Prior Satisfiability)	$K \cup \{O^-\}$ is consistent,
(Posterior Satisfiability)	$K \cup E \cup \{O^-\}$ is consistent,
(Prior Necessity)	$K \not\models O^+$,
(Posterior Sufficiency)	$K \cup E \models O^+$,

where O^- is a negative observation.

Comparing each definition, abduction (1) corresponds to the posterior sufficiency in induction. The prior necessity is not explicitly stated, but it is usually assumed in abduction. Thus, differences between abduction and induction exist in the prior/posterior satisfiability. Namely, traditional abduction takes care of computing explanations for positive observations, while induction considers not only positive ones but negative observations. To fill the gap, the abductive framework is extended as follows.

A knowledge base K is defined as a consistent first-order theory, which is given as a set of *clauses*

$$Head \leftarrow Body$$

where $Head$ is a disjunction and $Body$ is a conjunction of atoms. A clause with the empty head is called an *integrity constraint*, and a clause with the empty body is called a *fact*. A set of atoms $\mathcal{A}$ from K is called *abducibles*.

Let O^+ be an atom representing a *positive observation*, and O^- a denial $\leftarrow A$ with an atom A, representing a *negative observation*. Given a knowledge base K, which is consistent with O^-, a set of atoms $E \subseteq \mathcal{A}$ is an *explanation*[1] of those observations if it satisfies the following conditions:

1. $K \cup E \models O^+$, (2)
2. $K \cup E \cup \{O^-\}$ is consistent.

Namely, E, together with K, implies O^+ and $K \cup E$ satisfies O^-. Note that in traditional abduction (1) the second condition is simply stated as $K \cup E$ is consistent.[2] Similar extensions are in [Ade and Denecker, 1995; Inoue and Sakama, 1995; Dimopoulos and Kakas, 1996].

Example 2.1 Consider the knowledge base K

$$\begin{aligned} &\mathit{wet\text{-}shoes} \leftarrow \mathit{wet\text{-}grass},\\ &\mathit{wet\text{-}grass} \leftarrow \mathit{sprinkler\text{-}on},\\ &\mathit{wet\text{-}grass} \leftarrow \mathit{rained},\\ &\mathit{wet\text{-}umbrella} \leftarrow \mathit{rained}, \end{aligned}$$

with $\mathcal{A} = \{\mathit{rained}, \mathit{sprinkler\text{-}on}\}$. Suppose that we observe that shoes are wet but the umbrella is not wet. The situation is represented as the positive observation $O^+ = \mathit{wet\text{-}shoes}$ and the negative observation $O^- = \leftarrow \mathit{wet\text{-}umbrella}$. Then, these observations have the unique explanation $E = \{\mathit{sprinkler\text{-}on}\}$.

[1] In this paper, explanations mean minimal explanations (wrt set inclusion) unless stated otherwise.

[2] Or O^- is viewed as an integrity constraint in K.

Thus, the abductive framework is extended to handle negative observations. An essential difference between abduction and induction is, as addressed in the introduction, in the process of hypotheses generation.

In abduction, abductive hypotheses E are selected from the abducibles. Abducibles are usually supposed to be any atom appearing only in the body of some clause in K (i.e., dead-end atoms), or any atom appearing in the head of some fact in K (i.e., extensional atoms). Introducing abducibles makes computation of abduction easier, but it restricts possible hypotheses in advance.

In induction, on the other hand, hypotheses E is *newly* produced from K with respect to a given observation. In contrast to abduction, no candidate hypotheses are given in advance, and stress is put on the discovery of new knowledge. However, induction algorithms are generally expensive and will produce many candidate hypotheses in a large knowledge base.

To relax the restriction of abduction, introducing a mechanism of abducing not only pre-supposed facts but new rules will be useful. Such an inductive extension enhances the reasoning ability of abduction. From the induction side, inductive extension of abduction is viewed as strengthening inductive *biases* to compute abductive explanations. It will reduce non-determinism in induction and make it much feasible. With this background, in the next section we present a method of computing inductive generalization through abduction.

3 Abduction + Partial Evaluation = Inductive Generalization

Consider the knowledge base

$$\begin{aligned} K_1: \quad & \mathit{fly}(x) \leftarrow \mathit{bird}(x), \mathit{have\text{-}wings}(x),\\ & \mathit{bird}(\mathit{polly}) \leftarrow,\\ & \mathit{bird}(\mathit{tweety}) \leftarrow, \end{aligned}$$

with $\mathcal{A} = \{\mathit{bird}(x), \mathit{have\text{-}wings}(x)\}$.[3] Given the positive observation $O^+ = \mathit{fly}(\mathit{polly})$, abduction computes the explanation $E = \{\mathit{have\text{-}wings}(\mathit{polly})\}$.

Thus, abduction makes O^+ derivable from K_1 by explicitly introducing E. In contrast, we consider achieving the same effect by *implicitly* incorporating hypotheses to the knowledge base.

To this end, we use the technique of *partial evaluation*. Partial evaluation is a program optimization which specializes a program wrt known data. Here we apply the technique to specialize a knowledge base wrt computed explanations.

The process proceeds as follows. In K_1, the first clause has the abducible $\mathit{have\text{-}wings}(x)$ whose instance is in E, then we evaluate K_1 with $\mathit{have\text{-}wings}(x)$ (i.e., unfolding the clause by the abducible). The residual knowledge base becomes

$$K_1^+: \quad \mathit{fly}(x) \leftarrow \mathit{bird}(x),$$

[3] Any abducible with variables is identified with its ground instances.

$$bird(polly) \leftarrow,$$
$$bird(tweety) \leftarrow .$$

Then $fly(polly)$ is implied in K_1^+ without introducing E.

The produced knowledge base K_1^+ is an inductive generalization of K_1. In fact, $fly(x) \leftarrow bird(x)$ is a generalization of $fly(x) \leftarrow bird(x), have\text{-}wings(x)$, which is obtained by *dropping conditions* [Michalski, 1983].

Note that K_1 is evaluated not by $have\text{-}wings(polly)$ but by $have\text{-}wings(x)$, which has the effect of introducing the explanation E in its *general* form. As a by-product, K_1^+ implies facts that are not implied by $K_1 \cup E$, i.e., $K_1 \cup E \not\models fly(tweety)$ but $K_1^+ \models fly(tweety)$. Such an increase of proven facts other than observations is called an *inductive leap*, which is the unique feature of inductive generalization.

The procedure is generally given as follows. Let K be a knowledge base and O^+ a positive observation. Then,

1. Compute an abductive explanation E of O^+.
2. For any clause C from K whose condition has atoms unifiable with abducibles in E, produce the new clause C^+ by dropping all those atoms from the body of C.
3. Replace C by C^+.

The resulting knowledge base K^+ is called an *inductive generalization of K wrt O^+*.

The procedure has the following properties. First, inductive generalization K^+ is computed in a *constructive* manner using abduction, which improves extensive computation of hypotheses in induction. Second, it extends classical abduction in the sense that it constructs general rules which explain an observation. Third, dropping abductive conditions has justification that any rule containing hypotheses is incomplete and subject to revision. Formally, we have the following result.

Theorem 3.1 *$K \cup E \models O^+$ and $O^+ \notin E$ imply $K^+ \models O^+$.*

Namely, the produced knowledge base K^+ can explain the positive observation without introducing extra facts E. Here, if $O^+ \in E$ then O^+ is explained by itself. Such a trivial case is excluded in the above theorem.

4 Abduction + Disjunction = Specialization

Inductive generalization often produces an overly general clause and a knowledge base must be specialized when it implies some negative observation.

Suppose that the negative observation $O^- = \leftarrow fly(tweety)$ is given to the knowledge base

$$K_2: \quad fly(x) \leftarrow bird(x),$$
$$bird(polly) \leftarrow,$$
$$bird(tweety) \leftarrow .$$

Then, $K_2 \cup \{O^-\}$ is inconsistent. To recover consistency, K_2 must be *specialized* to satisfy O^-.

There are several ways to specialize K_2 and block the derivation of $fly(tweety)$. Here we consider weakening abducible facts which cause inconsistency in a knowledge base. In K_2 the fact $bird(tweety) \leftarrow$ implies $fly(tweety)$, then we specialize the fact by introducing *disjunction* as follows:

$$K_2^-: \quad fly(x) \leftarrow bird(x),$$
$$bird(polly) \leftarrow,$$
$$bird(tweety) \vee ab_bird(tweety) \leftarrow .$$

Here, $ab_bird(tweety)$ is a newly *invented* abducible which is uniquely associated with $bird(tweety)$. It has the effect of weakening the cause of O^- to make $K_2 \cup \{O^-\}$ consistent. Such rewriting is called *head refinement* in ILP [Raedt and Lavrac, 1993].

Semantically, the disjunctive clause represents two alternative worlds: one in which the fact $bird(tweety)$ holds and the other in which it does not hold (or, *tweety* is an abnormal bird). In ALP, such disjunctive clauses are useful to incorporate abductive hypotheses in a program [Inoue and Sakama, 1994]. Note that in ILP a new atom is also operationally invented to recover consistency, but its associated meaning is not necessarily clear.

The above procedure is generally given as follows. Let K be a knowledge base and $O^- = \leftarrow A$ a negative observation such that $K \cup \{O^-\}$ is inconsistent. Then,

1. Compute a minimal set of facts $F \subseteq K$, called a *cause* of O^-, such that $K \setminus F \not\models A$.
2. For any fact C in F, produce a disjunctive clause C^- which is obtained by introducing a new abducible to the head of C.
3. Replace C by C^-.

The resulting knowledge base K^- is called a *specialization of K wrt O^-*.

The procedure has the following properties. First, a cause of O^- is computed by abduction in a constructive manner, i.e., a set of facts F is an explanation of A in K (though it is not minimal). Second, it extends classical abduction in the sense that it introduces a mechanism of refining a knowledge base wrt negative observations. Third, weakening abducible facts in a knowledge base seems much safer than specializing non-hypothetical clauses. In the context of database update, abducible facts are considered extensional data which are subject to change. Formally, we have that

Theorem 4.1 *$K^- \cup \{O^-\}$ is consistent.*

Back to the example, notice that the previous positive observation $O^+ = fly(polly)$ is still derived from K_2^-. That is, given the initial knowledge base K_1, inductive generalization produces $K_2 = K_1^+$ which explains $fly(polly)$, then specialization produces K_2^- which is consistent with $O^- = \leftarrow fly(tweety)$.

Generally, given a knowledge base K, a positive observation $O^+ (\notin \mathcal{A})$, and a negative one O^-, let $K^\pm$ be a knowledge base obtained by combining the procedures

of inductive generalization and specialization. When O^+ has an explanation E in K, a minimal set of facts $F \subseteq K$ such that $(K \cup E) \setminus F \not\models O^+$ is called a *cause* of O^+ in K. Also, K is *range-restricted* if any variable in a clause occurs in the body of the clause. Then, the next result holds.

Theorem 4.2 *If a cause of O^+ in K and a cause of O^- in K^+ have no atom in common, and K^+ is range-restricted and consistent, then there is a knowledge base $K^\pm$ such that*

1. *$K^\pm \models O^+$, and*
2. *$K^\pm \cup \{O^-\}$ is consistent.*

Namely, the knowledge base $K^\pm$ explains O^+ and is consistent with O^-. Compare the above results with the extended abduction (2) in Section 2.

The need of range-restricted condition in the above theorem is illustrated below. Consider the knowledge base

$$K_3: \quad H(x) \leftarrow A(x), \quad A(x) \leftarrow B(x).$$

Given $O^+ = H(a)$, K_3^+ becomes

$$K_3^+: \quad H(x) \leftarrow A(x), \quad A(x) \leftarrow .$$

Next, given $O^- = \leftarrow H(b)$, $K_3^\pm$ becomes

$$K_3^\pm: \quad H(x) \leftarrow A(x), \quad A(x) \vee ab_A(x) \leftarrow .$$

Then, $K_3^\pm \cup \{O^-\}$ is consistent, but $K_3^\pm \not\models O^+$. Notice that K_3^+ is not range-restricted.

5 Discussion

In induction problems, it is common that learning concepts do not necessarily have definitions in a prior knowledge base. However, under such general setting induction produces many possible hypotheses and most of which might be useless. In contrast, we assumed a prior knowledge base having rules defining the learning concept and introduced a "modest" extension of abduction. Although the proposed methods are still restrictive compared with general induction systems, they realize constructive computation of induction and enhance the reasoning ability of abduction.

Ade and Denecker [1995] propose a method of integrating abduction and induction. In their approach, inductive hypotheses called *inducibles* as well as abducibles are used in the process of abduction. These inducibles are produced by a general induction algorithm. Dimopoulos and Kakas [1996] provide a method of constructing inductive hypotheses to explain observations. They use abduction to restrict a background theory to a relevant part on which induction is based. Such restriction will help to decrease the number of generated inductive hypotheses.

It is well-known that abduction is closely related to belief revision and database update. However, we think that inductive generalization is not necessarily appropriate for theory revision. This is because inductive generalization generally produces inductive leaps, which do not agree with the policy of minimal change in theory revision. On the other hand, the specialization technique presented in this paper is used for theory contraction. Contraction retracts previously believed knowledge, and the same effect is achieved by introducing disjunction. With this transformation, information at the initial stage is preserved in the revised knowledge base, which would be helpful when one wants to recover the prior knowledge base. This is in contrast to the approaches in [Kakas and Mancarella, 1990; Inoue and Sakama, 1995] where abduction causes removal of facts from a knowledge base.

References

[Ade and Denecker, 1995] H. Ade and M. Denecker. AILP: abductive inductive logic programming. In: *Proc. IJCAI-95*, pages 1201–1207, Morgan Kaufmann.

[Dimopoulos and Kakas, 1996] Y. Dimopoulos and A. Kakas. Abduction and inductive learning. In: L. De Raedt (ed.), *Advances in Inductive Logic Programming*, IOS Press/Ohmsha, 1996.

[Flach, 1996] P. A. Flach. On the logic of induction. Research Report, Tilburg University, 1996.

[Inoue and Sakama, 1994] K. Inoue and C. Sakama. On positive occurrences of negation as failure. In: *Proc. KR'94*, pages 293–304, Morgan Kaufmann.

[Inoue and Sakama, 1995] K. Inoue and C. Sakama. Abductive framework for nonmonotonic theory change. In: *Proc. IJCAI-95*, pages 204–210, Morgan Kaufmann.

[Kakas and Mancarella, 1990] A. Kakas and P. Mancarella. Database updates through abduction. In: *Proc. VLDB'90*, pages 650–661, Morgan Kaufmann.

[Kakas, *et al.*, 1992] A. Kakas, R. A. Kowalski, and F. Toni. Abductive logic programming. *Journal of Logic and Computation*, 2:719–770, 1992.

[Michalski, 1983] R. S. Michalski. A theory and methodology of inductive learning. In: R. S. Michalski, *et al.* (eds.), *Machine Learning: An Artificial Intelligence Approach*, pages 83–134, Tioga Publishing, 1983.

[Muggleton and Raedt, 1994] S. Muggleton and L. De Raedt. Inductive logic programming: theory and methods. *Journal of Logic Programming*, 19&20:629–679, 1994.

[Peirce, 1932] C. S. Peirce. *Collected Papers of Charles Sanders Peirce*, vol.2, C. Hartshorne, *et al.* (eds.), Harvard University Press, 1932.

[Raedt and Lavrac, 1993] L. De Raedt and N. Lavrac. The many faces of inductive logic programming. In: *Proc. ISMIS'93, Lecture Notes in Artificial Intelligence*, 689, pages 435–449, Springer.