Abduction and its relation to constrained induction

R.G. (Randy) Goebel
Department of Computing Science
University of Alberta
Edmonton, Alberta
Canada
goebel@cs.ualberta.ca
WWW: http://www.cs.ualberta.ca/˜goebel

Abstract

Abduction is an unsound rule of inference of the form "from P and $Q \rightarrow P$, infer Q." To emphasize the distinction between abduction and the use of *sound* rules of inference, Q is typically called a *hypothesis*. When combined with backward chaining goal-directed theorem provers, abduction provides the basis for hypothetical reasoning systems which treat observations on an environment as goals to be proven, subsequently identifying necessary hypotheses for their derivation.

As a style of artificial intelligence programming, abductive systems provide an implementation basis for an incredibly wide variety of tasks. Included are analogical reasoning, case-based reasoning, inductive logic programming, plan recognition, and many other styles or modes of reasoning. These different styles of reasoning can be "programmed" in terms of abduction, by isolating their desired results in terms of constraints on the structure of formal language used within the abductive process.

1 Introduction

Abduction is an unsound rule of inference of the form "from P and $Q \rightarrow P$, infer Q." To emphasize the distinction between abduction and *sound* or deductive rules of inference, Q is sometimes called a *hypothesis*, thus abduction provides the basis for hypothetical reasoning systems.

Within any logical system, the formal language component determines what can be expressed. For example, if one can express universal generalizations like $\forall X p(X)$ and particular facts like $p(a)$, then abduction can be used to abduce $\forall X p(X)$, by verifying a relationship like $\forall X p(X), \forall X p(X) \supset p(a) \vdash p(a)$. The form of the language thus determines whether abduction is producing what might be typically recognized as inductive generalizations—a form of induction.

Similary, what is abducible (cf. [Poo89; KKT93]) defines the logical forms preferable during the abductive reasoning process, so, for example, isolating formal statements about similarity amongst collections of sentences provides an abductive basis for case-based reasoning and analogy ([HA87; Goe89]). In summary, whether a reasoning system is perceived as doing abduction or not has a lot to do with the form of the theory from which reasoning proceeds.

2 What is induction?

Induction is generally agreed to be a non-deductive reasoning strategy, and like abduction, distinct from deductive reasoning strategies whose rules of inference are truth preserving. But as even a sketch of abduction shows, it is extremely easy to conceive rules of inference that are not truth preserving—even for informal discussions one requires a more formal framework for identifying induction as a object of study.

As non-deductive reasoning, induction is often described in terms of non-deductive propositional inference rules. For example, from p, infer $p \supset q$, or $q \supset p$. Model theoretically, there are fewer possible interpretations for $T_1 = \{p, p \supset q\}$ than there are for only $T_0 = \{p\}$. In otherwords, we have drawn a conclusion from some evidence, p, that allows us to conclude something more about the way the intended interpretation *might* be. Of course it may be that the intended interpretation is among those models for T_0 but not for T_1. The intended interpretation might just as well have $p, \neg q$, in which case the "guess" that $p \supset q$ is wrong—and is perhaps an inductive inference. In this case, we have made an inductive leap by using a common logical constant $\supset$ to form a tentative sentence that we might guess as being true in the intended interpretation. The question arises as to which of infinitely many such new sentences we might guess in this context? How are we to decide which of many choices to make? Abduction is typically formulated as a inductive framework in which this potential infinity of sentences is conventionally prescribed as potentially acceptable hypotheses or "abducibles."

In addition to constraining the set of acceptable hypotheses, any (further) decision making based on the results of this non-deductive reasoning requires further clarity regarding preference or bias on preferred states of affairs described by possible hypotheses. Intuitively, required is some prescription for what kinds of guesses are intuitively better guesses about the way things might be, given some kind of observations as evidence. For example, as described briefly above, we can shift to a first order language, and argue that from $p(a)$ one might conclude $\forall X, p(X)$. This kind of leap is typically called universal generalization, (or just generalization). If we see one white swan, we might guess that all swans are white. Since this is not the case, at least in our world, the inference is faulty. But there is a certain appeal to it, arising from a desire to use some kind of inference to explain more than just evidence already accumulated. There is a clear sense of induction as a method to theorize, or to cautiously (or from a deductive viewpoint, cavalierly) endeavour to guess about extending truth boundaries in order to predict the way things might be.

3 Abduction as a programming framework for applied reasoning

The programming foundation that this view of abduction-induction provides is naturally expressed within the traditional logic programming framework, where the relationship between a background theory F, a set of hypotheses H and a set of observations O is given as

$$F \cup H \vdash Q \qquad (1)$$

Logic programming in Prolog provides one possible implementation of such a framework, at least for the definite clause subset of first order logic. But it is relatively straightforward to use the same logical architecture for more powerful systems (e.g., [PGA87; Poo88]). Given the generality of the the more general logic programming architecture (e.g., [KKT93]), it is hard to sympathize with research that artificially seeks to distinguish, for example, analogical, case-based, and explanation-based reasoning as so distinct when such a simple underlying framework provides the basis for their programming specification.

In fact, it is simple to see that *any* form of induction, including all learning theories and any form of inductive logic programming, can be viewed as some more or less complex constraints on the suitability of a hypothesis set H, in the context of the simple framework provide by the logical relationship specified as equation (1).

4 Conclusion

Abduction is a logical method for isolating interesting hypotheses, and so is naturally applicable in *every* situation where goal-directed reasoning proceeds within the context of uncertain or incomplete information. What is typically identified as a kind of "bias" in learning research can be at least partially identified with preferences for what kinds of sentences are abducible. This simple framework helps focus on the problem of identifying preferred hypotheses, as an alternative to promoting various new reasoning strategies as being distinct forms of reasoning.

Acknowledgements

Koichi Furukawa has been especially influential in his patient guidance regarding the fundamental importance of abduction in hypothetical reasoning and inductive logic programming. The work of the Imperial College logic programming "family" (you know who you are) has provided a solid scholarly foundation for the logic programming foundation of abductive reasoning.

References

[Goe89] R. Goebel. A sketch of analogy as reasoning with equality hypotheses. In K. Jantke, editor, *Analogical and Inductive Inference*, volume 397 of *Lecture Notes in Computer Science*, pages 243–253. Springer Verlag, Berlin, 1989.

[HA87] M. Haraguchi and S. Arikawa. Reasoning by Analogy as a Partial Identity between Mo dels. In K.P. Jantke, editor, *Analogical and Inductive Inference*, volume 265 of *Lecture Notes in Computer Science*, pages 61–87. Springer-Verlag, New York, 1987.

[KKT93] A.C. Kakas, R.A. Kowalski, and F. Toni. Abductive logic programming. *Journal of Logic and Computation*, 2(6):719–770, 1993.

[PGA87] David Poole, Randy Goebel, and Romas Aleliunas. Theorist: a logical reasoning system for defaults and diagnosis. In N.J. Cercone and G. McCalla, editors, *The Knowledge Frontier: Essays in the Representation of Knowledge*, pages 331–352. Springer Verlag, 1987.

[Poo88] David Poole. A logical framework for default reasoning. *Artificial Intelligence*, 36:27–47, 1988.

[Poo89] D.L. Poole. Explanation and prediction. *Computational Intelligence*, 5(2):97–110, 1989.