On the Difference between Abduction and Induction: A Model Theoretic Perspective

Marc Denecker and **Bern Martens**[1] and **Luc De Raedt**[2]

Departement Computerwetenschappen
Katholieke Universiteit Leuven
Celestijnenlaan 200A, B-3001 Heverlee, Belgium
e-mail : {marcd,bern,lucdr}@cs.kuleuven.ac.be

Abstract. In this paper, we show how an appropriate possible world semantics exhibits the fundamental differences between various forms of logical reasoning such as deduction, abduction and induction. The ensuing model theoretic formalisation provides a solid foundation for exact further discourse on the issue, such as deciding whether a particular instance of reasoning should be classified as abduction or rather as induction.

1 Introduction

In logic and logic programming [17, 13], both inductive and abductive problems are usually phrased as follows:

Given a theory T and (an) observation(s) O,
find a theory Δ such that $T \cup \Delta \models O$.

Many attempts [1, 4, 8, 9] to distinguish between induction and abduction are based on the syntactical conditions Δ has to satisfy. In an *abductive* context, one often (but not always) requires Δ to consist of ground atoms. The label *induction*, on the other hand, is usually (but not exclusively) reserved for applications where Δ contains more complex formulas.

Clearly, the above distinction is not very convincing and it seems that a similar criticism can be levelled against other syntax based criteria. Indeed, any such criterion always suffers from exceptions to the extent that its classification power is virtually annihilated. For example, some systems "abduce" general formulas [6], and, similarly, "induction" sometimes is restricted to ground atoms [19, 18]. As a consequence, it has been advocated that there is no true distinction between abduction and induction or that the latter should simply be viewed as a generalisation of the former (and even deduction) [4].

In this paper, we show how a clear-cut conceptual distinction between deduction, abduction and induction as different reasoning principles *can* be made on *semantical* rather than syntactical grounds.

The remainder of this brief position paper is structured as follows. In Section 2, we first establish a distinction between two different kinds of knowledge one can discern in most knowledge representation applications. We then build upon this foundation to introduce a semantical characterisation of deduction, abduction and induction in Section 3. This exercise will allow us to clearly distinguish the fundamentally different nature of the three forms of inference. Finally, Section 4 illustrates the operation of the proposed framework.

2 Domain Knowledge and Scenario Knowledge

We consider settings where knowledge and reasoning deal with a class of scenarios all obeying the same general laws. In other words, some world has to be modelled, and a *scenario* can be considered as one possible state (in some contexts literally one possible sequence of events, hence the term "scenario") of this world. It is then possible, and even natural, to distinguish between *domain and scenario knowledge*. The former embodies the general laws that have to be satisfied in all possible scenarios. The latter, on the other hand, is more specific and singles out a subclass of scenarios (possibly even just one) reflecting this particular information.

In many areas of Artificial Intelligence and Knowledge Representation, information is structured in this way, and the distinction between domain and scenario knowledge makes sense.

- In Inductive Logic Programming [17], one distinguishes between background theory and rules to be learned, in other words domain knowledge, on the one hand, and scenario knowledge in examples, on the other hand.
- In Planning [16] in the event calculus, domain knowledge comprises possible actions, their pre-conditions, and the way they affect given properties. Scenario information is available as certain initial and final situations, and has to be further derived as the plan to get from the one to the other.
- Model-Based Diagnosis [11, 20] features a general theory describing the system's functioning in various circumstances, while specific fault observations provide scenario knowledge.
- In relational databases [22], the distinction general vs. specific is embodied in integrity constraints vs. relation exten-

[1] Partially supported as a postdoctoral fellow by the K.U.Leuven Research Council, Belgium, and partially by the HCM Network "Logic Program Synthesis and Transformation", contract nr. CHRX-CT-93-00414, University of Padova, Italy.
[2] Postdoctoral Fellow of the Belgian National Fund for Scientific Research.

ECAI 96. *12th European Conference on Artificial Intelligence*
Edited by W. Wahlster
Published in 1996 by John Wiley & Sons, Ltd.

sions. In deductive databases [23, 3], the definitions of implicitly defined relations usually represent general domain knowledge.

- We would like to stress that our classification in domain knowledge and scenario knowledge is not based on a distinction between facts and rules. Though scenario knowledge is by definition more specific than general domain knowledge, rules and formulas may represent scenario knowledge while facts may represent general domain knowledge.

Sometimes, a clear-cut distinction between general versus scenario knowledge is not easy; rather it seems that a human expert's knowledge is organised in a hierarchy, from very secure, general knowledge gradually down to more concrete knowledge. For lack of space, we do not elaborate this more general scheme here. We expect it to be easy to upgrade the framework presented in the next section in such a way that it can cope with such a knowledge hierarchy.

3 Possible Worlds Exhibit Abduction vs. Induction

A widespread intuition on the difference between abduction and induction is that the first derives concrete hypotheses representing scenario knowledge while the second derives general theories representing general domain knowledge (e.g. [8]). The distinction between domain and scenario knowledge can be formalised through the use of Kripke *possible world models* [14]. The latter are Kripke models where no restrictions are imposed on the accessibility relation between different worlds (or, in classical terminology, interpretations): It is the total binary relation on the set of possible worlds or interpretations comprising the model.

In our context, one possible world or interpretation can be understood as a mathematical abstraction of a particular scenario, while an entire possible world model reflects the given domain knowledge through the restrictions on the interpretations it contains, in other words through deciding exactly which worlds are possible.

Readers less familiar with Kripke models, may want to think of possible world models as the set of all models of a logical formula (hence, a possible world model captures the semantics of the formula), and of a possible world as a logical interpretation. Below, $\mathcal{K}$ denotes a possible world model, M a logical interpretation, F a logical formula, and $M \models F$ states that the interpretation M is a model for the formula F.

The difference between deduction, abduction and induction now becomes clear:

- *deduction*
  Given a possible world model $\mathcal{K}$ and a formula F, verify whether $\forall M \in \mathcal{K} : M \models F$.
- *abduction*
  Abduction is the mechanism for deriving explanations for given observations F. Given a possible world model $\mathcal{K}$ and a formula F, determine a possible world $M \in \mathcal{K}$ such that $M \models F$ or, more generally, determine a subclass $\mathcal{K}' \subset \mathcal{K}$ such that $\forall M \in \mathcal{K}' : M \models F$. In other words, ***abduction refines a given K.***
  Often, one imposes certain conditions on M or $\mathcal{K}'$, such as minimality or maximality.
  The above characterisation is not entirely satisfactory. Our view on abduction is stronger than defined above. As [12] argues, "in general, we usually think that [abductive] explanations give **causes** ..". The role of abduction is to find "causes" for certain observations. Therefore, abductive reasoning applies in contexts where the general domain knowledge includes explicit, and in general, domain dependent *causality information*. This information can be used to reduce or to explain given observations in terms of its causes. Virtually all applications of abduction in AI are of this kind and contain explicit causal information: diagnosis, planning, natural language interpretation, etc..
  This view on abduction (partially) characterises the nature of the information (the general domain knowledge) which is available in an abductive problem. In the absence of causality information, abduction collapses to model generation, or, more generally, to satisfiability checking, which can be characterised in our framework, in the same terms as the above characterisation of abduction.
  A possible critique of our view is that at the moment there is no formal model of what is causality information. Also our framework does not provide a direct answer to that question. However, the gap in the understanding of causality is rapidly closing. Causality is a research topic in which recently much progress has been made, for example in the context of the ramification problem [21, 15, 7].
- *induction*
  In this setting, an entire class Ψ of possible world models is given. Every $\mathcal{K} \in \Psi$ represents one possible combination of general laws that can hold in the actual, "real" world. A set of examples $\mathbf{E}$ (possibly divided in positive and negative ones) renders concrete scenario knowledge. The goal of induction can then be understood as finding a $\mathcal{K} \in \Psi$ that *explains* the examples. Various different notions of explanation have been employed in inductive learning, resulting in different frameworks for induction, most notably ***learning from interpretations*** (as studied in Valiant's pac-learning framework, [24, 2], and in propositional or attribute-value approaches to learning, but see [5]), and ***learning from entailment*** (as in inductive logic programming [17] and e.g. [10]).
  In learning from interpretations, the aim is to induce a hypothesis (a logical formula) from a set E consisting of interpretations that are (resp. are not) a model for the unknown target theory. Learning from E can be formalised as follows in our framework: find a $\mathcal{K} \in \Psi$ such that $\forall e \in E : e \in \mathcal{K}$ (resp. $e \notin \mathcal{K}$ for a negative example e).
  Learning from entailment aims at inducing a hypothesis from a set E of formulae that are (resp. are not) entailed by the unknown target theory. This translates into finding a $\mathcal{K} \in \Psi$ such that for each (positive) example $e \in E$: $\forall M \in \mathcal{K} : M \models e$ (resp. $\neg\forall M \in \mathcal{K} : M \models e$ for a negative example e).
  Notice that in both cases, ***induction refines Ψ.***
  $\mathcal{K}$ can again be subjected to extra conditions like minimality or maximality.

4 Some Applications

4.1 Bias

In an inductive problem, the term *bias* refers to the set of restrictions imposed on the kind of laws that the expert or user allows to be learned. In our framework, the notion of bias can be declaratively defined as the phenomenon that Ψ imposes restrictions on the combinations of general laws that can hold in the real world; i.e. that only certain possible world models $\mathcal{K}$ belong to Ψ and others do not. In practice, the expert has different mechanisms to define the declarative bias. One way is by stating some background theory T; then for all $\mathcal{K} \in \Psi : \mathcal{K} \models T$. Another way is by defining syntactical restrictions on the form of the formulae that can be learned. Such syntactical restrictions may not only be imposed to reduce the search space of the learner but also to assure that simple rather than complex explanations are generated, or that ad hoc explanations are not generated (e.g. often, but not always, an unacceptable ad hoc explanation of a training set E is E itself).

The declarative notion of bias for induction, as defined in our framework, cannot exist for abduction, since in an abductive problem, the possible world $\mathcal{K}$, representing some actual combination of laws is fixed. However, there is an analogous phenomenon. In general, the class of possible explanations returned by an abductive solver is restricted; these restrictions can be of a semantical (due to constraints) or of a syntactical nature.

4.2 Independence of examples

A condition for correct induction is that there are enough independent examples available. This could be guaranteed by random selection of ***statistically independent*** examples. In the framework of section 3, a logical notion of dependence and independence can be defined in a simple and elegant way. An example e depends upon a set of other examples E with respect to Ψ if and only if each $\mathcal{K} \in \Psi$ satisfying E satisfies also e. As a result, if some example e from a training set E depends on $E \backslash \{e\}$, it is possible to drop e from E without losing solutions. This notion of dependence, or rather its inverse notion of independence, can be seen as the logical counterpart of statistical independence.

We intend to illustrate how other meta-reasoning aspects of induction and abduction can be captured in our framework, in a final version of this paper.

5 Conclusion

In this brief "position paper", we have outlined how in many fields of Artificial Intelligence and Knowledge Representation, it is appropriate to distinguish between general domain knowledge and specific scenario information. We have subsequently shown how this distinction can be formalised through a possible worlds semantics. In the latter setting, a formal semantic characterisation of deduction, abduction and induction is feasible. The resulting framework provides a clear and elegant formalisation of the deeply rooted intuition that the above mentioned forms of inference are of a quite different nature indeed.

Finally, we conjecture that a similar approach can contribute to a better understanding of various complex forms of reasoning such as theory revision, knowledge assimilation, model generation, and how they relate to each other.

Acknowledgements

We thank Maurice Bruynooghe and Peter Flach for interesting discussions on the topic of this paper.

REFERENCES

[1] H. Adé and M. Denecker. Abductive Inductive Logic Programming. In *Proceedings of the 14th International Joint Conference on Artificial Intelligence.* Morgan Kaufmann, 1995.

[2] D. Angluin, M. Frazier, and L. Pitt. Learning conjunctions of horn clauses. *Machine Learning*, 9:147–162, 1992.

[3] S. K. Das. *Deductive Databases and Logic Programming.* Addison-wesley, 1992.

[4] L. De Raedt. *Interactive Theory Revision: an Inductive Logic Programming Approach.* Academic Press, 1992.

[5] L. De Raedt and S. Džeroski. First order *jk*-clausal theories are PAC-learnable. *Artificial Intelligence*, 70:375–392, 1994.

[6] M. Denecker and D. De Schreye. SLDNFA: an abductive procedure for abductive logic programs. draft, K.U.Leuven, Submitted to JLP, 1995.

[7] Marc Denecker, Kristof Van Belleghem, and Daniele Theseider Dupré. Dependencies and Ramifications in an Event-based Language. Draft, submitted, 1996.

[8] Y. Dimopoulos and T. Kakas. Abduction and inductive learning. In L. De Raedt, editor, *Advances in inductive logic programming.* IOS Press/Ohmshma, 1996.

[9] P. Flach. *Simply logical - Intelligent Reasoning by Example.* John Wiley & Sons, 1994.

[10] M. Frazier and L. Pitt. Learning from entailment. In *Proceedings of the 9th International Conference on Machine Learning*, 1993.

[11] W. Hamscher, L. Console, and J. de Kleer, editors. *Readings in Model-Based Diagnosis.* Morgan Kaufmann, 1992.

[12] John Josephson. Inductive Generalizations are Abductions. In *To appear at the ECAI96 workshop on Abductive and Inductive Reasoning*, 1996.

[13] A. C. Kakas, R. A. Kowalski, and F. Toni. Abductive logic programming. *Journal of Logic and Computation*, 2(6):719–770, 1992.

[14] S.A. Kripke. Semantical Consideration on Modal Logic. In *Reference and Modality*, pages 63–72. Oxford University Press, 1971.

[15] Fangzhen Lin. Embracing causality in specifying the indirect effects of actions. In C.S. Mellish, editor, *Proceedings of the International Joint Conference on Artificial Intelligence*, pages 1985–1991, 1995.

[16] Lode R. Missiaen, Marc Denecker, and Maurice Bruynooghe. CHICA, an abductive planning system based on event calculus. *Journal of Logic and Computation*, 5(5):579–602, September 1995.

[17] S. Muggleton and L. De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming*, 19/20:629–680, 1994.

[18] S. Muggleton and C. Feng. Efficient induction of logic programs. In *Proceedings of the 1st conference on algorithmic learning theory*, pages 368–381. Ohmsma, Tokyo, Japan, 1990.

[19] J.R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.

[20] R. Reiter. A theory of diagnosis from first principles. *Artificial Intelligence*, 32(1):57–96, 1987.

[21] Y. Shoham. Nonmonotonic reasoning and causation. *Cognitive Science*, 214:213–252, 1990.

[22] J. D. Ullman. *Database and Knowledge-Base Systems, Volume I.* Computer Science Press, 1988.

[23] J. D. Ullman. *Database and Knowledge-Base Systems, Volume II.* Computer Science Press, 1989.

[24] L. Valiant. A theory of the learnable. *Communications of the ACM*, 27:1134–1142, 1984.