Modelling Inductive Learning for Knowledge Acquisition Tasks

Birgit Tausend [1]

Abstract. In this paper, we discuss several models of inductive learning systems to be used in knowledge acquisition. We propose a new, more elaborate model including a set of induction parameters that allow for controlling the learning behaviour of the system. As an example for the declarative representation of induction parameters, we give an overview of representing the language bias in MILES-CTL.

1 Introduction

Inductive learning methods are used in many knowledge acquisition tools, for example to learn rules or strategic knowledge, as in LEAP [10], ASK [3], MOBAL [11], or DISCIPLE [5]. An important question is which model of the learning algorithm should be provided to the knowledge engineer. On the one hand, the model should not be too detailed and it should abstract from details of the implementation. However, if the model is too simplified, it is not very useful because of several reasons. First, a simple model is only appropriate as long as the behaviour of the learning method matches the behaviour expected by the knowledge engineer. A differing behaviour is not comprehensible to a knowledge engineer because he does not know all of the factors that influence the output produced from a particular input. A more elaborate model could not only be useful for a knowledge engineer to explain the system behaviour but also to support learning with additional knowledge. Often the knowledge engineer or the domain expert have some vague idea of rules or know heuristics in the application domain. With a model that reveals more of the influence factors, e.g., how preliminary hypotheses or the background knowledge are treated by the learning method, this knowledge could be exploited. Another argument in favour of more elaborate models of inductive learning is that revealing more parameters of the systems makes it easier to control and adapt a system. A further advantage of developing more elaborate models of inductive learning is that these models could be used to describe inductive learning systems at the knowledge level [2].

In this paper, we aim to develop a more elaborate model of inductive learning. We focus on the first two arguments, i.e. on models that help to understand the behaviour of the system and to use additional knowledge of the domain. First, we show two models often used in machine learning, and we point out some of their weaknesses. Because the development of a more elaborate model requires the investigation of the factors that influence the behaviour of the system, we discuss the factors commonly used in machine learning to describe the bias of a learning system. As these factors are only partially appropriate to solve our problem, we propose a more elaborate model in section 4. In section 5, we present our approach to representing the hypothesis language as an example for the declarative representation an influence factor in the more elaborate model. The last section contains some concluding remarks.

2 Models of Inductive Learning

A very abstract model of the task to be solved by the inductive learning method is described in [6] by:

Given

- the examples E (observational statements),
- a preliminary hypothesis H' (tentative inductive observation),
- the background knowledge B,

find

- a hypothesis H that together with the background knowledge B explains the examples E.

Though this model shown in figure 1 is widely used in inductive learning, it is inappropriate. First, this model works only well as long as the learning method behaves as it is expected by the knowledge engineer. But if it results in hypotheses that are not plausible or comprehensible to the knowledge engineer, he will hardly accept the learning method as a tool to

[1] Institut für Informatik, Universität Stuttgart, Breitwiesenstr. 20-22, D-70565 Stuttgart, Germany

ECAI 94. *11th European Conference on Artificial Intelligence* Edited by A. Cohn
Published in 1994 by John Wiley & Sons, Ltd.

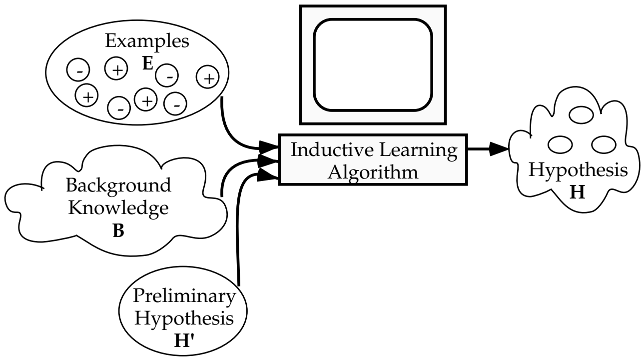

Figure 1. A simple model of inductive learning from examples

support the knowledge acquisition process and will try to use other tools or to solve the problems by hand.
Second, this model does not include the details important for a knowledge engineer. Without knowing the representation language of the input and the output, it is difficult to specify the interface to other tools. In this case, it is difficult to provide faultless inputs to the learning tool and to process its output. Thus, at least the representation language of the examples, the background and the hypotheses should be part of the model. Another detail not clearly specified in this model is the acceptance criterion. In general, it is not sufficient to require that the hypothesis has to explain the examples. In most cases, we need specific acceptance criteria. For example, in the presence of noise or for particular applications, acceptance criteria weaker than strong consistency have to be specified.
These problems can be solved by a more elaborate model similar to that described in [1]:

Given

- the representation language of the examples L_E,
- the representation language of the background knowledge L_B,
- the representation language of the hypotheses L_H,
- the examples E in L_E,
- the background knowledge B in L_B,
- a preliminary hypothesis H' in L_H,
- the acceptance criterion A,

find

- a hypothesis H in L_H that fulfills the acceptance criterion A.

Though this model shown in figure 2 takes into account the representation languages and the acceptance criterion, it is still incomplete with respect to explaining the learning behaviour of an inductive system. There are several other factors that influence which hypotheses are constructed by the system, e.g,. the search procedure used to search the hypothesis space or the order of the hypotheses in the space. These factors are also very important if inductive learning systems should be adaptable to varying applications, and also for the description at the knowledge level.

3 Bias in Machine Learning

In inductive learning, the factors that influence the output of the learning system are summarized by the term bias [7].

> ..., we use the term bias to refer to any basis for choosing one generalization over another, other than strict consistency with the observed training examples.

A similar definition can be found in [17]:

> ...all factors that influence hypothesis selection constitute bias.

A first problem is that this term is quite misleading as it suggests that the system is restricted in some way. However, the intended meaning is contrary. In fact, bias can be seen as the set of parameters that makes the system produce generalizations of the input as hypotheses.

> In retrospect, it is not surprising that an unbiased generalization system cannot make classifications of instances other than the training instances. An unbiased system is one whose inferences logically follow from the training instances, whereas classifications of new training instances do not logically follow from the classification of the training instances. If totally unbiased generalization systems are incapable of making the inductive leap to characterize new instances, the power of a generalization system follows directly from its biases.

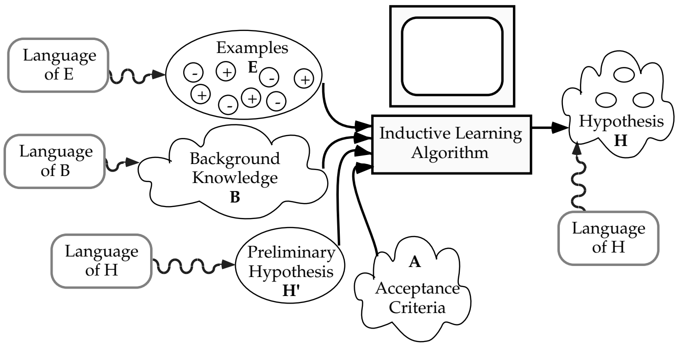

Figure 2. Another model of inductive learning

Thus, we prefer to use the term induction parameters instead of the term bias in this paper.
The influence factors have been more precisely defined in [17].

> ...these factors include the following:
>
> 1. The language in which the hypotheses are described.
> 2. The space of hypotheses that the program can consider.
> 3. The procedures that define in what order hypotheses are to be considered.
> 4. The acceptance criteria that define whether a search procedure may stop with a given hypothesis or should continue searching for a better choice.

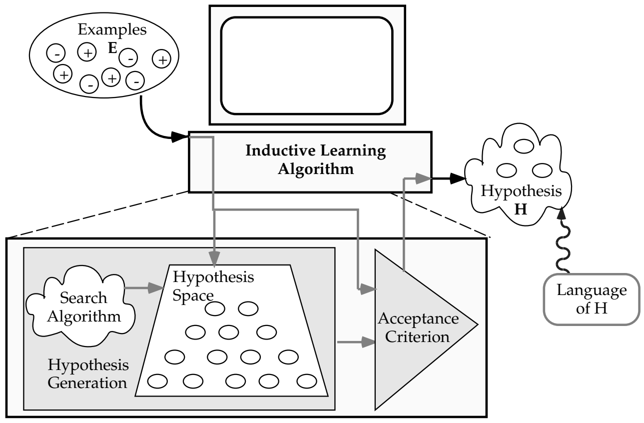

Figure 3. The model according to Utgoff

This definition sketched in figure 3 has several weaknesses. The reason for the first one is historical. Since the definition is closely related to the approaches it models, e.g., the candidate elimination approach in [8] or LEX [9], and these systems do not use background knowledge in learning, the role of the background knowledge has to not be taken into account.
Another shortcoming of the definition is that it describes rather the results of changing the parameters of a learning system than of describing the factors themselves. For example, the hypothesis space is determined by the hypothesis language and the criterion used to order the hypotheses in the space. Both factors influence the learning behaviour of the system as they vary in the systems developed for inductive learning. For example, in Inductive Logic Programming, four different criteria can be used to order the hypotheses space [4], and there is even a greater variety of hypothesis languages used in these systems [15].
These problems are the motivation to try a new definition of a more elaborate model that takes into account the background knowledge and describes the parameters instead of their results.

4 A More Elaborate Model

To define a new, more elaborate model, we combine the second model in section 2 with a set of more appropriate parameters. As in the former model, the examples, the background knowledge, the hypotheses and their languages and the acceptance criteria have to be specified. Since the new model is based on the learning-as-search paradigm, as is the second model in section 2, the parameters of the search procedure have to be specified, e.g., the width of the beam for beam search or selection criteria for heuristic search, e.g., best-first search. The initial hypothesis in the model in section 2 is a parameter that influences the boundary of the search space. Thus, we use the parameter starting hypothesis to subsume preliminary hypothesis and starting hypothesis generated by the learning system. A further parameter is a criterion to order the hypothesis space as this space is determined by the hypothesis language and this criterion. To define the space that is searched by the learning method, we have to specify the operators that are used to proceed in the hypothesis space, i.e. , to generalize or specialize hypotheses.
Thus, the new model can be defined as follows:

Given

- the examples E in L_E,
- the background knowledge B in L_B,
- and the setting of the induction parameters including
  - the representation language of the examples L_E
  - the representation language of the background knowledge L_B,
  - the representation language of the hypotheses L_H,
  - the parameters S^* of the search procedure S,
  - the operators O to proceed in the hypothesis space,
  - the initial hypotheses H',
  - the criterion $\leq_H$ to order the hypotheses,
  - the acceptance criterion A.

Find

- a hypothesis H in L_H that fulfills the acceptance criterion A.

This model sketched in figure 4 seems to be quite complicated but in most cases several parameters are fixed in a particular learning method and cannot be controlled by the knowledge engineer. On the

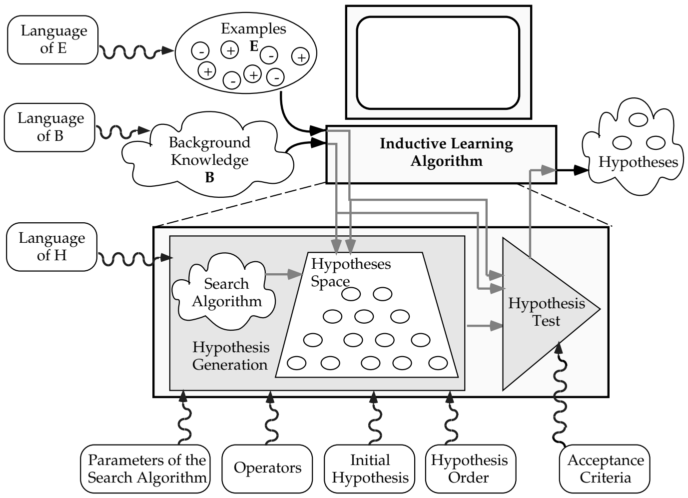

Figure 4. A new, more elaborate model of inductive learning

other hand, some of the parameters are very important and have to be elaborate in more detail. For example, we have to find appropriate representations for the hypothesis language [16] and investigate the influence of restricting the hypothesis language to the system behaviour. Another important observation is that the parameters are not independent. For example, the criterion to order the hypotheses or the acceptance criteria depend on the hypothesis language. Thus, a user has to be quite careful in changing the parameters and we should aim to improve the model in order to describe these relations.

5 Hypothesis Language as an Induction Parameter

Among the induction parameters, the hypothesis language is very important because many other parameters depend on it. Another reason for its importance is that the behaviour of the system in terms of learning speed and learning success strongly varies with respect to the hypothesis language. In particular, powerful hypothesis languages as Horn logic in Inductive Logic Programming (ILP) systems [13] have to be strictly controlled in order to avoid these problems.
Since controlling a parameter requires its declarative representation, we developed a representation for the hypothesis language of ILP systems. This representation MILES-CTL [16] has been implemented as part of the system MILES [14], a modular system for simulating, experimenting and testing ILP methods. The main goals of MILES-CTL are to provide a unifying representation for the various language biases in ILP [15] that is easy to understand and to extend, and enables a concise description of the hypothesis language.

MILES-CTL is a scheme-based approach similar to the approach used in MOBAL [12]. The hypothesis language is described by a set of schemes for the hypothesis clauses. Each scheme called clause template in MILES-CTL consists of literal templates for the head and the body literals of clauses. A literal template consists of several items describing the set of literals it covers, with respect, e.g., to their predicate name, arguments, arity, predicate and argument types or depth in the clause. These items form a record-like structure, i.e., each item consists of an identifier followed by a scheme constant, a scheme variable, or a scheme variable the instantiation of which is restricted by conditions.
For example, the following clause template T covers hypotheses that include an arbitrary head literal because its template includes only scheme variables, and two body literals. The arity of both body literals has to be less than two. The predicate type of the second has to be *comp*, and the first includes only arguments of type *list* or *number*.

$$T: \begin{bmatrix} predicate : P1, \\ arguments : A1 \end{bmatrix} \leftarrow$$

$$\begin{bmatrix} predicate : P2, \\ arguments : A2, \\ arity : Ar2 \| (Ar2 \leq 2) \\ argument\_types : T2 \| (T2 \in \{list, number\}) \end{bmatrix},$$

$$\begin{bmatrix} predicate : P3, \\ arguments : A3, \\ predicate\_type : comp \\ arity : Ar3 \| (Ar3 \leq 2) \end{bmatrix}.$$

A hypothesis language defined in MILES-CTL can be easily adapted by a knowledge engineer by adding and deleting clause templates or by adapting scheme variables, conditions and scheme constants of their literal templates. This representation is useful as well for shifting the bias in order to recover from overly strong biases.

6 Conclusion

In this paper we discussed the need for a more elaborate model of inductive learning methods if these are used in knowledge acquisition. We recalled several models that describe the learning process at different levels of abstraction. In order to overcome the weaknesses of these model, we defined a new, more elaborate model that takes into account the background knowledge and includes the influence factors instead of their effects. An interesting open question is whether this model is useful for describing inductive learning at the knowledge level and for better controlling and adapting learning systems.

ACKNOWLEDGEMENTS

This work has been partially supported by ESPRIT BRA 6020 ILP.

REFERENCES

[1] L. De Raedt and M. Bruynooghe, 'A unifying framework for concept-learning algorithms', *The Knowledge Engineering Review*, **7**(3), 251–269, (1992).

[2] T.G. Dietterich, 'Learning at the knowledge level', *Machine Learning*, **1**(3), 287–316, (1986).

[3] T. Gruber, 'Automated knowledge aquistion for strategic knowledge', *Machine Learning*, **4**(3/4), 293–336, (Dec. 1989).

[4] B. Jung, *Generalisierungsoperatoren bei der induktiven logischen Programmierung*, Diplomarbeit Nr. 940, Fakultät Informatik, Universität Stuttgart, 1992.

[5] Y. Kodratoff and G. Tecuci, 'DISCIPLE-1: Interactive apprentice system in weak theory fields', in *Proc. of the 4th International Conference on Machine Learning*. Morgan Kaufmann, (1987).

[6] R. S. Michalski, 'A theory and methodology of inductive learning.', in *Machine Learning: An Artificial Intelligence Approach*, eds., R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, Tioga, (1983).

[7] T. M. Mitchell, 'The need for biases in learning generalizations', Technical Report TR-117, Department of Computer Science, Rutgers University, (1980).

[8] T.M. Mitchell, 'Version spaces: A candidate elimination approach to rule learning', in *Proc. of 5th International Joint Conference on AI*. Morgan Kaufmann, (1977).

[9] T.M. Mitchell, 'Generalization as search', *Artificial Intelligence*, **18**(2), 203–226, (1982).

[10] T.M. Mitchell, S. Mahadevan, and L.I. Steinberg, 'LEAP: A learning apprentice system for VLSI design', in *Proc. 9th International Joint Conference on AI*. Morgan Kaufmann, (1985).

[11] K. Morik, 'Sloppy modelling', in *Knowledge Representation and Organization in Machine Learning*, ed., K. Morik, Springer, (1987).

[12] K. Morik, S. Wrobel, J. Kietz, and W. Emde, *Knowledge Aquisition and Machine Learning: Theory Methods and Applications*, Academic Press, 1994.

[13] S. Muggleton, 'Inductive logic programming', in *Proc. of 1st Conference on Algorithmic Learning Theory*, Tokio, (October 1990). Ohmsha.

[14] I. Stahl and B. Tausend, 'MILES – a Modular Inductive Logic Programming Experimentation System'. Deliverable STU1.2 of ESPRIT BRA 6020: Inductive Logic Programming (ILP), 1993.

[15] B. Tausend, 'Biases and their effects in inductive logic programming', in *Machine Learning: ECML-94, European Conference on Machine Learning, Catania, Italy*. Springer, (1994).

[16] B. Tausend, 'Representing biases for inductive logic programming', in *Machine Learning: ECML-94, European Conference on Machine Learning, Catania, Italy*. Springer, (1994).

[17] P.E. Utgoff, 'Shift of bias for inductive concept learning', in *Machine learning: An artificial intelligence approach*, eds., R.S. Michalski, J.G. Carbonell, and T.M. Mitchell, Morgan Kaufmann, (1986).