The Knowledge Engineering Review, Vol. 10:1, 1995, 77–81

Integration of Machine Learning and Knowledge Acquisition

CLAIRE NÉDELLEC

Laboratoire de Recherche en Informatique, Groupe Inférence et Apprentissage, Université Paris Sud, bât 490, F-91405 Orsay, France

1 Introduction

"Integration of Machine Learning and Knowledge Acquisition" may be a surprising title for an ECAI-94 workshop, since most machine learning (ML) systems are intended for knowledge acquisition (KA). So what seems problematic about integrating ML and KA? The answer lies in the difference between the approaches developed by what is referred to as ML and KA research. Apart from some major exceptions, such as learning apprentice tools (Mitchell et al., 1989), or libraries like the Machine Learning Toolbox (MLT Consortium, 1993), most ML algorithms have been described without any characterization in terms of real application needs, in terms of what they could be effectively useful for. Although ML methods have been applied to "real world" problems few general and reusable conclusions have been drawn from these knowledge acquisition experiments. As ML techniques become more and more sophisticated and able to produce various forms of knowledge, the number of possible applications grows. ML methods tend then to be more precisely specified in terms of the domain knowledge initially required, the control knowledge to be set and the nature of the system output (MLT Consortium, 1993; Kodratoff et al., 1994).

As opposed to this pragmatic aim, a more theoretical approach to knowledge acquisition has recently started investigating how ML techniques can be taken into account in model-based knowledge acquisition methodologies such as KADS (Van de Velde & Aamodt, 1992), or the generic tasks of Chandrasekaran (1989). According to Van de Velde and Aamodt (1992), a description of the functionality of the ML method at the knowledge level is a prerequisite, whether the ML technique is a modelling tool used during the acquisition phase, or part of the final knowledge-based system (KBS). The role of the domain knowledge, the goals and the operations must be described independently from the implementation.

2 Integration

Notable efforts have been undertaken recently to integrate both approaches from an application point of view (Tecuci et al., 1993), as well as from a modelling point of view (Anjewierden et al., 1992; Van de Velde et al., 1994). The work presented at the ECAI-94 workshop attempted to overcome this dichotomy by drawing some general results from industrial applications and motivating theoretical results by practice. The ML applications presented at the workshop illustrate the integration of ML tools into complex knowledge acquisition processes that are more or less automated. They overcome the naive view that a single and isolated ML system would be able to generate, in one shot, a complete and correct knowledge base to be simply transferred into the final KBS.

The themes that were discussed during the workshop are representative of what seems to be the main emerging issues in ML and KA integration. Some work shows how the integration of ML tools into the knowledge acquisition process is specific to the application goal (section 3). The knowledge engineer has then to specify by himself how knowledge-based tools may cooperate together to develop a given application. Only tools carrying out a single function are considered in this approach. In opposition to this view, two types of integrated approaches are proposed. On the one hand, a *vertical integration* of ML methods in sophisticated systems supports the whole KBS lifecycle (section 4). On the other hand, an *horizontal integration* of ML methods results in libraries or configurable and generic systems (section 5).

Both approaches are described from a problem-solving point of view, but with very different perspectives. Work on vertical integration studies how the problem solving methods of the application may control the knowledge acquisition process. In contrast, horizontal integration requires ML techniques to be described themselves as problem-solving methods to allow precise comparisons. The main underlying theoretical issue thus concerns a knowledge level modelling of ML (section 6) in terms of explicit input, output and goals.

3 ML as KA tool

In the work presented by I. Bratko, ML techniques are applied to knowledge discovery in discrete event simulation models by using both simulators and ML methods. Mladenic et al. (1994) report the result of the analysis and interpretation of the output of a discrete event simulator by a FOL-based and two attribute-based learning systems. The same goal was pursued in three different application domains (supermarket, pub and steelworks domains), namely improving *understanding of the domain expert* in terms of relationships between parameters of the simulated model, the resource usage and potential bottlenecks location. The results obtained by applying ML methods are considered as satisfactory by the domain experts not only because they allow direct improvement of the simulated systems, but also because the analysis of given experiments led to the formation of new target problems and the conducting of new experiments with the learning systems.

G. Bisson, with the ML system KBG, provides a way to support these kind of experiments (Bisson, 1994). KBG provides facilities for the knowledge engineer to modify the input of KBG (e.g. examples and a weighted domain theory) whenever the learning results are not considered as satisfactory. KBG provides traces of the learning phases in the form of *explanations* so that the user may understand how the input knowledge has been interpreted by KBG, and modify it as a consequence.

F. Schmalhofer also presented an application where the ML method participates in the KA process among other knowledge-based tools that are here integrated into a *cooperative KA architecture* called COEX. COEX also provides editors and graphers. Knowledge acquisition with COEX is entirely controlled by the user. COEX has been successfully applied to inducing causal relations in medical diagnosis.

4 Integrated architecture for problem-solving

On the way towards more integrated systems, both approaches proposed by K. Börner and D. Janetzko, and APT systems include in the same *learning apprentice architecture* acquisition tools of training examples, interactive learning tools, and the final knowledge-based system. The work of Börner and Janetzko (1994) is devoted to design, and is applied to industrial building. Task-structures are abstracted from the resolution traces provided by a domain expert, that represent sequences of resolution steps. A generalized case is then learned from each resolution step by case-based learning methods.

APT illustrates more specifically how the whole lifecycle of a KBS can be supported by a cooperative architecture providing knowledge acquisition, validation and maintenance facilities for problem-solving. Learning is performed to improve the performances of the integrated problem-solver by completing and refining the problem-solving task structures and the domain theory. Guidance for learning and knowledge acquisition is alternatively provided by the system and the user depending on the learning task to be performed (Nédellec et al., 1994).

Thus, in both approaches (Börner & Janetzko, 1994; Nédellec & Causse, 1992), learning is strongly guided by the problem-solving task. Learning techniques are viewed as support tools in the final problem-solving system rather than as participants in the initial acquisition process as in KBG or COEX.

The same issue of guiding ML with the help of the problem-solving methods of the domain expert (or the resulting KBS) has been studied by J. Thomas and Y. T. Park. Problem-solving methods are explicitly represented as ML biases in Thomas' system ENIGME. They are modelled as a KADS conceptual model in order to ease the integration of ENIGME as a learning tool into the KADS methodology (Ganascia et al., 1993). A given problem-solving method is represented by a KADS task structure and composed of inference steps. It forms a bias to guide learning in the sense that the example base is split up into subsets that correspond to the various inference steps. The learning algorithm is then applied to each subset of examples. Experimentation on the first bid of the card game bridge compared the unbiased strategies of ID3 and CN2 with the biased strategy of ENIGME using the Charade system as its ML tool. This empirical evaluation clearly showed the superiority of the latter in terms of accuracy.

KADS inference level is also used as a bias for learning in the recent work of Dompseler and Van Sommeren (1994). Y. T. Park and S. H. Kim do not explicitly consider problem-solving methods as bias, but similarly constrain knowledge refinement by metarules that represent strategic domain knowledge (Park & Kim, 1994). They assume that a "domain knowledge gap" can be identified and recovered from, by identifying the failure of a strategic metarule at a higher level of abstraction. The meta-rule base is here somewhat similar to the KADS task level.

5 Flexible ML systems

5.1 Bias language

As opposed to this *vertical* integration of learning techniques into the KA process and more generally into the KBS lifecycle, libraries, multistrategy systems and configurable systems represent a *horizontal* integration of various ML techniques that are able to meet a wide range of KA goals. Flexibility requires an explicit representation of the control level that determine what are the suitable learning operations to apply to a given problem. Eliciting the control of the learning process referred to as *declarative bias* has always been a fundamental issue in ML because it strongly affects the complexity of the learning process and the learning results (Mitchell, 1991; Utgoff, 1986; Grosof & Russel, 1990). Three of the works presented at the workshop also demonstrate the utility of a bias language as a concise and powerful way for the user to explicitly parameterize ML systems instead of tuning low level knowledge such as examples and domain theory.

As shown by B. Tausend, in her system MILES-CTL, language biases that constrain the language of learnable concepts is particularly valuable to restrict drastically the search space generated by expressive concept languages such as first order logic (Tausend, 1994). B. Tausend proposes a language of bias based on second-order schemata to learn Horn clauses as concept definitions in an inductive logic programming framework. This language allows the user to set the form of the clauses to be learned such as the number of literals in the body, the type of the predicates, etc. R. Feldman also proposes a bias language for the system FRST, that is not concerned with the concept definition language but with the condition under which learning operators must apply (Feldman, 1994). The FRST language allows the user to define the learning context in which a given revision operator will be preferably applied to the domain theory. The context includes false/true positive and false/true negative example coverage, topological description of the theory, etc.

5.2 Problem-solving view of ML

P. Albert, as an invited speaker, presented a generic architecture called HAIKU to configure generate-and-test ML algorithms (Rouveirol & Albert, 1994; Nédellec & Rouveirol, 1994). The control level of HAIKU is composed of a set of three types of bias called language, search and validation biases, and a set of learning operators. The instantiation of the biases, plus the example source and the background knowledge completely determine the behaviour of the resulting ML algorithm, and thus the learnable knowledge and performance of the ML algorithm. The modelization of the whole system as a problem-solving method in KADS allows a close integration of HAIKU with the KADS methodology that will make the cooperation with other KA tools easier, while developing a given application.

This approach differs from the learning apprentices or ENIGME, since the descriptions of the biases do not directly stem from the problem-solving methods of the application domain, but is rather determined by the ML algorithm capabilities. On the other hand, the great level of flexibility of this generic architecture enables the simulation of very different methods and knowledge of different types to be learnt. In both cases, giving the user the possibility to set control knowledge leads to two kinds of requirements. First, the control knowledge the user is asked to set *a priori* must be clearly related to his concern so that the resulting learning process can meet his goals and respects his constraints as in COEX, APT and ENIGME. Second, the ML algorithm must be structured in a way that makes explicit the basic learning operations and the control parameters to be set, and how they affect the operations. Work on MILES and HAIKU focus on this last requirement by modelling learning as a problem-solving method at the knowledge level.

6 Knowledge level modelling of ML

As first claimed by Van de Velde and Aamodt (1992), modelling learning at the knowledge level will allow both, to make explicit and implicit learning strategies, and to compare ML methods on a common base for KA purposes. Learning in HAIKU is described at the knowledge level as a two stage operation. The first one consists in defining by an initial bound the hypothesis space to be searched taking into account language biases. The second one consists in searching the hypothesis space by applying a learning operator to the current bound, taking into account examples, search and validation biases. This model allows the representation of the different types of biases that play a role in generate-and-test ML methods and to explicate and study their interaction and their influence on the learning result. B. Tausend and H. Adé works on bias (Adé et al., 1995) belong to the same theoretical framework, since they propose operational definitions and typologies of bias for machine learning.

7 Conclusion

The work presented in Amsterdam is representative in some way of the evolution of both ML and KA communities towards more integrated approaches of theoretical issues such as model-based methodologies, and more practical issues such as the effective application of cooperating KA and ML methods to real life problems.

A common problem-solving view, both to guide the knowledge acquisition process, and to describe the knowledge acquisition process itself, also seems to be a key issue for a strong cooperation of both approaches (Fensel et al., 1995).

References

Adé, H, De Raedt, L and Bruynooghe, M, 1995. "Declarative bias for specific-to-general ILP systems" *Machine Learning* (to appear).

Anjewierden, A, Shadbolt, N and Wielinga, BJ, 1992. "Supporting knowledge acquisition: The acknowledge project" In: *Enhancing the Knowledge Engineering Process—Contributions from ESPRIT,* pp 143–172, Elsevier.

Bisson, G, 1994. "From inductive learning to knowledge acquisition through explanations" In: *Proc. ECAI-94 Workshop on Integration of ML and KA*, pp 1–11.

Börner, K and Janetzko, D, 1994. "Case-based learning for knowledge-based design support" In: *Proc. ECAI-94 Workshop on Integration of ML and KA,* pp 12–18.

Chandrasekaran, B, 1989. "Task-structures, knowledge acquisition and learning" *Machine Learning* **4** 251–346. Kluwer Academic Publishers.

Consortium MLT, 1993. "Final Public Report." Esprit II Project 2154.

Dompseler, HJH and Van Someren, M, 1994. "Using models of problem solving as bias in automated knowledge acquisition." In: AG Cohn (ed.), *Proc. ECAI-94,* pp 499–503.

Feldman, R, 1994. "FRST – an interactive revision system for forward chaining rule bases" In: *Proc. ECAI-94 Workshop on Integration of ML and KA,* pp 19–28.

Feldman, R and Nédellec, C, 1994. "A framework for specifying explicit bias for revision of approximate knowledge bases." In: B Gaines and M Musen (eds.), *Proc. Knowledge Acquisition Workshop (KAW-94),* pp 15.1–15.19.

Fensel, D, Ganascia, JG, Nédellec, C, Plaza, E, Rouveirol, C, Van de Velde, W and Van Someren, M (eds.), 1995. *Knowledge Level Modelling and Machine Learning,* Heraklion, Greece.

Ganascia, JG, Thomas, J and Laublet, P, 1993. "Integrating models of knowledge and machine learning." In: PB Brazdil (ed.), *Proc. ECML 93,* pp 396–401, Vienna.

Grosof, B and Russell, S, 1990. "Shift of bias as non-monotonic reasoning" In: PB Brazdil and K Konolige (eds.), *Machine Learning, Meta-Reasoning and Logics.* Kluwer Academic.

Kodratoff, Y, Moustakis, V and Graner, G, 1994. "Can machine learning solve my problem?" *Applied Artificial Intelligence Journal* **8** (1) 1–31. Taylor & Francis Publishers.

Mladenic, D, Bratko, I, Paul, RJ and Grobelnik, M, 1994. "Knowledge acquisition for discrete event systems using machine learning" In: *Proc. ECAI-94 Workshop on Integration of ML and KA,* pp 29–36.

Mitchell, TM, 1991. "The need for biases in learning generalizations" In: JW Shavlik and TG Dietterich (eds.), *Readings in Machine Learning,* pp 184–191. Morgan Kaufmann.

Mitchell, TM, Mahadevan, S and Steinberg, LI, 1989. LEAP: A learning apprentice for VLSI design. In: *Machine Learning III: An Artificial Intelligence Approach*, pp 271–289, Morgan Kaufmann.

Nédellec, C and Causse, K, 1992. "Knowledge refinement using knowledge acquisition and machine learning methods." In: T Wetter et al. (eds.), *Proc. 6th European Knowledge Acquisition Workshop (EKAW-92),* pp 171–190, Springer-Verlag.

Nédellec, C, Ferreira, JL, Correia, J and Costa, E, 1995. "Machine learning goes to the bank." *Applied Artificial Intelligence Journal* (Special Issue on Machine Learning), Y. Kodratoff (ed.). Taylor & Francis Publishers.

Nédellec, C and Rouveirol, C, 1994. "Specification of the HAIKU system." Report 928, University of Paris-South.

Park, YT and Sung-Hee, K, 1994. "Knowledge base refinement by failure driven approach." In: *Proc. ECAI-94 Workshop on Integration of ML and KA,* pp 37–43.

Rouveirol, C and Albert, P, 1994. "Knowledge level model of a configurable learning system." In: L. Steels et al. (eds.), *Proc. 8th European Knowledge Acquisition Workshop (EKAW-94),* pp 374–393, Springer-Verlag.

Tausend, B, 1994. "Modelling inductive learning for knowledge acquisition tasks." In: *Proc. ECAI-94 Workshop on Integration of ML and KA,* pp 45–49.

Tecuci, G, Kedar, S and Kodratoff, Y, 1993. *Proc. IJCAI-93 Workshop on Machine Learning and Knowledge Acquisition: Common Issues, Contrasting Methods, and Integrated Approaches.*

Thomas, J, 1994. "Learning within a problem solving method: An empirical evaluation." In: *Proc. ECAI-94 Workshop on Integration of ML and KA,* pp 51–57.

Utgoff, PE, 1986. "Shift of bias for inductive concept-learning" In: RS Michalski, JG Carbonell, and TM Mitchell (eds.), *Machine Learning: An artificial intelligence approach*, pp 107–148, Morgan Kaufmann.

Van de Velde, W and Aamodt, A, 1992. "Machine learning issues in CommonKADS, ESPRIT Project KADS-II".

Van de Velde, W, Fensel, D, Plaza, E and Van Someren, M (eds.), 1994. *Learning and Knowledge Level, MLnet Workshop,* Catania, Italy.