Industrial Applications of ML: Illustrations for the KAML Dilemma and the CBR Dream

Yves Kodratoff and Hervé Mignot

LRI, Équipe Inférence et Apprentissage
Université Paris-Sud
F-91405 Orsay
email: {yk,mh}@lri.fr

Abstract.
This paper presents several industrial applications of ML in the context of their effort to solve the "KAML problem", i.e., the problem of merging knowledge acquisition and machine learning techniques. Case-based reasoning is a possible alternative to the problem of acquiring highly compiled expert knowledge, but it raises also many new problems that must be solved before really efficient implementations are available.

Introduction

There are many sides to the description of what an industrial application is. In a recent paper (Kodratoff, Graner, & Moustakis 1994) we summarized some of the experience gained during the CEC project MLT in counselling a user on which of the many types of machine learning (ML) to use for his special application. In this presentation, we shall consider two of the main subfields of the ones that need merging for an industrial application, seemingly the richest in generating future research problems: validation of KBS and merging of ML into a knowledge acquisition (KA) method. The first one is almost untouched by specialists in ML, while the second one led to much work, some of it will be reported in the rest of the paper.

As just said, real-world applications require validation of the programs used. Let us speak briefly of what means validation in our context, and what ML can have to do with it.

It seems that "validation" takes three different meanings in the context of KBS. All different types of knowledge originate from the expert's knowledge which is not directly accessible, thus the KA system helps the expert to gather his knowledge in the KA system knowledge level. In KADS' knowledge level, one finds the models, such as the model of tasks, the model of expertise etc. In the model of expertise, one finds knowledge about the strategies, the tasks, the inference, and the domain. All these kinds of knowledge are usually considered validated because they issue "directly" from the expert. This gives us a first kind of validation, by which an expert reconsiders his own knowledge at the knowledge level, and checks its validity. This is not enough in reality since experts do make mistakes from time to time, and even when they agree on the actions to take, they also often disagree on the reasons (that is, what knowledge to use) why these actions are to be taken. It is always good to compare such validated knowledge to the real world. The knowledge must thus be translated to the symbol level, i.e., a language into which programs can be written, to be checked against real applications. During this process, many mistakes are possible, and we have here need for a second kind of validation, the classical one in software engineering, that the knowledge level (the "specification") matches correctly the symbol level (the "algorithm"). During the verification process, the expert will find misbehaviours of the system, that will request some changes. This is the also known as the classical "trial and error" validation technique. Notice however the complication arising because transformations can be performed either at the symbol, or at the knowledge level.

Validation can make use of ML techniques, for both incompleteness and incorrectness. The knowledge to be considered is threefold: the rules of expertise, a deeper kind of knowledge given by a semantic-net and a set of integrity constraints, and a set of examples. When anomalies are detected, the correction is performed according to sets of positive and negative examples of the concept to revise (Lounis 1993).

We will give a few examples of industrial applications in the following. What must be kept in mind, though, is that all real-real-world applications met very nicely the requirements of the KA + ML workshops, because they had to solve this problem in the first place. All considered what is the essential difference between an academic and an industrial work? The academic chooses the data in a repertory of such available data, while the industrial receives data from his users, often demanding ones. These repertories, at least in ML, tend to contain quite a variety of data of various levels of difficulty, but for all of them, the KA phase has been completed beforehand. Thus, the industrial is not only under pressure of his users, but he has also to count on them to perform the KA phase which is a crucial one as we all know. In the following, we will refer to this problem as to the KAML problem, with this mild joke that we indeed need camels to help us crossing the desert that expands between the fertile plains of industrial applications, and the nice oases of academic research.

As a coarse view, one can say that with respect to KAML, academics have produced one very interesting approach, known as knowledge refinement. On another hand, we personally dug five different ways of integrating ML and KA out of the solutions of the people that tackled real-world applications of ML. This paper is mainly devoted to the description of solutions to the KAML problem, together with the industrial applications that led to these solutions. We shall successively speak of the following: the knowledge refinement approach, how existing ML technique must be adapted to meet real life requirements, what kind of knowledge can be acquired from a human expert in order to obtain good ML results, why KA needs ML to enhance the rate of the acquisition, the problem if finding new representation schemes to meet experts' requirements, how to acquire compiled knowledge, and finally the promises and challenges of the CBR approach.

Knowledge refinement

We shall not give here any detailed description of the different revision techniques since they received already considerable acknowledgement in the academic community. We just recall some of the earlier work that we did on knowledge refinement roughly around 1985, for an application to air-traffic control, and on DISCIPLE.

Air traffic control

Our work on air traffic control has been published first in 1988 (Cannat & Vrain 1988), and more recently under a more detailed form (Kodratoff & Vrain 1993).

In this application, we used a refinement cycle which makes explicit the role of the user. It includes the steps necessary to translate the knowledge given by the expert and the learned knowledge. The importance and difficulty of these translation stages are generally underrated by academics, while they are the very condition at which an application can take place (Cannat & Vrain 1988, Kodratoff & Vrain 1993).

DISCIPLE

The main idea behind DISCIPLE (Kodratoff & Tecuci 1987), and behind more recent versions, APT (Nédellec 1992, Nédellec & Causse 1992) and Neo-DISCIPLE (Tecuci 1992, Tecuci & Duff 1994) is user-driven revision, with the idea that experts are better at checking solutions than building theories. DISCIPLE thus proposes a solution to a problem that will become a positive example if the user agrees with the solution, and a negative example if he disagrees, and the possible causes for his agreement or disagreement. These causes are used to refine logically the existing rules, by adapting (i.e., generalizing or particularizing depending on the cases) the conditions of the rules to the positive and negative examples. This has been used in a bank application in order to help eliciting the knowledge of experts, but it also requests a patient user that accepts to "play" with the system in order to build the initial data base that can be quite extensive.

Adapting ML to meet real-life requirements

In a yet unpublished paper, (Schmalhofer et al. to appear) report a very thorough experience on solving the KAML problem for specific industrial needs. These authors report finding it necessary to adapt both conceptual clustering and explanation-based learning, by integrating expert consultation inside the ML algorithms. See also (Esposito, Malerba, & Semeraro 1993, 1994).

More generally, it seems that a multistrategic approach is needed for many applications that work already quite well by using statistical methods, but that can be still enhanced when some more symbolic treatment is also performed.

ML Solves KA Problems

Ml can be used to solve KA problems in which the representation of knowledge is so complex that traditional KA becomes unbearable.

Example 1. Fault detection in helicopter blades
Detection and repair of faults of an helicopter blade is not only a technical process, but also a judicial since the repairing person is responsible in case of an accident. The system has to give an argument to explain a given repair, and authors (El Attar & Hamery 1994) has to produce rules that would combine symbolic and numeric data in the way best suited to the existing validation procedures. Secondly, feature intervals in which detection and repair of faults take place were not immediately available from the experts, they had to be directly acquired from the examples, in such a way that the results were still understandable by (and agreeable to) the experts.

Example 2. Learning on French justice system
This problem is a special version of the classical one of the missing values with "don't care" values that are not significant to the solution. It appears that the French judicial knowledge requires that one has to deal explicitly with this problem. Tree generating procedure cannot deal very well with this kind of problem, this is why (Venturini 1994) adapted genetic algorithms technique to rule generation.

Example 3. Refining rules for a production system (Terano & Muro 1994).
Once rules have been learned by in a classical way, it is always possible to consider that one can improve them by searching in the space of all possible rules, in the vicinity of the existing rules. Genetic Algorithms are obviously well-adapted to solve this kind of search problem since strong parents are already available.

Helping ML by some additional knowledge

Biasing the ML mechanism with acquired knowledge

It makes explicit which information is essential to decrease the combinatorics of ML, and acquires this knowledge by elicitation.

Example 4. Learning to design VLSI.
Learning the most specific generalization form a set of examples is one of the basic problems of ML since it is of the best ways to reduce the complexity of the set of examples, while keeping their common properties. While relatively trivial at the zeroth order, in first order representations this problem becomes quickly very difficult (Kodratoff & Ganascia 1986) because of a combinatorial explosion. Some work has been done to

deal with this complexity, as by using Bisson's distance measure, for instance (Kodratoff & Bisson 1992).

Application to VLSI design requires a relational representation (Hermann & Beckmann 1994) since it must represent the bindings of the different parts of the circuit. These authors present an original solution making use of the user's knowledge of what parts can be possibly bound to other parts. This is a knowledge which is well-known by the experts, but which is not acquired classically in KA systems because they perform no learning and do not need this information. In other words, it is very clear here that a new kind of knowledge has to be acquired because of the learning component.

The ML mechanism is impossible without manually acquired knowledge

ML requirements make it necessary to ask more information, or a new kind of information to the expert.

Example 5. Prediction of cylinder banding
In the printing industry, banding is a nuisance known for long. Bob Evans (Evans & Fisher 1993) signalled that the usual causal KA has been failing to solve this problem. Driven by the requirements of his induction algorithm, Evans promoted a new approach, a more pragmatic one, by which he proposes to find only conditions at which banding will appear. He reported having many difficulties having the specialists answering his "trivial" questions, until unexpected features appeared in the prediction for banding, and were confirmed by experience.

Adapting KA to meet real-life requirements

Example 6. EBL learning of operational rules in the Pilot Assistant
The continuous maintenance of the knowledge of these six interconnected large expert systems by a varying set of experts make necessary to have a global knowledge repository of easily understandable and maintainable knowledge, to be translated into the language of each expert system. This top knowledge is gathered in a systematic way by means of an EBL component. Domain knowledge is acquired in a classical way, it is then transformed into a standard representation by means of one example, and of a criterion of operationality (Miller & Levi 1994). In this very case, one can see than ML became a new way of gathering expert knowledge by merging theoretical knowledge and examples.

Example 7. Road and train traffic control
The introduction of a ML component, together with the difficulties due to the domain complexity forced (Arciszewski et al. 1994) to develop an original KA process. In short, one has to make a careful decision on which simplifications to the real-world problem will lead to a model which still realistic and with which one can still work.

Develop new representations to allow experts to express their knowledge

Example 8. Improving manufacturing processes (Riddle, Segal, & Etzioni 1994)
The Boeing company decided to use ML in order to improve some of its manufacturing processes. These authors find five problems, all more or less of KA nature, to solve prior to applying the induction mechanism. One of them was choosing instances, a "pure KA" problem which is no addressed by classical KA, and that must be solved for KAML with new KA methods to represent the information necessary to the ML algorithms. An other was finding relevant attributes, problem which is better known than the former one, and is indeed addressed by KA techniques. Unfortunately application of subsequent ML techniques request more attention to the problem of irrelevant attributes than it is usual in KA.

Example 9. KAML for decision under constraints
The problem is to recognize plans and intentions of an enemy for a decision making assistant (Barès et al. 1994). Humans handle this problem by merging three kinds of different knowledge, general principles of tactics, intelligence information, and the doctrine of both sides. In order to merge these three kinds of information in a retrievable form, we had to develop a special knowledge representation framework, inspired from Schank's XPs (Schank 1986), in which a special part is devoted to the slow emergence of a plan when some partial information confirms its activation.

Use ML to acquire knowledge usually compiled by experts

Acquiring perceptual chunks

Example 10. Solving geometry problems.
This is a difficult application, if not an industrial one. The work of (Suwa & Motoda 1994) addresses the particular problem of gathering chunks of a perceptual nature for problem solving in such a way that their preconditions are easy to detect and discriminating. Besides, these chunk may well act as simple hints that drive the solution in the good direction, without reaching the desired goal directly. PCLEARN learns such perceptual information that experts are often unable to provide directly, by analyzing success proof trees and selecting the objects are recognizable to its recognition rules to build a chunk of them.

Acquire plan abstractions

In (Schmalhofer & Tschaitschian 1993, Schmalhofer et al. to appear), the authors describe a methodology (which makes use of a user-controlled conceptual clustering and explanation-based learning, as already noticed) for acquiring production plans in mechanical engineering. The experts obviously are able to provide concrete plans on how to solve a specific problem, but the mentioned system can be viewed as helping them to generate also abstract plans that make their experience easier to apply to new problems.

CBR solutions and problems

CBR is used in planning to provide a set of plans that have already been used and in which problems appearing when building a plan from scratch have been worked out. Using these plans (or cases) reduces the computational effort necessary in planning to deal with protections and preconditions, requiring minor changes to adapt the plan recalled to the new situation. It also allows to exploit simultaneous goal achievements, found for goal conjunctions in the past and used in the plans stored in the case base. Finally, case-based planning can be used to deal with reactive planning, i.e. when the world in which the planner operates evolves independently, like the real world. Examples of case-based planning applications can be found in (Alterman 1988, Hammond 1989, Kolodner 1993, Veloso 1992).

The goal of CBR is efficient reuse of knowledge, NOT the building of causal theories. The negative side effect of deep causal theories is that it transforms the story of the case so much that the user no longer recognizes the case. On the contrary, CBR has indeed the negative effect that the causal theory is kept implicit throughout the reasoning process, but the positive one that the story of the case is easy to recognize.

In practice, as we suppose each reader knows, CBR works by extracting from the base of cases the cases whose description are the nearest to the description of the target, then use the solution of these nearest cases to find a solution for the target. This is very similar to rules by the substitution (description --> condition, solution --> action), but as opposed to rules, there is no explicit causal link between the descriptions and the solutions.

Knowledge acquisition issues

A first obvious problem is the definition of the knowledge representation of the cases themselves. A case contains all the information necessary to be able to recompose what has been happening in the external world. A "case", as opposed to the more traditional knowledge representation systems, is never defined in general, it depends on the application it describes. This relaxation on the constraints will request that some extra knowledge is asked to the domain expert, as we shall see in the following.

The domain expert, as is almost always the case of AI oriented applications, is requested to build an ontology of his application domain: What are the features of the domain, when are they significant, what are the semantics of these features, what are their domain of values, what are the relations among features (i.e., organizing the knowledge of the domain).

In order to define an efficient similarity measure, the knowledge acquisition process must include, besides the "plain" field knowledge, seven types of knowledge, particular to CBR. It is interesting to identify those types in order to avoid bad surprises in applications. They are:

a - the similarity measure itself,

b - domain knowledge to be used for similarity assessment, which often appears as rules of two kinds:

b1 - domain theory (used for instance to saturate the description of a case),

b2 - rules allowing to compute local similarity,

c - domains of significance (validity) of features,

d - contexts in which a given similarity measure is efficient, being understood that in most cases, a similarity measure is efficient when it uses significant features,

e - domain integrity constraints to ensure the global soundness of the measure made upon separated features,

f - transfer functions, for translating the solutions of the base into the possible ones of the target,

g - and finally, constraints relative to the application of the cases in order to avoid absurd solutions.

The above special requirements represent the extra knowledge necessary to keep available the story of the case. In other words, they constitute the form deep knowledge takes in CBR: it is a very unusual form, certainly a non-causal one.

A conclusion on using CBR

It is quite usual to consider that the predictive power of Science comes from the building of causal theory allowing to explain successions of events. With the CBR approach, the link between predictability and causality is no longer deemed compulsory, but simply desirable. Predictability itself is achieved through the recognition of a conjunction of values of shallow features.

CBR involves such a non-causal knowledge acquisition, taking into account, in a new way, the deep knowledge:

- define domains of significant features in order to know in which context which features should be used to compute the similarity,
- define domains of similarity measures in order to know in which context which similarity measure should be used,
- use classical CBR to select a few cases that might be useful, and select among them those that are really to be used by
    - ruling out those that violate known integrity constraints,
    - rule out those cases that are not confirmed by given rules that might have another form than integrity constraints
    - checking the consequences of the choice of a case, i.e., select according to the applicability of the cases,
- refine the application of the case.

It will be often the case that CBR is more efficient than a causal model, especially when the formalization of the field is still incomplete. In order to increase its efficiency, it may be necessary to collect supplementary knowledge which is very different from the one normally considered as causal by scientists.

Final Conclusion

As a first conclusion, we would like to stress that our experience tells us that ML is not an easy solution for KA, but that the solution of the KAML problem goes through improvement of both KA and ML, thus it still needs much research work. Industrial applications are playing the role of pointer to academic research to problems that are somewhat underestimated nowadays.

We presented here some examples of seven solutions to the problems of integrating ML and KA, there might be more problems and more solutions that we did not meet yet.

References

Alterman, R. 1988. Adaptative planning. *Cognitive Science* 12: 393—422.

Arciszewski T., Khasnabis S., Khurshidul Hoda S., Ziarko W. "Machine Learning in Transportation Engineering: A Feasibility Study," *Applied Artificial Intelligence*, special issue on applications of ML, 1994.

Barès M., Canamero D., Delannoy J. F., Kodratoff Y. "XPlans: Case-Based Reasoning for Plan Recognition," *Applied Artificial Intelligence*, special issue on applications of ML, 1994.

Cannat J. J., Vrain C. "Machine Learning Applied to Air Traffic Control" Colloque *International Homme-Machine et Intelligence Artificielle dans les domaines de l'Aéronautique et de l'Espace*, Toulouse, France, 28-30 September 1988, pp. 265-274.

El Attar M., Hamery X. "An Industrial Expert System Acquired by Machine Learning", *Applied Artificial Intelligence*, special issue on applications of ML, 1994., 1994..

Esposito F., Malerba D., Semerano G. "A Multistrategy Learning Approach to Document Recognition," *Applied Artificial Intelligence*, special issue on applications of ML, 1994.

Evans B., Fisher D. "Mitigating Delays in Printing with Interactive Machine Induction," unpublished communication at the IMLC'93 workshop on industrial applications of ML, Amherst, July 1993.

Hammond, K.J. 1989. *Case-based planning: Viewing planning as a memory task.* Boston: Academic Press.

Hermann J., Beckmann R. "LEFT - A System that Learns Rules about VLSI-Design from Structural descriptions", *Applied Artificial Intelligence*, special issue on applications of ML, 1994.

Kodratoff Y., Bisson G. "The Epistemology of Conceptual Clustering: KBG, an Implementation", *J. of Intelligent Information Sys. 1*, 57-84, 1992.1992?

Kodratoff Y., Ganascia, J. G. "Improving the generalization step in learning", in *Machine Learning: An artificial intelligence approach, Vol. 2*, Michalski R. S., Carbonell J. G., Mitchell T. M. (eds.), Morgan Kaufmann , pp. 215-244, 1986

Kodratoff Y., Graner N., Moustakis V. S. "Can Machine Learning Solve my Problem?", *Applied Artificial Intelligence*, special issue on applications of ML, 1994.

Kodratoff Y., Tecuci G. : "DISCIPLE-1 : Interactive Apprentice System in weak theory Fields", *Proc. IJCAI-87*, Milan Aug. 87, pp. 271-273.

Kodratoff Y., Vrain C. "Acquiring First Order Knowledge about Air Traffic Control," *Knowledge Acquisition 5*, pp. 1-36, 1993.

Kolodner, J.L. 1993. *Case-based reasoning.* San Mateo, CA: Morgan Kaufmann.

Lounis H. "Integrating Machine-Learning Techniques in Knowledge-Based Systems Verification", ISMIS'93, 15-18 June 1993, Trondheim, Norway. (Proceedings to appear in LNCS, Springer-Verlag, 1994).

Miller C. A., Levi K. R. "Linked-Learning for Knowledge Acquisition: A Pilot's Associate Case Study", *Knowledge Acquisition*, special issue on merging KA and ML, 1994.

Morik K., Potamias G., Moustakis V. S. , Charissis G. "Knowledgeable Learning Using MOBAL: A Medical Case Study", *Applied Artificial Intelligence*, special issue on applications of ML, 1994.

Nédellec C. "How to specialize by Theory Refinement", in *Proceedings of the 10th European Conference on Artificial Intelligence (ECAI 92)*, August 1992.

Nédellec C., Causse K. "Knowledge Refinement Using Knowledge Acquisition and Machine Learning Methods", *Current Developments in Knowledge Acquisition: EKAW-92*, T. Wetter & al Eds., Springer-Verlag, May 1992.

Riddle P., Segal R., Etzioni O. "Representation Design and Brute-Force Induction in a Boeing Manufacturing Domain," *Applied Artificial Intelligence*, special issue on applications of ML, 1994.

Schank R., *Explanation Patterns: Understanding mechanically and creatively*, Lawrence Erlbaum 1986.

Schmalhofer F. Tschaitschian B. "The acquisition of a procedure schema from text and experience", *Proc. 15th An. Conf. of the Cognitive Science Soc.*, Boulder CO, pp. 883-888, June 1993.

Schmalhofer F., Reinartz T., Tschaitschian B. "A Unified Approach to Learning for Complex real World Domains", to appear, a version has been made available during the IJCAI'93 workshop on KA + ML.

Suwa M., Motoda H. "PCLEARN: A Computer Model for Learning perceptual Chunks," *AI Communications*, special issue on applications of ML, 1994.

Tecuci G., "Automating Knowledge Acquisition as Extending, Updating and Improving a Knowledge Base," *IEEE Transactions on Systems, Man, and Cybernetics*, 22:6, pp. 1444 - 1460, 1992.

Tecuci G., Duff D. "A Framework for Knowledge Base Refinement through Multistrategy Learning and Knowledge Acquisition," to appear in *Knowledge Acquisition*, 1994.

Terano T., Muro Z. "On-the Fly Knowledge Base Refinement by a Classifier System," *Proc. 2nd World Congress on Expert Systems*, Lisbon Jan. 1994, and *AI Communications*, special issue on applications of ML, 1994..

Veloso, M. 1992. Learning by analogical reasoning in general problem solving. Carnegie Melon University, School of Computer Science Technical Report no. CMU-CS-92-174.

Venturini G. "Analyzing French Justice with a Genetic-Based Inductive Algorithm", *Applied Artificial Intelligence*, special issue on applications of ML, 1994.