Chapter 1

Invited Papers

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

Y. Kodratoff

LRI, Equipe Inference et Apprentissage,
Univ. Paris-Sud F-91405 Orsay

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.

1 Introduction

There are many sides to the description of what an industrial application is. In a recent paper (Kodratoff, Graner, and Moustakis, 1994) we summarized some of the experience gained during the CEC project MLT in counseling 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 misbehaviors 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, 1993a, 1993b). Let us underline that very little, besides the cited work of Lounis, has been done in this direction. One can however consider that (Morik et al., 1994) have used MOBAL as a validation system in a medical application. They make use of the deductive abilities of MOBAL in order to find contradictions among the rules learned so far, and then solve by induction these contradictions. MOBAL is not a validation system but it presents so many functionalities that it can be also used in that way, and for this particular medical application, it seems it has been a very efficient way to use it.

During the last year, there have been a good many workshops on industrial applications of ML, and on the merging of ML into a KA method. We will give a few examples of these 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, I 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.

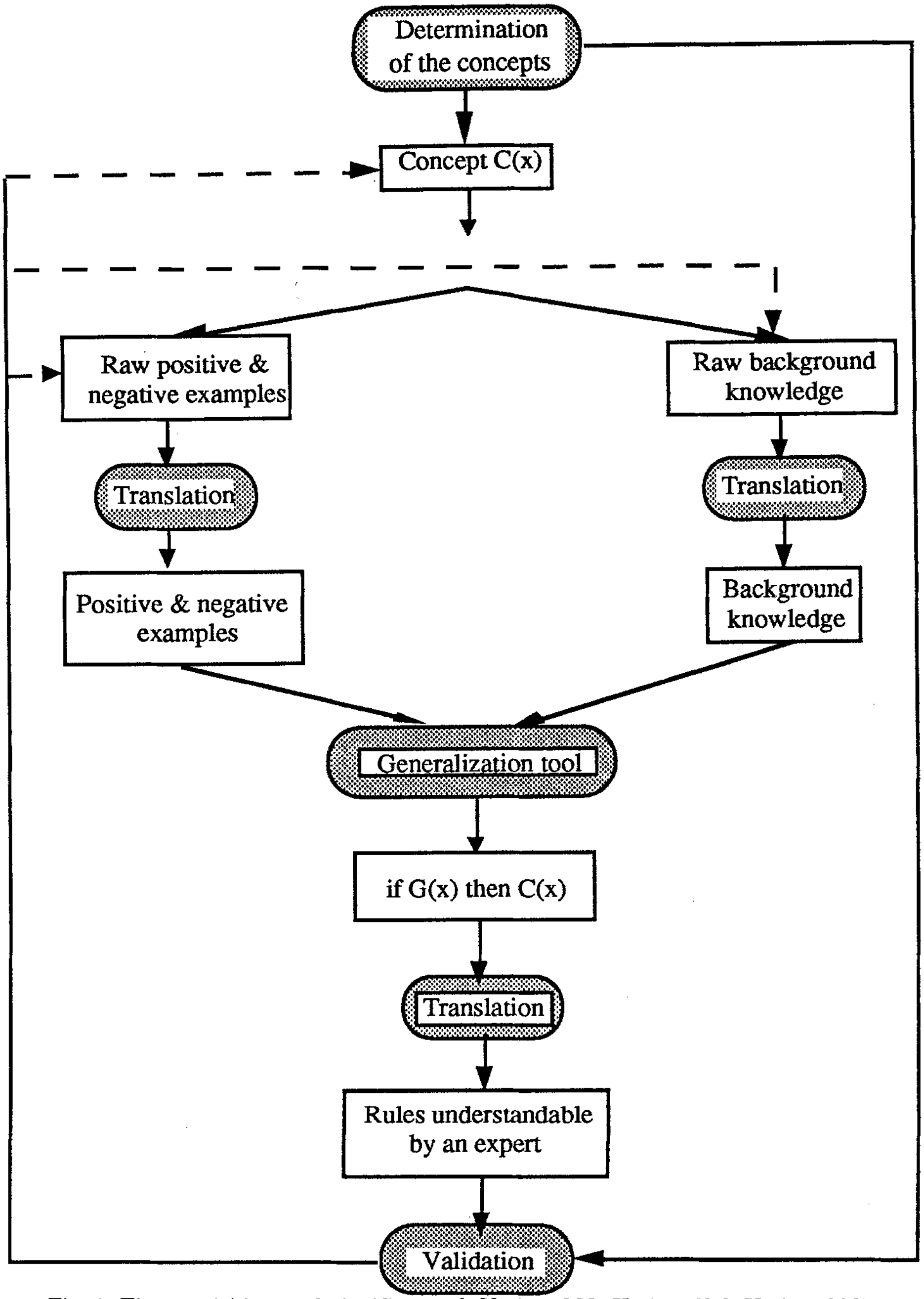

Fig. 1. The acquisition cycle in (Cannat & Vrain, 1988; Kodratoff & Vrain, 1993)

2 Knowledge refinement

We shall not give here any detailed description of the different revision techniques since they received already considerable acknowledgment in the academic community. We shall however give in section 2.3 a brief account of recent research on the topic, and 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.

2.1 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). Very simply put, we used a refinement cycle which makes explicit the role of the user, as indicated in figure 1 below. It includes the steps necessary to transform the knowledge given by the expert and the learned knowledge. The examples and the knowledge base given by the expert must be translated into a representation adapted to the generalization tool. In order to make the validation step easier, the learned knowledge must be translated back into a representation understandable by the expert. 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. In principle, the main difficulty comes from that humans do not understand easily relations in first order logic, even when their knowledge, being strongly relational, makes non-stop use of it (Kodratoff & Vrain, 1993).

2.2 DISCIPLE

The main idea behind DISCIPLE (Kodratoff & Tecuci 1987a, 1987b), and behind more recent versions, APT (Nédellec, 1992; Nédellec and Causse, 1992) and Neo-DISCIPLE (Tecuci 1993; Tecuci and Duff, 1994) is exactly the one of 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. He then chooses among possible causes for his agreement or disagreement, and 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 (Nédellec et al., 1994), 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.

2.3 Some Recent Results - and Problems

As implicitly said above, applications of ML systems show that the initial domain theory provided by the user and possibly completed by the ML system tends to be incomplete and incorrect. Furthermore, ML systems may add incorrect concept definitions they have learned from incorrect example descriptions or insufficient domain theory. This explains the increasing interest in automatically revising incorrect and incomplete domain theories. Some of the propositional systems are described in (Ginsberg, 1989; Ourston and Mooney, 1990; Craw and Sleeman, 1990; Cain, 1991; Towell, Shavlik, and Noordewier, 1991; Wilkins, 1991), and some of the first order revision systems are found in (Shapiro, 1983; Sammut and Banerji, 1986; Pazzani, 1988; Muggleton and Buntine, 1988; Bergadano and Giordana, 1989; Duval, 1991; Wogulis, 1991; Richards and Mooney, 1991; Rouveirol, 1992; DeRaedt, 1992; Nédellec, 1992; Nédellec and Causse, 1992; Esposito et al., 1993; Morik et al., 1993; Feldman and Nédellec, 1994).

The revision problem can be described as containing two problems, two phases, and two strategies.

The two problems are the completeness problem: given a theory T, and an example E of the concept C, such that T does not recognize the positive example E of C as an example of C, build a theory T' that recognizes E, and the correctness problem: given a theory T and a negative example NE of the concept C, such that T recognizes the negative example E of C as an example of C, build a theory T' such that T' does not recognize NE.

The two phases are the localization of the culprit clause and the refinement itself.

The two strategies are the monotonic and the non-monotonic one. In the monotonic approach, the version space is reduced by eliminating inconsistent hypotheses for ever, while non monotonic algorithms may reconsider pruned hypotheses. One can find several examples of these two strategies in (Nédellec and Rouveirol, 1993)

3 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.
Also typical of this approach is (Esposito, Malerba, and Semeraro, 1993, 1994) which describes an application to document layout recognition in the context of a software environment for office automation distributed by Olivetti. These authors extended a star algorithm (Michalski, 1983) by a flexible matching algorithm, and used a data analysis discriminant system to speed up the star algorithm.
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.

3.1 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
This application asked the authors (El Attar and Hamery, 1994) to deal with two different problems. First, detection an repair of faults of an helicopter blade is not only a technical process, but also a kind of judicial one, since the repairing person is responsible in case of an accident. The system must thus also give the best possible "argument" to explain why a given repair is suitable. The data being of a symbolic/numeric nature, the authors had to produce the rules that would combine these two natures in the way the best suited to the existing validation procedures. Secondly, one of the most important features to determine is a set of intervals into which detection and repair of faults takes place. These intervals 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
The problem met here is the classical one of the missing values, in its very special version where there are many "don't care" values that are not significant to the solution, as for instance, the values of the attribute "color of the feathers" for a mammal. One usually simply tries to cut the problem in pieces that avoid such kind of problems, but it appears that the existing (French) judicial knowledge is under such a form at present 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 and 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.

4 Helping ML by some additional knowledge

4.1 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. At the zeroth order, this problem is relatively trivial since it amounts to Michalski's "dropping rule" (Michalski, 1983), together with some of his "climbing hierarchies" rule when the dropping is done cleverly. In first order representations (actually, we speak here of relational representations making use of universally quantified variables), however, this problem becomes quickly very difficult (Kodratoff and Ganascia, 1986) because each subpart of an object can be bound to any other subpart of another object, thus leading to a combinatorial explosion. Some work has been done to deal with this complexity, as by using Bisson's distance measure, for instance (Kodratoff and Bisson, 1992). Application to VLSI design requires a relational representation (Hermann and 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.

4.2 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. In his ICML presentation, Bob Evans (Evans and Fisher, 1993) signaled that the usual approach is to understand the causes of banding, and failed. In that sense, 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.

5 Adapting KA to meet real-life requirements

Example 6. EBL learning of operational rules in the Pilot Assistant
Honeywell has developed a pilot's associate that contains as much as possible of the knowledge relative the military piloting skills. This system contains six interconnected large expert systems. Each of them needs continuous maintenance since the methods for fight and counter-fight are constantly evolving, and maintenance must be done by a varying set of experts. These constraints make it necessary to have a kind of global knowledge repository of easily understandable and maintainable knowledge, from which it will 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 and 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 problem of transportation engineering makes use of complex mathematical models that limit their practical interest, especially in case of rapid changes. 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.

6 Develop new representations to allow experts to express their knowledge

Example 8. Improving manufacturing processes (Riddle, Segal, and Etzioni, 1994)
The Boeing company decided to use ML in order to improve some of its manufacturing processes. There are many problems to solve prior to applying the induction mechanism, and they are all more or less of a KA nature. These authors find five problems to address to begin with, and we shall point at two of them here.

Choosing instances: what do we exactly call an example of behavior we are going to learn from? In the case of manufacturing, is an instance a set of alarms with the associated events, the history of a rejection of manufactured part, etc.? This a "pure KA" problem which is no addressed by classical KA, and that must be solved for KAML.

Finding relevant attributes. Thus problem is better known than the former one, and it is indeed addressed by KA techniques. Unfortunately, humans are much better than machines to deal with irrelevant attributes, and application of subsequent ML

techniques request more attention to the problem of irrelevant attributes than it is usual in KA.
Particularly for the first of these problems, one thus notices that new KA methods must be developed in order to represent the information necessary to the ML algorithms.

Example 9. KAML for decision under constraints
We participate in the development of a part of a decision making assistant in which we are in charge of the most "interpretative" part, by which want the plans of the enemy are recognized, and its intentions accordingly interpreted (Barès et al., 1994). The way humans handle this problem is 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; Schank and Kass, 1990), in which a special part is devoted to the slow emergence of a plan when some partial information confirms its activation.

7 Use ML to acquire knowledge usually compiled by experts

7.1 Acquiring perceptual chunks

Example 9. Solving geometry problems.
This is a difficult application, if not an industrial one. In many cases, finding an elegant solutions depends on finding the correct "perceptual chunk" that triggers a particularly efficient strategy. This is a problem similar to acquiring search control knowledge as addressed by PRODIGY (Minton et al., 1989). The work of (Suwa and Motoda, 1994) however 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 discriminant. Besides, these chunk may well act as simple hints that drive the solution in the good direction, without reaching the desired goal directly. In general, experts are unable to provide directly such perceptual information. PCLEARN learns them by analyzing success proof trees, and selecting the objects that are recognizable to its recognition rules to build a chunk of them.

7.2 Acquire plan abstractions

In (Schmalhofer and 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.

7.3 Hopes raised by CBR

Guy Boy (1987, 1991) shows with some detail how becoming an expert is transferring causal knowledge to the recognition of situations in which an immediate solution is available. In other words, a beginner solves the problems with deep domain knowledge, while the expert is able to recognize the situations where a solution is available without the need of a complex reasoning. CBR does mimic this attitude, and should be thus explored also as a good way of representing very compiled expert knowledge.

8 CBR solutions and problems

The goal of CBR is efficient reuse of knowledge, NOT the building of causal theories. This statement stresses the difference between classical Data Base (DB) approaches, and this new way to deal with shallower information. It tells to users wanting to build or to express causal theories that they have better to make use of classical DB, not of CBR. 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 other words, taking into account shallow information has the positive features of being fast and easy to understand for user, and the negative one of failing when a deep model is actually needed. We shall see how more interrogation of the user can gather extra-knowledge which will be used in order to represent deep knowledge, in another way than the classical one.
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. For example, a case of an plane accident can well contain the color of the eyes of the pilot, or any other spurious information.

8.1 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. For aircraft incidents, for example, one could define a case by the information necessary to rerun the incident on a simulator. As one can see, 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 - rules possibly included in the similarity measure. These rules can be of two kinds b1 - domain theory (used for instance to saturate the description of a case)

b2 - rules allowing to compute local similarity measures

c - integrity constraints of the field,

d - domains of significance of the features,

e - 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 (see d above).

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

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

We shall give more detail on all these points below, recalling first that the similarity measure is often a simple syntactic measure of the type Hamming measure, as used in Data Analysis. Using such a simple measure is problematic since retrieving a good case is essential to CBR, and since cases contain shallow knowledge: a simple measure will tend to recognize shallow similarities of limited interest. 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.

8.2 Verification and validation

By its principle, CBR gathers information coming directly from the user, to be used directly by him. It is thus always possible that incorrect information is provided. A controller of some sort is necessary in order to verify the cases. In the context of CBR, validation has several meanings.

a - One must verify the coherency of the information contained in the descriptions of the cases. A pilot may, for instance, inverse in time the sequence of events, all in good faith. When shown a simulation, he will willingly acknowledge his error.

b - One must validate the solution proposed by the expert. Some solutions, considered as satisfactory by a given expert, will be validated also by an expert of "higher level", and used in order to give future advice on how to behave in a similar situation. Some other solutions will be considered as unsatisfactory by the expert of "higher level", and used in order to give future advice on how not to behave in a similar situation.

c - One must validate the whole system, by performing systematic studies of its application to yet "unknown" cases.

8.3 More on the similarity measure

It is used to retrieve cases similar to the target case; it computes the distance between the descriptions of the target case, and the descriptions of the base cases, preferably giving more importance to more significant features.
It may be completed by a rule-based choice: similarity measure is used to select the k nearest cases, and rules choose among them which are the best. This leads to complex similarity measures that are not simply the computation of a distance but also take into account rules and/or integrity constraints in order to optimize the choice.

It may be necessary to define even more complex measures by taking into account the consequences of the proposed solutions (see an example below in section 8.4).
One further point is that a given measure is usually not valid on the whole set of examples, it is necessary to set up sub-domains, or regions, and a similarity measure valid for each region. In satellite diagnostic, ESA contractors developed such a context sensitive similarity measure. Another solution can be performing an initial clustering leading to an organization of the base, giving in turn an initial evaluation of the possible regions. Defining these regions can be also seen as part of the interrogation of the expert. Notice how different from classical requests is knowledge elicitation in the context of CBR.

8.4 Transfer of base solution to target

There at least three ways to transfer the base solution to the target.

- No transfer procedure is necessary because the cases are directly submitted to user.
- A trivial transfer: the solution of the base is applied to the target with no change. This is quite dangerous and can lead to very absurd solutions, like proposing to a ship to "go down" because it has been a good solution for a plane.
- A real transfer which implies the existence of a pattern matching and of a transfer function. When a base case similar to the target case has been retrieved, its solution cannot be usually applied straightforwardly to the target, some transformations must be done in order to adapt the solution to the descriptions of the target. This transformation is obviously obtained as follows.

> Compute the set of substitutions unifying the descriptions of the target with those of the base (they are called "replacements" when a constant is replaced by a constant). Apply the substitutions or replacements found in the descriptions to the solution of the base, this gives the solution of the target.

This operation is obviously always necessary in order to avoid trivial mistakes in the target solution. It can be made to comprise some deep knowledge by recording transfer functions associated to some replacements. For instance, suppose that the case base is relative to cars, and that the solution recommends to slow down to some speed in some descriptions. Suppose that the target case is relative to mopeds, then one must replace car by moped in the target solution, and one must also say that such a replacement must be associated to a function linking the changes in speeds of cars to the changes in speeds of mopeds. This function obviously expresses deep knowledge about driving cars and mopeds. It is clear that all these functions compress the information much less than a formal theory of the speeds for each vehicle. It is very useful when such a theory is not known, and when partial knowledge only is available.
In order to transfer a solution to the target, two different kinds of rules can be used.
The first kind is of the form

IF	(in descriptions)	A --> B
THEN	(in solution)	A --> B AND C --> D

where the added part C --> D expresses the changes due to the substitution A --> B.
For instance, such a rule can be:

IF	(in descriptions)	woman --> man
THEN	(in solution)	woman --> man AND pregnant --> future father

The second kind is relative to the transfer of parameters in functions appearing in the solution. It has the form:

Knowing that the substitutions (x/x', and y/y') occur between base and target,

IF	(in base)	parameter x is in description, and parameter y is in solution and y = f(x)
THEN	(in target)	parameter x' is in description, and parameter y' is in solution then y' = f(x')

This adaptation may happen to lead to absurd conclusion, this is why, one must define conditions on the limits of the changes to bring into a solution when transferring from base to target. For example, suppose that the base contains a slow growth process to happen in conditions X, and the target contains a very rapid growth process to happen in exactly same conditions X, then one can apply a transfer rule of the form

IF	(in descriptions)	slow --> fast
THEN	(in solution)	increase speed of measurements.

Nevertheless, this rule must be restricted by the condition that if the recommended speed becomes higher than what any kind of material can stand then this base case must be ruled out how high the similarity might be.
All these rules are difficult to build, and they are necessary each time causality is not implicitly taken into account in the shallow representation.

8.5 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. This attitude leads to discard many non-causal facts thus looked upon as spurious. In some cases, as for examples bases of air-traffic incidents, this leads to forget circumstances linked to the "story" of the events, as they have been lived by the expert. Retrieval of past cases becomes tedious for the expert. More generally, we would like to point at the new attitude of accepting in some conditions to rely on shallow, non-causal knowledge, either because it is important, or because causal knowledge seems to evade the scientist's efforts. In other words, 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. The price to pay for this attitude is obvious. It may well happen that cases superficially very near to each other are actually very different. There are two solutions to this problem. The most obvious one is to use hybrid systems that stick to a causal model as far as possible, and switch to CBR when failing on a problem (or the reverse: Do CBR, and check if the causal model validates the findings of CBR). One should notice however that this is not suitable when the shallow knowledge is really needed. This is why we would like to explore here the second solution which is to stay in the CBR framework, but to improve it so as to take 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.

To conclude, 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.

9 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.

Acknowledgments

The ESPRIT contract MLT has been the first to show how many applications of many different kinds can be done of ML. The help of the CEC is thus gratefully acknowledged. This paper is based on the workshops on industrial applications of ML we organized with Pat Langley at EMLC'93 and IMLC'93 , the workshop on KA + ML we organized with George Tecuci and Smadar Kedar at IJCAI'93, and finally on a "Special Interest Group" on CBR taking place at the Human Machine Interactions and Artificial Intelligence in Avionics and Space congress held in Toulouse, Sept. 28-30, 1993. During this last workshop, we received much interesting information from Michele Visciola from ALENIA, Richard Kraft from NASA Ames, François Allard kindly shared his ESA experience, and we had a long discussion on flight incidents and CBR with Jean Claude Wanner in preparation of the group.

References

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.

Bergadano F., Giordana A., "Guiding Induction with Domain Theories", in *Machine Learning III: An Artificial Intelligence Approach*, pp. 475-492, Kodratoff Y. & Michalski R. (eds.), Morgan Kaufmann, 1989.

Boy G. A. *Intelligent Assistant Systems*, Academic Press, London UK, 1991

Boy G. A. "Operator Assistant Systems," *International Journal of Man-Machine Studies* 27, 1987, pp. 541-554.

Cain T., "The DUCTOR: A Theory Revision System for Propositional Domains", in *Proceedings of the 5th International Machine Learning Workshop (IWML-91)*, Evanston, June 1991.

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.

Craw S., Sleeman D. "Automating the Refinement of Knowledge-Based Systems," *Proc. ECAI90 Conference*, Aiello L (Ed.), pp. 167-172, Pitman, London 1990.

De Raedt L., *Interactive Theory Revision, an inductive logic programming approach*, Academic Press, 1992.

Duval B., "Abduction for Explanation-Based Learning", in *Proceedings of the European Working Session on Learning (Machine Learning, EWSL-91)*, pp. 348-360, Lecture Notes in Artificial Intelligence, Y. Kodratoff (Ed.), Springer Verlag, March 1991.

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., Semeraro G., "Specialization in Incremental Learning: The Negation Operator", in *Proceedings of the AAAI Spring Symposium on Training Issues in Incremental Learning*, Stanford University, March 1993.

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.

Feldman R., Nédellec C., to appear in IKAW'94, Banff.

Ginsberg A., Theory Revision via prior Operationalization", in *Proceedings of National Conference on Artificial Intelligence*, pp. 590-595, Aug. 1989.

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., Tecuci G. : "What is an Explanation in DISCIPLE," *Proc. Intern. Workshop in ML*, Irvine 1987, pp. 160-166.

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

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).

Lounis H. "Knowledge-based Systems Verification: A Machine-Learning-Based Approach", 2nd European Symposium On Verification & Validation (EUROVAV'93), 24-26 March 1993, P. de Mallorca, Spain.

Michalski, R. S., "A Theory and Methodology of Inductive Learning," in *Machine Learning: An Artificial Intelligence Approach, Vol. 1*, R. S. Michalski, J. G. Carbonell, T. M. Mitchell (Eds.), Tioga, Palo Alto, CA, pp. 83-134, 1983.

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.

Minton S., Carbonell J. G., Knoblock C. A., Kuokka D. R., Etzioni O., Gil Y. "Explanation-based learning: a problem solving perspective," *Artificial Intelligence 40*, pp. 63-118, 1989.

Morik K., Wrobel S., Kietz J. U., Emde W. *Knowledge Acquisition and Machine Learning - Theory, Methods, and Applications*, Academic Press, 1993.

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.

Muggleton S., Buntine W. "Machine Invention of First Order Predicates by Inverting Resolution", in *Proceedings of the 5th International Machine Learning Workshop*, pp. 287-292. Morgan Kaufmann, 1988.

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

Nédellec C., Correia J., Ferreira J. L., Costa E. "Machine Learning goes to the bank," *Applied Artificial Intelligence*, special issue on applications of ML, 1994.

Nédellec C. et Rouveirol C., "Hypothesis Selection Biases for Incremental Learning", in Proceedings of the AAAI Spring Symposium on Training Issues in Incremental Learning, Stanford University, March 1993.

Ourston D., Mooney R. J., "Changing the Rules : A Comprehensive Approach to Theory Refinement" in *Proceedings of National Conference on Artificial Intelligence*, pp. 815-820, Aug. 1990.

Pazzani M. J., "Detecting and Correcting Errors of Omission after Explanation-Based Learning", in *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pp. 713-718, Detroit, Aug. 1989.

Richards B. L., Mooney R. J. "First-order Theory Revision," *Proc. 8th Intern. ML Workshop*, pp. 447-451, Evanston IL, 1991.

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.

Rouveirol C. "Completeness of inductive procedures", in *Proc. of the 8th International Machine Learning Workshop (IWML-91)*, Morgan Kaufmann, June 1991.

Sammut C. A., Banerji R. B., "Learning Concepts by asking Questions", in *Machine Learning II : An Artificial Intelligence Approach*, pp. 167-192, R. S. Michalski J. G. Carbonell and T. M. Mitchell (Eds.) Morgan Kaufmann, 1986.

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

Schank R. C., Kass A. "Explanations, Machine Learning, and Creativity," in *Machine Learning: An Artificial Intelligence Approach, Volume III*, Y. Kodratoff and R. S. Michalski (Eds.), Morgan Kaufmann, San Mateo, pp. 31-48, 1990.

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.

Shapiro E. Y. *Algorithmic program debugging*, MIT press, Cambridge, MA, 1983.

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

Tecuci G. "Plausible Justification Trees: A Framework for the Deep and Dynamic Integration of Learning Strategies," *Machine Learning 11*, 1993.

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..

Towell G. G., Shavlik J., Noordewier M. O. "Refinement of Approximately Correct Domain Theories by Knowledge-Based Neural Networks," *Proc. Eighth Nat. Conf. on AI*, pp. 861-866, Boston 1990.

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

Wilkins D. C., Knowledge Base Refinement as Improving an Incorrect and Incomplete Domain theory, in *Machine Learning III: An Artificial Intelligence Approach*, pp. 492-513, Kodratoff Y. & Michalski R. (eds.), Morgan Kaufmann, 1989.

Wogulis J "Revising Relational Domain Theories," *Proc. 8th Inter. Workshop on ML*, Morgan Kaufmann, June 1991.