PROCEEDINGS OF THE INTERNATIONAL CONFERENCE
ON FIFTH GENERATION COMPUTER SYSTEMS 1992,
edited by ICOT. © ICOT, 1992

Applications of Machine Learning: Towards Knowledge Synthesis

Ivan Bratko

Faculty of Electr. Eng. and Computer Sc., Ljubljana University
and J. Stefan Institute, Slovenia

Abstract

This paper shows, by presenting a number of Machine Learning (ML) applications, that the existing ML techniques can be effectively applied in knowledge acquisition for expert systems, thereby alleviating the known knowledge acquisition bottleneck. Analysis in domains of practical interest indicates that the performance accuracy of knowledge induced through learning from examples compares very favourably with the accuracy of best human experts. Also, in addition to accuracy, there are encouraging examples regarding the clarity and meaningfulness of induced knowledge. This points towards automated knowledge synthesis, although much further research is needed in this direction. The state of the art of some approaches to Machine Learning is assessed relative to their practical applicability and the characteristics of a problem domain.

1 Introduction

Machine Learning is one of the most active areas of Artificial Intelligence. In the view of the technical results of this area, and the well known knowledge acquisition bottleneck in expert systems, sometimes known as the Feigenbaum bottleneck, it is surprizing that Machine Learning has not had a stronger impact on the practice of knowledge acquisition for expert systems. Even some known authorities on expert systems occasionally express a reserved view regarding automatic knowledge acquisition through machine learning. For example, Chandrasekaran (1991) in a recent discussion posed the question: "It is often proposed that a way to avoid teasing expertise from experts is to automatically learn from examples. Have you found this a useful strategy?" The answer from a leading practitioner from the commercial side of expert systems technology was: "... I have yet to see a situation where that is an effective way to go forward, especially when you're starting with somebody who knows something. ..."

The practice of AI applications in some laboratories and companies shows, however, that this expresses an overly pessimistic view. This paper presents examples of ML applications in which *existing* techniques were effectively applied.

The paper does not aspire to be in any way a complete survey of the state-of-the-art ML techniques and their applications. However, the example applications and programs discussed are generally illustrative of the practically oriented ML research done at many AI laboratories.

An early demonstration of the usefulness of Machine Learning from examples in knowledge acquisition was induction-assisted knowledge base construction for diagnosing soybean diseases (Chilausky and Michalski 1976; Michalski and Chilausky 1980). A comparison between a manually constructed knowledge base and one constructed with the assistance of an inductive learning program showed the advantages of the latter approach.

Michie (1989) describes another early interesting experience concerning the construction of a small expert system to decide whether a Space Shuttle pilot should lend manually or automatically. The decision depends on the current information about the stability, altitude and velocity estimates of the vehicle etc. This project was an early demonstration of the experts' difficulty in explicitly formulating decision rules although all the relevant information was in their heads. Experience shows that experts' difficulty of this kind is a rather typical phenomenon.

Michie (1989) writes:

"Early in 1984, to address a NASA requirement, the autolander's chief designer, Mr Roger Burke, with engineering colleagues, attempted to construct a computer program to map the real-time values of monitored variables to the alternative decisions *use-auto* and *noauto*. Such a program running on an on-board computer was needed to display continually updated advice to the pilot. After some months of (noninductive) programming they concluded that further effort would not be rewarding. The trouble was later shown to have stemmed not from any intrinsic difficulty of the decision task but from the disability from which every expert suffers in articulating what he or she knows, whether about plant pathology, about medical diagnosis, about process control, about how to play lightening chess or about the movements of the stockmarket.

Mr Burke and his colleagues then attended a course in inductive programing given by Radian Corporation in Austin, Texas, based on the commercial induction software RuleMaster (Michie et. al. 1984). Relieved from the struggle to read the needed rules directly from inside their own heads, they were able (...) to construct the solution ..."

Although a not very sophisticated tool was applied to a not very difficult problem, the NASA experience is very instructive. It illustrates the phenomenon concerning the difficulty of eliciting explicit rules about a domain even if (1) there are experts that can solve concrete problems in the domain quite well, and (2) there is nothing inherently difficult about the domain. Even so, extracting explicit rules from the user turns out to be difficult. When the knowledge elicitation process is aided by a learning tool, the process suddenly appears trivial. Finally, when the actual simple looking solution becomes obvious, there is typically a somewhat embarassing impression that "clearly, the problem should have been possible to solve without the use of machine learning". However, experience confirms that often only when a machine learning tool is eventually applied, the problem solution emerges as obvious.

Another early and similar example of this phenomenon is W. Leech's (1986) application of ML to the synthesis of control rules for process control at a Westinghouse nuclear fuel processing plant. Control rules synthesised from examples using another early ML tool ExpertEase improved the yield drammatically. When analysing the project that led to this innovation, the company officially confirmed that the discovery of the new control rules only occurred when ML was used and the discovery would have been highly unlikely without it.

A review (Urbančič, Kononenko and Križman 1991) of AI applications done by my laboratory in Ljubljana also contains many applications with similar scenario. Among over sixty AI applications included in the review, almost half of them critically rely on the use of ML techniques. One more or less randomly chosen example among these applications, illustrating the same point as the NASA and Westinghouse experience, is from the Jesenice Steel Mill, Slovenia. Their problem was the control of the quality of the rolling emulsion for the Sendzimir rolling mill. The quality of rolling critically depends on the properties of emulsion. An expert therefore daily measured various parameters of emulsion in the rolling mill (concentration of iron, ashes, presence of bacteria, etc.) and decided on the appropriate action (e.g. change emulsion, add anti-bacteria oil, no action, etc.). When the expert was expected to leave the company they attempted to construct an expert system, extracting his decision knowledge from him in the dialogue fashion. Only when after half a year there was no clear progress, they were prepared to apply a ML tool (Assistant Professional in this case; Cestnik et al. 1987) using example decisions from the expert's practice as learning examples. The resulting decision tree, implemented as an expert system, is now used regularly and completely substitutes the decisions that were previously entirely made by the expert.

The most practically successful form of learning has been *attribute-based* learning exemplified by the TDIDT approach (top-down induction of decision trees, e.g. Quinlan 1986). The next section presents results of applications of attribute-based learning in various domains of medical diagnosis and prognosis. These results are interesting also in that they enable a quantitative comparison of the performance of human experts and ML-based diagnostic systems. Although very effective in many domains of practical interest, attribute-based learning has some clear limitations, pointed out in Section 3. These limitations are being overcome by the development of another generation of learning systems, implementing *relational learning*, such as ILP (Inductive Logic Programming, Muggleton 1991). Section 4 presents an example application where the ability of relational learning is essential. ILP, although less mature than attribute-based learning, shows great potentials in application problems that are hard to tackle with attribute-based learning. Section 5 discusses the future of ML with respect to knowledge synthesis.

2 Applications in medical domains

Along with the development of various learning methods in the Ljubljana AI Laboratories, these methods were applied to a number of medical diagnosis/prognosis problems. These applications also served as a source of useful new ideas for further improvements of the learning methods. Some of our medical data (in particular the diagnosis in lymphography, location of primary tumor, and prognosis in breast cancer) were made available to other researchers and were used by many for experimentation and direct comparison of various learning algorithms.

This section presents some results obtained in Ljubljana with various learning systems in several medical domains. Most of this work in medical applications was done with the Assistant system although other programs were also used, including GINESYS (Gams 1988) and LogArt (Cestnik and Bratko 1988). Assistant belongs to the TDIDT family of learning programs (top down induction of decision trees, Quinlan 1986). Assistant is a successor of Quinlan's ID3 (Quinlan 1979) with a number of addidional mechanisms. Early experiments with a version of ID3 in learning of diagnostic rules for lymphatic cancers (Bratko and Mulec 1979) provided encouragement that led us to further exloration and substantial refinements of this approach that were implemented in Assistant. The new mechanisms, motivated and discovered through experiments in medical domains, include: automatic selection of good examples for learning, handling partially specified objects (missing data), forward pruning of decision trees, post pruning (Niblett and Bratko 1986, Cestnik and Bratko 1991), binarisation of attributes (Kononenko et. al. 1985; Bratko and Kononenko 1987). It should be noted that these techniques among some other important improvements to the basic TDIDT learning were contributed or independently discovered by other researchers, for example in the C4 program (Quinlan et al. 1989) and the CART system (Breiman et al. 1984). Mingers (1989a; 1989b) reviews various related techniques and makes an attempt at their comparison.

TDIDT programs belong to attribute-based learning. They accept learning examples in the form of attribute-value vectors. Similarly, both GINESYS and LogArt are attribute-based learning programs. GINESYS generates if-then rules. The innovation of GINESYS was *confirmation rules* that accompany the "main" rule and enable the system to exploit *redundancy* in the attribute-value data. Redundancy is in principle useful in noisy domains, such as medicine, as a means for filtering out errors. The idea of exploiting redundancy (rule bases with redundancy) was later accepted as generally useful in learning in noisy domains, but GINESYS (Gams 1989) was probably the first to build explicitly on this principle in Machine Learning.

Unlike most other systems, LogArt (Cestnik and Bratko 1988) generates *elimination rules*. The rules are ordered according to their statistical credibility. In diagnosis, rules are applied in this order to eliminate all but one of the diagnostic possibilities. When this is not possible and there are more than one residual diagnostic possibilities, the Bayes classifier is employed as a tie-breaker. The credibility of induced rules is measured simply as the number of confirming observations in the learning data. These rules are extremely simple and thus also useful for straightforward explanation of the diagnostic decision. Despite this almost unbelievable simplicity, LogArt compares extremely well with other learning systems in respect of diagnostic accuracy. The key to LogArt's performance lies in high number of simple elimination rules for each application which, similar to GINESYS, facilitates the use of redundancy. This makes LogArt very robust with respect to noise in the learning data and also enables it to cope easily with missing data, that is unspecified attribute values. On the theoretical side, it was shown that LogArt's classification procedure can be viewed as a special strategy of evaluating the Bayes classification rule without the attribute independence assumption (Cestnik and Bratko 1988). LogArt's classification procedure tends to use those conditional probabilities for whose estimation the learning data provides most evidence.

Table 1 summarises the properties of eight medical domains in which these learning systems have been applied. The domains are characterised by: the number of known examples (patients), the number of classes (that is: possible diagnoses), the number of attributes, the average number of possible attribute values per attribute. More detailed description of these applications can be found for example in (Bratko and Kononenko 1987), (Pirnat et al. 1989) and (Roškar et al. 1986).

Table 2 shows results of these applications in terms of diagnostic accuracy of learned diagnostic

domain	examples	classes	majority class	attributes	average no. values	entropy (bit)
lymphography 1	148	4	55 %	18	3.3	1.23
lymphography 2	150	7	46 %	18	3.3	2.11
primary tumor	339	22	25 %	17	2.2	3.64
breast cancer	288	2	80 %	10	2.7	0.72
hepatitis	155	2	79 %	19	3.6	0.74
thyroid	884	4	56 %	15	15.7	1.59
rheumathology	355	6	66 %	32	9.1	1.70
urinary tract m	1843	9	21 %	44	3.8	2.91
urinary tract f	3580	9	25 %	45	6.5	2.59

Table 1: Properties of the medical application domains.

rules by the three systems. The performance of medical experts is also included for comparison in the cases when their performance has been experimentally estimated on the same data as used by the systems. In one case (lymphography), the physicians' performance is an expert's own estimate and was not experimentally confirmed. It is probably an optimistic over-estimate. Systems' accuracy on new data was estimated in the usual way: 70% of the available data was randomly chosen for learning, and the remaining 30% was diagnosed by the learned rule. The system's diagnoses on the "new" data were then compared with the known physician's diagnoses. This was repeated several times (usually ten times, to reduce statistical fluctuation) and the figures in Table 2 are the average of these repeated experiments. For comparison, the performance of "naive Bayes" (that is Bayes classification under the assumption of attribute independence) is also included. It should be noted that this straightforward application of Bayes has the disadvantage that it does not support the usual style of explanation in expert systems. It is therefore avoided in expert systems, although Michie (1990) describes a way to overcome this difficulty.

Some accuracy results in Table 2 are surprizing as in some cases the system's or expert's accuracy are lower than the percentage of the majority class. For example, in the breast-cancer domain the specialists' performance is 64% and Assistant's performance is 77%. These performances are both below the 80% percent likelihood of the majority class, so an almost uninformed clasifier, always just predicting the majority class, would score better than both human experts and the learning programs. This reflects a drawback of simple accuracy measure as the criterion of success of a classifier. The accuracy criterion does not take into account the relative difficulty of predicting particular classes and is therefore misleading, particularly in domains where the probabilities are extremely unequally distributed between classes, as in the breast cancer domain. This problem with accuracy as a performance measure is discussed in (Kononenko and Bratko 1991), and an information-based criterion is proposed. Therefore classifiers' *information scores* (in bits) are also given wherever they were available. The information scores are in all cases positive, indicating that the clasifiers are in fact always doing better than an uninformed classifier (which would, always classifying into the majority class, by definition of the information-based performance measure score zero).

One conclusion indicated by Table 2 is that the knowledge bases induced from no more than a few hundreds of examples of patients in some narrow diagnostic domain, perform better than medical doctors, including best specialists. Such a conclusion has been empirically confirmed by several other studies. This result should, of course, be taken with some qualifications. Namely, the criterion of performance here is only in terms of classification accuracy (or information score) under the condition that both the human expert and the induced classifier are given the same information. In practice, the human expert might be able to use extra information. Also, the medical doctor would typically have a much better global understanding of the problem and be capable of deeper explanation of the particular cases.

3 Attribute-based learning *vs.* relational learning

Applications of Machine Learning described above all rely on the use of attribute-based learning. Both learning examples and induced concept descriptions employ global attributes of objects and not rela-

domain	doctors nonspec.	doctors specialists	Assistant	GINESYS	LogArt	naive Bayes
lymphography 1			76%		84%	79%
lymphography 2	60% (D)	85% (D)	65% (A)	70% (C)		67%
primary tumor	32% , 0.95 bit	42% , 1.22 bit	44% , 1.38 bit	52% (C)	44% (B)	49% , 1.59 bit
breast cancer	64% , 0.03 bit	64% , 0.05 bit	77% , 0.07 bit	74% (C)	78% (B)	79% , 0.06 bit
hepatitis			83%		85%	84%
thyroid		64% , 0.59 bit	73% , 0.86 bit			68% , 0.70 bit
rheumathology		56% , 0.26 bit	61% , 0.46 bit			57% , 0.28 bit
urinary tract m			70% (A)			67%
urinary tract f			80% (A)			79%

Table 2: Performance in terms of classification accuracy and information score (in bits) on new data of the three learning systems, physicians (specialists and non-specialists), and the Bayes classifier evaluated under the assumption of attribute independence. Labels A, B, C, D in the table mean: A - old implementation of Assistant on DEC-10; B - in the case that more than one class remain un-eliminated by rules, naive Bayes is applied as tie-break; C - original data preprocessed so that unknown attribute values in data are replaced by the most likely value; D - expert physician's estimate (not measured experimentally).

tions among their parts. Well known families of such learning programs are TDIDT (e.g. Quinlan 1986), AQ (e.g. Michalski 1983), CN2 (Clark and Niblett 1989). Attribute-based learning is a relatively simple approach to learning and is therefore most widespread and widely used. The following advantages of attributional learning contribute to its success in practical applications:

- Computational efficiency
- Attributional learning is relatively well understood
- Attributional learning process is easy to understand by the users and it is straightforward to apply
- The attribute-value language is natural in many domains and many users are used to this representation
- It is well understood how to handle noisy and incomplete data in attributional learning; there are methods that handle noise very well

However, attribute-based learning also has strong limitations:

- Background knowledge can be expressed in rather limited form
- Lack of relational descriptions makes the concept description language inappropriate for some domains

Attribute-based descriptions are essentially equivalent to propositional logic. This is not sufficiently expressive for describing concepts in some application areas. An example of such a problem area is the finite-element mesh design which is described in detail in the next section.

The realization of the limitations of attribute-based learning led to a number of recent developments towards learning at the level of first-order predicate logic, including programs CIGOL (Muggleton and Buntine 1988), FOIL (Quinlan 1990), GOLEM (Muggleton and Feng 1990) and LINUS (Lavrač, Džeroski and Grobelnik 1991). This led to the establishment of a special area of Machine Learning, named by Muggleton (1990) *Inductive Logic Programming* (ILP; see also Muggleton 1992). The learning problem in ILP is formalised as: given some background knowledge B expressed as a set of predicates, some examples E and some negative examples N, find a logic formula H, such that:

$$B \wedge H \vdash E$$

and

$$B \wedge H \nvdash N$$

The following section describes an application that illustrates the suitability of this approach.

4 Application of ILP to finite-element mesh design

Dolšak and Muggleton (1991) describe an application of ILP to a problem for which the attribute-

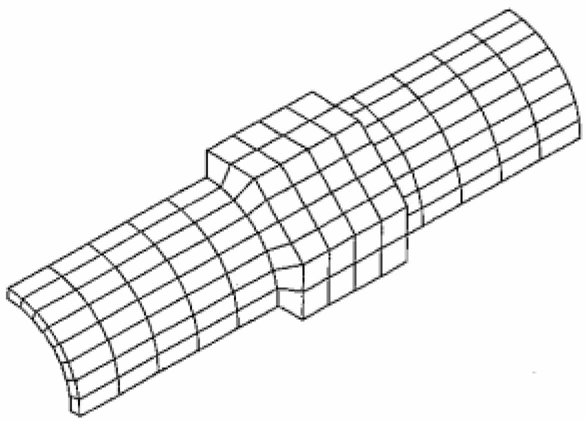

Figure 1: A cylindrical object partitioned by mesh suitable for the finite element computation. (Dolšak 1991)

based learning is unsuitable, and the relational representation appears natural. Here we illustrate this application with more recent results reported in (Dolšak 1991).

The problem of finite-element mesh design arises in numerical computation. Given, for example, a machine part and forces acting on it, the problem is to compute the pressure and deformations throughout the object. The finite-element method involves the partitioning of the given object into finite elements. Figure 1 shows an example. The resulting partition is called a finite element mesh.

For each element of the mesh, constraints in the form of equations are stated. The constraints approximately state the physical laws modelling the behaviour of the individual elements. These approximations are sufficiently accurate if the elements are sufficiently small. Generally, the finer the mesh, the smaller the error. However, a dense mesh results in a large number of equations, leading to a lengthy computation when solving the corresponding system of equations. The complexity of computation is often measured in days or weeks of CPU time and can easily become prohibitive. The problem, then, is to find a suitable compromise between the density and coarseness of the mesh.

Normally some regions of the object require denser mesh whereas in other regions a coarser mesh still suffices for good approximation. There is no known general method that would enable automatic determination of optimal, or reasonably good meshes. However, expert users of finite element methods are capable of making good guesses about proper density of the mesh in various regions of the objects. Unfortunately, the experts have difficulties in forming general rules that would enable the automation of such guesses.

In general, the mesh depends on the geometric properties of the object and forces acting on it. As pressure is transmitted between adjacent elements, the mesh density in a region of the object depends also on the adjacent regions. These general considerations were captured in Dolšak's application as background knowledge for the ILP learning in the form of properties and relations, such as:

```
short( Edge)
usual_length( Edge)
loaded( Edge)
not_loaded( Edge)
two_side_fixed( Edge)
neighbour_xy( Edge1, Edge2)
neighbour_xz( Edge1, Edge2)
...
```

The meaning of these relations is straightforward. For example, an edge is "two_side_fixed" if it is fixed at both ends. `neighbour_xy( Edge1, Edge2)` means that the edges are adjacent and they are in the xy-plane.

In an experiment to learn a characterisation of the density of a mesh in terms of these relations, five meshes known to work well were used as sources of examples for learning (Figure 2). The relation to be learned was:

```
mesh( Edge, N)
```

where Edge is the name of an edge in the structure, and N is the recommended number of finite elements along this edge. The target definition of this relation is to be learned in terms of the properties and relations in the given structure. All the five meshes used comprised altogether 278 edges, that is 278 positive examples for learning. The number of finite elements along the edges varied between 1 and 17. In edges with high partition, say 10, it was assumed that a similar partition would still make a good mesh, so 10 ± 1 was considered acceptable and sometimes used as another positive example. Negative examples were generated according to the closed-world assumption: if the given partitioning of an edge was 3, say, then partitionings such as 4, 5, etc. were taken as negative examples. This finally gives the following number of facts for learning in

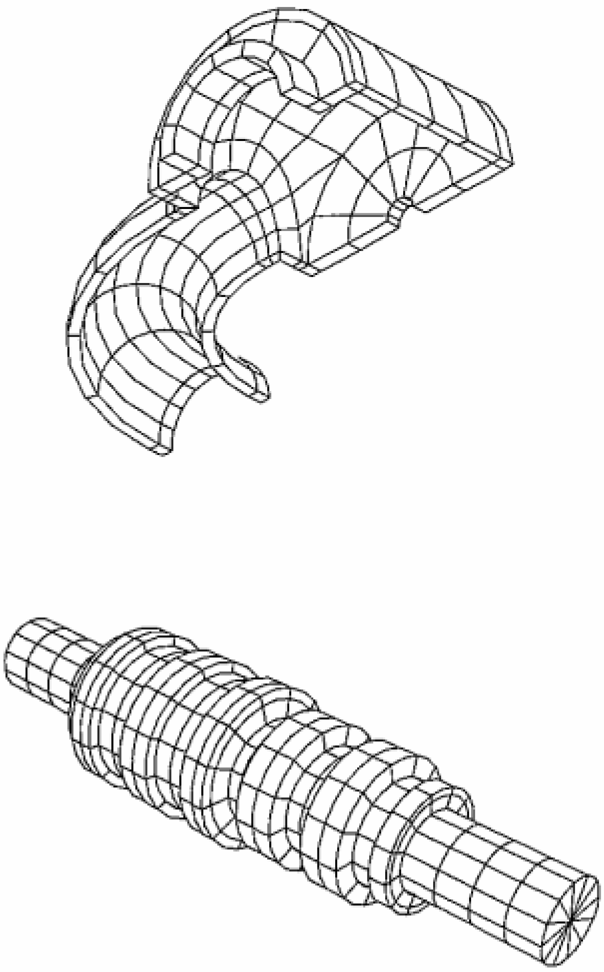

Figure 2: Two of the five meshes used for learning (Dolšak).

this experiment (Dolšak 1991):

357 positive examples
2840 negative examples
2132 background facts

Several relational learning algorithms were tried on this data: GOLEM (Muggleton and Feng 1990), LINUS (Lavrač, Džeroski and Grobelnik 1991) and FOIL (Quinlan 1990). The results obtained with GOLEM were judged to be the most satisfactory. GOLEM generated a large number of rules, some of them being practically irrelevant. For example, although logically correct, they were computationally useless when applied to classifying new edges. On the other hand, some rules appeared useful. Fortunately it was possible to formalise the criteria for distinguishing useful rules from the others. These criteria were implemented as a short Prolog program (Dolšak 1991) for postprocessing the rules generated by GOLEM.

The so resulting set of rules were of interest to expert users of the finite element methods. According to their comments, these rules reveal interesting relational dependences. The following is an example of such a generated rule (the generated syntax is that of Prolog clauses):

```
mesh( Edge, 7)  :-
  usual_length( Edge),
  neighbour_xy( Edge, EdgeY),
  two_side_fixed( EdgeY),
  neighbour_zx( EdgeZ, Edge),
  not_loaded( EdgeZ).
```

This rule says that an appropriate partitioning of `Edge` is 7 if `Edge` has a neighbour `EdgeY` in the xy-plane so that `EdgeY` is fixed at both ends, and `Edge` has another neighbour `EdgeZ` in the xz-plane so that `EdgeZ` is not loaded.

The following is a recursive rule also generated by GOLEM:

```
mesh( Edge, N)  :-
  equal( Edge, Edge2),
  mesh( Edge2, N).
```

This observes that an edge's partition can be determined by looking for an edge of the same length and shape in the same object. Of course, for this rule to be computationally useful, at least some of such equivalent edges has to have its partition determined by its own properties and those of its neighbours.

The accuracy of the induce knowledge base was estimated by a cross-validation mehod. Thereby a subset of 10 % of the example edges was effectively removed from the training set. The remaining 90 % of the data was used for rule induction, and the so induced rules were applied to the removed 10 % of the data now used as a test set. This was repeated ten times.

The results can be summarised as follows. On the average, the classification on the test set was correct in 78 % of the tested edges, incorrect in 2 % of the edges, and the edge remained unclassified (partition unknown) in 20 % of the test edges. An edge remains unclassified if there is no induced rule covering the edge.

In another, more practically realistic evaluation attempt, the generated knowledge base was applied to determining a mesh for a completely new structure, one not used for learning (shown in Figure 1). In this case, 67 % of the edges were classified correctly, 22 % incorrectly, and 11 % remained unclassified.

These results were input into a commercial automatic mesh generator as a partial specification of the mesh. The partial mesh was then completed automatically by the mesh generator, resulting in the mesh shown in Figure 3a. This mesh is close to the known good mesh of Figure 1, but unfortunately not quite acceptable with respect to the resulting numerical errors. Figure 3b shows the mesh generated by the commercial generator without any guidance from the user. This mesh is certainly fine enough with respect to the numerical erros, but completely unacceptable with respect to the computational complexity it requires. Figure 3c is again generated by the commercial generator, only this time guided by the user's advice regarding the "global" size of the elements in the mesh. This is again a deficient mesh which illustrates the generator's inability to adjust the density of the mesh in various regions of the object according to the criticality of the region. Comparing the meshes in Figures 3a-c it becomes clear that the induced knowledge base does "understand" the criticality of varios regions of the object and tries to adjust the density accordingly.

The mesh resulting from the induced knowledge base can actually be easily improved. There is a well known rule of thumb in mesh design that in a rectangular mesh the ratio between the length and width of elements should not exceed 2. Applying this rule to mending the mesh of Figure 3a in fact results in the very good mesh of Figure 1.

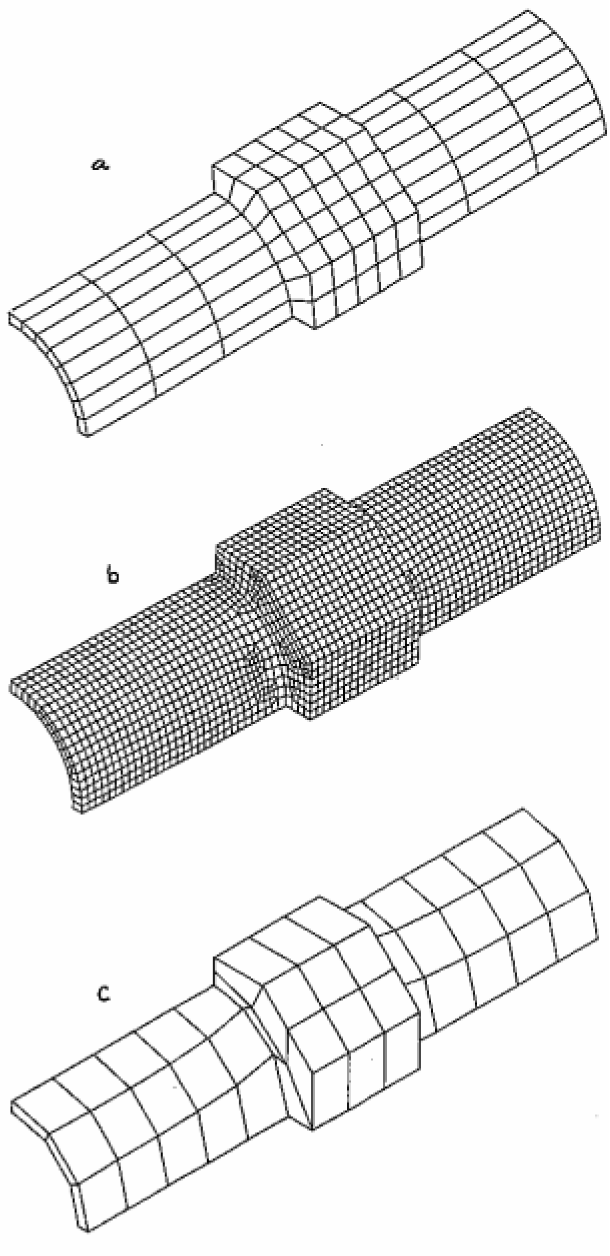

Figure 3: (a) A mesh generated by the induced knowledge base and completed by a commercial generator. (b) The mesh generated by the automatic mesh generator completely autonomously, without any guidance from the user. (c) A mesh, generated by the mesh generator, guided by the user's advice recommending the "global granularity" of 150 mm. (Dolšak 1991)

5 Towards knowledge synthesis

As illustrated by the applications described in this paper, and concluding from many other applications, ML techniques have proved to be a useful tool for efficient construction of expert systems for tasks like classification, prediction, decision making etc. In our experience, for example in the medical domains, employing ML it was possible to inductively construct competent diagnostic systems in the matter of months, weeks or even days (including time for defining the problem, choice of attributes, preparation of learning data, etc.) when it would take much longer without learning.

Muggleton (1991) and Clark et al. (1991) describe another comparison between dialogue-based and induction-based knowledge acquisition for large expert systems with thousands or tens of thousands of rules. That comparison showed that in projects employing ML the knowledge acquisition effort in man years (relative to the number of rules) was one or two orders of magnitude lower than in dialogue-based acquisition. It should be admitted that the basis for comparison was simply the number of rules in the knowldge-base per man-year invested. The quality of rules was not considered. Although the inductively constructed knowledge bases perform accurately, the question still remains whether automatically synthesised knowledge represents symbolically meaningful information. That is, does it tell the humans something about the problem domain in a transparent way that also fits nicely into the human's normal understanding of the domain. In other words, whatever has been induced from examples, does it deserve to be called *knowledge*?

In ML there has been strong awareness of the importance of this comprehensibility criterion (for example Michie 1986 and 1988). There exist some standard techniques that help in this respect. For example, tree pruning in induction of decision trees, in addition to supressing noise, often improves the transparency of induced trees enormously by simply reducing the tree size to, say, 10% of its original size. It should be admitted, however, that compactness is only one measure that is usually correlated with meaningfulness. Human experts often prefer less compact, possibly redundant descriptions, because thay better correspond to the way the problem domain is structured in their heads, or to the way that the knowledge is to be used. The use of knowledge may require not only classification, but for example the achievement of certain goals, explanation, planning, or making decision on the basis of incomplete information. Criteria to decide whether given information deserves to be called *knowledge* are intricate. Of course, these criteria do not exactly correspond to simple measures of accuracy or compactness of induced rules. Identifying and formalising these criteria is an important research topic. Still, there has already been some success in the direction of automatically inducing meaningful information. Knowledge has been generated through ML that was of interest and revealing to human experts.

I will illustrate this by an example from the KARDIO project (Bratko, Mozetič and Lavrač, 1989). In KARDIO, a deep qualitative model of the heart was compiled for efficiency reasons into a large shallow diagnostic knowledge base. This was then compressed, using ML techniques, into a small number of equivalent prediction and diagnostic rules. It was interesting to compare these mechanically synthesised descriptions with human-synthesised descriptions that can be found in the medical literature.

Here is an example of a synthesised prediction rule which tells what are the characteristic features in the ECG signal in the case of the disorder called AV block of the third degree (avb3 for short, possibly combined with any number of other defects in the heart):

```
[av_conduct  =  avb3]  is characterised by
   [rhythm_QRS  =  regular]  and
   [relation_P_QRS  =  independent_P_QRS]
```

This rule is in the VL1 formalism, normally used in the AQ family of programs (Michalski 1983). The propositions have the form `[attribute = value]`. Figure 4 illustrates what essentially happens in the case of the avb3 defect.

For comparison, one of the classical books on ECG (Goldman 1976) describes this arrhythmia as follows: "In this condition the atria and ventricles beat entirely **independently** of one another. ... The ventricular **rhythm** is usually quite **regular** but at a much slower rate (20-60)." Some words here are in bold face to help the comparison between Goldman's description and the machine synthesised description. It is easy to notice strong similarities between both descriptions. It is nice that even the same qualitative descriptors, such as **independent** or **regular** appear in both descriptions. Goldman notices that the ventricular rate is usually much lower (20-60) which is not mentioned in the machine generated description. This is in fact

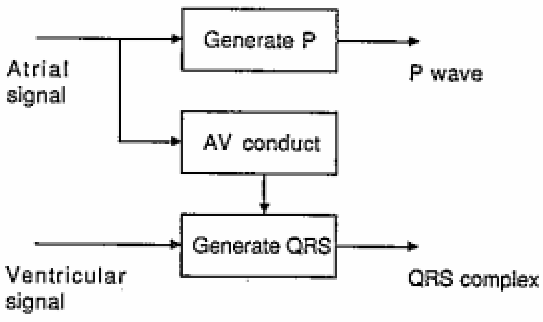

Figure 4: The mechanism of the heart disorder called av-block of the third degree. In the normal heart, the atrial signal reaches the ventricles through the AV-conductance and affects the QRS complex. In the case of the av-block, the atrial signal cannot propagate to the ventricles and has no effect on the QRS complex.

the only essential difference between both descriptions. The reason that the ventricular rate is not mentioned in the machine-generated description is that it is redundant with respect to distinguishing between those conditions of the heart in which avb3 appears, and those in which it does not. Another authority on ECG, Phibbs (1973) describes avb3 as: "(1) The atrial and ventricular **rates are different**: the atrial rate is faster; the ventricular rate is slow and **regular**. (2) There is **no consistent relation between P waves and QRS complexes.**" Again, some descriptors are in bold face to facilitate comparison with the machine-generated description. The comparison is rather straightforward in this case as well.

The example above shows how well some of the synthesised descriptions correspond to those in the standard medical literature. On the other hand, some of the synthesised descriptions are considerably more complex than those in the literature. Machine-generated descriptions in such cases give much more detail that may not be necessary for an intelligent reader with a physiological background. Such a reader can usually infer the missing detail from the background knowledge. Making induced descriptions appealing to humans requires adding some redundancy or leaving out some information that can be usually recovered from background knowledge. How to add and leave out just the right amount is an open research problem.

6 Conclusions

A large number of ML applications confirm the practical importance of this technology. Experience shows that inductive knowledge-acquisition is typically an iterative process whereby the representation, background knowledge and example sets are gradually refined through experiments and feedback obtained from the domain expert. ML tools are repeatedly applied. Induction from examples can be viewed as a way of compiling a high level specification where the specification consists of examples and background knowledge. The practical advantage of this approach lies in the fact that it is often easier to obtain examples (e.g. from the domain expert) than to extract from the expert explicit general laws about the domain.

Until now, attribute-based learning has enjoyed most success in practice. However, the recent important developments in inductive logic programming (ILP) go beyond the limitations of the attribute-based learning. Recent applications of ILP include, in addition to the mesh design described in this paper, the prediction of protein secondary structure (Muggleton et al. 1992). Another exciting area facilitated by ILP is automated construction of qualitative models from observed behaviours. Work that has been done in this direction includes (Mozetič 1987a,b; also described in Bratko et al. 1989), (Coiera 1989), (Bratko et al. 1991) and (Krann et al. 1991)

Acknowledgements

A number of medical doctors helped in the applications described in this paper. I would like to thank S. Hojker, G. Klanjšček, J.Lamovec, V. Pirnat, M. Soklič and M. Zwitter of the University Medical Center, Ljubljana, P. Abrams and M. Torrens of the Clinical Investigation Unit, Ham Green Hospital, Bristol, UK, and G. Gong of Carnegie-Mellon University, Pittsburgh, who helped in the medical applications of induction either by the data for learning or by advice. M. Horvat, B. Čerček, A. Grad and P. Rode assisted with the cardiological expertise in the KARDIO project. I would also like to acknowledge the cooperation of my AI colleagues whose names appear in the references. Special thanks to Igor Kononenko for his work in medical domains, Bojan Dolšak for lending the mesh design figures, and Sašo Džeroski who helped in the preparation of this paper.

This paper was written while the author was visiting the Strathclyde University and Turing Institute, Glasgow, Scotland

References

Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J. (1984) *Classification and Regression Trees.* Belmont, CA: Wadsworth.

Bratko, I., Kononenko, I. (1987) Learning diagnostic rules from incomplete and noisy data. In: *Interactions in AI and Statistics* (ed. B. Phelps) London: Gower Technical Press.

Bratko, I., Mozetič, I, Lavrač, N. (1989) *KARDIO: A Study in Deep and Qualitative Knowledge for Expert Systems.* Cambridge, Massachusetts: The MIT Press.

Bratko,I., Muggleton, S., Varšek, A. (1991) Learning qualitative models of dynamic systems, *Proc. Inductive Logic Programming ILP-91*, Viana do Castelo, Portugal, March 1991. (An abbreviated version also in *Machine Learning: Proc. Eighth Int. Workshop*, eds. L.A. Birnbaum and G. Collins, San Mateo, CA: Morgan Kaufmann).

Bratko, I., Mulec, P. (1979) An experiment in automatic learning of diagnostic rules. *Informatica*, Vol. 4, No. 4, 353-359.

Cestnik, B., Bratko, I. (1988) Learning redundant rules in noisy domains, *Proc. ECAI 88, European Conf. on Artificial Intelligence*, Muenchen, August 1988

Cestnik, B., Kononenko, I., Bratko, I. (1987) ASSISTANT 86: a knowledge elicitation tool for sophisticated users. In *Progress in Machine Learning* (eds. I. Bratko, N. Lavrač; Proc. Second European Working Session on Learning. Bled, Slovenia, 1987) Wilmslow, England: Sigma Press, 1987.

Chandrasekaran, B. (1991) Interview with Frederick Hayes-Roth and Richards Fikes, *IEEE Expert*, October 1991, pp. 3-14.

Chilausky, R., Jacobsen, B., Michalski, R.S. (1976) An application of variable-valued logic to inductive learning of plant disease diagnostic rules. *Proceedings of the sixth international symposium on multiple-valued logic.* May 25-28, 1976, Logan, Utah.

Clark, P., Cestnik, B., Sammut, C., Stender, J. (1991) Applications of machine learning: notes from the panel members. *Machine Learning – EWSL-91; Proc. European Working Session on Learning* (Porto, Portugal, March 1991) Springer-Verlag.

Clark, P., Niblett, T. (1989) The CN2 induction algorithm. *Machine Learning*, Vol. 3, no. 4, 261-284.

Coiera, E. (1989) Generating qualitative models from example behaviours. DCS Report No. 8901, School of Electr. Eng. and Computer Sc., Univ. of New South Wales, Sydney, Australia.

Dolšak, B., Muggleton, S. (1991), The application of Inductive Logic Programming to Finite Element Mesh Design. Proc. ILP-91, Viana do Castelo, Portugal.

Dolšak, B. (1991) Determining the geometric model of finite element meshes using AI methods. University of Maribor: CAD Center (M.Sc. thesis) Maribor, Slovenia.

Gams, M. (1989) New measurements highlight the importance of redundant knowledge, *Proc. 4th European Working Session on Learning* (ed. K. Morik; Montpellier, France) Pitman and Morgan Kaufmann.

Kononenko, I., Bratko, I. (1991) Information based evaluation criterion for classifier's performance. *Machine Learning*, Vol. 6, 67-80.

Krann, I., Richards, B., Kuipers, B. (1991) Automatic abduction of qualitative models, *Proc. QR-91*, Austin, Texas, May 1991.

Lavrač, N., Džeroski, S, Grobelnik, M. (1991) Learning nonrecursive definitions of relations with LINUS. *EWSL '91: Machine Learning: Proceedings of the European Working Session on Learning* , Porto, Portugal (ed. Y. Kodratoff) Springer-Verlag, Lecture Notes in Artificial Intelligence.

Leech, W.J. (1986) A rule based process control method with feedback. In: Advances in Instrumentation, Vol 41, Part 1, 69 - 175.

Michalski, R.S. (1983) A theory and methodology of inductive learning. In: *Machine Learning: an Artificial Intelligence Approach* (eds. R.S. Michalski, J.G. Carbonell and T.M. Mitchell) Palo Alto, CA: Tioga.

Michalski, R.S., Chilausky, R.L. (1980) Learning by being told and learning from examples: an experimental comparison of the two methods of knowledge acquisition in the context of developing an expert system for soybean disease diagnosis. *International Journal of Policy Analysis and Information Systems*, Vol. 4, 125-161.

Michie, D. (1986), The superarticulacy phenomenon in the context of software manufacture. *Proceedings of the Royal Society A*, London, Vol. 405, 185 - 212.

Michie, D. (1988) Machine learning in the next five years. *EWSL-88 - Proc. 3rd European Working Session on Learning*, Glasgow, 1988. London: Pitman.

Michie, D. (1989) Problems of computer-aided concept formation. In *Applications of Expert Systems, Vol. 2* (ed. J.R. Quinlan) Turing Institute Press in association with Addison-Wesley)

Michie, D. (1990) Personal models of rationality. *Journal of Statistical Planning and Inference*, Vol. 25, 381-399.

Michie, D., Muggleton, S., Riese, C., Zubrick, S. (1984) RuleMaster: a second generation knowledge engineering tool, *Proc. 1st Conf. Artificial Intelligence Applications, IEEE Computer Society.*

Mingers, J. (1989a) An empirical comparison of selection measures for decision tree-induction. *Machine Learning*, Vol. 3, No. 4, 319-342.

Mingers, J. (1989b) An empiricial comparison of pruning methods for decision-tree induction. *Machine Learning*, Vol. 4, No. 2.

Mozetič, I. (1987a) Learning of qualitative models. In *Progress in Machine Learning* (eds. I. Bratko, N. Lavrač; Proc. 2nd European Working Session on Learning, Bled, Slovenia) Wilmslow: Sigma Press.

Mozetič, I. (1987b) The role of abstractions in learning qualitative models. *Proc. Fourth Int. Workshop on Machine Learning*, Irvine, CA, June 1987. Morgan Kaufmann.

Muggleton, S. (1991), Inductive logic programming, *New generation computing*, Vol. 8, 295 - 318.

Muggleton, S. (1992, ed.) *Inductive Logic Programming*, Academic Press.

Muggleton, S., Buntine, W. (1988) Machine invention of first order predicates by inverting resolution. *Proc. 5th Int. Conf. Machine Learning*, Ann Arbor, MI, 1988.

Muggleton, S., Feng, C. (1990) Efficient induction of logic programs. *Proc. First Conf. Inductive Learning Theory*, Tokyo: Japanese Society for Artificial Intelligence, 1990.

Muggleton, S., King, R.D., Sternberg, M.J.E. (1992) Protein secondary structure using Logic (submitted for publication).

Niblett, T., Bratko, I. (1986) Learning decision trees in noisy domains. In: *Expert Systems 86: Proc. Expert Systems 86 Conf.* (ed. M. Bramer) Cambridge Univ. Press.

Pirnat, V., Kononenko, Janc, T., Bratko, I., Medical analysis of automatically induced diagnostic rules, *Proc. Third Int. Conf. AI in Medicine*, London, August 1989.

Quinlan, J.R. (1979), Discovering rules by induction from large collections of examples. In *Expert systems in the Microelectronic Age* (ed. D. Michie) Edinburgh University Press.

Quinlan, J.R. (1986) Induction of decision trees. *Machine Learning Journal*, Vol. 1, 81-106.

Quinlan, J.R. (1990) Learning logical definitions from relations. *Machine Learning Vol. 5*, pp 239-266.

Quinlan, J.R., Compton, P., Horn, K.A., Lazarus, L. (1989) Inductive knowledge acquisition: A case study. In *Applications of Expert Systems, Vol. 2* (ed. J.R. Quinlan) Turing Institute Press in association with Addison-Wesley.

Roškar, E., Abrams, P., Bratko, I., Kononenko, I., Varšek, A., MCUDS - an expert system for the diagnostics of lower urinary tract, *Journal of Biomedical Measurement Informatics and Control*, Vol. 1, No. 4, 201-204 (1986).

Urbančič, T., Kononenko, I., Križman, V. (1991) Review of AI applications at Ljubljana AI Labs. Ljubljana, J. Stefan Institute: Technical Report DP-6183.