Relationship between Machine Learning and Some Research Lines at ICOT

Report on Visit to ICOT in October 1991

Pavel B. Brazdil
LIACC, University of Porto
Portugal

Introduction

During my visit to ICOT which covered the period between 14-26 October I had many interactions with the members of the 5th Laboratory (Theorem Proving). This is due to the fact that I am interested not only in machine learning, but also in applying machine learning methods to the process of development of logic programs. These interests are closely related to the concerns of this laboratory.

Apart from my contacts with the 5th Laboratory I have also met some members of the 4th Laboratory (Constraint Logic Programming) and 7th Laboratory (Parallel Applications) which enabled me to get a more complete picture of ICOT and its activities.

Acknowledgements

I am grateful to Dr. Koichi Furukawa for suggesting this interesting possibility to me during the EWSL-91 Conference in Porto (Portugal), and to Dr. Kazuhiro Fuchi for turning this possibility into a real fruitful event .

I am grateful also to Dr. Iwata and Kumiko Karakawa for resolving numerous administrative problems, and to Tadashi Kawamura for arranging an interesting schedule of meetings. Last but not least, I would like to thank everyone else I met for the kind reception, support and many stimulating discussions during my stay in ICOT.

Lectures and other Presentations

Invited Lecture on *Learning in Multi-agent Environments*
24 Oct '91 at the Workshop on Algorithmic Learning Theory (ALT-91).

Talk on *Learning in Multi-agent Environments*
22 Oct '91, in the monthly series of talks on Parallel Theorem Proving organized by ICOT Res. Center. Seminar on a similar topic at ICOT Res. Center, 17 Oct '91.

Program Synthesis and Inductive Methods

I was most interested to discuss the Dr. Furukawa's *Program Synthesis based on Induction and Verification* and contrast it with our own ideas on program development using *Inductive Engineering*. I have also had a discussion with Ishizaka on generalization of terms and assisted a demonstration of his system. This system was developed on the basis of H. Arimura, T. Shinohara and S. Otsuki paper on *Unions of Tree Patterns*. In addition I had talks with M. Fujita who has described the work on *Program Synthesis* in ICOT.

Dr. Furukawa's work on *Program Synthesis based on Induction and Verification* includes many lines that interest me personally. Our approach been formulated recently in the form of a proposal for research in *Inductive Engineering,* prepared in collaboration with other European universities and institutions. Both proposals share the idea that induction could be used to develop programs, but do not limit the study to this issue only. Both proposals propose to exploit specifications, and also, to incorporate some verification techniques. Because of these common interests, further collaboration on these lines would be of interest to us.

The work on program synthesis has been carried out by Dr. R. Hasegawa, H. Fujita and M. Fujita employs a MGTP (model generation based theorem prover) which attempts to extract a program from a proof. The techniques bear some similarity to the ones used in explanation based learning (EBL). The EBL method constructs an explicit representation of the proof in the process of conducting the proof. The explicit representation is then manipulated to obtain the required (more efficient) version of the original program.

A similar strategy is used in the work on program synthesis. However, the system based on MGTP is more powerful. First, it uses a richer language to represent knowledge (FOL). Secondly, it can incorporate inductive schemata in the proof and in that way construct new recursive definitions. As the language is richer (the the system may employ various rules of inference), some of the steps in the program extraction phase are not so transparent to someone not so familiar with the system.

Program Transformation and Inductive Methods

T. Kawamura has described his system based on *unfold/fold* method. I looked for similarities between this approach and the existing work in machine learning. As has been noted earlier (e.g. during S.Muggleton's visit to ICOT) the operation of *folding* can be related to *inverse resolution* of Muggleton and Buntine. This point was later reiterated during a discussion with Dr. Furukawa, who has also drawn my attention to other earlier work done in ICOT in the area of program transformation (e.g. paper of Seki and Furukawa).

It is interesting to note that in most work on program transformation the program is manipulated in the abstract, that is, without recourse to examples. In inductive programming, however, examples are exploited to a great extent and this seems to endow these methods with a great power. As T.Kawamura pointed out, this idea has already been exploited by Bruynooghe, de Raedt and de Schreye (IJCAI-89). Although, this method is applicable only to a subclass of problems, it would seem that this idea is worth exploring further.

In the system described by Bruynooghe, de Raedt and de Schreye only one example is used to guide the process of transformation. This falls short to what is common in inductive programming. Examples help in checking whether the induced program, or any intermediate version *covers* all (or most) the given examples and none (or few) negative examples.

The inductive learning system that is being developed at LIACC laboratory in Porto (by L. Torgo) attempts to search the space of possible generalizations / specializations in an incremental way, that is by performing modifications of rules after each new example has become available. This visit could prompt us to set up a new line of research in which we would attempt to extend our system to perform some unfold/fold like transformations in an incremental way.

Hypothetical Reasoning and Inductive Methods

During my visit to ICOT I was also introduced to the work on hypothetical reasoning, abduction and analogy which I have followed with great interest, as these areas too share some concerns with machine learning. Katsumi Inoue explained a system for *Hypothetical reasoning in logic programs;* Yoshihiko Ohta described *Forward chaining hypothetical reasoner linked to ATMS;* Jun Arima described *Logical Analysis of Relevance in Analogy.*

K. Inoue's system follows Poole's framework in which *default knowledge H* is separated from a *theory T* about the world. One of the main tasks of his knowledge system is to find a maximal subset E of H such that a consistent answer set (extension) could be found for the extended program E U H. As K.Inoue has pointed out, if adding assumptions causes inconsistencies, a minimal set of assumptions should be ignored to restore consistency. In this work it is assumed that theory T incorporates the part of knowledge that we are quite confident about, and hypothesis H represent knowledge that is less certain (default knowledge or possible assumptions). A question arises on what grounds (other than intuitive ones) we can make this separation. This is where inductive methods can come in.

A typical rule induction system (including the one written in Porto) associates two kinds of estimates with each rule generated. One provides an estimate of the probability that the rule will be correct in the new relevant case (i.e whenever the rule will apply). The second gives an estimate of the generality of the rule, which is determined by looking at the proportion of cases of the domain that are covered by this rule. The first estimate is often called the estimate of *correctness*, and the second one estimate of *completeness* or *coverage*. These two estimates could provide us with the basis for deciding whether the given rule should be placed in T or H (or neither). Theory T could include all rules with very high correctness. Default knowledge could include all rules with high coverage and reasonable level of correctness. We note that some rules may not come under any of the two categories (these are often useful in further refinements).

We could envisage that the systems capable of hypothetical reasoning (such as the one discussed by K.Inoue) could be used in conjunction with an inductive system in the following way. The hypothetical system could be used to identify potential inconsistencies and suggest ways of removing them.

We observe that the process of omitting the probabilities represents a kind of *qualitative abstraction* which is easier for us to handle. It seem easier to construct an *extension* for a given T and H, than to calculate how plausible it actually may be. The schemes for hypothetical reasoning delimit the space of plausible extensions (e.g. an extension must not contain inconsistencies), and usually are not concerned with more refined (e.g. quantitative) distinctions of plausibility.

The introduction of *soft constraints* (in the context of CLP) by K. Satoh and Dr. A. Aiba provides more refined graduation of plausibility without having the disadvantage of going to the full quantitative model. An important question is what kind of abstraction is really justified in a given situation. The answer to this question is not easy, as this issue is related to the difficult *representation problem* that AI did not get a good grasp of yet.

Constraint Logic Programming

In this series of meetings *Ken Satoh* described the method for *Computing Soft Constraints by Hierarchical Constraint Logic Programming (HCLP)* (work done together with Dr. A. Aiba) and some fascinating applications, e.g. the problem of gear box design; Dr. Aiba explained also some theoretical background of *Constraint Logic Programing* (joint work with K. Sakai). I have also got familar with the work of Y. Ohta and K. Inoue in the area of *upside-down meta-interpretation*.

As CLP methods appear to be very powerful, but is limited to sets of algebraic equations, it is tempting to explore these ideas further and fuse them with inductive logic programming. It seems attractive that one could describe, in a very declarative manner, the obvious relationships in a given domain, without concern to which facts are known, or what the goals are. Then one would attend to the given facts and goals and let the system transform this representation into a suitable procedural form.

There is still another possibility, however. Instead of trying to transform the given declarative form into a procedural one, one could look for intelligent ways of interpreting the given declarative knowledge. Some interpreters of this kind do exist already (e.g. upside-down meta-interpretation). Future research could show how these could be best exploited.

Meetings with Seventh Laboratory in ICOT

During my stay in ICOT I also met N. Ichiyoshi and Dr. K. Nitta who have set up a video session for me describing the main features of the Multi-PSI and PIM systems, and the applications that are being developed to run on parallel hardware. Dr. K. Nitta has then supplemented this session with additional explanations. The legal application seemed of particular interest to me, as it was conceivable that one could apply machine learning techniques to derive a decision in a particular case. One difficulty that I could see, however, was that this application contained a relatively small number of cases. This seems to be due to the fact that each case is rather complex and hence difficult to code and enter into the machine.

Curriculum Vitae
of
Pavel B. Brazdil

Pavel Brazdil was born in Czechoslovakia and completed his first degree in Brno in 1968. During 1969-81 he was an employee of IBM in Great Britain where he worked on software design etc. Pavel Brazdil has done his postgraduate studies at the Dept. of Artificial Inteligence in Edinburgh where completed his Ph.D. in 1981. Since 1982 he has been lecturing at the University of Porto in Portugal.

Dr. Brazdil has presented about 15 papers at various International Workshops and Conferences. Many of these were published in books. He has also edited a book called *Machine Learning, Meta-reasoning and Logics* which was published by Kluwer Academic Press.

In 1990 he was invited to give a Summary talk at the European Conference on Artificial Intelligence (ECAI-90) in Stockholm, Sweden. Dr. Brazdil was also invited to lecture at two International Machine Learning Summer Schools (in 1988, Les Arcs, France; in 1989, Urbino, Italy).

Dr. Brazdil has worked also in Programme Committees of about 10 international Workshops and Conferences (including IJCAI).

Dr. Brazdil has organized several workshops and conference: *EPIA-86* in Porto; Programme Chair of *Machine Learning, Meta-reasoning and Logics,* Sesimbra, Portugal, 1988; Local Chair of *European Working Session on Machine Learning (EWSL-90),* Porto, 1990. Dr. Brazdil was elected to Chair the next European Machine Learning Conference *(EWSL-92)* that will take place in Viena, Austria. Dr. Brazdil's group is taking part in three European Projects (2 Esprit and a Tempus Project).