Machine Learning, 23, 109–111 (1996)

A Reply to Pazzani's Book Review of "Inductive Logic Programming: Techniques and Applications"

NADA LAVRAČ nada.lavrac@ijs.si
SAŠO DŽEROSKI saso.dzeroski@ijs.si
J. Stefan Institute, Jamova 39, 61000 Ljubljana, Slovenia

Editor: Alberto Segre

1. Introduction

We are pleased to reply to Michael Pazzani's thorough review of our book on Inductive Logic Programming (ILP). The book gives an introduction to this new and fast growing field. We would like to emphasize that it also gives an in-depth account of the most established and applicable techniques within the field and several applications of these techniques. Our reply presents the view we took when writing the book; this answers most of the specific points made by the reviewer.

Inductive Logic Programming (ILP) is concerned with learning first-order rules formulated in the language of logic programs. ILP systems use this language for representing background knowledge, examples and hypotheses. The main motivation for using this language is its clear syntax and semantics, as well as the sound theoretical and practical methods for deductive inference in it. Early relational learning systems, such as ARCHES, INDUCE, and ML-SMART, which use other representation formalisms, are not generally considered ILP systems. Thus, they are not discussed in the book.

We have consciously biased the book toward ILP techniques and systems that have reached a certain degree of applicability to practical problems. This has strongly influenced the choice of topics covered by the book. For example, while learning recursive rules (inverted implication) and predicate invention have received a lot of attention within the ILP community, few practical results exist so far. Furthermore, while learnability results are important, they are not of immediate practical interest. Consequently, we have decided not to discuss learnability, including, for example, the learnability results obtained by transforming ILP problems to propositional form (Dzeroski et al., 1992).

The practical orientation of the book explains the omission of several techniques and systems mentioned in the review, including interactive ILP systems (e.g., CLINT). Although the first ILP systems were of an interactive nature and important developments in this area have been made, few interactive ILP systems have been applied to practical problems. Interaction with the user (oracle) is in fact demanding and so are the practical applications of interactive ILP systems. The empirical ILP setting, on the other hand, resembles the well-understood propositional learning setting as used in the widely-used ID3 and AQ systems. In short, most existing ILP applications involve empirical ILP systems; hence the focus of our book is on empirical ILP.

In addition to giving an introduction to ILP, the book is also an in-depth study on multi-class learning and the handling of real numbers, imperfect data and irrelevant features, which are central topics of our own ILP research. These techniques are embedded in the ILP systems LINUS, DINUS, and mFOIL. Our research has focused on adapting techniques from propositional learning for use in ILP. Transforming an ILP problem to propositional form (as done in LINUS and DINUS) enables the use of a variety of propositional learning techniques. Moreover, these techniques can also be adapted for direct use in ILP systems (e.g., for noise-handling in mFOIL). In the transformation approach, background knowledge (in the form of a logic program) is used to introduce new attributes. Although this may look simple (as in the medical application), it suffices for learning the important class of determinate logic programs, which is also the hypothesis language of GOLEM (Muggleton & Feng 1990). Besides its practical applicability, the transformation approach is important for understanding the relation of propositional learning to ILP and the computational complexity of learning.

Although many ILP applications are still proof-of-the-principle applications, some are getting closer to real use. The accuracies achieved in the finite element mesh design domain are, in fact, much higher now (Dolsak et al., 1994) than originally reported in the book. The molecular biology ILP applications by Stephen Muggleton and his colleagues (Muggleton et al., 1992; Srinivasan et al., 1994) are important in the sense that they have produced new knowledge published in top scientific journals in the respective application areas. Two of these applications, drug design and protein secondary structure prediction, are summarized in the book. The recent application to predicting mutagenicity of chemical compounds from their chemical structures is especially noteworthy, as this is a problem that can not be addressed by either propositional systems or determinate ILP systems (Srinivasan et al., 1994). We are pleased to observe that applications more interesting than those presented in the book are emerging every day.

The lack of a conclusion chapter that would recommend particular ILP systems for particular problem classes and give information on how to obtain ILP systems is partly due to the fact that ILP is a young and fast-growing field. It is difficult to make definite conclusions and recommendations about which algorithm is appropriate for which class of problems; this is a difficult task even for the propositional learning area, which has been active for a long time. While few ILP systems were publicly available at press time, several are now available via anonymous ftp.[1]

We believe that despite its inevitably incomplete coverage of the field, our book is a timely introductory text that contributes to the development of ILP and its accessibility to the outside world.

Notes

1. ILP software, datasets, and publications can be obtained by anonymous ftp from Bonn, Germany at `ftp://ftp.gmd.de/MachineLearning/ILP/public` in the subdirectories `software`, `data`, and `papers`. Some can also be found in the UCI Machine Learning Repository. Additional information on ILP research can be accessed through the World Wide Web server located in Ljubljana, Slovenia at `http://www-ai.ijs.si/ilpnet.html`.

References

Dolšak, B., Bratko, I., and Jezernik, A. (1994). Finite-element mesh design: an engineering domain for ILP application. In S. Wrobel (Ed.) *Proc. Fourth Int. Workshop on Inductive Logic Programming (ILP-94)* (pp. 305–320) Bad Honnef/Bonn. GMD Studien Number 237. Bonn: GMD.

Džeroski, S., Muggleton, S., and Russell, S. (1992). PAC-learnability of determinate logic programs. In *Proc. Fifth ACM Workshop on Computational Learning Theory* (pp. 128–135). New York: ACM Press.

Muggleton, S., and Feng, C. (1990). Efficient induction of logic programs. In *Proc. First Conference on Algorithmic Learning Theory* (pp. 368–381) Ohmsha, Tokyo.

Muggleton, S., King, R.D., and Sternberg, M.J.E. (1992). Protein secondary structure prediction using logic. *Protein Engineering, 5*, 647–657.

Srinivasan, A., Muggleton, S.H., King, R.D., and Sternberg, M.J.E. (1994). Mutagenesis: ILP experiments in a non-determinate biological domain. In S. Wrobel (Ed.) *Proc. Fourth Int. Workshop on Inductive Logic Programming (ILP-94)* (pp. 217–232) Bad Honnef/Bonn. GMD Studien Number 237. Bonn: GMD.