Guest Editors' Introduction

STEPHEN MUGGLETON

steve@comlab.ox.ac.uk

*Oxford University Computing Laboratory
Wolfson Building
Parks Road
Oxford, OX1 3QD
United Kingdom*

DAVID PAGE

dpage@comlab.ox.ac.uk

*Oxford University Computing Laboratory
Wolfson Building
Parks Road
Oxford, OX1 3QD
United Kingdom*

Since its inception in 1991 (Muggleton, 1991), inductive logic programming (ILP) has been distinguished from other areas of automated inductive learning by its choice of representation language. The representation language for ILP is definite clause logic, as embodied in the logic programming language Prolog. Because of its richness and simple declarative form, definite clause logic is suitable for areas of computer science as diverse as relational databases and graph algorithms, and for application areas as diverse as organic chemistry/molecular biology and natural language. Indeed ILP already has seen successes in these application areas (Bratko & Muggleton, 1995, Zelle & Mooney, 1996).

As a result of its representation language, ILP brings the theoretical and implementational emphases of logic programming together with the applications-oriented experimental nature of machine learning. Consequently, it has been the aim of ILP researchers to drive their theoretical and implementational work by practical applications, so that theory, implementations, and applications can develop together. This aim was prominent at a 1995 planning meeting for a European Community-wide research effort into ILP, where ILP researchers identified key scientific topics for further research. These topics were chosen specifically because they are critical to further success in major application areas including those mentioned in the first paragraph. The most important scientific topics include issues in learning logical theories (hypotheses) that involve

1. multiple—and mutually-dependent—predicates,
2. nontrivial use of floating point numbers,
3. complex clauses,
4. large and/or complex background theories (initial logical theories provided to a learning algorithm, which the algorithm augments to produce the final theory).

Each of the papers in this special issue makes a contribution to work on one or more of these difficult issues.

The first paper, by DeRaedt and DeHaspe, contributes to the issue of learning theories involving multiple predicates. This issue is important, for example, in discovery within relational databases, where a database typically consists of several different relations (predicates).

The second and third papers both address the issue of learning theories involving floating-point numbers. The second paper, by Karalic and Bratko, examines this issue in the context of a wide variety of applications, while the third paper, by Hau and Coiera, focuses on the context of qualitative simulation in a medical domain.

Much of logic programming research in recent years has gone into extending the boundaries of the representation, for example by moving to higher-order logic or removing the definite-clause restriction within certain contexts. The fourth paper in this special issue, by Idestam-Almquist, examines the theoretical issues involved in learning general clauses as opposed to definite clauses only.

The final paper, by Morales, describes one approach to learning in the presence of large, complex background theories. The paper exemplifies the approach with applications to learning concepts in chess, qualitative simulation, and music.

In closing, we would like to caution the reader that the purpose of this special issue is to present state-of-the-art research results regarding some of the most crucial issues for ILP. It is not to provide a survey of earlier research results or a history of the field. That purpose is already served by two earlier special issues, to which we refer the reader (Muggleton, Mizoguchi & Furukawa, 1995, Muggleton, 1994).

References

- S. Muggleton (Guest Editor). Special issue on inductive logic programming. *SIGART Bulletin*, 5(1), 1994.
- I. Bratko & S. Muggleton. Applications of inductive logic programming. *Communications of the ACM*, 38(11):65–70, 1995.
- S. Muggleton. Inductive logic programming. *New Generation Computing*, 8(4):295–318, 1991.
- S. Muggleton, F. Mizoguchi, & K. Furukawa (Guest Editors). Special issue on inductive logic programming. *New Generation Computing*, 13, 1995.
- J. Zelle & R. Mooney. Comparative results on using inductive logic programming for corpus-based parser construction. In S. Wermter, E. Riloff, and G. Scheler, editors, *Symbolic, Connectionist, and Statistical Approaches to Learning for Natural Language Processing*. Springer-Verlag, 1996.