Inductive Logic Programming: issues, results and the LLL challenge

Stephen Muggleton [1]

Abstract. Inductive Logic Programming (ILP) [9, 11] is the area of AI which deals with the induction of hypothesised predicate definitions from examples and background knowledge. Logic programs are used as a single representation for examples, background knowledge and hypotheses. ILP is differentiated from most other forms of Machine Learning (ML) both by its use of an expressive representation language and its ability to make use of logically encoded background knowledge. This has allowed successful applications of ILP [1] in areas such as molecular biology [12, 10, 6, 5] and natural language [7, 3, 2] which both have rich sources of background knowledge and both benefit from the use of an expressive concept representation languages. For instance, the ILP system Progol has recently been used to generate comprehensible descriptions of the 23 most populated fold classes of proteins [14], where no such descriptions had previously been formulated manually. In the natural language area ILP has not only been shown to have higher accuracies than various other ML approaches in learning the past tense of English [8] but also shown to be capable of learning accurate grammars which translate sentences into deductive database queries [15]. In both cases, follow up studies [13, 4] have shown that these ILP approaches to natural language problems extend with relative ease to various languages other than English.

The area of Learning Language in Logic (LLL) is producing a number of challenges to existing ILP theory and implementations. In particular, language applications of ILP require revision and extension of a hierarchically defined set of predicates in which the examples are typically only provided for predicates at the top of the hierarchy. New predicates often need to be invented, and complex recursion is usually involved. Similarly the term structure of semantic objects is far more complex than in other applications of ILP. Advances in ILP theory and implementation related to the challenges of LLL are already producing beneficial advances in other sequence-oriented applications of ILP. In addition LLL is starting to develop its own character as a sub-discipline of AI involving the confluence of computational linguistics, machine learning and logic programming.

ACKNOWLEDGEMENTS

This work was supported partly by the Esprit Long Term Research Action ILP II (project 20237), EPSRC grant GR/K57985 on Experiments with Distribution-based Machine Learning and an EPSRC Advanced Research Fellowship held by the author.

[1] Department of Computer Science, University of York, Heslington, York, YO1 5DD, United Kingdom.

ECAI 98. *13th European Conference on Artificial Intelligence*
Edited by Henri Prade
Published in 1998 by John Wiley & Sons, Ltd.

REFERENCES

[1] I. Bratko and S. Muggleton, ‘Applications of inductive logic programming’, *Communications of the ACM*, **38**(11), 65–70, (1995).

[2] J. Cussens, ‘Part-of-speech tagging using Progol’, in *Proc. of the 7th International Workshop on Inductive Logic Programming (ILP-97)*, pp. 93–108. Springer-Verlag, (1997). LNAI 1297.

[3] James Cussens, David Page, Stephen Muggleton, and Ashwin Srinivasan, ‘Using Inductive Logic Programming for Natural Logic Processing’, in *ECML’97 – Workshop Notes on Empirical Learning of Natural Language Tasks*, eds., W. Daelemans, T. Weijters, and A. van der Bosch, pp. 25–34, Prague, (1997). University of Economics. Invited keynote paper.

[4] S. Džeroski and T. Erjavec, ‘Induction of Slovene nominal paradigms’, in *Proceedings of the 7th International Workshop on Inductive Logic Programming*, eds., N. Lavrac and S. Džeroski, (1997). LNAI 1297.

[5] P. Finn, S. Muggleton, D. Page, and A. Srinivasan, ‘Pharmacophore discovery using the inductive logic programming system progol’, *Machine Learning*, **30**, 241–271, (1998).

[6] R. King, S. Muggleton, A. Srinivasan, and M. Sternberg, ‘Structure-activity relationships derived by machine learning: the use of atoms and their bond connectives to predict mutagenicity by inductive logic programming’, *Proceedings of the National Academy of Sciences*, **93**, 438–442, (1996).

[7] R.J. Mooney, ‘Inductive logic programming for natural language processing’, in *Proceedings of the Sixth International Workshop on Inductive Logic Programming*, ed., S. Muggleton, 3–21, Springer-Verlag, Berlin, (1997). LNAI 1314.

[8] R.J. Mooney and M.E. Califf, ‘Induction of first-order decision lists: Results on learning the past tense of english verbs’, *Journal of Artificial Intelligence Research*, **3**, 1–24, (1995).

[9] S. Muggleton, ‘Inductive logic programming’, *New Generation Computing*, **8**(4), 295–318, (1991).

[10] S. Muggleton, R. King, and M. Sternberg, ‘Protein secondary structure prediction using logic-based machine learning’, *Protein Engineering*, **5**(7), 647–657, (1992).

[11] S. Muggleton and L. De Raedt, ‘Inductive logic programming: Theory and methods’, *Journal of Logic Programming*, **19,20**, 629–679, (1994).

[12] M. Sternberg, R. King, R. Lewis, and S. Muggleton, ‘Application of machine learning to structural molecular biology’, *Philosophical Transactions of the Royal Society B*, **344**, 365–371, (1994).

[13] C.A. Thompson, R.J. Mooney, and L.R. Tang, ‘Learning to parse natural language database queries into logical form’, in *Workshop on Automata Induction, Grammatical Inference and Language Acquisition*, (1997).

[14] M. Turcotte, S.H. Muggleton, and M.J.E. Sternberg, ‘Protein fold recognition’, in *Proc. of the 8th International Workshop on Inductive Logic Programming (ILP-98)*. Springer-Verlag, (1998). (To appear).

[15] J. Zelle and R. Mooney, ‘Learning to parse database queries using Inductive Logic Programming’, in *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pp. 1050–1055, Portland, Oregon, (1996). AAAI Press/MIT Press.