Machine Learning, 44, 207–209, 2001

Editorial: Inductive Logic Programming is Coming of Age

PETER FLACH Peter.Flach@bristol.ac.uk
University of Bristol, Woodland Road, Bristol BS8 1UB, UK

SAŠO DŽEROSKI Saso.Dzeroski@ijs.si
Jozef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia

This issue of the Machine Learning journal is devoted to inductive logic programming (ILP). The Ninth International Workshop on ILP (ILP'99) was held in June 1999 in Bled (Slovenia). After the workshop, we invited authors who presented their work at ILP'99 to consider submitting a suitably extended version of their paper to the Machine Learning journal. As ILP workshops are not (indeed, should not be) the only place where work on ILP can be found, we scrutinised other proceedings such as ICML, ECML, COLT, KDD, and UAI for ILP-related papers and wrote to their authors with a similar request. By the end of March 2000 we had received 12 submissions, each of which was reviewed by 3 reviewers. On the basis of the reviews, we selected the four papers that are included in this issue.

In its most common form, ILP is concerned with inducing rules from examples and background knowledge, all of which are expressed as Prolog programs. This uniformity of representation is relatively unique within the diverse field of machine learning, and has contributed significantly to the identity and coherence of inductive logic programming as a field of research. However, one should not confuse the contingencies of syntax with the essentials of representation. What is crucial about ILP is not that rules are written with the conclusion preceding the conditions, or that variables in rules are to begin with an uppercase character, but that the underlying logic is first-order predicate calculus, which means that the objects classified by these rules can have a deeply nested yet flexible structure. Hierarchical structures are required whenever the objects to be classified have more structure than can be expressed by an attribute-value vector. Flexible structures are required, e.g., whenever only part of an object is responsible for its classification, but it is unknown in advance which part (as in the multiple instance problem (Dietterich, Lathrop, & Perez, 1997), or when the objects are sequences (as in bio-informatics or natural language domains)). So, in a more general sense, ILP encompasses the application of machine learning methods to domains with flexible nested structures. It should therefore come as no surprise that one of the papers in this issue is clearly addressing ILP issues, even though it does not present a single line of Prolog.

While its origins can be traced back to Plotkin's work in the early 70s and Shapiro's work in the early 80s, ILP started to claim its place in the world as a separate branch of machine learning when the first ILP workshop was organised in 1991. We think it is appropriate to characterise the 10 years that have elapsed as ILP's adolescence, and we are happy to say that the papers in this issue show that ILP is coming of age. This can be no better illustrated than by pointing out that none of the papers is a 'pure' ILP paper, but that each of them explores relations with other disciplines or research areas, be they neural networks, cost-sensitive classification, computational learning theory, or probabilistic representations. Rather than summing up what has been achieved in 10 years, each of the papers points to promising lines of research, thereby demonstrating the vitality of ILP as a research area.

The paper *On exact learning of unordered tree patterns* by Thomas Amoth, Paul Cull, and Prasad Tadepalli investigates the computational complexity of learning in a setting where both the objects and the rules considered are flexible nested structures and have the form of unordered trees. They consider different ways of matching rules to objects, i.e., mapping tree patterns onto and into instance trees, and show that unordered tree patterns are not exactly learnable from equivalence and subset queries for the one-to-one onto mapping, whereas they are exactly learnable from equivalence and membership queries for the one-to-one into mapping. While not using Prolog to represent objects and rules, this work is clearly related to inductive logic programming as shown by the authors: a class of tree patterns called clausal trees that includes non-recursive single-predicate Horn clauses is shown to be learnable from equivalence and membership queries.

Stochastic logic programs (Muggleton, 1996) extend logic program clauses with labels determining the probability with which the clause is chosen to resolve with a matching query, and establish a promising approach to combining logical and probabilistic knowledge. Learning stochastic logic programs requires learning the clauses and estimating their labels. James Cussens concentrates on the latter problem in his paper *Parameter estimation in stochastic logic programs.* He presents a new algorithm called failure-adjusted maximisation, which is an instance of the EM algorithm providing a closed-form for computing parameter updates within an iterative maximisation approach.

The next paper, *Approximate match of rules using backpropagation neural networks* by Boonserm Kijsirikul, Sukree Sinthupinyo and Kongsak Chongkasemwongse, is concerned with improving the performance of rules learned by inductive logic programming methods by using approximate matching of first-order rules to instances. The approximate matching is implemented by first decomposing the given rules into elementary features, then training a neural network to do the approximate matching. As demonstrated by a variety of experiments, each of the two stages contributes to improving the performance of the original rules.

In the fourth and final paper in this issue, *Extracting context-sensitive models in Inductive Logic Programming*, Ashwin Srinivasan applies ROC analysis (Provost & Fawcett, 1997) to ILP, with the goal to obtain a solution that contains models that are optimal under different misclassification costs, or contexts. The author uses a version of the Parcel feature subset selection algorithm (Scott et al., 1998) to identify parts of the background knowledge that are most relevant to a particular context. This suggests a possible approach to assessing the relevance of background predicates in ILP.

In our view, these papers demonstrate that inductive logic programming is firmly embedded in machine learning, and that we can look forward to more exciting work exploring the connections between ILP and other machine learning approaches.

References

Dietterich, T. G., Lathrop, R. H., & Lozano-Perez, T. (1997). Solving the multiple instance problem with axis-parallel rectangles. *Artificial Intelligence, 89*, 31–71.

Muggleton, S. (1996). 'Stochastic logic programs'. In: L. de Raedt (ed.). *Advances in Inductive Logic Programming*. IOS Press, pp. 254–264.

Provost, F. & Fawcett, T. (1997). Analysis and visualization of classifier performance. Comparison under imprecise class and cost distributions. In *Proceedings of the Third International Conference on Knowledge Discovery and Data Mining* (*KDD-97)* (pp. 43–48). AAAI Press.

Scott, M. J. J., Niranjan, M., & Prager, R. W. (1998). Parcel: feature subset selection in variable cost domains. Technical Report CUED/F-INFENG/TR.323, Cambridge University Engineering Department, Cambridge, UK. Version available at url. http://svr-www.eng.cam.ac.uk/reports/people/niranjan.html.