Recent Progress in Learning Horn Expressions with Queries*

Roni Khardon
Division of Informatics
University of Edinburgh
Edinburgh, EH9 3JZ
Scotland
roni@dcs.ed.ac.uk

Abstract

Work in recent years has shown that if we allow the learner to ask questions then reasonably large subsets of first order Horn clause languages are efficiently learnable. The talk will disucuss theoretical and practical aspects of these results.

Introduction

Early work in Inductive Logic Programming (ILP) included several systems that allowed the learner to ask questions in the learning process [24, 23, 18, 8]. Most of recent work, however, tends to use learning from examples as the main paradigm (e.g. [20, 17, 9, 5]). Since only limited classes of expressions are learnable from examples [6], heuristics are used to obtain good results in practice. One recent strand of theoretical work has shown that larger classes of expressions are learnable if the learner is allowed to ask questions [12, 21, 4, 22, 16, 14, 13, 3, 15]. These works use standard oracles from learning theory [1] as well as new types of questions appropriate for the first order setting. Two main challenges remain in this area. One is to further clarify which classes are learnable and with what complexity. The other is to establish applications of query based algorithms. The talk will discuss both aspects trying to give an insight into algorithmic issues and illustrate possible applications.

Learning with Queries

Angluin's so-called minimally adequate teacher [1] is often used in this context. In this setting the learner is allowed two types of questions. With Equivalence Queries the learner presents a hypothesis and requests either confirmation that it is correct or otherwise a counter example. With Membership Queries the learner can construct an arbitrary description of a hypothetical example and ask whether the example is a member of the concept being learned. Several forms of examples (i.e. atoms, clauses, interpretations) have been used before in ILP [7] and each of these yields a different learning model when combined with the query setting. Some authors use additional query types to allow for more efficient learning. Arimura [4] and Reddy and Tadepalli [22] used "derivation order" queries to identify which atom is the first one to be used when deriving a particular conclusion. Krishna Rao and Sattar [16] use subsumption queries to find out whether hypothesised clauses syntactically match the concept being learned.

Despite the variation in example types and question types the algorithms in [12, 21, 4, 22, 16, 14, 13, 3] share a common structure, which is already reflected in learning propositional expressions [1, 2, 10]. A similar structure is used for learning description-logic expressions in [11]. The algorithms maintain multi-clause hypotheses, and learn all clauses simultaneously. Given a new uncovered example, the algorithm tries to use the example to "refine" one of its current clauses. If this succeeds then a new hypothesis is formed by replacing this clause with its refined version, and leaving other clauses unchanged. Otherwise the new example is used to add a new clause to the hypothesis. Clearly, a careful treatment

*Partly supported by EPSRC grant GR/M21409.

of "successful refinement" is needed. All algorithms use some form of products for refinement, either Plotkin's LGG [19] or direct product of first order structures. Since these operations generate large clauses further size-reducing operations are needed. While equivalence queries are used to find new uncovered examples, membership queries and other oracles are used to identify successful refinements and safe size-reducing operations.

Applications

There are several ways to apply such algorithms in practical situations. The natural approach is to develop interactive systems relying on users to answer questions [24, 8, 15]. These can be useful in tools supporting the development of logic programs. A challenging possibility arises in domains where one can perform experiments e.g. lab tests in chemical domains to answer membership queries. If this is feasible then one might be able to automate the entire learning process. A similar idea can be used when one can simulate such an experiment; Reddy and Tadepalli [21] use this idea in an AI planning problem. To test whether a certain operator is useful, they use it directly in their planner and test the results.

Finally these ideas can also contribute new algorithms for learning from examples only. In recent work we obtained answers to queries simply by evaluating the queries on a given data set [15]. The resulting algorithm is quite different from most current systems in that it performs specific to general search and uses large refinement steps in its search process. These ideas have been tested in several domains including learning list manipulation programs and natural language grammar learning where good performance was demonstrated [15].

References

[1] D. Angluin. Queries and concept learning. *Machine Learning*, 2(4):319–342, 1988.

[2] D. Angluin, M. Frazier, and L. Pitt. Learning conjunctions of Horn clauses. *Machine Learning*, 9:147–164, 1992.

[3] M. Arias and R. Khardon. A new algorithm for learning range restricted Horn expressions. In *International conference on Inductive Logic Programming*. Springer, 2000.

[4] H. Arimura. Learning acyclic first-order Horn sentences from entailment. In *Proceedings of the International Conference on Algorithmic Learning Theory*, Sendai, Japan, 1997. Springer-verlag. LNAI 1316.

[5] H. Blockeel and L. De Raedt. Top down induction of first order logical decision trees. *Artificial Intelligence*, 101:285–297, 1998.

[6] W. Cohen. PAC-learning recursive logic programs: Negative results. *Journal of Artificial Intelligence Research*, 2:541–573, 1995.

[7] L. De Raedt. Logical settings for concept learning. *Artificial Intelligence*, 95(1):187–201, 1997. See also relevant Errata (forthcoming).

[8] L. De Raedt and M. Bruynooghe. An overview of the interactive concept learner and theory revisor CLINT. In S. Muggleton, editor, *Inductive Logic Programming*. Academic Press, 1992.

[9] L. De Raedt and W. Van Laer. Inductive constraint logic. In *Proceedings of the International Conference on Algorithmic Learning Theory*, 1995.

[10] M. Frazier and L. Pitt. Learning from entailment: An application to propositional Horn sentences. In *Proceedings of the International Conference on Machine Learning*, pages 120–127, Amherst, MA, 1993. Morgan Kaufmann.

[11] M. Frazier and L. Pitt. CLASSIC learning. *Machine Learning*, 25:151–193, 1996.

[12] D. Haussler. Learning conjunctive concepts in structural domains. *Machine Learning*, 4(1):7–40, 1989.

[13] R. Khardon. Learning function-free Horn expressions. *Machine Learning*, 37(3):241–275, 1999.

[14] R. Khardon. Learning range-restricted Horn expressions. In *Proceedings of the Fourth European Conference on Computational Learning Theory*, pages 111–125, Nordkirchen, Germany, 1999. Springer-verlag. LNAI 1572.

[15] R. Khardon. Learning Horn expressions with LogAn-H. In *International Conference on Machine Learning*, 2000.

[16] M. Krishna Rao and A. Sattar. Learning from entailment of logic programs with local variables. In *Proceedings of the International Conference on Algorithmic Learning Theory*, Otzenhausen, Germany, 1998. Springer-verlag. LNAI 1501.

[17] S. Muggleton. Inverse entailment and Progol. *New Generation Computing*, 13:245–286, 1995.

[18] S. Muggleton and W. Buntine. Machine invention of first order predicates by inverting resolution. In S. Muggleton, editor, *Inductive Logic Programming*. Academic Press, 1992.

[19] G. D. Plotkin. A note on inductive generalization. In B. Meltzer and D. Michie, editors, *Machine Intelligence 5*, pages 153–163. American Elsevier, 1970.

[20] J. R. Quinlan. Learning logical definitions from relations. *Machine Learning*, 5:239–266, 1990.

[21] C. Reddy and P. Tadepalli. Learning Horn definitions with equivalence and membership queries. In *International Workshop on Inductive Logic Programming*, pages 243–255, Prague, Czech Republic, 1997. Springer. LNAI 1297.

[22] C. Reddy and P. Tadepalli. Learning first order acyclic Horn programs from entailment. In *International Conference on Inductive Logic Programming*, pages 23–37, Madison, WI, 1998. Springer. LNAI 1446.

[23] C. Sammut and R. Banerji. Learning concepts by asking questions. In R. Michalski, J. Carbonell, and T. Mitchell, editors, *Machine Learning : An AI Approach, Volume II.* Morgan Kaufman, 1986.

[24] E. Y. Shapiro. *Algorithmic Program Debugging.* MIT Press, Cambridge, MA, 1983.