Machine Learning, 23, 103–108 (1996)

Review of "Inductive Logic Programming: Techniques and Applications" by Nada Lavrač, Sašo Džeroski

MICHAEL PAZZANI pazzani@ics.uci.edu
University of California, Irvine, CA 92715

Editor: Alberto Segre

1. Introduction

Inductive Logic Programming (ILP) is an important and active subfield of machine learning. Unlike most of machine learning, ILP is concerned with learning first-order (or relational) rules, a representation that is more expressive than the attribute-value representation typically used by decision trees and neural networks. Due to the recent emergence of this field, most of the important work is scattered across a number of conference papers, journal papers, and theses. Most papers use slightly different notation, terminology, and definitions, which results in unnecessary confusion to newcomers to this research area. *Inductive Logic Programming: Techniques and Applications* is the first text that attempts to present an overview of the field, and it performs admirably at this task. While there are other books on ILP, some are edited volumes that do not provide extensive introductory material (e.g., Muggleton, 1992) and others (e.g., De Raedt, 1992; Morik, Wrobel, Kietz, & Emde, 1993) provide detailed reports on a single research project.

Inductive Logic Programming: Techniques and Applications is appropriate as an introductory graduate text. It contains sufficient background material to gently introduce someone to the field, and it provides detailed descriptions of recent research contributions to the field. Since many ILP systems have sound theoretical foundations, the book contains a number of definitions and introduces some new notation. The definitions are illustrated with numerous examples that help to make the concepts concrete. The book does not suffer from a flaw I have seen in some theoretical treatments. There is no formalism for its own sake in this book. The definitions and notations introduced are used later in the book and help to clarify important concepts.

2. Overview

Inductive Logic Programming: Techniques and Applications consists of fourteen chapters organized into four parts: an introduction to ILP, a description of empirical ILP, handling imperfect data in ILP, and applications of ILP.

The first part consists of three chapters that provide an introduction to ILP, describing the goals of the field, some background material and introductory definitions. It is here that the authors introduce two important subfields of ILP:

- **empirical ILP** in which a learner is given a set of examples E represented as ground facts for a predicate p and some background knowledge B and learns a hypothesis H such that H is complete and consistent with respect to E and B.
- **interactive ILP** which is related to empirical ILP except that the learner is given a single example e, a current hypothesis H which may be incorrect, and an oracle that can answer membership queries. In this case, the learner produces a new hypothesis H' by asserting and retracting clauses from H.

Generally, these chapters are clearly written, and amply illustrated. I would have preferred an additional example showing a Q-subsumption lattice to illustrate the difference between the greatest lower bound and the least upper bound of two clauses. The book would be more comprehensive if additional ILP problems, such as batch theory revision with a set of training examples (e.g., FORTE, Richards & Mooney, in press), were incorporated in the unifying framework presented in the first part of the book. Nonetheless, this part of the text provides an important and useful unifying framework for the field of ILP and the remainder of the book.

The book focuses on empirical ILP, and this is the topic of the second part consisting of four chapters. The first chapter of this part provides an overview of important contributions such as FOIL, GOLEM, and MOBAL. I was quite happy with the detailed discussion of FOIL. GOLEM is described a bit more succinctly. Although there are references to extensions of GOLEM (e.g., to handle noise), the text does not elaborate on the methods and hence, misses an opportunity to clarify these extensions and describe them using the same notation.

The next two chapters describe the authors' work on LINUS, a relational learner that operates by first converting a relational problem to an attribute value one, learning with an attribute value learner, and then converting the learned description to relational rules. Due to its use of a propositional learner, LINUS can learn only constrained clauses (i.e., clauses whose bodies only contain variables used in the head of clause) and clauses that contain determinate literals. A series of experiments is reported comparing the accuracy and time of LINUS and FOIL. Although for the most part the experiments are well done, the comparison of execution times provides little useful information, since these systems are implemented in different computer languages and run on different computers. The final chapter of this part describes and compares the search space explored by LINUS and FOIL and gives complexity results of the two algorithms. Occasionally, I questioned the utility of devoting so much attention to LINUS in a text on ILP, since it appears that LINUS pushes attribute-value learners as far as they can go toward solving relational problems. However, these chapters may be useful to readers who want detailed information on decision tree and propositional rule learning algorithms. In addition, they serve to illustrate the trade-off between the expressiveness of the learned language and the complexity of the learning process.

The third part contains three chapters that deal with handling imperfect data in ILP. They present mFOIL (the authors' extension to FOIL for handling noise) and report on a number of experiments comparing various alternatives for handling noise in ILP systems. These chapters also review noise handling in attribute-value learners (used directly by LINUS) and show how these methods (e.g., statistical and encoding length) can be adapted to ILP systems. A series of experiments are reported comparing mFOIL, FOIL, and LINUS by introducing noise in a noise-free data set. In these experiments, mFOIL tends to have higher accuracy than other methods. My only criticism of this section is that I wish experiments were run with other systems (e.g., GOLEM), other methods (e.g., Reduced Error Pruning), and other data sets.

The last part of the book contains four chapters on applications of ILP. Here applications are described in detail. The first chapter discusses an application of LINUS (using CN2 as an attribute-value learner) on learning rules for diagnosis of rheumatic diseases. Although interesting, this experiment really doesn't seem to require ILP techniques, since the only use of background knowledge is to create 5 new attributes. The second chapter discusses an application to finite-element mesh design using LINUS, GOLEM, and FOIL. Although this problem clearly requires learning relational rules, the accuracy of the learned rules are too low to be useful to practitioners, probably due to the small number of examples. The third chapter discusses the ability of GOLEM and mFOIL to identify the correct qualitative model of dynamic systems. The last chapter summarizes other ILP applications, including the use of GOLEM to learn rules that predict the secondary structure of proteins and the biological activity of chemical compounds.

3. Critique

Although I do highly recommend the book, it contains several shortcomings in that it failed to cover several important aspects of relational learning. First, little attention is given to learning recursive rules (e.g., Aha, Ling, Matwin & Lapointe, 1993), which is a problem that has received a great deal of attention. Second, although theoretically oriented, this book does not go into any detail on the findings of computational learnability (e.g., Cohen, 1994). In inductive logic programming the trade-off between the expressiveness of the representation language and the learnability of concepts is an important recurring theme. Third, predicate invention, an area that is still in its infancy, is an area that would have benefited by being covered within the framework of the text. The few papers on this topic (e.g., Kijsirikul, Numao & Shimuera, 1992; Muggleton & Buntine,1988) use different notations and require some effort to read. If predicate invention methods were summarized in the text, more researchers would be able to incorporate them in future systems, understand their limitations, and extend them. Similarly, I wish more detail were included on additional ILP systems, such as CLINT and ML-SMART (Bergadano & Giordana, 1988) that have made important contributions. Early relational learning systems are virtually ignored.

Another area not covered in depth in this text is multi-class learning. Most ILP systems are designed for binary classification problems. This occasionally leads to simplifying some problems. For example, predicting the secondary structure of proteins is treated as a two-class problem (α-helix and others), while many systems treat this as a three class problem (α-helix, β-coil, and other). The authors imply that a problem with n classes may be treated as a multiple predicate problem (i.e., n binary problems). However, since classes are typically mutually exclusive and exhaustive, the situations when more than one class is predicted and when no class is predicted by the learned rules must be addressed (Ali & Pazzani, 1993). Methods to address this issue are not discussed in the text.

A final issue that is not addressed in the text is the role of background knowledge in learning. I'm paying more attention to this omission than others, since it is an area that I'm most familiar with. In this text, two different types of background knowledge tend to be used:

- Additional relations that elaborate on the example. For example, to learn the predicate $aunt(X)$, a learner is given positive examples such as "mary" and negative examples such as "john". The learner can use the background knowledge $female(mary)$ to assist in learning the concept definition. For the most part, background relations of this sort correspond to the attributes used by attribute-value learners. Of course, much of the power of ILP comes from the fact that examples might be distinguished not because their attributes differ, but because the attributes of objects associated with the examples differ. For example, in diagnosing some medical disorders that are hereditary, it is common for a doctor to want to know the medical history of people related to the patient. At a recent visit to the orthodontist to determine whether my daughter needed braces, I was surprised when the orthodontist scrutinized my teeth.
- Causal or diagnostic knowledge that might be the knowledge-base for a rule-based expert system.

The authors make no attempt to distinguish the different types of knowledge, and this leads to some confusion over whether ILP systems are "knowledge-intensive" learners. In my view, if only descriptive background knowledge is present, an ILP system isn't "knowledge-intensive," while if causal knowledge is present, it is. The failure to distinguish these types of background knowledge may be an advantage if, for example, the same technique that handles noise in the example descriptions could handle incorrect causal knowledge or if methods for dealing with missing attribute values were applicable to incomplete background knowledge. However, different methods are applied to each of these problems, and it is important for an ILP system that wishes to address all of these issues to distinguish between the different types of background knowledge. Mooney and Zelle (1994) provide an overview of methods for dealing with incorrect and incomplete background knowledge.

The issue of irrelevant background knowledge is not addressed. It is well known that irrelevant attributes can reduce the accuracy of inductive attribute-value learning systems, so one would expect that irrelevant descriptive background knowledge would reduce the accuracy of ILP systems. Knowledge-intensive learning methods, such as explanation-based learning, were intended to reduce the sensitivity of learners to irrelevant attributes. However, as used in most ILP systems, correct, but irrelevant, causal background knowledge may decrease accuracy. This occurs because background knowledge is used in a bottom-up fashion, to derive additional information concerning each example, and irrelevant background knowledge just adds irrelevant information to each example. In contrast, systems such as ML-SMART, FOCL (Pazzani & Kibler, 1992), and GRENDEL (Cohen, 1991) can use causal background knowledge in a top-down manner, insuring that the additional derived information is relevant. These systems can distinguish relevant from irrelevant causal knowledge, because, like EBL systems, they are given an addition input: a target concept. A target concept is an abstract (non-operational) definition of the concept to be learned. The causal background knowledge indicates how instances of the target concept might be identified by examining the descriptive information of the examples. Since the causal background knowledge may be incorrect or incomplete, these knowledge-intensive learners also include an inductive learning component.

The text discusses the promise of ILP systems for a variety of applications, but I was somewhat disappointed with many of the results presented in the application section. For example, although in theory it would be possible to take a large relational database (e.g., an employee records database) and learn predicates from this database (e.g., the profile of employees who are likely to leave the company), the text does not report on any attempts to address such problems with ILP systems. The unifying framework and clear introduction to the field presented in the text could inspire some to tackle such problems. However, the book missed an opportunity to encourage such experimentation by not including information on how a reader might get a copy of some of the major algorithms or the same databases used in the application section. If such information were available, readers could easily try existing algorithms on new problems and compare the results of new algorithms on existing databases.

The most glaring omission of the text is the lack of a conclusion chapter. The text does a good job of describing a variety of different algorithms, and evaluates some of these algorithms experimentally, but the authors do not make recommendations about which algorithm is appropriate for which class of problems.

4. Conclusion

Inductive Logic Programming: Techniques and Applications is a clear introduction to the field of ILP. It would serve as an ideal text for an introductory course on this topic. Its most important asset is its description of several important ILP systems together with the background theory needed to understand how these systems operate. Its most significant shortcoming is that it is not as comprehensive as one might hope, and would therefore require supplementation with a few journal and conference papers, such as those mentioned in this review.

References

Aha, D., Ling, C., Matwin, S., and Lapointe, S. (1993). Learning singly recursive relations from small datasets. In *Proceedings of the IJCAI-93 Workshop on Inductive Logic Programming*. Available as AIC Report 93-017 (`http://www.aic.nrl.navy.mil/papers/1993/AIC-93-017.ps.Z`). Navy Center for Artificial Intelligence Research.

Ali, K. & Pazzani, M. (1993). HYDRA: A noise-tolerant relational concept learning algorithm. *Proceedings of the International Joint Conference on Artificial Intelligence* (pp. 1064–1071) Chambery, France.

Bergadano, F., & Giordana, A. (1988). A knowledge intensive approach to concept induction. *Proceedings of the Fifth International Conference on Machine Learning* (pp. 305–317). Ann Arbor, MI: Morgan Kaufmann.

Cohen, W. (1991). Grammatically biased learning: Learning Horn Theories using an explicit antecedent description language. AT&T Bell Laboratories Technical Report 11262-910708-16TM.

Cohen, W. (1994) Grammatically biased learning: Learning Horn theories using an explicit antecedent description language. *Artificial Intelligence, 68*, 303–366.

De Raedt, L (1992). *Interactive Theory Revision: An Inductive Logic Programming Approach.* London: Academic Press.

Kijsirikul, B., Numao, M., & Shimuera, M. (1992). Discrimination-based constructive induction of logic programs. *Proceedings of the National Conference on Artificial Intelligence* (pp. 44–49). San Jose, CA: AAAI Press.

Morik, K., Wrobel, S., Kietz, J. & Emde, W. (1993) *Knowledge Acquisition and Machine Learning.* London: Academic Press.

Mooney & Zelle, R. (1994). Integrating ILP and EBL. *SIGART Bulletin, 5,* 12–21.

Muggleton, S. (1992). *Inductive Logic Programming.* Academic Press: London.

Muggleton, S. & Buntine, W. (1988). Machine invention of first-order predicates by inverting resolution. *Proceedings of the Fifth International Conference on Machine Learning* (pp. 291–298). Ann Arbor, MI: Morgan Kaufmann.

Pazzani, M., & Kibler, D. (1992). The utility of knowledge in inductive learning. *Machine Learning, 9,* 57–94.

Richards, B. & Mooney, R. (1995). Refinement of first-order Horn-clause domain theories. *Machine Learning, 19,* 95–132.