Inductive Logic Programming

Techniques and Applications

Nada Lavrač and Sašo Džeroski

Jožef Stefan Institute
Ljubljana, Slovenia

This book was published by Ellis Horwood, New York in 1994.

Please reference this book as follows:

N. Lavrač and S. Džeroski.
Inductive Logic Programming: Techniques and Applications.
Ellis Horwood, New York, 1994.

Contents

Foreword

This book is about inductive logic programming (ILP), which is a new technology combining principles of inductive machine learning with the representation of logic programming. The new technology aims at inducing general rules starting from specific observations and background knowledge.

Let me explain why inductive logic programming is currently regarded as (one of) the most important recent development(s) in machine learning research. In my view, ILP $=$ I $\cap$ LP, i.e., ILP is the intersection of techniques and interests in induction and logic programming. This makes inductive logic programming more powerful than traditional techniques that learn from examples, namely, inductive logic programming uses an expressive first-order logic framework instead of the traditional attribute-value framework, and facilitates the use of background knowledge. The first point is important because many domains of expertise cannot be formulated in an attribute-value framework. The second point is also significant because background knowledge is the key to success in nearly all applications of artificial intelligence. At the same time, inductive logic programming has room for both theory and practice. Inductive logic programming has a strong theoretical basis as it inherited many results from logic programming and computational learning theory, and adapted (and continues to adapt) these to fulfill the needs of a new discipline. Inductive logic programming has also very impressive applications in scientific discovery, knowledge synthesis and logic programming. To mention an important achievement: Stephen Muggleton has recently produced — using general purpose inductive logic programming systems — new scientific knowledge in drug design and protein engineering, which is now published in scientific journals [King et al. 1992, Muggleton et al. 1992a].

The authors of this book, Nada Lavrač and Sašo Džeroski, are members of the AI Laboratory of the Jožef Stefan Institute (Ljubljana, Slovenia) directed by Ivan Bratko. The laboratory and its members have an outstanding reputation in applying both machine learning and logic programming to real-life knowledge engineering problems. Nada Lavrač has already co-authored books in both areas: *Prolog through Examples: A Practical Programming Guide* (by Igor Kononenko and Nada Lavrač, Sigma Press, 1988) and *KARDIO: A Study in Deep and Qualitative Knowledge for Expert Systems* (by Ivan Bratko, Igor Mozetič and Nada Lavrač, The MIT Press, 1989). It is therefore no surprise that Nada Lavrač, now with Sašo Džeroski, combines her two interests in this book.

The book is written from an engineering perspective with applications in mind. Indeed, after an introduction to and a survey of inductive logic programming in Part I, the authors focus on the empirical setting for inductive logic programming in Part II. The empirical setting is currently the one best suited for applications as it aims at synthesizing knowledge from potentially large sets of observations. In this second part, the authors present the first important contribution: the idea that inductive logic programming can benefit from the well-understood traditional induction techniques such as top-down induction of decision trees. This idea forms the basis of the system LINUS (and its extension DINUS) which inherits its efficiency from the traditional learning algorithms. The efficiency of LINUS is very relevant, not only to the users, but also to computational learning theory. Indeed, it was recently discovered that some of the assumptions of LINUS/DINUS are closely related to the class of polynomially learnable predicates in computation learning theory (PAC-learnability). Another significant contribution of LINUS is that it is the first inductive logic programming system able to handle numerical data. In Part III, the authors tackle another key problem when doing real-life applications: imperfect data. To this aim, they investigate and evaluate new heuristics for inductive logic programming, and show in this way that inductive logic programming is ready to handle imperfect data and therefore also real-life applications. Real-life applications are the subject of Part IV, which contains some impressive results in medicine (the award-winning application in rheumatology at the Third Scandinavian Conference on Artificial Intelligence, SCAI-91), in engineering (finite element mesh design), and qualitative modeling.

Besides giving a good introduction to the field of inductive logic programming, this book also shows what (knowledge) engineers can achieve with this new technology. Furthermore, it is easily readable and accessible. Therefore, I believe it will be of interest to a wide audience, including graduates, researchers and practitioners of artificial intelligence and computer science; in particular, those interested in machine learning, knowledge engineering, knowledge discovery in databases and logic programming.

Sint-Amandsberg, April 1993 *Luc De Raedt*

Preface

Inductive logic programming (ILP) is a research area at the intersection of machine learning and logic programming. It aims at a formal framework as well as practical algorithms for inductive learning of relational descriptions in the form of logic programs. From logic programming, ILP has inherited its sound theoretical basis, and from machine learning, an experimental approach and orientation towards practical applications. ILP has already shown its application potential in the following areas: knowledge acquisition, inductive program synthesis, inductive data engineering, and knowledge discovery in databases.

This book is an introduction to this exciting new field of research. It should be of interest to the following readership: knowledge engineers concerned with the automatic synthesis of knowledge bases for expert systems; software engineers who could profit from inductive programming tools; researchers in system development and database methodology, interested in techniques for knowledge discovery in databases and inductive data engineering; and researchers and graduate students in artificial intelligence, machine learning, logic programming, software engineering and database methodology.

As ILP is a relatively young field, the ILP literature consists mainly of conference and workshop proceedings. A book collection of papers [Muggleton 1992a], covering the whole field of ILP, is also available, as well as a book on theory revision in the context of interactive ILP [De Raedt 1992]. Interactive ILP, one of the two major subfields of ILP, is closely related to program debugging and theory revision where a small number of examples is available. The second major subfield of ILP, called empirical ILP, places an emphasis on the extraction of regularities from a large number of possibly erroneous examples. The spirit of empirical ILP is much closer to the existing successful learning systems, and is therefore more suitable for practical applications than interactive ILP. This book extensively covers empirical ILP techniques and applications.

The book is divided into four parts. Part I is an introduction to the field of ILP. Part II describes in detail empirical ILP techniques and systems. Part III is concerned with the problem of handling imperfect data in ILP, whereas Part IV gives an overview of empirical ILP applications.

Part I comprises Chapters 1 to 3. Chapter 1 touches upon the topics covered by the book. Chapter 2 introduces the basic concepts and terminology of logic programming, deductive databases and inductive logic programming. The basic generalization and specialization techniques used in ILP are described in Chapter 3.

Chapters 4 to 7 constitute Part II. Chapter 4 gives an overview of several empirical ILP systems with an emphasis on FOIL [Quinlan 1990], which has been the basis for much of our work. The central chapter of this part, Chapter 5, gives a detailed description of the ILP system LINUS. This includes a description of the LINUS environment, the propositional learning systems incorporated into LINUS, the transformation algorithm, a specification of the hypothesis language and a complexity analysis. It is also shown how LINUS can be extended to generate hypotheses in a more expressive hypothesis language. Chapter 6 describes several experiments in learning relations with LINUS. In Chapter 7, the framework of refinement graphs is used to compare the hypothesis languages and the search complexity of LINUS and FOIL.

Part III starts with Chapter 8, which first describes the problem of handling imperfect data in attribute-value and ILP systems and then gives an overview of techniques and heuristics used in handling imperfect data. Chapter 9 presents the ILP system mFOIL, which incorporates techniques for handling imperfect data from attribute-value systems into the FOIL framework. Its language bias (search space) and search bias (heuristics, search strategy and stopping criteria) are described in detail. An experimental comparison of LINUS, FOIL and mFOIL in a chess endgame domain with artificially introduced errors (noise) in the examples is made in Chapter 10.

Although experiments in artificial domains under controlled amounts of noise reveal important performance aspects of ILP systems, the ultimate test is their application to real-life problems. Part IV gives a detailed description of the applications of LINUS and mFOIL to three different practical learning problems. Comparisons are also made with FOIL and GOLEM [Muggleton and Feng 1990]. Chapter 11 describes the application of LINUS to the problem of learning rules for medical diagnosis, where examples and specialist background knowledge were provided by a medical expert. Chapter 12 compares the performance of mFOIL, FOIL and GOLEM on the task of inducing rules for determining the appropriate resolution of a finite element mesh. Chapter 13 describes the induction of qualitative models from example behaviors and describes the application of mFOIL to this problem, making also a comparison to other systems. Finally, Chapter 14 concludes with an overview of several applications of GOLEM to illustrate the potential of ILP for practical applications.

Acknowledgements

The book is a result of several years of research. Most of it was done at the Artificial Intelligence Laboratory of the Computer Science Department, Jožef Stefan Institute in Ljubljana. We would like to thank Ivan Bratko, the head of the Artificial Intelligence Laboratory, for guiding our research interests towards challenging research topics in AI. Our thanks goes also to Marjan Špegel, the head of the Computer Science Department, for his support during all these years of research.

Our research was based on the tradition of the Artificial Intelligence Laboratory in the areas of machine learning and qualitative modeling. The KARDIO project [Bratko et al. 1989] has been a motivation and a source of ideas for our work. The special purpose learning system QuMAS [Mozetič 1987], which was used in KARDIO for reconstructing a qualitative model of the heart, was based on the idea of transforming a relation learning problem into propositional form. This idea has been further developed into our general ILP environment called LINUS [Lavrač et al. 1991a]. This approach has provided us with a possibility to use a variety of learning techniques, including mechanisms for handling imperfect data, developed in propositional learning systems. Besides using the noise-handling techniques within the LINUS framework, we have also adapted them for direct use in our ILP learner mFOIL [Džeroski and Bratko 1992a], an extension of the FOIL [Quinlan 1990] system. LINUS and mFOIL transcend the approaches of QuMAS and

FOIL mainly by the use of more sophisticated mechanisms for handling imperfect data, which is one of the main topics of this book. We are grateful to Igor Mozetič for his support in the development of LINUS, and to Bojan Cestnik who contributed his expertise and advice in the selection of the noise-handling mechanisms described in this book.

We wish to thank Peter Flach, Igor Kononenko, Stephen Muggleton, Tanja Urbančič and, in particular, Luc De Raedt who contributed significantly to this book by providing helpful comments and advice. We would further like to thank Marko Grobelnik and Dunja Mladenič, who contributed to the development of LINUS and a VAX/VMS version of ASSISTANT incorporated into LINUS, respectively; Bruno Stiglic and Robert Trappl for their support and interest in this work; Ross Quinlan for making his FOIL system available for experimental comparisons; Michael Bain, Ivan Bratko, Bojan Dolšak, Vladimir Pirnat and Ross Quinlan for making their experimental data available; Bojan Orel for his TEXpertise used in the preparation of this book; and our colleagues in the Artificial Intelligence Laboratory for their contributions to the views expressed in this book.

Our research has been continuously supported by the Slovenian Ministry of Science and Technology. The research was also part of the ESPRIT II Basic Research Action no. 3059 ECOLES and the ESPRIT III Basic Research Project no. 6020 Inductive Logic Programming. The Slovenian Ministry of Science and Technology supported Nada Lavrač during her three-month visit to the University of Illinois at Urbana-Champaign, Urbana, Illinois in 1985, and to the George Mason University, Fairfax, Virginia in 1988. The Belgian State-Science Policy Office grant enabled her a six-month stay at the Katholieke Universiteit Leuven, Belgium in 1991–1992. The Slovenian Ministry of Science and Technology and the British Council supported Sašo Džeroski during his eight-month stay at the Turing Institute, Glasgow, UK, also in 1991–1992. We are grateful to Ryszard Michalski (George Mason University), to Maurice Bruynooghe and Luc De Raedt (Katholieke Universiteit Leuven), and to Stephen Muggleton (The Turing Institute) for their support of our work in the stimulating research environments of their institutions.

The inductive logic programming community, gathered in the ESPRIT III project Inductive Logic Programming, provided a stim-

ulating environment for the exchange of ideas and results in the field; this book is a contribution to the body of knowledge developed within this community. We are grateful to Stephen Muggleton and Luc De Raedt for their contributions to ILP research and for their organizational efforts which have made this research group so lively.

Copyright acknowledgements

The following material was taken from the book *Inductive Logic Programming*, edited by Stephen Muggleton, with the kind permission of Academic Press: Figure 5.1, Figure 6.6, Sections 7.4, 7.5, 7.6 and Figure 14.9.

A major part of Chapter 11 appears in *Applied Artificial Intelligence* 7: 273–293, 1993, under the title 'The utility of background knowledge in learning medical diagnostic rules'. It is republished with the permission of the publisher Taylor & Francis.

Thanks to Bojan Dolšak for Figure 12.1, Dunja Mladenić and Aram Karalič for Figure 14.3, and Ashwin Srinivasan for Figure 14.1.

Ljubljana, March 1993 *Nada Lavrač* and *Sašo Džeroski*