Computational Logic and Machine Learning: A roadmap for Inductive Logic Programming

DRAFT, April 1999

Nada Lavrač
J. Stefan Institute
Jamova 39, 1000 Ljubljana, Slovenia
nada.lavrac@ijs.si

Abstract

Computational logic has already significantly influenced (symbolic) machine learning through the field of inductive logic programming (ILP) which is concerned with the induction of logic programs from examples and background knowledge. In ILP, the shift of attention from program synthesis to knowledge discovery resulted in advanced techniques that are practically applicable for discovering knowledge in relational databases. Machine learning, and ILP in particular, has the potential to influence computational logic by providing an application area full of industrially significant problems, thus providing a challenge for other techniques in computational logic. This paper gives a brief introduction to ILP, presents state-of-the-art ILP techniques for relational knowledge discovery as well as some research and organizational directions for further developments in this area.

1 Introduction

Inductive logic programming (ILP) [45, 49, 39] is a research area that has its backgrounds in inductive machine learning and logic programming. ILP research aims at a formal framework as well as practical algorithms for inductive learning of relational descriptions that typically have 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 research has been strongly influenced also by Computational learning theory, and recently, by Knowledge Discovery in Databases (KDD) [23] which led to the development of new techniques for relational data mining.

In general, an ILP learner is given an initial theory B (background knowledge) and some evidence E (examples), and its aim is to induce a theory H (hypothesis) that together with B *explains* some properties of E. In most cases the hypothesis H has to satisfy certain restrictions, which we shall refer to as the *bias*. Bias includes prior expectations and assumptions, and can therefore be considered as the logically unjustified part of the background knowledge. Bias is needed to reduce the number of candidate hypotheses. It consists of the language bias L, determining the hypothesis space, and the search bias which restricts the search of the space of possible hypotheses.

The background knowledge used to construct hypotheses is a distinctive feature of ILP. It is well known that relevant background knowledge may substantially improve the results of learning in terms of accuracy, efficiency and the explanatory potential of the induced knowledge. On the other hand, irrelevant background knowledge will have just the opposite effect. Consequently, much of the art of inductive logic programming lies in the appropriate selection and formulation of background knowledge to be used by the selected ILP learner.

This paper first gives a brief introduction to ILP and presents a selection of recently developed ILP techniques for relational knowledge discovery. The overview is restricted to techniques satisfying the strong criterion formulated for machine learning by Michie [41] that requires explicit symbolic form of induced descriptions. The paper proceeds with future research directions for ILP and concludes with some suggestions regarding the organization of the research in this area and its commercialization as a methodology of data mining for large relational databases.

2 State-of-the-art in ILP

This section presents the state-of-the art in ILP. If briefly introduces two basic theoretical settings, gives pointers to successful ILP applications and presents recent technological developments in the area, categorized into the two main theoretical settings.

2.1 ILP problem specification

An inductive logic programming task can be formally defined as follows:

Given:
- a set of examples E
- a background theory B
- a language bias L that defines the clauses allowed in hypotheses
- a notion of *explanation*

Find: a hypothesis $H \subset L$ which explains the examples E with respect to the theory B.

This definition needs to be instantiated for different types of ILP tasks [49]. The instantiation will concern the representation of training examples, the choice of a hypothesis language and an appropriate notion of explanation.

By explanation we here refer to an acceptance criterion of hypotheses: the hypothesis explains the data if it satisfies a certain user-defined criterion w.r.t. the data. We will discuss some formal acceptance criteria used in different ILP settings, but we also need to bear in mind that ILP aims at the induction of hypotheses that are expressed in an explicit symbolic form, that can be easily interpreted by the user/expert and may contribute to the better understanding of the problem addressed, ideally forming a piece of new knowledge discovered from the data.

2.2 ILP settings

The state-of-the-art ILP settings are overviewed below. For the underlying theory see [49, 50]. For a practical introduction to ILP see [39]. Transparencies of the Summer School in ILP and KDD, held in Prague in September 1997, giving an introduction to ILP can be found at:
`http://www-ai.ijs.si/~ilpnet/ilpkdd`.

2.2.1 Predictive ILP

Predictive ILP is the most common ILP setting, ofter referred to as *normal ILP*, *explanatory induction*, *discriminatory induction*, or *strong ILP*. Predictive ILP is aimed at learning of classification and prediction rules. This ILP setting typically restricts E to ground facts, and H and B to sets of definite clauses. The strict notion of explanation in this setting usually denotes coverage and requires global completeness and consistency.

Global completeness and consistency implicitly assume the notion of *intensional coverage* defined as follows. Given background theory B, hypothesis H and example set E, an example $e \in E$ is (intensionally) covered by H if $B \cup H \models e$. Hypothesis H is (globally) complete if $\forall e \in E^+ : B \cup H \models e$. Hypothesis H is (globally) consistent if $\forall e \in E^- : B \cup H \not\models e$.

Given the restriction to definite theories $T = H \cup B$, for which there exists a unique least Herbrand model $M(T)$, and to ground atoms as examples, this is equivalent to requiring that all examples in E^+ are true in $M(B \cup H)$ [49].

By relaxing the notion of explanation to allow incomplete and inconsistent theories that satisfy some other acceptance criteria (predictive accuracy, significance, compression), the predictive ILP setting can be extended to include learning of classification and prediction rules from imperfect data, as well as *learning of logical decision trees* [2]. In a broader sense, predictive ILP incorporates also *first-order regression* [33] and *constraint inductive logic programming* [53] for which again different acceptance criteria apply.

2.2.2 Descriptive ILP

Descriptive ILP is sometimes referred to as *confirmatory induction*, *non-monotonic ILP*, *description learning*, or *weak ILP*. Descriptive ILP is usually aimed at learning of clausal theories [13]. This ILP setting typically restricts

B to a set of definite clauses, H to a set of (general) clauses, and E to positive examples. The strict notion of explanation used in this setting requires that all clauses c in H are true in some preferred model of $T = B \cup E$, where the preferred model of T may be, for instance, the least Herbrand model $M(T)$. (One may also require the completeness and minimality of H, where completeness means that a maximally general hypothesis H is found, and minimality means that the hypothesis does not contain redundant clauses.)

By relaxing the strict notion of explanation used in clausal discovery [13] to allow for theories that satisfy some other acceptance criteria (similarity, associativity, interestingness), descriptive ILP can be extended to incorporate *learning of association rules* [8], *first-order clustering* [12, 21, 36], *database restructuring* [24, 54] *subgroup discovery* [58], *learning qualitative models* [31] and *equation discovery* [19].

2.2.3 An example illustrating predictive and descriptive ILP

Consider a problem of learning family relations where the predictive knowledge discovery task is to define the target relation `daughter(X,Y)`, which states that person `X` is a daughter of person `Y`, in terms of relations defined in background knowledge B. Let the training set E consist of positive and negative examples for the target predicate `daughter/2`. A positive example $e \in E^+$ provides information known to be true and should be entailed by the induced hypothesis. A negative example $e \in E^-$ provides information that is known not to be true and should not be entailed.

$$
\begin{aligned}
E^+ &= \{\texttt{daughter(mary,ann)}, \texttt{daughter(eve,tom)}\} \\
E^- &= \{\texttt{daughter(tom,ann)}, \texttt{daughter(eve,ann)}\} \\
B &= \{\texttt{mother(ann,mary)}, \texttt{mother(ann,tom)}, \texttt{father(tom,eve)}, \\
&\quad \texttt{father(tom,ian)}, \texttt{female(ann)}, \texttt{female(mary)}, \texttt{female(eve)}, \\
&\quad \texttt{parent(X,Y)} \leftarrow \texttt{mother(X,Y)}, \texttt{parent(X,Y)} \leftarrow \texttt{father(X,Y)}, \\
&\quad \texttt{male(pat)}, \texttt{male(tom)}\}
\end{aligned}
$$

If the hypothesis language L contains all definite clauses using the predicate and functor symbols appearing in the examples and background knowledge, a predictive ILP system can induce the following clause from E^+, E^- and B:

```
daughter(X,Y) ← female(X), parent(Y,X).
```

Alternatively, a learner could have induced a set of clauses:

```
daughter(X,Y) ← female(X), mother(Y,X).
daughter(X,Y) ← female(X), father(Y,X).
```

In descriptive knowledge discovery, given E^+ and B only, an induced theory could contain the following clauses:

```
← daughter(X,Y), mother(X,Y).
female(X) ← daughter(X,Y).
mother(X,Y); father(X,Y) ← parent(X,Y).
```

One can see that in the predictive knowledge discovery setting classification rules are generated, whereas in the descriptive setting database regularities are derived.

2.2.4 Other ILP settings

There has been a suggestion [16] of how to integrate the two main settings of predictive and descriptive ILP. In this integrated framework the learned theory is a combination of (predictive) rules and (descriptive) integrity constraints that restrict the consequences of these rules.

Other ILP settings have also been investigated, the most important being *relational instance-based learning* [22]. Excellent predictive results have been achieved by the relational instance-based learner RIBL [22] in numerous classification and prediction tasks. Recently, *first-order reinforcement learning* [20] and first-order Bayesian classifier [27] have also been studied. Since these ILP settings do not involve hypothesis formation in explicit symbolic form, the developed techniques do not qualify as techniques for relational knowledge discovery.

2.3 ILP applications

Most of the current real-world ILP applications involve predictive knowledge discovery, in particular the induction of classification and prediction rules from a given database of examples and the available background knowledge. Successful ILP applications include drug design [35], protein secondary structure prediction [47], mutagenicity prediction [55], carcinogenesis prediction [56], medical diagnosis [43], discovery of qualitative models in medicine [31], finite-element mesh design [17], telecommunications [54], natural language processing [44], recovering software specifications [7], and many others.

There are many overview papers on ILP applications, including an early overview [3]. A tutorial on ILP applications is available at
`http://www-ai.ijs.si/~ilpnet/ilpkdd/`.
A report providing short descriptions of applications developed as part of the ESPRIT IV project ILP2 is available at
`http://www-ai.ijs.si/~ilpnet/ilpkdd/appsdel/`.
ILP applications bibliography is available at:
`http://www-ai.ijs.si/~ilpnet/ilpkdd/ilpab.ps`.

A sample ILP application on river water quality evaluation is outlined, illustrating the predictive and descriptive ILP settings.

Given a list of biological indicators present at a river water sampling site and their abundance levels (300 samples), the predictive knowledge discovery task is to classify the sample into one of the five classes $B1a$, $B1b$, $B2$, $B3$ and $B4$ [18]. Knowledge about biological indicators (families of macro-benthis invertebrates) consisted of eighty predicates of the form `family(X,A)`, each denoting that `family` is present in sample `X` with aboundance level `A` (e.g., `tipulidae(X,A)`, `asellidae(X,A)`, etc.) and the background knowledge about the relations between abundance levels was applied.

The ILP system Golem [48] induced three rules for class $B1a$, fourteen for $B1b$, sixteen for $B2$, two for $B3$ and none for $B4$. According to river ecologist expert evaluation, many of the rules were interesting and agreed with the expert knowledge. Here are two examples of such rules.

```
b1a(X) ← leuctridae(X,A).
b1b(X) ← ancylidae(X,A), gammaridae(X,B), hydropsychidae(X,C),
   rhyacophilidae(X,D), greater_than(B,A), greater_than(B,D).
```

The first rule, covering forty-three positive and four negative examples, states that a sample belongs to the best water quality class $B1a$ if Leuctridae is present. Indeed, according to the expert, Leuctridae is an indicator of good water quality. The second rule states that Gammaridae in abundance is a good indicator of class $B1b$, along with the other families present.

The same data was used for descriptive knowledge discovery by an ILP system Claudien [13]. Claudien checks all possible clauses, within a given language bias, for consistency w.r.t. the given examples. For instance, the first clause below specifies more than one quality class for which the family Perlodidae is indicative. If, in addition, Rhyacophilidae are also present, then this indicates best water quality class $B1a$, as shown in the second rule.

```
b1a(X); b1b(X) ← perlodidae(X,A).
b1a(X) ← perlodidae(X,A), rhyacophilidae(X,A).
```

The river quality experts particularly liked the clauses induced by Claudien due to the coverage of more than one class by a single clause; this is important since the classification results were achieved by a discretization of the originally continuous clasification problem.

2.4 ILP techniques

This section reviews the state-of-the-art ILP techniques most of which have already shown their potential for use in real-life applications. The overview is limited to recent ILP developments, aimed at data mining from real-life databases [40]. These developments have a marketing potential in the prosperous new areas of Data Mining and Knowledge Discovery in Databases. Further information on some of the systems can be found at http://www-ai.ijs.si/~ilpnet/, and more details on some of the systems recently developed as part of the ESPRIT IV program on ILP2 at: http://www-ai.ijs.si/~ilpnet/ilpkdd/sysdel/.

It is worthwhile noticing that none of the reviewed techniques belongs to programming assistants which have a much smaller marketing potential and a limited usefulness for solving real-life problems in comparison with ILP data mining tools and techniques.

2.4.1 Predictive ILP

Learning of classification rules. This is the standard ILP setting that has been used in numerous successful predictive knowledge discovery applications. The well-known systems for classification rule induction include Foil [51][1], Golem [48] and Progol [46]. Foil is efficient and best understood due to its similarity to Clark and Niblett's CN2. On the other hand, Golem and Progol are champions concerning successful ILP applications, despite the fact that they are substantially less efficient. Foil is a top-down learner, Golem is a bottom-up learner, and Progol uses a combined search strategy. All are mainly concerned with single predicate learning from positive and negative examples and background knowledge; in addition, Progol can also be used to learn from positive examples only. They use different acceptance criteria: compression, coverage/accuracy and minimal description length, respectively.

Induction of logical decision trees. The system Tilde [2] belongs to Top-down induction of decision tree algorithms. It can be viewed as a first-order upgrade of Quinlan's C4.5, employing logical queries in tree nodes which involves appropriate handling of variables. The main advantage of Tilde is its efficiency and capability of dealing with large numbers of training examples, which are the well-known properties of Tilde's propositional ancestors. Hence Tilde currently represents one of the most appropriate systems for predictive knowledge discovery. Besides the language bias, Tilde allows for lookahead and prepruning (according to the minimal number of examples covered) defined by parameter setting.

First-order regression. The relational regression task can be defined as follows: Given training examples as positive ground facts for the target predicate $r(Y, X_1, ..., X_n)$, where the variable Y has real values, and background knowledge predicate definitions, find a definition for $r(Y, X_1, ..., X_n)$, such that each clause has a literal binding Y (assuming that $X_1, ..., X_n$ are bound). Typical background knowledge predicates include less-or-equal tests, addition, subtraction and multiplication. An approach to relational regression is implemented in the system FORS (First Order Regression System) [33] which performs top-down search of a refinement graph. In each clause, FORS can predict a value for the target variable Y as the output

[1] The successor of Foil, the system Ffoil, can successfully be used for inducing relational definitions of functions.

value of a background knowledge literal, as a constant, or as a linear combination of variables appearing in the clause (using linear regression).

Inductive Constraint Logic Programming. It is well known that Constraint Logic Programming (CLP) can successfully deal with numerical constraints. The idea of Inductive Constraint Logic Programming (ICLP) [53] is to benefit from the number-handling capabilities of CLP, and to use the constraint solver of CLP to do part of the search involved in inductive learning. To this end a maximally discriminant generalization problem in ILP is transformed to an equivalent constraint satisfaction problem (CSP). The solutions of the original ILP problem can be constructed from the solutions of CSP, which can be obtained by running a constraint solver on CSP.

2.4.2 Descriptive ILP

Learning of clausal theories and association rules. In discovering full clausal theories, as done in the system Claudien [13], each example is a Herbrand model, and the system searches for the most general clauses that are true in all the models. Clauses are discovered independently from each other, which is a substantial advantage for data mining, as compared to the learning of classification rules (particularly learning of mutually dependent predicates in multiple predicate learning). In Claudien, search of clauses is limited by the language bias. Its acceptance criterion can be modified by setting two parameters: the requested minimal accuracy and minimal number of examples covered. In another clausal discovery system, Primus [26], the best-first search for clauses is guided by heuristics measuring the "confirmation" of clauses. The Claudien system was further extended to Warmr [8] that enables learning of association rules from multiple relations.

First-order clustering. Top-down induction of decision trees can be viewed as a clustering method since nodes in the tree correspond to sets of examples with similar properties, thus forming concept hierarchies. This view was adopted in C0.5 [12], an upgrade of the Tilde logical decision tree learner. A relational distance-based clustering is presented also in [36]. An early approach combining learning and conceptual clustering techniques was implemented in the system Cola [21]. Given a small (sparse) set of classified training instances and a set of unclassified instances, Cola uses Bisson's conceptual clustering algorithm KBG on the entire set of instances, climbs the hierarchy tree and uses the classified instances to identify (single or disjunctive) class descriptions.

Database restructuring. The system Fender [54] searches for common parts of rules describing a concept, thus forming subconcept definitions to be used in the refurmulation of original rules. The result is a knowledge base with new intermediate concepts and deeper inferential structure than the initial "flat" rulebase. The system Index [24] is concerned with the problem of determining which attribute dependencies (functional or multivalued) hold in the given relational database. The induced attribute dependencies can be used to obtain a more structured database. Both approaches can be viewed as doing predicate invention, where (user selected) invented predicates are used for theory restructuring.

Subgroup discovery. The subgroup discovery task is defined as follows: given a population of individuals and a property of those individuals we are interested in, find the subgroups of the population that are statistically "most interesting", i.e., are as large as possible and have the most unusual statistical (distributional) characteristics with respect to the property of interest. The system Midos [58] guides the top-down search of potentially interesting subgroups using numerous user-defined parameters.

Learning qualitative models of dynamic systems. The automated construction of models of dynamic system may be aimed at qualitative model discovery. A recent qualitative model discovery system [31], using a Qsim-like representation, is based on Coiera's Genmodel to which signal processing capabilities have been added.

Equation discovery. The system LAGRANGE [19] discovers a set of differential equations from an example behavior of a dynamic system. Example behaviors are specified by lists of measurements of a set of system variables, and background knowledge predicates enable the introduction of new variables as time derivatives, sines or cosines of system variables. New variables can be further introduced by multiplication.

3 Analysis of the state-of-the-art in ILP

The analysis of the state-of-the-art in ILP is to a large extent based on the analysis of papers accepted to the International ILP Workshops 1991–1998, as presented in the paper *A perspective on inductive logic programming* by Luc De Raedt [9]. The paper is available at:
`http://www.cs.kuleuven.ac.be/~lucdr/shaking.ps`,
and the accompanying transparencies at:
`http://www.cs.kuleuven.ac.be/~lucdr/Shaking.ppt`.

3.1 Theory, techniques and applications

The analysis shows that since 1991 about 40% of papers were concerned with theoretical ILP issues, about 38% with ILP techniques and implementations, and about 22% of papers reported on experiments and applications. The proportion of applications is significant to categorize ILP as a useful science (as pointed out by David Cornwell, concerning the ESPRIT III project on ILP (1992–1995) [9]). The relatively large proportion of application papers is the result of the research orientation of the groups involved in the two ESPRIT projects on ILP (ILP project 1992–1995, and ILP2 project 1996–1999), as well as the policy of the organizers of ILP Workshops who always encouraged papers on applications of ILP.[2]

3.2 The influence of CL on the theory of ILP

The analysis of theoretical papers accepted to ILP workshops by De Raedt [9] indicates that about 31% of these papers are related to logic programming, the main issues studied being inference rules - 30.8%, program synthesis 23.1%, other issues 19.2%, negation 7.7%, constraint logic programming 7.7%, abduction 5.8%, and implementation 5.8%. An analysis of these proportions is given in [9]. One important remark is that the theory of ILP does not follow the recent developments in logic programming and computational logic but, to a large extent, uses the well known results developed in early LP research. It remains an open question whether more advanced LP techniques are not needed due to the recent orientation of ILP towards data mining and KDD. On the other hand, advanced LP techniques may become increasingly important once ILP starts seriously addressing difficult learning problems in natural language processing, where recursion, negation and other issues requesting a strong theoretical foundation in LP will come into play again.

3.3 Marketing potential for ILP software

The section on the state-of-the-art ILP techniques gave an overview of the most recent technological developments in ILP. These developments have in most cases already shown their potential for solving real-life application problems. Some of the reviewed ILP systems have also a marketing potential within the increasingly developing SW market of tools for data mining. Some tools like Kepler (developed at GMD, and marketed by Dialogis GmbH) have a number of ILP tools incorporated in their KDD toolbox. ILP tools will soon be incorporated also in other commercial data mining packages. Clementine, developed and marketed by Intelligent Solutions Ltd., will (as a result of the ESPRIT IV project ALADIN) soon incorporate ILP data mining algorithms Progol, Tilde and Claudien.

3.4 Influence of different research areas on ILP

The research areas that have strongly influenced ILP are machine learning, computational learning theory, computational logic and, recently, knowledge discovery in databases.

Machine learning. ILP has its roots in machine learning, and most of ILP researchers have done machine learning research before entering ILP.

[2] It is worth noticing, however, that such encouragements are not always fruitful. In 1997 there was a strong emphasis on applications and a Summer school on ILP for KDD was organized to strengthen the link with Data Mining and KDD; despite these efforts, in 1997 the number of theoretical papers reached its maximum, and the number of accepted application papers dropped in comparison with 1996.

Machine learning has always provided the basic research philosophy where experimental evaluation and applications play a key role in the development of novel techniques and tools. This was the case in the early days of ILP as well as now in its mature stage.

Computational learning theory. COLT has helped ILP to better understand the learnability issues and provided some basic learning algorithms that were studied and adapted for the needs of ILP. However, COLT has not influenced ILP research as strongly as other areas reviewed in this section.

Computational logic. Computational logic and logic programming in particular played an extremely important role in early ILP research. Besides providing a framework that helped in the development of the theory of ILP, it provided the established representational formalisms and an initially challenging application area of program synthesis. Due to the difficulty of this application task that can not be solved without very strong biases and restrictions on the hypothesis language, program synthesis has become substantially less popular in the recent years.

Knowledge discovery in databases. The role that computational logic played in the early days of ILP has recently to a large extent been taken by the new challenging area knowledge discovery in databases and data mining. ILP is namely very well suited in filling the gap between the current KDD tools capable of dealing with single data tables and the need of dealing with large data collections stored in multiple relational tables. The influence of KDD on ILP is expected to grow substantially in the coming years.

4 Future challenges for ILP

This section first presents the connections among the areas of Computational Logic that may prove to be fruitful in future research on ILP, some application challenges for ILP, and continues with technological advances that will be needed to be able to deal with these challenges. Challenges addressed as short-term research orientation for ILP are complemented with longer-term challenges for ILP and machine learning/KDD in general. Long-term challenges namely address the problem of dealing with a variety of data sources including images, sound and video inputs.

This section is to a large extent influenced by the panel discussion organized by John W. Lloyd as part of the workshop "Logic Programming and Machine Learning: A Two-way Connection", organized at JICSLP'98 in Manchester in June 1998 as a yearly Compulog Net meeting of the area "Computational Logic and Machine Learning". Besides my personal views expressed at this panel, this section tries to reflect the views of the other panel speakers: Luc De Raedt and Stephen Muggleton.

4.1 Challenging application areas

4.1.1 Application areas related to Computational Logic

Compulog Net has identified some of the main research directions for computational logic, having in mind their industrial relevance in solving real-life problems. These areas include inductive logic programming (ILP), constraint logic programming (CLP) and natural language processing (NLP). The identified areas can be viewed as a challenge in future research directions for ILP. Initial ILP research connecting these areas has already started.

Constraint logic programming. As shown in the overview of techniques for predictive ILP in Section 2.4.1, the connection between ILP and CLP has already been established through the work on *Inductive Constraint Logic Programming.* ILP has recognized the potential of CLP number-handling and of the CLP constraint solver to do part of the search involved in inductive learning. Early work in this area by Page and Frisch, and Mizoguchi and Ohwada in 1992, and recent work by Sebag and Rouveirol [53] show the potential of merging ILP and CLP that has to be explored to a larger extent in the future. Due to the industrial relevance of both areas of computational logic it is expected that the developments in this area may result in products of great industrial benefit. It is unfortunate to note, however, that (to the best of my knowledge) no systematic effort in joining the research efforts of groups in ILP and CLP is envisioned in the near future, e.g., in the framework of ESPRIT V.

Natural language processing. Stephen Muggleton has recently proposed to devote substantial research efforts into the research area *Learning Language in Logic* (LLL). The intention is to apply ILP in various areas of natural language processing, including the syntactical and morphological analysis of corpus data, using the existing large grammars as background knowledge. To this end, a project proposal will be submitted in the framework of ESPRIT V, proposed by some of the partners of the ILP2 project consortium and the most important NLP research groups. Two preparatory meeting of this project have been organized already, and numerous research results already achieved (see the ILP applications bibliography refered to in Section 2.3, as well as a special bibliography on this area being collected by James Cussens - contact `jc@cs.york.ac.uk`).

Abduction. Other initiatives among areas of computational logic have recently also identified the potential for mutual benefits. One of these is the *Abduction and Induction* initiative of Peter Flach and Antonis Kakas that resulted in a series of workshops on this topic and the fortcoming edited volume *Abduction and Induction: Essays on their Relation and Integration* [25]. Early research in this direction by Luc De Raedt (the system CLINT) and more recent work by Dimopoulos and Kakas [15] show the potential for the merging of these technologies. A new ILP framework and system, called ACL [32] for abductive concept learning has been developed and used to study the problems of learning from incomplete background data and of multiple predicate learning. More work in this area is expected in the future.

Higher-order logic. Some work towards the use of higher-order reasoning and the use of functional languages has also started recently, in particular using the declarative programming language Escher (facilitating the use of higher-order features) for learning and hypothesis representation [28]. This work may be a first step towards a larger research initiative in using higher-order features in ILP.

Deductive databases. A tighter connection with deductive database technology has been recently advocated by Luc De Raedt [10, 11] introducing an inductive database mining query language that integrates concepts from ILP, CLP, deductive databases and meta-programming into a flexible environment for relational knowledge discovery in databases. Since the primitives of the language can easily be combined with Prolog, complex systems and behaviour can be specified declaratively. This type of integration of concepts from different areas of computational logic can prove extremely beneficial for ILP in the future. It can lead to a novel ILP paradigm of inductive logic programming query languages whose usefulness may be proved to be similar to those of constraint logic programming.

Other LP based advances. The use of more advanced LP techniques may be due to the recent interest of LP researchers that have already contributed to the use of a more expressive formalism which enables learning together the positive and negative part of a concept where the learned theory is an extended logic programs with classical negation (as opposed to normal programs) [14, 37, 38] where a potential inconsistency in such a theory is resolved by learning priorities amongst contradictory rules.

A lot of work on LP semantics in the past twelve years, culminating in the definition of well-founded semantics and stable model semantics, and subsequent elaborations could be considered in future ILP research, since they allow dealing with non-stratified programs, and 3-valuedness. A lot of work on knowledge representation and non-monotonic reasoning has developed using such semantical basis. Also, recent work on constructive negation would allow inducing rules without fear of floundering, and generating exceptions to default negations which could then be generalized.

Argumentation semantics and procedures are also likely to be useful for composing rules learnt separately from several sources, algorithms, strategies.

The work in LP on preferences [5, 6] is bound to be of interest when combining rules. And all the more so because user preferences might be learnt form instances of user choice and rejection. This will be crucial for information gathering on the basis of user preferences. Fuzzy LP will become important in the future for fuzzyfying such induced preference rules, as well as generalized annotated programs (GAPs) [34] which allow for different degrees of contradiction to be expressed.

Moreover, the implementational techniques of tabling in LP have matured and prove quite useful [59]. In ILP they will save a lot of recomputation because results are memoized in an efficient way. Indeed, in ILP each time a clause is abstracted or refined it has to be tested again with the evidence, though many literals in the clause, and surely the background, are the same, so that repetition of the same computation occurs. This is all the more important when learnt programs become deeper, ie not shallow.

4.1.2 Other challenging application areas for ILP

Molecular biology. At present, molecular biology applications of ILP have come closest to practical relevance. Among the first was predicting protein secondary structure, followed by predicting drug activity through modeling structure-activity relations and predicting the mutagenicity of aromatic and heteroaromatic nitro-compounds. On these problems, which are of immediate practical interest, results that are better or equal to the best previously known results have been obtained, in addition to understandable and relevant new knowledge. Recent ILP applications in the area of molecular biology include prediction of rodent carcinogenicity bioassays, modeling structure-activity relations for modulating transmembrane calcium movement, pharmacophore discovery for ACE inhibition and diterpene structure elucidation. At present and in the immediate future PDB/SCOP and SwissProt data sources can be used for data analysis. In the future there is a lot of potential for ILP applications using data of the human genome project, producing excessive amounts of data that needs to be analysed.

Agents, personalized SW applications and skill acquisition. Internet is an immense source of information and data analysis of internet resources is one of the greatest present challenges. Internet data analysis agents are thus one of the big challenges for future research. This challenge is further explained and elaborated in Section 4.2.2, under the header 'Continuous learning from global datasets'. In addition, personalized SW agents will be developed, by learning from past user behaviour (e.g., user's habits when browsing the web). There are also challenging skill acquisition applications, including control tasks, such as learning from simulation data to learn the operational skill needed for controlling an aircraft, or the robots skills needed for playing footbal in a RoboCup competition.

Medical consultants. In the fifth framework documents medicine is pointed out as an important issue in making the information society more friendly to the citizen. In this sense, learning and answering queries from medical resources available on the Web will become increasingly important. This involves telemedicine for home monitoring of patients with chronic diseases, as well as 'second opinion' consultancy for citizens when dealing with their daily health problems, and experts when dealing with rare and difficult cases.

4.2 Challenges in ILP research

4.2.1 Short-term research issues

The following advancements of ILP will be needed to successfully deal with the above application challenges.

ILP as a methodology for first-order learning. As shown in Section 2.4, ILP has already developed numerous useful techniques for relational knowledge discovery. A recent research trend in ILP is to develop algorithms implementing all the most popular machine learning techniques in the first-order framework. Already developed techniques upgrading propositional learning algorithms include first-order decision tree learning [2], first-order clustering [12, 36], relational genetic algorithms [30], first-order instance-based learning [22], first-order reinforcement learning [20] and first-order Bayesian classifier [27]. It is expected that the adaptation of propositional machine learning algorithms to the first-order framework will continue also in the areas for which first-order implementations still do not exist. This should provide a full scale methodology for relational data mining based on future ILP implementations of first-order Bayesian networks, first-order neural networks, possibly first-order fuzzy systems and other ILP upgrades of propositional machine learning techniques.

Improved robustness and scaling-up of ILP algorithms. This involves the development of robust learning algorithms (w.r.t. noise, missing information, ...), the development of standards for data and knowledge representation, standards for parameter settings, on-line transformers between different data formats, improved efficiency of learners, and the capacity of dealing with large datasets.

Multi-strategy learning and integration. The present data mining applications already require data analysis to be performed by different machine learning algorithms, aimed at achieving best learning results. Multistrategy learning has shown that best results can be achieved by a combination of learning algorithms or by combining the results of multiple learners. Current simple and popular approaches involve the well-known *bagging* and *boosting* that employ redundancy to achieve better classification accuracy [4, 29, 52].

More sophisticated approaches will require the integration of different learners into KDD tools, standard statistical tools and SW packages (like EXCELL) and into SW packages standardly used in particular applications. Integrated machine learning will have to be based also on a better understanding of the different types of problem domains and characteristics of learning algorithms best suited for the given data characteristics.

Mixing of different rules by the use of LP techniques also allows for combining multi-strategy and multi-source learning in a declarative way. Some of the existing techniques are inspired on contradiction removal methods originated in LP, others rely on recent work on updating LP programs with one another [1]. LP based combination techniques may become more important in the near future.

Hierarchically structured learning and predicate invention. Learning from 'flat' datasets nowadays typically results in 'flat' hypotheses that involve no intermediate structure and no constructive induction/predicate invention. Despite substantial research efforts in this area challenging results can still be expected.

Criteria for the evaluation of hypotheses. Except for the standard accuracy measure, other evaluation measures need to be developed. This is of particular importance for descriptive ILP systems that lack such measures for the evaluation of results. Measures of similarity, distance measures, interestingness, precision, measures for outlier detection, irrelevance, and other heuristic criteria need to be studied and incorporated into ILP algorithms.

Criteria for the relevance of background knowledge. Background knowledge and previously learned predicate definitions should be stored for further learning in selected problem areas. One should be aware, however, that an increased volume of background knowledge may have also undesirable properties: not only that learning will become less efficient, but given irrelevant information the results of learning will be less accurate. Therefore it is crucial to formulate criteria for evaluating the relevance of background knowledge predicates before they are allowed to become part of a library of bakground knowledge predicates for a given application area.

Learning from temporal data. ILP is to some extent able to deal with temporal information. However, specialized constructs should be developed for applications in which the analysis of a current stream of time labelled data represents an input to ILP. Experience from the area of *temporal data abstraction* could be used to construct higher-level predicates summarizing temporal phenomena.

4.2.2 Long-term research issues

Some of the issues discussed in this section are relevant to ILP only, whereas others are relevant to machine learning in general. Some of these issues have been pointed out as important already by Tom Mitchell in the article published in the Fall 1997 issue of the AI Magazine [42].

Analysis of comprehensibility. It is often claimed that for many applications comprehensibility is the main factor if the results of learning are to be accepted by the experts. Despite these claims and some initial investigations of intelligibility criteria for symbolic machine learning (such as the standard Occam's razor and minimal description length criteria) there are few research results concerning the intelligibility evaluation by humans.

Building specialized learners and data libraries. Particular problem areas have particular characteristics and requirements, and not all learning algorithms are capable of dealing with these. This is a reason for starting to build specialized learners for different types of applications. In addition, libraries of 'cleaned' data, background knowledge and previously learned predicate definitions should be stored for further learning in selected problem areas. Notice that such libraries are currently being established for selected problem areas in molecular biology. This approach will lead to the reusability of components and to extended example sets that should be achieved also as the result of systematic query answering and experimentation that is part of 'continuous' learning as discussed below.

Continuous learning from 'global' datasets. Under this heading we understand the requirement for *learning from various data sources*, where data sources can be of various types, including propositional and relational tables, textual data, and hypermedia data including speech, images and video. This involves also the issue of *globality*, i.e., learning from local datasets as well as referencial datasets collected and maintained by the world's best experts in the area, referential case bases of 'outlier' data as well as data that is publicly available via WWW. Achieving the requirement of continuous and global learning will require also *learning agents* for permanent learning by theory revision from updated world-wide data, as well as the development of *query agents* that will be able to access additional information from WWW via query answering (invoked either by experts or by automatically extracting answers from WWW resources, possibly by invoking learning and active experimentation). Query agents may involve dynamic abductive querying on WWW.

5 Organizational aspects

5.1 Current and future European projects and networks

In the fourth framework the following projects and networks relevant to ILP are being funded:

ILP2 (1996–1999) LTR, a follower of the ILP project (1992–1995), supporting fundamental developments of the theory and techniques of ILP, as well as applications. More information at `http://www.cs.kuleuven.ac.be/~ml/esprit/esprit.ilp2.20237.html`.

ALADIN (1998–2001) applied research project aimed at applications of ILP and the incorporation of ILP algorithms into commectial data mining software Clementine, marketed by the Intelligent Solutions Ltd.

KESO (1996-1998) project Knowledge Extraction for Statistical Offices has as its goal to construct a versatile, efficient, industrial strength knowledge extraction/data mining system prototype that satisfies the needs of providers of large-scale statistical data. The work in this project triggered the development of the Midos system by GMD. More information at `http://www-fit-ki.gmd.de/projects/ml/keso.html`.

ILPNet2 (1998-2001) a network of excellence in ILP, a follower of ILPNET (1993–1996) `http://www-ai.ijs.si/ilpnet.html`, providing infras-

tructure support and mobility of researchers in ILP. More information at
`http://www.cs.bris.ac.uk/~ilpnet2/`.

CompulogNet3 (1996–1999) a network of excellence in Computational logic, a follower of two previous networks, providing infrastructure support and mobility of researchers in computational logic. More information at `http://www.compulog.org/net-www`.

MLNet2 (1999-2001) a network of excellence on Machine Learning, a follower of the ML Net network of excellence, providing infrastructure support and mobility of ML researchers. Compulog Net and ML Net have previously successfully collaborated in supporting various activities of mutual interest and this collaboration is expected to continue in the future.

One of the strategic orientations for LTR in the forthcoming fifth framework, reflected in the green paper proposal for the fifth framework suggests an overall theme of 'User friendliness in an information society', and stresses the need for research and development into generic technologies. One of the three generic technologies mentioned is for *representation, creation and handling of knowledge* for which ILP is directly relevant.

It is planned to propose, as a follower of ILP2, a project on Learning Language in Logic (LLL) by a consortium of ILP and NLP partners. A project proposal on ILP for KDD is also planned, aimed at advancing ILP techniques towards learning from large relational databases.

The tight collaboration of three networks of excellence is of great importance for the ILP research area. The networks can help each other in the promotion of this research field and in providing the research infrastructure (including collecting the ILP bibliography, on-line access to ILP systems and databases, as well as joint education and training activities and industrial liaisons).

5.2 Means of self-assessment via organized competitions

One of the most interesting and objective ways of evaluating research results is by submitting the results of learners to international competitions organized for selected problem areas. In those 'blind test' competitions (results of which become known only by the end of the competition period, and are not known to the public before the final evaluation of the results) algorithms compete among each other, as well as with human experts. These competitions want to answer a question whether AI programs can indeed participate in and contribute to scientific discovery.

Some of these competitions have been announced also as 'AI challenges' at IJCAI97 [57]. The challenges relevant to ILP include:

PTE2 is the Second Predictive Toxicology Evaluation challenge aimed at better understanding of environmentally-induced cancers and the mechanisms of chemical carcinogenesis. The rodent carcinogenicity tests conducted within the US National Toxicology Program by the National Institute of Environmental Health Sciences (NIEHS) resulted in a large database of compounds classified as carcinogens or otherwise. The Predictive-Toxicology Evaluation project of the NIEHS provides the opportunity to compare carcinogenicity predictions on previously untested chemicals. This has resulted in two blind trials: PTE-1 and the ongoing PTE-2.

The Predictive Toxicology Evaluation Challenge has been devised to provide Machine Learning programs an opportunity to participate in carcinogensis prediction. A Prolog representation of the data for the carcinogenesis problem was made available, and the prediction for each chemical (cancerogenic or non-cancerogenic) was required to enter the competition. Entries submitted by Mid November 1998 were to be evaluated for chemical relevance by Doug Bristol (US National Institute of Environmental Health Sciences) in terms of accuracy and explanatory power. For more information see `http://www.comlab.ox.ac.uk/oucl/groups/machlearn/PTE`.

CASP3 is the Third Community Wide Experiment on the Critical Assessment of Techniques for Protein Structure Prediction. Protein crystallographers and NMR spectroscopists provided the information on structures for which they expected to know the results by October 1998. The goal of the competition was to compare the submitted protein structure predictions of experts and algorithms with the actual results. The evaluation meeting was held in Asilomar, CA in December 1998. For more information see `http://PredictionCenter.llnl.gov/casp3/doc/casp3-description.html`.

LLL is a challenge in Language Learning in Logic, using a FraCaS dataset which contains a grammar created as a part of the FraCaS project aimed at developing a standard framework for computational semantics - see the project deliverables at `http://www.cogsci.ed.ac.uk/~fracas/deliverables.html`. FraCaS grammar is an attribute-value grammar able to produce the semantic interpretations of the covered sentences in the form of quasi logical forms (QLF). Given samples of the FraCaS grammar and language L, the LLL challenge is to learn a grammar G' that generates a language L' as close as possible to L if only the correct sentences generated by the FraCaS grammar are used as input. The data and the document on the FraCaS dataset and the LLL challenge were distributed to the interested parties in 1998. For more information contact `{kazakov,stephen}@cs.york.ac.uk`.

International challenges seem to be the best possible evaluation method for ILP. Their characteristics, particularly important for the evaluation of algorithms participating in the challenges, are namely: involvement in genuine scientific discovery; participation in true blind trials; availability of a large database with expert-certified classifications; strong competition from methods used by domain experts; and independent adjudication by world-reknowned scientists.

These challenge competitions can be considered as a model for the evaluation of other computational logic methods as well.

Best application award. Ways of showcasing important industrial applications of CL were already discussed at the Compulog Net executive board meetings. One of the ideas was to follow the practice of Erudit that had successfully organised a *competition for best applications* and it was agreed that CLN should do the same. It seems worthwhile to launch a competition for ten best applications, selected by a chosen jury, and construct a special web section to feature these. The top 4 or 5 could then be selected for featuring in an industrial brochure as well as an article for newspapers and magazines. Criteria for evaluation need to be decided, which may (besides 'saved millions for the company') incorporate the selection of such applications that are large, fast, safety-critical, or appeal in other ways (see the 30-APR-1997 mail to executive board members of CLN). It would be possible to launch such a competition as part of the planned CL 2000 conference in London.

Best PhD thesis award. The award for the best PhD thesis in Computational logic could be launced, and awarded on an annual/bi-annual basis at future CL conferences, starting with CL 2000. A similar initiative has been recently launched by ECCAI for the best European PhD thesis in AI.

5.3 Education, training and technology transfer

Since 1995, ILP has been aware of the importance of education, training and technology transfer. As part of ILP2 and ILPNet2, end-user clubs have been organized, their members being the potential end-users of the ILP technology. Specialized seminars and meeting of the end-user club have been organized. These include an ILP2 end-user meeting in Prague in September 1997, and ILP2 end-user workshop in York in December 1997. The project proposal on LLL is a result of such an end-user club initiative. The first ILPNet2 end-user club meeting was organized in Caminha, Portugal as part of the Summer School on KDD organized in September 1998.

The previous successful collaboration between the CompulogNet and ILPNet can be enhanced by, for instance, jointly supporting the education and training activities of the Center for Knowledge Transfer in Information

Technologies at IJS (Ljubljana) that is involved in these activities for ILP-Net2. These activities include also the industrial seminars (one of which the Center organized already for Compulog Net in 1997: 'Biomedical applications of Computational Logic and Machine Learning').

Forms of technology transfer and marketing have been studied by the IJS Center for Knowledge Transfer in Information Technologies and Studio Phi (a marketing company) as part of the marketing initiative in the field of Decision support and Knowledge discovery. A feasibility study, financed by venture capital investing in this area, showed marketing potential in the area of Decision support and KDD and proposed an appropriate organizational structure and an activity plan to reach this aim. This could be seen as a model for a similar marketing study for the commercialization of ILP (and other areas of computational logic), possibly by engaging area experts as well as experts in marketing.

6 Benefits for the European industry and public sector

The gap between data generation and its understanding is increasing. This is true for the industrial sector as well as for the public sector. This is also true for data whose collection is aimed at new scientific discoveries.

The increased amounts of data have been recognized as a challenge by the research communities doing machine learning, data visualization and statistics. They reacted by proposing a new, commercially attractive name aimed at addressing the market in need for these technologies for effective analysis of data. The new areas of data mining and knowledge discovery in databases have successfully found their way to the commercial market. This holds both for data mining products as well as for the applications of data mining and KDD.

ILP is a mature technology that enables efficient analysis of relational data. As such it well complements other KDD and data mining technologies since it offers a unique property of being able to deal with structured data and data stored in relational databases. This provides ILP with an advantage on the market when compared with other data mining techniques that are only able to deal with flat data stored in single tables. ILP has to find the means of using this advantage in the KDD market in the future. Moreover, next to information retrieval, ILP is very appropriate also for the analysis of textual data whose role is becoming increasingly important with the advent of Internet. ILP has a huge marketing potential in this area as well. ILP has also a great potential for scientific discovery in various areas of science.

It is well known that Europe has a leading role in ILP research. Europe has also the advantage of having joint research projects and scientific networks funded by the Commission of the EU. The potential of ILP and KDD should be explored by supporting European research projects in the area ILP for KDD, supporting applied projects for integrating ILP tools into commercial KDD software, and by further supporting scientific networks providing infrastructure for scientific and commercial developments of ILP. Industrial liaisons can be promoted via user-clubs of European projects, in applied projects as well as in direct industrial contracts. Further commercialization of the technology of ILP may be the topic of a commercial feasibility study aimed at the integration of commercializing efforts in this area and the systematic promotion of this technology in industry and in the public sector.

Acknowledgments

The section on the state-of-the-art ILP techniques reflects some of the research results achieved in the EU funded projects no. 6020 ILP (1992–95) and no. 20237 ILP2 (1996–98). The selection is based on the research in ILP directed towards practical algorithms for relational knowledge discovery used in data mining of real-world relational datasets. I am grateful to John W. Lloyd, Luc De Raedt and Stephen Muggleton for enabling me to base the ideas for future developments in ILP also on their views expressed at the panel discussion of the Compulog Net area meeting with the title "Logic Programming and Machine Learning: A Two-way Connection" organized by John W. Lloyd at JICSLP'98 in Manchester. I am grateful to Luc De Raedt, Sašo Džeroski, Peter Flach and Stefan Wrobel for their collaboration in previous joint overviews of ILP that strongly influenced this work and its presentation. Thanks goes also to Luis Moniz Pereira and Antonis Kakas for their comments on the draft of this paper.

References

[1] J. J. Alferes, J. A. Leite , L. M. Pereira, H. Przymusinska and T. C. Przymusinski. Dynamic Logic Programming, In A. Cohn, L. Schubert and S. Shapiro (eds.) *Proc. of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pp. 98–109, Morgan Kaufmann, 1998.

[2] H. Blockeel and L. De Raedt. Top-down induction of logical decision trees. Submitted to *DAMI, Special Issue on Inductive Logic Programming*, 1998.

[3] I. Bratko, S. Muggleton. Applications of inductive logic programming. *Communications of the ACM*, Vol. 38, 1995.

[4] L. Breiman. Bagging predictors. *Machine Learning* 24(2): 123–140, 1996.

[5] G. Brewka. Well-founded semantics for extended logic programs with dynamic preferences. *Journal of of AI Research*, 4:19–36, 1996.

[6] G. Brewka and T. Eiter. Preferred answer sets. In A. Cohn, L. Schubert and S. Shapiro (eds.) *Proc. of the Sixth International Conference on Principles of Knowledge Representation and Reasoning (KR'98)*, pp. 89–97, Morgan Kaufmann, 1998.

[7] W. Cohen. Recovering software specifications with inductive logic programming. *Proc. Twelfth National Confeernce on Artificial Intelligence*, The MIT Press, 1994.

[8] L. Dehaspe, L. De Raedt. Mining association rules in multiple relations. *Proc. Seventh Int. Workshop on Inductive Logic Programming*, Springer, LNAI 1297, pp. 125–132, 1997.

[9] L. De Raedt. A perspective on inductive logic programming. Invited lectures at *The Workshop on Current and Future Trends in Logic Programming*, Shakertown, to appear in Springer LNCS, 1999. Available at: `http://www.cs.kuleuven.ac.be/~lucdr/shaking.ps`.

[10] L. De Raedt. A relational database mining query language. In *Proc. Fourth Workshop on Artificial Intelligence and Symbolic Computation*, Springer LNAI, 1998 (in press).

[11] L. De Raedt. An inductive logic programming query language for database mining (extended abstract). In *Proc. JICSLP'98 post-conference workshop Compulog Net Area Meeting on Computational Logic and Machine Learning*, pp. 27–34, 1998.

[12] L. De Raedt, H. Blockeel. Using logical decision trees for clustering. *Proc. Seventh Int. Workshop on Inductive Logic Programming*, Springer, LNAI 1297, pp. 133–140, 1997.

[13] L. De Raedt, L. Dehaspe. Clausal discovery. *Machine Learning*, 26(2/3): 99–146, 1997.

[14] Y.Dimopoulos, A.C. Kakas. Learning Non-monotonic Logic Programs: Learning exceptions. In *Proc. of 8th European Conference on Machine Learning ECML-95*, 122–138, Springer LNAI 912, 1995.

[15] Y. Dimopoulos, A.C. Kakas. Abduction and inductive learning. In L. De Raedt, ed., *Advances in Inductive Logic Programming*, pp. 144–171, IOS Press, 1996.

[16] Y. Dimopoulos, S. Džeroski and A.C. Kakas. Integrating Explanatory and Descriptive Induction in ILP. In *Proc. of the 15th International Joint Conference on Artificial Intelligence (IJCAI97)*, pp. 900–907, 1997.

[17] B. Dolšak, S. Muggleton. The application of inductive logic programming to finite-element mesh design. In S. Muggleton, ed., *Inductive Logic Programming*, pp. 453–472. Academic Press, 1992.

[18] S. Džeroski, L. De Haspe, B. Ruck, W. Walley. Classification of river water quality data using machine learning. In *Proc. of the Fifth Int. Conference on the Development and Application of Computer Technologies to Environmental Studies, Vol. I: Pollution Modelling.*, pp. 129–137. Computational Mechanics Publications, Southampton, 1994.

[19] S. Džeroski, L. Todorovski. Discovering dynamics: From inductive logic programming to machine discovery. *Proc. Tenth Int. Conference on Machine Learning*, pp. 97–103, Morgan Kaufmann, 1993.

[20] S. Džeroski, L. De Raedt, H. Blockeel. Relational reinforcement learning. In D. Page (Ed.) *Proc. Eighth Int. Conference on Inductive Logic Programming*, pp. 11–22, Springer, LNAI 1446, 1998.

[21] W. Emde. Learning of characteristic concept descriptions from small sets to classified examples. *Proc. Seventh European Conference on Machine Learning*, LNAI 784, pp. 103–121, Springer, 1994.

[22] W. Emde, D. Wettschereck. Relational instance-based learning. *Proc. Thirteenth Int. Conference on Machine Learning*, pp. 122–130, Morgan Kaufmann, 1996.

[23] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy, eds. *Advances in Knowledge Discovery and Data Mining.* The MIT Press, 1995.

[24] P.A. Flach. Predicate invention in inductive data engineering. *Proc. Sixth European Conference on Machine Learning*, Springer, LNAI 667, pp. 83-94, 1993.

[25] P. Flach, A. Kakas (eds.) *Abduction and Induction: Essays on their Relation and Integration*, Kluwer, 1999 (in press).

[26] P. Flach, N. Lachiche. Cooking up integrity constraints with Primus. Technical report, University of Bristol, 1998.

[27] P.A. Flach and N. Lachiche. 1BC: A first-order Bayesian classifier. In *Proc. of the 9th International Workshop on Inductive Logic Programming (ILP'99)*, pp. 92–103, Springer LNAI 1634, 1999.

[28] P. Flach, C. Giraud-Carrier, J.W. Lloyd. Strongly typed inductive concept learning. In D. Page (Ed.) *Proc. Eighth Int. Conference on Inductive Logic Programming*, pp. 185–194, Springer, LNAI 1446, 1998.

[29] Y. Freund and R.E. Shapire. Experiments with a new boosting algorithm. In *Proc. Thirteenth International Machine Learning Conference ICML'96*, 148–156, Morgan Kaufmann, 1996.

[30] A. Giordana, C. Sale. Learning structured concepts using genetic algorithms. In *Proc. Ninth Int. Workshop on Machine Learning*, pp. 169–178, 1992.

[31] D.T. Hau and E.W. Coiera. Learning qualitative models of dynamic systems. *Machine Learning*, 26(2/3): 177–212, 1997.

[32] A.C. Kakas and F. Riguzzi. Learning with Abduction. In *Proc. of the 7th International Workshop on Inductive Logic Programming (ILP97)*, pp. 181-189. Springer LNAI 1297, 1997.

[33] A. Karalič, I. Bratko. First order regression. *Machine Learning*, 26(2/3): 147–176, 1997.

[34] M. Kifer and V.S. Subrahmanian. Generalized annotated logic programs. *Journal of Logic Programming*, 1992.

[35] R.D. King, S. Muggleton, R. Lewis, M.J.E. Sternberg. Drug design by machine learning: The use of inductive logic programming to model the structure-activity relationships of trimethoprim analogues binding to dihydrofolate reductase. *Proc. of the National Academy of Sciences of the USA* 89(23): 11322–11326, 1992.

[36] M. Kirsten, S. Wrobel. Relational distance-based clustering. In D. Page (Ed.) *Proc. Eighth Int. Conference on Inductive Logic Programming*, pp. 261–270, Springer, LNAI 1446, 1998.

[37] E. Lamma, F. Riguzzi and L. M. Pereira. Strategies in Combined Learning via Logic Programs. *Machine Learning*, 1999 (in press).

[38] E. Lamma, F. Riguzzi and L. M. Pereira. Agents Learning in a Three-valued Logical Setting. In A. Panayiotopoulos (ed.)*Proc. of the Workshop on Machine Learning and Intelligent Agents*, in conjunction with *Machine Learning and Applications, Advanced Course on Artificial Intelligence (ACAI-99)*, Chania, Greece, 1999 (in press).

[39] N. Lavrač, S. Džeroski. *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood, 1994.

[40] N. Lavrač. Inductive logic programming for relational knowledge discovery. Invited lecture at *The 1998 Joint International Conference and Symposium on Logic Programming*, pp. 7–22, MIT Press, 1998.

[41] D. Michie. Machine learning in the next five years. *Proc. Third European Working Session on Learning*, pp. 107–122. Pitman, 1988.

[42] T. Mitchell. Does machine learning really work? *AI Magazine* 18 (3): 11–20, 1997.

[43] F. Mizoguchi, H. Ohwada, M. Daidoji, S. Shirato. Using inductive logic programming to learn classification rules that identify glaucomatous eyes. In N. Lavrač, E. Keravnou, B. Zupan, eds., *Intelligent Data Analysis in Medicine and Pharmacology*, Kluwer, pp. 227–242, 1997.

[44] R.J. Mooney, 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.

[45] S. Muggleton, ed. *Inductive Logic Programming.* Academic Press, 1992.

[46] S. Muggleton. Inverse Entailment and Progol. *New Generation Computing, Special issue on Inductive Logic Programming*, 13(3-4), 1995.

[47] S. Muggleton, R.D. King, M.J.E. Sternberg. Protein secondary structure prediction using logic. *Protein Engineering* 7: 647–657, 1992.

[48] S. Muggleton, C. Feng. Efficient induction of logic programs. *Proc. First Conference on Algorithmic Learning Theory*, pp. 368–381. Ohmsha, Tokyo, 1990.

[49] S. Muggleton, L. De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming* 19/20: 629–679, 1994.

[50] S.H. Nienhuys-Cheng, R. de Wolf. *Foundations of inductive logic programming.* Springer LNAI 1228, 1997.

[51] J.R. Quinlan. Learning logical definitions from Relations. *Machine Learning* 5:239-266, 1990.

[52] J.R. Quinlan (1996) Boosting, bagging, and C4.5 . In *Proc. Thirteenth National Conference on Artificial Intelligence*, 725–730, AAAI Press.

[53] M. Sebag, C. Rouveirol. Constraint Inductive Logic Programming. In L. De Raedt, ed., *Advances in Inductive Logic Programming*, pp. 277–294, IOS Press, 1996.

[54] E. Sommer. Rulebase stratifications: An approach to theory restructuring. *Proc. Fourth Int. Workshop on Inductive Logic Programming*, GMD-Studien 237, pp. 377-390, 1994.

[55] A. Srinivasan, S. Muggleton, R.D. King, M.J.E. Sternberg. Mutagenesis: ILP experiments in a non-determinate biological domain. *Proc. Fourth Int. Workshop on Inductive Logic Programming*, GMD-Studien 237, pp. 217–232, 1994.

[56] A. Srinivasan, R.D. King, S. Muggleton, M.J.E. Sternberg. Carcinogenesis prediction using inductive logic programming. In N. Lavrač, E. Keravnou, B. Zupan, eds., *Intelligent Data Analysis in Medicine and Pharmacology*, Kluwer, pp. 243–260, 1997.

[57] A. Srinivasan, R.D. King, S. Muggleton, M.J.E. Sternberg. The Predictive Toxicology Evaluation Challenge. In *Proc. Fifteenth Int. Joint Conf. on Artificial Intelligence*, pp. 4–9, Morgan Kaufmann, 1997.

[58] S. Wrobel. An algorithm for multi-relational discovery of subgroups. *Proc. First European Symposium on Principles of Data Mining and Knowledge Discovery*, pp. 78–87. Springer, 1997.

[59] XSB Group Home Page: `http://www.cs.sunysb.edu/~sbprolog/`.