Relational Data Mining and Subgroup Discovery

Nada Lavrač

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

Abstract. In Inductive Logic Programming (ILP), the recent shift of attention from program synthesis to knowledge discovery resulted in advanced relational data mining techniques that are practically applicable for discovering knowledge in relational databases. This paper gives a brief introduction to ILP, presents the state-of-the-art ILP techniques for relational knowledge discovery and outlines recent approaches to relational subgroup discovery.

1 Introduction

Inductive logic programming (ILP) [33, 36, 26, 12] is a research area that has its roots 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, also by Knowledge Discovery in Databases (KDD) [15] 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 a *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.

This paper first gives a brief introduction to ILP and presents a selection of recently developed ILP techniques for relational data mining [12], followed by an outline of recent approaches to relational subgroup discovery. The overview is restricted to techniques satisfying the strong criterion formulated for machine learning by Michie [31] that requires explicit symbolic form of induced descriptions.

2 State-of-the-art in ILP

This section 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 [36]. 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 [36, 37]. For a practical introduction to ILP see [26].

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 $M(B \cup H)$ [36].

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* [1]. In a broader sense, predictive ILP incorporates also *first-order regression* [22] and *constraint inductive logic programming* [40] for which again different acceptance criteria apply.

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 [7]. 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 [7] to allow for theories that satisfy some other acceptance criteria (similarity, associativity, interestingness), descriptive ILP can be extended to incorporate *learning of association rules* [2], *first-order clustering* [6, 13, 23], *database restructuring* [16, 42] *subgroup discovery* [45], *learning qualitative models* [21] and *equation discovery* [10].

An illustrative example 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.

Other ILP settings There has been a suggestion [8] 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* [14]. Excellent predictive results have been achieved by the relational instance-based learner RIBL [14] in numerous classification and prediction tasks. Recently, *first-order reinforcement learning* [11] and first-order Bayesian classifier [18] 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.

3 Relational Data Mining Techniques

This section reviews the state-of-the-art relational data mining techniques most of which have already shown their potential for use in real-life applications. The overview is limited to recent Relational Data Mining developments, aimed at the analysis of real-life databases [27, 12]. These developments have a marketing potential in the prosperous new areas of Data Mining and Knowledge Discovery in Databases. 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.

3.1 Predictive RDM techniques

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 [39][1], Golem [35] and Progol [34]. 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. The system LINUS [25, 26], developed from a learning component of QuMAS [32], introduced the propositionalization paradigm by transforming an ILP problem into a propositional learning task.

Induction of logical decision trees. The system Tilde [1] 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) [22] 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 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) [40] 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

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

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.

3.2 Descriptive RDM techniques

Learning of clausal theories and association rules. In discovering full clausal theories, as done in the system Claudien [7], 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 [17], the best-first search for clauses is guided by heuristics measuring the "confirmation" of clauses. The Claudien system was further extended to Warmr [2] 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 [6], an upgrade of the Tilde logical decision tree learner. A relational distance-based clustering is presented also in [23]. An early approach combining learning and conceptual clustering techniques was implemented in the system Cola [13]. 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 [42] 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 [16] 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 [45] 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 [21], using a Qsim-like representation, is based on Coiera's Genmodel to which signal processing capabilities have been added.

Equation discovery. The system LAGRANGE [10] 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.

Inductive databases. A tighter connection with deductive database technology has been recently advocated by Luc De Raedt [4, 5] 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 RDMn 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.

3.3 Some challenges in RDM research

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 [1], first-order clustering [6, 23], relational genetic algorithms [20], first-order instance-based learning [14], first-order reinforcement learning [11] and first-order Bayesian classifier [18]. 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.

4 Relational Subgroup Discovery and Related Work

4.1 Relational Subgroup Discovery

In the newly emerging field of subgroup discovery two most important systems for discovering subgroups are Explora [24] and Midos [45]. The first system treats the learning task as a single relation problem, i.e., all the data are assumed to be available in one table (relation), while the second one extends this task to multi-relational databases, which is related to a number of other learning tasks [7, 30, 46], mostly in the ILP [12, 26]. In both systems the propositional (attribute-value) language is used to describe the induced hypotheses, i.e., discovered subgroups are defined as conjunctions of features (attributes values). The most important features of Explora and Midos concern the use of heuristics for subgroup discovery; the heuristics are outlined below.

We have developed a relational subgroup discovery system RSD [28] on principles that employ the following main ingredients: exhaustive first-order feature construction, elimination of irrelevant features, an implementation of a relational rule learner, the weighted covering algorithm and incorporation of example weights into the weighted relative accuracy heuristic, probabilistic classification, and area under ROC rule set evaluation.

As the input, RSD expects (a) a relational database containing one main table (relation) where each row corresponds to a unique *individual* and one attribute of the main table is specified as the *class* attribute, and (b) a mode-language definition used to construct first-order features.

The main output of RSD is a set of subgroups whose class-distributions differ substantially from those of the complete data-set. The subgroups are identified by conjunctions of symbols of pre-generated first-order features. As a by-product, RSD also provides a file containing the mentioned set of features and offers to export a single relation (as a text file) with rows corresponding to individuals and fields containing the truth values of respective features for the given individual. This table is thus a propositionalised representation of the input data and can be used as an input to various attribute-value learners.

An important feature of the RSD algorithm is the use of the weighted covering algorithm. In the classical covering algorithm of rule-set induction, only the first few induced rules may be of interest as subgroup descriptors with sufficient coverage, since subsequently induced rules are induced from biased example subsets, i.e., subsets including only positive examples not covered by previously induced rules. This bias constrains the population for subgroup discovery in a way that is unnatural for the subgroup discovery process which is, in general, aimed at discovering interesting properties of subgroups of the entire population. In contrast, the subsequent rules induced by the weighted covering algorithm allow for discovering interesting subgroup properties of the entire population.

The weighted covering algorithm is implemented in such a way that covered positive examples are not deleted from the current training set. Instead, in each run of the covering loop, the algorithm stores with each example a count how many times (with how many rules induced so far) the example has been covered. Weights of covered examples decrease according to the formula $e(i) = \frac{1}{i+1}$, where $e(i)$ is the weight of an example being covered i times.

A variant of the weighted covering algorithm has been used also in the context of the SD subgroup discovery algorithm [19], and in the CN2-SD subgroup discovery algorithm [28].

4.2 Measures of Interestingness

Various rule evaluation measures and heuristics have been studied for subgroup discovery, aimed at balancing the size of a group (referred to as factor g in [24]) with its distributional unusualness (referred to as factor p). The properties of functions that combine these two factors have been extensively studied (the "p-g-space"). Similarly, the weighted relative accuracy heuristic, defined as $WRAcc(Class \leftarrow Cond) = p(Cond).p(Class|Cond) - p(Class))$ and used in [44], trades off generality of the rule ($p(Cond)$, i.e., rule coverage) and relative accuracy ($p(Class|Cond) - p(Class)$). Besides such 'objective' measures of interestingness, some 'subjective' measure of interestingness of a discovered pattern can be taken into the account, such as actionability ('a pattern is interesting if the user can do something with it to his or her advantage') and unexpectedness ("a pattern is interesting to the user if it is surprising to the user") [41].

4.3 Subgroup Evaluation Measures

Evaluation of induced subgroups in the ROC space [38] shows classifier performance in terms of false alarm or *false positive rate* $FPr = \frac{FP}{TN+FP}$ (plotted on the X-axis) that needs to be minimized, and sensitivity or *true positive rate* $TPr = \frac{TP}{TP+FN}$ (plotted on the Y-axis) that needs to be maximized. The ROC space is appropriate for measuring the success of subgroup discovery, since subgroups whose TPr/FPr tradeoff is close to the diagonal can be discarded as insignificant. The standard approach is to use the area under the ROC convex hull defined by subgroups with the best TPr/FPr tradeoff as a quality measure for comparing the success of different learners.

Acknowledgments

This work was supported by the Slovenian Ministry of Education, Science and Sport, and the EU funded project Data Mining and Decision Support for Business Competitiveness: A European Virtual Enterprise (IST-1999-11495). I am grateful to Luc De Raedt, Sašo Džeroski, Peter Flach and Dragan Gamberger for joint research in ILP and subgroup discovery.

References

1. H. Blockeel and L. De Raedt. Top-down induction of logical decision trees. Submitted to *DAMI, Special Issue on Inductive Logic Programming*, 1998.
2. 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.
3. L. De Raedt. A perspective on inductive logic programming. Invited lecture at *The Workshop on Current and Future Trends in Logic Programming*, Shakertown, to appear in Springer LNCS, 1999.
   Available at: `www.cs.kuleuven.ac.be/~lucdr/shaking.ps`.

4. L. De Raedt. A relational database mining query language. In *Proc. Fourth Workshop on Artificial Intelligence and Symbolic Computation*, Springer LNAI, 1998 (in press).
5. 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.
6. 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.
7. L. De Raedt, L. Dehaspe. Clausal discovery. *Machine Learning*, 26(2/3): 99–146, 1997.
8. 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.
9. 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.
10. 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.
11. 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.
12. S. Džeroski, N. Lavrač (eds.) *Relational Data Mining*. Springer, 2001.
13. 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.
14. W. Emde, D. Wettschereck. Relational instance-based learning. *Proc. Thirteenth Int. Conference on Machine Learning*, pp. 122–130, Morgan Kaufmann, 1996.
15. U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy, eds. *Advances in Knowledge Discovery and Data Mining*. The MIT Press, 1995.
16. P.A. Flach. Predicate invention in inductive data engineering. *Proc. Sixth European Conference on Machine Learning*, Springer, LNAI 667, pp. 83-94, 1993.
17. P. Flach, N. Lachiche. Cooking up integrity constraints with Primus. Technical report, University of Bristol, 1998.
18. 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.
19. Gamberger, D. & Lavrač, N. (2002) Descriptive induction through subgroup discovery: a case study in a medical domain. In *Proc. of 19th International Conference on Machine Learning (ICML2002)*, Morgan Kaufmann, in press.
20. A. Giordana, C. Sale. Learning structured concepts using genetic algorithms. In *Proc. Ninth Int. Workshop on Machine Learning*, pp. 169–178, 1992.
21. D.T. Hau and E.W. Coiera. Learning qualitative models of dynamic systems. *Machine Learning*, 26(2/3): 177–212, 1997.
22. A. Karalič, I. Bratko. First order regression. *Machine Learning*, 26(2/3): 147–176, 1997.

23. 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.
24. Klösgen, W. (1996) Explora: A multipattern and multistrategy discovery assistant. In U.M. Fayyad, G. Piatetski-Shapiro, P. Smyth and R. Uthurusamy (Eds.) *Advances in Knowledge Discovery and Data Mining*, 249–271. MIT Press.
25. N. Lavrač, S. Džeroski and M. Grobelnik. Learning nonrecursive definitions of relations with LINUS. In *Proc. Fifth European Working Session on Learning*, pp. 265–281. Springer.
26. N. Lavrač and S. Džeroski. *Inductive Logic Programming: Techniques and Applications.* Ellis Horwood, 1994.
27. N. Lavrač, S. Džeroski and M. Numao. Inductive logic programming for relational knowledge discovery. *New Generation Computing* 17 (1): 3–23, 1999.
28. N. Lavrač, F. Železný, P. Flach. Relational subgroup discovery: A propositionalization approach through first-order feature construction. *Proc. Int. Conference on Inductive Logic Programming, ILP-2002.* Springer, 2002 (in press).
29. N. Lavrač, P. Flach, B. Kavšek, L. Todorovski. Rule induction for subgroup discovery with CN2-SD. *Proc. Integrating Aspects of Data Mining, Decision Support and Meta-Learning*, Workshop at the ECML/PKDD-2002 Conference, in press, 2002.
30. H. Mannila, H. Toivonen. On an algorithm for finding all interesting sentences. In R. Trappl, ed., *Proc. Cybernetics and Systems'96*, pp. 973–978, 1996.
31. D. Michie. Machine learning in the next five years. *Proc. Third European Working Session on Learning*, pp. 107–122. Pitman, 1988.
32. I. Mozetič. Learning of qualitative models. In I. Bratko and N. Lavrač (eds.) *Progress in Machine Learning*, pp. 201–217. Sigma Press, 1987.
33. S. Muggleton, ed. *Inductive Logic Programming.* Academic Press, 1992.
34. S. Muggleton. Inverse Entailment and Progol. *New Generation Computing, Special issue on Inductive Logic Programming*, 13(3-4), 1995.
35. S. Muggleton, C. Feng. Efficient induction of logic programs. *Proc. First Conference on Algorithmic Learning Theory*, pp. 368–381. Ohmsha, Tokyo, 1990.
36. S. Muggleton, L. De Raedt. Inductive logic programming: Theory and methods. *Journal of Logic Programming* 19/20: 629–679, 1994.
37. S.H. Nienhuys-Cheng, R. de Wolf. *Foundations of inductive logic programming.* Springer LNAI 1228, 1997.
38. Provost, F. & Fawcett, T. (2001) Robust classification for imprecise environments. *Machine Learning*, 42(3), 203–231.
39. J.R. Quinlan. Learning logical definitions from Relations. *Machine Learning* 5:239-266, 1990.
40. M. Sebag, C. Rouveirol. Constraint Inductive Logic Programming. In L. De Raedt, ed., *Advances in Inductive Logic Programming*, pp. 277–294, IOS Press, 1996.
41. A. Silberschatz, A. Tuzhilin. On Subjective Measure of Interestingness in Knowledge Discovery. In *Proc. First International Conference on Knowledge Discovery and Data Mining (KDD)*, pp. 275–281, 1995.
42. E. Sommer. Rulebase stratifications: An approach to theory restructuring. *Proc. Fourth Int. Workshop on Inductive Logic Programming*, GMD-Studien 237, pp. 377-390, 1994.
43. 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.

44. L. Todorovski, P. Flach, N. Lavrač. Predictive Performance of Weighted Relative Accuracy. In Zighed, D.A., Komorowski, J. and Zytkow, J., editors, *Proc. of the 4th European Conference on Principles of Data Mining and Knowledge Discovery (PKDD2000)*, pp. 255–264, Springer, 2000.
45. 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.
46. S. Wrobel, S. Džeroski. The ILP description learning problem: Towards a general model-level definition of data mining in ILP. In K. Morik and J. Herrmann, eds, *Proc. Fachgruppentreffen Maschinelles Lernen (FGML-95)*, 44221 Dortmund, Univ. Dortmund, 1995.