Relational and Semantic Data Mining
— Invited Talk —

Nada Lavrač [1,2,3(✉)] and Anže Vavpetič[1,2]

[1] Jožef Stefan Institute, Jamova 39,
1000 Ljubljana, Slovenia
[2] Jožef Stefan International Postgraduate School, Jamova 39,
1000 Ljubljana, Slovenia
{nada.lavrac,anze.vavpetic}@ijs.si
[3] University of Nova Gorica, Nova Gorica, Slovenia

Abstract. Inductive Logic Programming (ILP) and Relational Data Mining (RDM) address the task of inducing models or patterns from multi-relational data. One of the established approaches to RDM is propositionalization, characterized by transforming a relational database into a single-table representation. After introducing ILP and RDM, the paper provides an overview of propositionalization algorithms, which have been made publicly available through the web-based ClowdFlows data mining platform. The paper concludes by presenting recent advances in Semantic Data Mining, characterized by exploiting relational background knowledge in the form of domain ontologies in the process of model and pattern construction.

Keywords: Inductive Logic Programming · Relational Data Mining · Semantic Data Mining · Propositionalization

1 Introduction

Standard machine learning and data mining algorithms induce hypotheses in the form of models or propositional patterns learned from a given data table, where one example corresponds to a single row in the table. Most types of propositional models and patterns have corresponding relational counterparts, such as relational classification rules, relational regression trees, relational association rules. Inductive Logic Programming (ILP) [23] and Relational Data Mining (RDM) [4,6] algorithms can be used to induce such relational models and patterns from multi-relational data, e.g., data stored in a relational database.

For relational databases in which data instances are clearly identifiable (the so-called individual-centered representation [7]), various techniques can be used for transforming a relational database into a propositional single-table representation [14]. After performing such a transformation [18], usually named *propositionalization* [12], standard propositional learners can be used, including decision tree and classification rule learners.

F. Calimeri et al. (Eds.): LPNMR 2015, LNAI 9345, pp. 20–31, 2015.
DOI: 10.1007/978-3-319-23264-5_3

The first part of the paper presents a survey of the state-of-the-art propositionalization techniques. Following an introduction to the propositionalization problem and a description of a number of propositionalization methods, we translate and unify the terminology, using a language that should be familiar to an analyst working with relational databases. Furthermore, we provide an empirical comparison of freely available propositionalization algorithms. Finally, we present our approach to making the use of propositionalization algorithms easier for non-experts, as well as making the experiments shareable and repeatable. The freely available state-of-the-art methods discussed in this paper were wrapped as reusable components in the web-based data mining platform ClowdFlows [13], together with the utilities for working with a relational database management system (RDBMS).

The second part of the paper addresses a more recent ILP setting, named *semantic data mining* (SDM), characterized by exploiting relational background knowledge in the form of domain ontologies in the process of model and pattern construction. The development of SDM techniques is motivated by the availability of large amounts of semantically annotated data in all domains of science, and biology in particular, posing requirements for new data mining approaches which need to deal with increased data complexity, the relational character of semantic representations, as well as the reasoning capacities of the underlying ontologies. The paper briefly introduces the task of semantic data mining, followed by a short overview of the state-of-the-art approaches. Finally, the paper presents the Hedwig semantic subgroup discovery algorithm [1,33] developed by the authors of this paper.

The paper is structured as follows. Section 2 gives an introduction to the propositionalization task, describes the state-of-the-art methods, and presents a number of reusable propositionalization workflows implemented in the ClowdFlows data mining platform. In Sect. 3 we introduce the SDM task, a quick state-of-the-art overview, and a recent semantic subgroup discovery approach Hedwig. Section 4 concludes the paper with a brief summary.

2 Propositionalization

Propositional representations (a single table format) impose the constraint that each training example is represented as a single fixed-length tuple. Due to the nature of some relational problems, there exists no elegant propositional encoding; for example, a citation network in general cannot be represented in a propositional format without loss of information, since each author can have any number of co-authors and papers. The problem is naturally represented using multiple relations, e.g., including the *author* and the *paper* relations.

Problems characterized by multiple relations can be tackled in two different ways: (1) by using a relational learner such as Progol [22] or Aleph [30], which can build a model or induce a set of patterns directly, or (2) by constructing complex relational features used to transform the relational representation into a propositional format and then applying a propositional learner on the transformed single-table representation. In this paper we focus on the latter approach,

called *propositionalization*. Propositionalization is a form of *constructive induction*, since it involves changing the representation for learning. As we noted before, propositionalization cannot always be done without loss of information, but it can be a powerful method when a suitable relational learner is not available, when a non-conventional ILP task needs to be performed on data from a given relational database (e.g., clustering), and when the problem at hand is *individual-centered* [7]. Such problems have a clear notion of an individual and the learning occurs only at the level of (sets of) individual instances rather than the (network of) relationships between the instances. As an example consider the problem of classifying authors into research fields given a citation network; in this case the author is an individual and learning occurs at the author level, i.e. assigning class labels to authors, rather than classifying the authors in terms of their citations in the citation network of other authors.

To illustrate the propositionalization scenario, consider a simplified multi-relational problem, where the data to be mined is a database of authors and their papers, with the task of assigning a research field to unseen authors. In essence, a complete propositional representation of the problem (shown in Table 1) would be a set of queries $q \in Q$ (complex relational features) that return value *true* or *false* for a given author. Each query describes a property of an author. The property can involve a rather complex query, involving multiple relations as long as that query returns either true or false. For example, a query could be "does author X have a paper published at the ECML/PKDD conference?".

Table 1. A sample propositional representation of authors table.

Author	q_1	q_2	...	q_m	Class
A1	1	1	...	1	C_1
A2	0	1	...	0	C_1
A3	1	0	...	0	C_2
...	...	...	...	...	...
An	0	1	0	0	C_1

While this transformation could be done by hand by a data miner, we are only interested in automated propositionalization methods. Furthermore, the transformation into a propositional representation can be done with essentially any ML or DM task in mind: classification, association discovery, clustering, etc.

2.1 Relational Data Mining Task Formulation

A relational data mining task can be formally defined as follows.

Given:

- evidence E (examples, given extensionally),
- an initial theory B (background knowledge, given extensionally or as sets of clauses over the set of background relations).

Find:

- a theory H (hypothesis, in the form of a set of logical clauses) that together with B explains the target properties of E.

where the target property can be a selected class label (the target class) or some other property of interest.

This is a typical ILP definition of the problem, given that numerous existing approaches to relational data mining and propositionalization were developed within the field of ILP. However, since real-world data is in most cases stored in some Relational Database Management System (RDBMS), we try to unify the terminology used across various approaches to be as familiar as possible also to researchers working with databases—which are likely the ones most interested in propositionalization techniques. The definition of a relational data mining task using a more conventional database terminology is given below.

Given:

– target table t, where each row is one example,
– related tables T, connected to t via foreign keys.

Find:

– a query Q (a set of sub-queries) that together with T describes the target properties of t.

In the rest of this paper we will focus on the classification (and subgroup discovery) tasks with a clear notion of the target property of interest (a selected class label), since we can effectively compare different approaches via the performance of the resulting classifier. Using propositionalization to tackle classification tasks must involve two independent steps: (1) preparing a single-table representation of the input database, and (2) applying a propositional learner on that table. In contrast, learners that directly use the multi-relational representations intertwine feature construction and model construction. In propositionalization, these two steps are separated. The workload of finding good features (which have large coverage of instances, or which best separate between different classes) is done by the propositionalization algorithm, while the work of combining these features to produce a good classification model is offloaded to the propositional learner.

The actual art of propositionalization is to generate a number of good, potentially complex features (binary queries), to be evaluated as *true* or *false* for each individual, which the learner will use to construct a classifier. In the model construction phase, the learner exploits these queries about each individual as features used to construct the model. For example, if a decision tree model is constructed, each node in the tree will contain a single query, with the two values (*true* and *false*) on the outgoing branches of this node. Note that propositionalization is not limited only to binary features—many approaches (e.g., [15] and [11]) also use aggregation functions to calculate feature values.

To classify unseen individuals, the classifier must then evaluate the queries that are found in the decision tree nodes on the unseen example and follow the branches according to their answers to arrive at a classification in the leave of the decision tree.

2.2 Overview of Propositionalization Algorithms

The best known propositionalization algorithms are first briefly described, followed by the experimental evaluation of the ones which are publicly available.

LINUS [18] is one of the first propositionalization approaches. It generates features that do not allow recursion and newly introduced variables. The second limitation is more serious and means that the queries cannot contain joins. An improvement of LINUS is SINUS [19] which incorporates more advanced feature construction techniques inspired by feature construction implemented in 1BC [7].

Aleph [30] is an ILP toolkit with many modes of functionality: learning theories, feature construction, incremental learning, etc. In this paper we are interested in its feature construction facility which can be used as a tool for propositionalization. Aleph uses mode declarations to define the syntactic bias. Input relations are defined as Prolog clauses: either extensionally or intensionally.

RSD [36] is a relational subgroup discovery algorithm composed of two main steps: the propositionalization step and the subgroup discovery step. The output of the propositionalization step can be used also as input to other propositional learners. RSD effectively produces an exhaustive list of first-order features that comply with the user-defined mode constraints, similar to those of Progol [22] and Aleph [30]. Furthermore, RSD features satisfy the connectivity requirement, which imposes that no feature can be decomposed into a conjunction of two or more features. Mode declarations define the algorithm's syntactic bias, i.e. the space of possible features.

HiFi [17] is a propositionalization approach that constructs first-order features with hierarchical structure. Due to this feature property, the algorithm performs the transformation in polynomial time of the maximum feature length. Furthermore, the resulting features are the smallest in their semantic equivalence class. The algorithm is shown to perform several orders of magnitude faster than RSD for higher feature lengths.

RelF [16] constructs a set of tree-like relational features by combining smaller conjunctive blocks. The novelty is that RelF preserves the monotonicity of feature reducibility and redundancy (instead of the typical monotonicity of frequency), which allows the algorithm to scale far better than other state-of-the-art propositionalization algorithms.

RELAGGS [15], which stands for *rel*ational *agg*regation, is a propositionalization approach that uses the input relational database schema as a basis for a declarative bias and it aims to use optimization techniques usually used in relational databases (e.g., indexes). Furthermore, the approach employs aggregation functions in order to summarize non-target relations with respect to the individuals in the target table.

Stochastic propositionalization [12] employs a search strategy similar to random mutation hill-climbing: the algorithm iterates over generations of individuals, which are added and removed with a probability proportional to the fitness of individuals, where the fitness function used is based on the Minimum Description Length (MDL) principle.

Safarii [11] is a commercial multi-relation data mining tool.[1] It offers a unique pattern language that merges ILP-style structural descriptions as well as

[1] http://www.kiminkii.com/safarii.html.

aggregations. Safarii comes with a tool called ProSafarii, which offers several pre-processing utilities—including propositionalization via aggregation.

Wordification [26,27] is a propositionalization method inspired by text mining, which can be seen as a transformation of a relational database into a corpus of text documents. Wordification aims at constructing simple, easy to understand features, acting as words in the transformed Bag-Of-Words representation.

Extensive description of the experimental evaluation of the available propositionalization algorithm is presented in [28]. Fully reproducing the experimental results is outside the scope of this paper. The evaluation of different propositionalization approaches was performed on binary classification tasks using seven datasets from five different relational domains. The Friedman test [8] using significance level $\alpha = 0.05$ and the corresponding Nemenyi post-hoc test [24] were applied. This evaluation approach was used as an alternative to the t-test, which is proven to be inappropriate for testing multiple algorithms on multiple datasets [5]. A birds's eye view of the results is shown in Fig. 1.

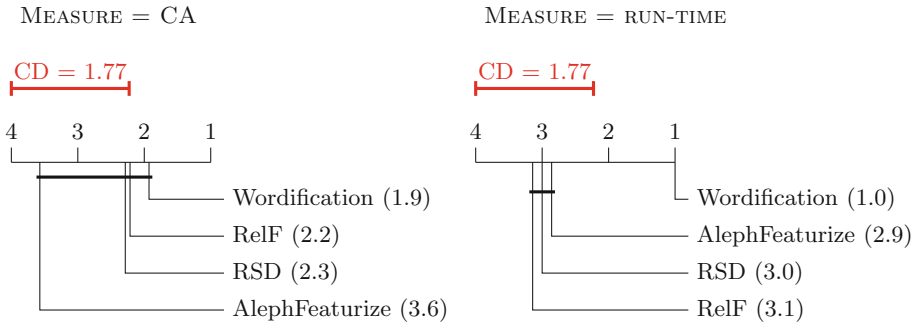

Fig. 1. Critical distance diagram for the reported classification accuracy (left; not enough evidence to prove that any algorithm performs better) and run-time (right; significant differences for $\alpha = 0.05$) results. The numbers in parentheses are the average ranks.

The statistical test was first performed using the J48 decision tree learner for classification accuracy and run-time. For classification accuracy, there is not enough evidence to prove that any propositionalization algorithm on average performs better than the others (Fig. 1 left, for significance level $\alpha = 0.05$), even though wordification achieves the best results on five out of seven benchmarks. We repeated the same statistical analysis for the LibSVM results, where the conclusion ended up the same. For run-time, however, the results are statistically significant in favor of wordification; see the critical distance diagram in the right part of Fig. 1. The diagram tells us that the wordification approach performs statistically significantly faster than other approaches, under the significance level $\alpha = 0.05$. Other approaches fall within the same critical distance and no statistically significant difference was detected.

2.3 ILP in the ClowdFlows Platform

The ClowdFlows platform [13] is an open-source, web-based data mining platform that supports the construction and execution of scientific workflows. This web application can be accessed and controlled from anywhere while the processing is performed in a cloud of computing nodes. A public installation of ClowdFlows is accessible at http://clowdflows.org. For a developer, the graphical user interface supports simple operations that enable workflow construction: adding workflow components (widgets) on a canvas and creating connections between the components to form an executable workflow, which can be shared by other users or developers. Upon registration, the user can access, execute, modify, and store the modified workflows, enabling their sharing and reuse. On the other hand, by using anonymous login, the user can execute a predefined workflow, while any workflow modifications would be lost upon logout.

We have extended ClowdFlows with the implementation of an ILP toolkit, including the popular ILP system Aleph [30] together with its feature construction component, as well as RSD [36], RelF [16] and Wordification [26] propositionalization engines. Construction of RDM workflows is supported by other specialized RDM components (e.g., the MySQL package providing access to a relational database by connecting to a MySQL database server), other data mining components (e.g., the Weka [34] classifiers) and other supporting components (including cross-validation), accessible from other ClowdFlows modules. Each public workflow is assigned a unique URL that can be accessed by any user to either repeat the experiment, or use the workflow as a template to design another workflow. Consequently, the incorporated RDM algorithms become handy to use in real-life data analytics, which may therefore contribute to improved accessibility and popularity of ILP and RDM.

Figure 2 shows some of the implemented ILP workflows using ILP and Weka module components. The first workflow assumes that the user uploads the files required by RSD as Prolog programs. Workflows constructed for the other three propositionalization approaches Aleph, RelF and Wordification, which are also made publicly available, assume that the training data is read from a MySQL database.

In terms of workflows reusability, accessible by a single click on a web page where a workflow is exposed, the implemented propositionalization toolkit is a significant step towards making the ILP legacy accessible to the research community in a systematic and user-friendly way. To the best of our knowledge, this is the only workflow-based implementation of ILP and RDM algorithms in a platform accessible through a web browser, enabling simple workflow adaptation to the user's needs. Moreover, the ILP toolkit widgets actually use a Python library called `python-rdm` which is available on GitHub[2]. The authors welcome extensions and improvements from the community.

[2] https://github.com/anzev/rdm/.

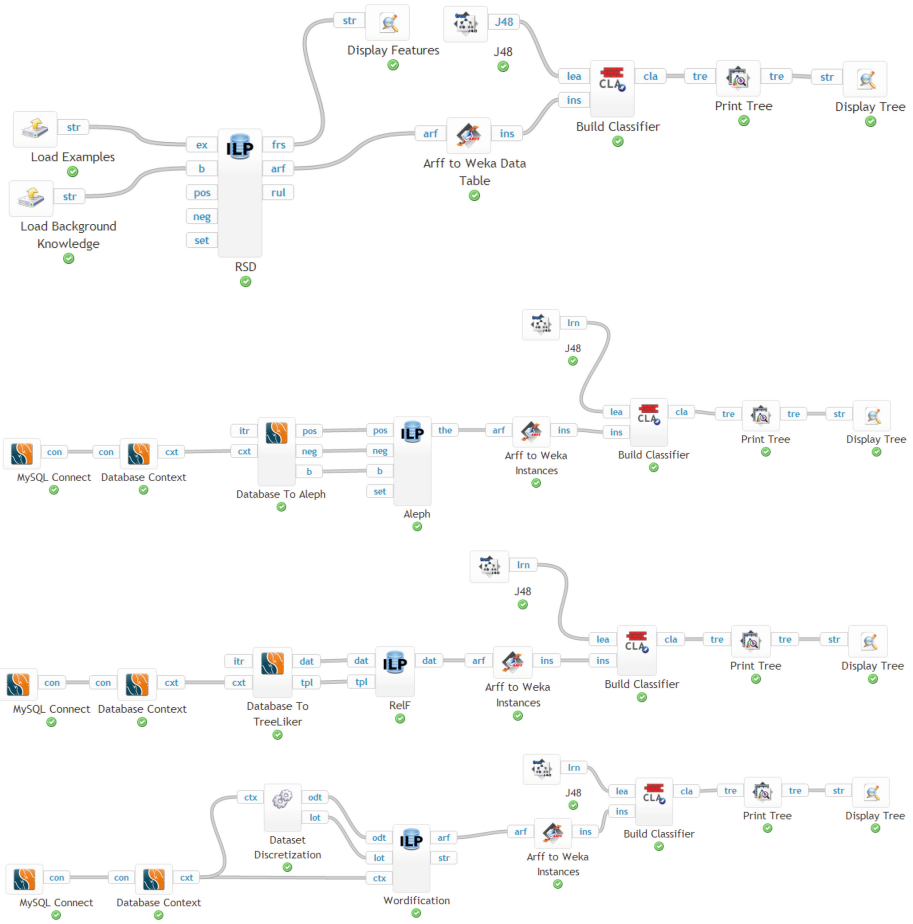

Fig. 2. First: RSD propositionalization workflow using ILP and Weka components is available online at http://clowdflows.org/workflow/471/ (the same RSD workflow, extended by accessing the training data using a MySQL database, is available at http://clowdflows.org/workflow/611/). Second: Aleph workflow available at http://clowdflows.org/workflow/2224/. Third: RelF workflow available at http://clowdflows.org/workflow/2227/. Fourth: Wordification workflow available at http://clowdflows.org/workflow/2222/.

3 Semantic Data Mining

Rule learning, which was initially focused on building predictive models formed of sets of classification rules, has recently shifted its focus to descriptive pattern mining. Well-known pattern mining techniques in the literature are based on association rule learning [2,29]. While the initial studies in association rule mining have focused on finding interesting patterns from large datasets in an unsupervised setting, association rules have been used also in a supervised setting, to learn pattern

descriptions from class-labeled data [20]. Building on top of the research in classification and association rule learning, subgroup discovery has emerged as a popular data mining methodology for finding patterns in the class-labeled data. Subgroup discovery aims at finding interesting patterns as sets of individual rules that best describe the target class [10,35].

Subgroup descriptions in the form of propositional rules are suitable descriptions of groups of instances. However, given the abundance of taxonomies and ontologies that are readily available, these can also be used to provide higher-level descriptors and explanations of discovered subgroups. Especially in the domain of systems biology the GO ontology [3], KEGG orthology [25] and Entrez gene–gene interaction data [21] are good examples of structured domain knowledge that can be used as additional higher-level descriptors in the induced rules.

The challenge of incorporating the domain ontologies in data mining was addressed in recent research on semantic data mining (SDM) [32]. See Fig. 3 for a diagram of the SDM process.

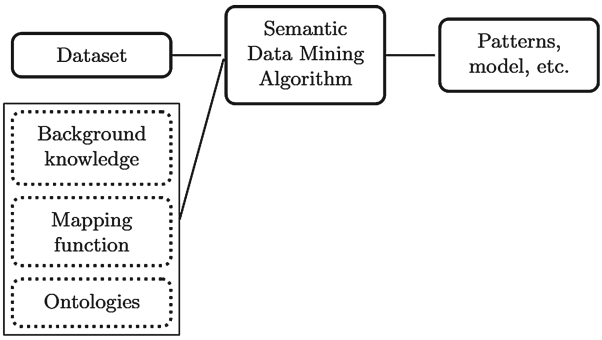

Fig. 3. The Semantic Data Mining (SDM) process illustration.

In [32] we described and evaluated the SDM toolkit that includes two semantic data mining systems: SDM-SEGS and SDM-Aleph. SDM-SEGS is an extension of the earlier domain-specific algorithm SEGS [31] which allows for semantic subgroup discovery in gene expression data. SEGS constructs gene sets as combinations of GO ontology [3] terms, KEGG orthology [25] terms, and terms describing gene–gene interactions obtained from the Entrez database [21]. SDM-SEGS extends and generalizes this approach by allowing the user to input any set of ontologies in the OWL ontology specification language and an empirical data collection which is annotated by domain ontology terms. SDM-SEGS employs ontologies to constrain and guide the top-down search of a hierarchically structured space of induced hypotheses. SDM-Aleph, which is built using the popular inductive logic programming system Aleph [30] does not have the limitations of SDM-SEGS, imposed by the domain-specific algorithm SEGS, and can accept any number of OWL ontologies as background knowledge which is then used in the learning process.

Based on the lessons learned in [32], we introduced a new system Hedwig in [33]. The system takes the best from both SDM-SEGS and SDM-Aleph. It uses a search mechanism tailored to exploit the hierarchical nature of ontologies. Furthermore, Hedwig can take into account background knowledge in the form of RDF triplets. Compared to [33], the current version of the system uses better redundancy pruning and significance tests based on [9]. Furthermore, the new version also support negations of unary predicates. Apart from the financial domain in [33], the approach was also applied on a multi-resolution dataset of chromosome abberrations in [1].

Hedwig is open-source software available on GitHub[3] and the authors welcome improvements from the community.

4 Conclusions

This paper addresses two lines of research of the authors, the propositionalization approach and the semantic data mining approach to RDM.

First, ILP and RDM are introduced, together with an overview of popular propositionalization algorithms. Next, the paper briefly presents the results of an experimental comparison of several such algorithms on several relational databases. These approaches have been made available through the web-based ClowdFlows data mining platform, together with repeatable and reusable workflows. The paper concludes by presenting recent advances in Semantic Data Mining, characterized by exploiting relational background knowledge in the form of domain ontologies in the process of model and pattern construction.

In further work, we will combine ILP and RDM approaches with the approaches developed in the network mining community, to address open challenges in linked data and heterogeneous information network analysis.

Acknowledgments. This work was supported by the Slovenian Ministry of Higher Education, Science and Technology [grant number P2-0103], the Slovenian Research Agency [grant number PR-04431], and the SemDM project (Development and application of new semantic data mining methods in life sciences) [grant number J2-5478].

References

1. Adhikari, P.R., Vavpetič, A., Kralj, J., Lavrač, N., Hollmén, J.: Explaining mixture models through semantic pattern mining and banded matrix visualization. In: Džeroski, S., Panov, P., Kocev, D., Todorovski, L. (eds.) DS 2014. LNCS, vol. 8777, pp. 1–12. Springer, Heidelberg (2014)
2. Agrawal, R., Srikant, R.: Fast algorithms for mining association rules in large databases. In: Bocca, J.B., Jarke, M., Zaniolo, C. (eds.) Proceedings of the 20th International Conference on Very Large Data Bases, pp. 487–499. Morgan Kaufmann Publishers Inc., San Francisco (1994)

[3] https://github.com/anzev/hedwig.

3. Gene Ontology Consortium: the Gene Ontology project in 2008. Nucleic Acids Res. **36**(Database-Issue), 440–444 (2008)
4. De Raedt, L.: Logical and relational learning. In: Zaverucha, G., da Costa, A.L. (eds.) SBIA 2008. LNCS (LNAI), vol. 5249, pp. 1–1. Springer, Heidelberg (2008)
5. Demšar, J.: Statistical comparison of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)
6. Džeroski, S., Lavrač, N. (eds.): Relational Data Mining. Springer, Heidelberg (2001)
7. Flach, P.A., Lachiche, N.: 1BC: a first-order Bayesian classifier. In: Džeroski, S., Flach, P.A. (eds.) ILP 1999. LNCS (LNAI), vol. 1634, pp. 92–103. Springer, Heidelberg (1999)
8. Friedman, M.: The use of ranks to avoid the assumption of normality implicit in the analysis of variance. J. Am. Stat. Assoc. **32**, 675–701 (1937)
9. Hämäläinen, W.: Efficient search for statistically significant dependency rules in binary data. Ph.D. thesis, Department of Computer Science, University of Helsinki, Finland (2010)
10. Klösgen, W.: Explora: a multipattern and multistrategy discovery assistant. In: Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P., Uthurusamy, R. (eds.) Advances in Knowledge Discovery and Data Mining, pp. 249–271. American Association for Artificial Intelligence, Menlo Park (1996)
11. Knobbe, A.J. (ed.): Multi-Relational Data Mining. Frontiers in Artificial Intelligence and Applications, vol. 145. IOS Press, Amestardam (2005)
12. Kramer, S., Pfahringer, B., Helma, C.: Stochastic propositionalization of non-determinate background knowledge. In: Page, D.L. (ed.) ILP 1998. LNCS, vol. 1446. Springer, Heidelberg (1998)
13. Kranjc, J., Podpečan, V., Lavrač, N.: ClowdFlows: a cloud based scientific workflow platform. In: Flach, P.A., De Bie, T., Cristianini, N. (eds.) ECML PKDD 2012, Part II. LNCS, vol. 7524, pp. 816–819. Springer, Heidelberg (2012)
14. Krogel, M.-A., Rawles, S., Železný, F., Flach, P.A., Lavrač, N., Wrobel, S.: Comparative evaluation of approaches to propositionalization. In: Horváth, T., Yamamoto, A. (eds.) ILP 2003. LNCS (LNAI), vol. 2835, pp. 197–214. Springer, Heidelberg (2003)
15. Krogel, M.-A., Wrobel, S.: Transformation-based learning using multirelational aggregation. In: Rouveirol, C., Sebag, M. (eds.) ILP 2001. LNCS (LNAI), vol. 2157, pp. 142–155. Springer, Heidelberg (2001)
16. Kuželka, O., Železný, F.: Block-wise construction of tree-like relational features with monotone reducibility and redundancy. Mach. Learn. **83**(2), 163–192 (2011)
17. Kuželka, O., Železný, F.: Hifi: tractable propositionalization through hierarchical feature construction. In: Železný, F., Lavrač, N. (eds.) Late Breaking Papers, the 18th International Conference on Inductive Logic Programming (2008)
18. Lavrač, N., Džeroski, S., Grobelnik, M.: Learning nonrecursive definitions of relations with LINUS. In: Kodratoff, Y. (ed.) EWSL 1991. LNCS, vol. 482, pp. 265–281. Springer, Heidelberg (1991)
19. Lavrač, N., Flach, P.A.: An extended transformation approach to Inductive Logic Programming. ACM Trans. Comput. Logic **2**(4), 458–494 (2001)
20. Liu, B., Hsu, W., Ma, Y.: Integrating classification and association rule mining. In: Proceedings of the 4th International Conference on Knowledge Discovery and Data mining (KDD 1998), pp. 80–86. AAAI Press, August 1998
21. Maglott, D., Ostell, J., Pruitt, K.D., Tatusova, T.: Entrez Gene: gene-centered information at NCBI. Nucleic Acids Res. **33**(Database issue), D54–D58 (2005)

22. Muggleton, S.: Inverse entailment and Progol. New Gener. Comput. **13**(3–4), 245–286 (1995). Special issue on Inductive Logic Programming
23. Muggleton, S. (ed.): Inductive Logic Programming. Academic Press, London (1992)
24. Nemenyi, P.B.: Distribution-free multiple comparisons. Ph.D. thesis (1963)
25. Ogata, H., Goto, S., Sato, K., Fujibuchi, W., Bono, H., Kanehisa, M.: KEGG: kyoto encyclopedia of genes and genomes. Nucleic Acids Res. **27**(1), 29–34 (1999)
26. Perovšek, M., Vavpetič, A., Cestnik, B., Lavrač, N.: A wordification approach to relational data mining. In: Fürnkranz, J., Hüllermeier, E., Higuchi, T. (eds.) DS 2013. LNCS, vol. 8140, pp. 141–154. Springer, Heidelberg (2013)
27. Perovšek, M., Vavpetič, A., Lavrač, N.: A wordification approach to relational data mining: early results. In: Riguzzi, F., Železný, F. (eds.) ILP 2012 Proceedings of Late Breaking Papers of the 22nd International Conference on Inductive Logic Programming, Dubrovnik, Croatia, 17–19 September 2012. CEUR Workshop Proceedings, vol. 975, pp. 56–61. CEUR-WS.org (2012)
28. Perovšek, M., Vavpetič, A., Kranjc, J., Cestnik, B., Lavrač, N.: Wordification: propositionalization by unfolding relational data into bags of words. Expert Syst. Appl. **42**(17–18), 6442–6456 (2015)
29. Piatetsky-Shapiro, G.: Discovery, analysis, and presentation of strong rules. In: Piatetsky-Shapiro, G., Frawley, W.J. (eds.) Knowledge Discovery in Databases. AAAI/MIT Press, Menlo Park (1991)
30. Srinivasan, A.: Aleph manual, March 2007. http://www.cs.ox.ac.uk/activities/machinelearning/Aleph/
31. Trajkovski, I., Lavrač, N., Tolar, J.: SEGS: search for enriched gene sets in microarray data. J. Biomed. Inform. **41**(4), 588–601 (2008)
32. Vavpetič, A., Lavrač, N.: Semantic subgroup discovery systems and workflows in the SDM-toolkit. Comput. J. **56**(3), 304–320 (2013)
33. Vavpetič, A., Novak, P.K., Grčar, M., Mozetič, I., Lavrač, N.: Semantic data mining of financial news articles. In: Fürnkranz, J., Hüllermeier, E., Higuchi, T. (eds.) DS 2013. LNCS, vol. 8140, pp. 294–307. Springer, Heidelberg (2013)
34. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques, 3rd edn. Morgan Kaufmann, Amsterdam (2011)
35. Wrobel, S.: An algorithm for multi-relational discovery of subgroups. In: Komorowski, J., Żytkow, J.M. (eds.) PKDD 1997. LNCS, vol. 1263, pp. 78–87. Springer, Heidelberg (1997)
36. Železný, F., Lavrač, N.: Propositionalization-based relational subgroup discovery with RSD. Mach. Learn. **62**(1–2), 33–63 (2006)