In *Multi-Relational Data Mining Workshop at KDD-2002.*

Towards Structural Logistic Regression: Combining Relational and Statistical Learning

Alexandrin Popescul[1], Lyle H. Ungar[1],
Steve Lawrence[2], and David M. Pennock[2]

[1] University of Pennsylvania
Department of Computer and Information Science
{popescul,ungar}@unagi.cis.upenn.edu
[2] NEC Research Institute
{dpennock,lawrence}@research.nj.nec.com

Abstract. Inductive logic programming (ILP) techniques are useful for analyzing data in multi-table relational databases. Learned rules can potentially discover relationships that are not obvious in "flattened" data. Statistical learners, on the other hand, are generally not constructed to search relational data; they expect to be presented with a single table containing a set of feature candidates. However, statistical learners often yield more accurate models than the logical forms of ILP, and can better handle certain types of data, such as counts. We propose a new approach which integrates structure navigation from ILP with regression modeling. Our approach propositionalizes the first-order rules at each step of ILP's relational structure search, generating features for potential inclusion in a regression model. Ideally, feature generation by ILP and feature selection by stepwise regression should be integrated into a single loop. Preliminary results for scientific literature classification are presented using a relational form of the data extracted by ResearchIndex (formerly CiteSeer). We use FOIL and logistic regression as our ILP and statistical components (decoupled at this stage). Word counts and citation-based features learned with FOIL are modeled together by logistic regression. The combination often significantly improves performance when high precision classification is desired.

1 Introduction

The structure of the data being modeled often dictates what model form is most appropriate. For example, when word counts are used for document classification, statistical modeling tools, such as logistic regression, maximum entropy inference or Naive Bayes, are more appropriate. These tools are capable of flexible evidence aggregation. In other situations, for example when using citation structure in document classification, statistical models are commonly either overwhelmed by sparsity or lack the functionality to navigate the underlying relational structure. In such situations, inductive logic programming (ILP) techniques provide functionality to navigate relational structure and generate potentially new forms of evidence, not readily available in a "flattened" one-table representation.

Unfortunately, the use of logic as a representational language makes them too inaccurate in many domains.

We propose an approach where first-order rules being constructed inside ILP's relational structure search loop are propositionalized [3] at each step to generate features which are immediately considered for possible inclusion into a regression model. The regression model may include propositionalized features in binary form (rule satisfied at least once) or as counts (the number of independent ways a training example can satisfy the first-order rule).

In this paper, we start to address the issues of combining ILP and regression by exploring some of the advantages and disadvantages of ILP using FOIL[4] [12, 13] and of logistic regression [9], separately, and in combination, by looking at the task of classifying scientific literature. We use the data from ResearchIndex,[5] an online digital library of computer science papers [1, 11]. It contains a rich set of relational data, including the text of titles, abstracts and documents, citation information, author names and affiliations, conference or journal names, and paper downloads. This data can be naturally represented as multi-relational domain knowledge, e.g. `author_of(author,document)`, `contains_word(document, word)`, `title_contains_word(document, word)`, `cites(document1, document2)` etc. We are also expanding this basic structure with derived clusters such as topics (clusters of words) and communities (clusters of people). For many tasks, it is useful to know, for example, what papers written by frequently cited authors cite a given paper.

Training and testing are performed on twelve document classification tasks using the data from ResearchIndex. We show that relational representation of citation structure results in higher performance than only using the "flat" citation information immediately available in a document. FOIL often achieves high precision levels while heavily compromising recall. Modeling words with logistic regression performs better when high recall is desired. Both can be improved by feature combination within logistic regression when high precision is preferred. Logistic regression modeling provides an additional flexibility through a natural choice of a decision threshold when adjusting the desired level of precision-recall.

2 Task and Data

We define twelve separate binary classification tasks on the ResearchIndex data formulated in relational form.[6] For each task, we:

- Select documents containing a specific query phrase (Table 1). These are included as positive labeled examples.
- Randomly select from the remaining collection three times as many negative examples.

[3] Represented as propositional features.
[4] We use FOIL6.4.
[5] `http://researchindex.org/`
[6] Only documents cited at least once are considered.

- Augment the collection by including all other documents that have incoming or outgoing citations from the positive or negative core documents. The new documents included at this stage participate in learning as unlabeled examples.
- Extract from the documents words and citation information (words used in a query phrase are excluded to avoid selection bias).
- Randomly split positive and negative examples into training and test sets, 2/3 and 1/3 of the total respectively.

Table 1 lists topics and collection sizes.

Table 1. Summary of twelve binary classification tasks

Query phrase	**# of $\oplus$ docs**	**# of all docs**
"association rules"	354	2399
"bayesian networks"	491	3418
"data mining"	1057	8273
"decision trees"	1156	9559
"digital libraries"	669	5149
"graphical models"	368	2561
"inductive logic programming"	280	1850
"information extraction"	713	5410
"knowledge discovery"	547	4061
"machine translation"	498	3543
"maximum entropy"	234	1396
"natural language processing"	1098	9269

The tasks are defined for the initial experimentation. One can include a richer background knowledge by considering documents that are more than one citation away from the positive or negative core documents in the citation graph. The positive and negative classes can have other priors and other validation schemes can be used, e.g. K-fold cross validation.

3 Methodology and Results

Our eventual goal is to use ILP-style search to generate rules, which can then be incorporated into a regression model. Since we want the best features for a regression (rather than a logic) model, this should ideally involve replacing the rule selection criteria used inside of an ILP structure search with feature selection criteria commonly used in regression analysis. We propose using the FOIL-like structure navigation loop, i.e. the search of refinement graphs, and consider the rules at each node of this search graph for inclusion in logistic regression model in a forward stepwise selection manner. As we will see below, a simpler

approach of generating a set of rules using FOIL, creating features from them (i.e., "propositionalizing" them), and feeding those features to a regression is suboptimal in the sense that the greedy search used to select the rules, including the logic-oriented practice of removing the covered positive tuples, often fails to generate the features desired for the regression. Feature generation and selection should be more tightly coupled.

For example, FOIL supplied with both word and citation information tends to learn very few rules involving citations, as words overwhelm the search. However, our experiments presented in later subsections show that citation information can be extremely useful when high precision classification is required. We have not yet implemented a fully integrated system; for now we present preliminary results exploring the circumstances under which pure logic or hybrid logic-regression methods give superior performance. We do this by looking separately at ILP and regression models predicting document class based on words, based on citations, and based on both words and citations.

3.1 Modeling Words and Citations Separately

Words and citations provide two extremely different examples of how logic-based and regression-based models can give different performance. Words tend to indicate the class of a document, but the best word-based models combine information from many words. Almost all documents have some relevant words, so recall is high, even if precision is not perfect. Citations, in contrast, give highly precise classification when they are present. Unfortunately, many documents do not have any citations useful for classification, so recall is often poor. These differences are reflected in the performances of FOIL and logistic regression.

The "classical" approach to document classification assumes a "flat" attribute-value representation of word counts. We have experimented with both FOIL and logistic regression for word-based classification.[7] FOIL uses background relations of type `has_word`*`WORD`*`(X)`, where *`WORD`* is a concrete word in the vocabulary. As expected, the logistic regression resulted in a more powerful predictive model, improving precision by 6.0%. The recall level remained the same, due to the restricted word feature set considered by logistic regression. Table 2 includes the details of the comparisons.

We also used FOIL to learn classification rules based purely on the citation structure of documents.[8] The following relations are considered in FOIL programs learning:

- Target relation:
  - `inclass(X)`
- Background relations:
  - `cites(X,Y)`,
  - `cites_doc`*`ID`*`(X)`,

[7] Here using only the words selected by FOIL.
[8] FOIL is used with its default settings.

- `citedby_doc`*`ID`*`(X)`,

where *ID* is an identification number of a concrete document, X and Y are variables of type *Document*. This representation would have been equivalent to using just one relation `cites(X,Y)`, and declaring all documents as theory constants, except in the situations where FOIL learned simple classification rules of the form `inclass(`*`ID`*`)`. This tended to overfit and produced higher out-of-sample classification errors.

To convince ourselves of the benefits of relational representation compared to using "flat" features alone, i.e. only immediately incoming and outgoing citations, we compared FOIL's performance with and without structural exploration. Predictive accuracy for all twelve tasks improves when structural relations are allowed. In this case FOIL also tended to learn shorter programs (average 162 vs. 124 clauses). Table 2 includes this comparison.

The precision is quite high in both cases (87.1% and 87.2%), more likely statistically the same. The average recall improved 4.6% from 52.7% to 57.3%. This recall level may still be unacceptably low in many applications.

The following is a typical example of learned rules for classification of documents into the class "inductive logic programming":[9]

```
ilp(A) :- cites(A,B), cites_doc222642(B).
ilp(A) :- cites(A,B), citedby_doc102608(B).
ilp(A) :- cites(B,A), cites_doc368053(B).
ilp(A) :- cites(A,B), cites_doc221578(B).
ilp(A) :- cites_doc18992(A).
ilp(A) :- cites(A,B), citedby_doc97299(B).
ilp(A) :- citedby_doc192387(A).
ilp(A) :- cites(A,B), citedby_doc179764(B).
ilp(A) :- cites(A,B), cites_doc180353(B).
ilp(A) :- cites(B,A), cites_doc94985(B).
```

For example, the first rule classifies document A as positive if there exists a document B such that A cites B and B cites document 222642, where document 222642 is a paper by S. Muggleton *Inductive Logic Programming* (1992), in the MIT Encyclopedia of the Cognitive Sciences, which is a very authoritative paper in the field (314 citations as counted by ResearchIndex).[10] The transitive nature of this rule makes it stronger than the "flat" alternative where a document is an ILP document if it directly cites this document,

```
ilp(A) :- cites_doc222642(A).
```

[9] Rules with more than two literals in the body were present in programs learned for other classes. They were relatively rare. Considering a larger citation graph neighborhood is likely to increase the number of more complex rules.

[10] Document ID's in this example correspond to internal ID's of ResearchIndex. The reader can check the details of other papers by entering their ID after the home URL of ResearchIndex, e.g. for document 222642, request `citeseer.nj.nec.com/222642.html`.

Modeling citations with logistic regression was less successful than FOIL due to the extreme sparsity of citation structure. Using propositionalized citation-based features generated by FOIL in logistic regression gives results virtually identical to those in FOIL, as the presence of at least one "strong" FOIL feature was enough numerically for positive classification in this domain.

Bibliometric Interpretation of Citation-Based Rules Some of the citation based rules learned by FOIL have a natural bibliometric interpretation. Bibliometrics studies the development of scientific disciplines by analyzing their citation structure [6, 15].

For example, the following rule discovers an instance of the concept of bibliographic coupling: [11]

```
ilp(A) :- cites(A,B), citedby_doc102608(B).
```

Document 102608 cites several highly authoritative papers in ILP, thus increasing the chance of document A belonging to the ILP class if A and document 102608 are bibliographically coupled, i.e. there exists a document B cited by them both.

The following rule discovers an instance of the concept of co-citation:

```
ilp(A) :- cites(B,A), cites_doc368053(B).
```

Document A is an ILP document if it is co-cited together with document 368053. The latter is a highly cited document in ILP.

3.2 Combination of Word and Citation Features

We model propositionalized citation-based FOIL features and word-counts with logistic regression. The combined model is compared to words-only logistic regression. The improvement at the *default 0.5 decision threshold* [12] is not uniform. The accuracy was improved by an average of 0.60%. The improvement in precision is more obvious (2.37%), while recall remained almost the same on average. The average classification error reduction is 5.1%. Formal statistical significance testing confirms that the improvement in average precision level is significant at the 0.05 level (the 95% confidence interval of the precision improvement is $2.37\% \pm 1.68\%$). The tests failed to show significant differences in average recall and accuracy levels between the two models. Table 2 includes the average performance of the combined model at the default decision threshold.

A more complete comparison requires precision-recall curve analysis. More refined precision-recall compromises may be needed in many applications. Logistic regression modeling provides an added benefit by allowing a natural choice of a tuning mechanism. We present precision-recall curve analysis in the next section.

[11] Bibliographic coupling is the degree of similarity between two documents based on documents cited in common.

[12] Predicted probability of the positive class.

Table 2. Average performance of FOIL and logistic regression on twelve tasks with word and citation features

	FOIL			LR	
	flat cites	**struct. cites**	**words**	**words**	**comb.**
Accuracy, %	86.2	87.3	86.4	88.2	88.8
Precision,%	87.1	87.2	73.8	79.8	82.2
Recall, %	52.7	57.3	70.7	70.8	70.7

4 Precision-Recall Analysis

Often, within a fixed learned model a decision threshold can be varied to achieve a desirable precision level at the expense of recall or vice versa. Logistic regression's predicted probability of a class is a natural choice of a decision threshold. In the previous section the examples are classified as positive if the decision threshold was higher than 0.5 and negative otherwise. Here we vary the decision threshold of both logistic regression models with words only and words-citations features to generate their precision-recall curves.

Unfortunately, FOIL does not offer a natural way to vary the decision threshold to generate the curves. Although search parameters can be tuned, that would correspond to merely learning a different FOIL program and is analogous to selecting a different set of features in logistic regression. Another way was suggested by Craven et al. [4] where an estimated accuracy of the first rule that matches an example is used as its confidence measure. This allows a precision-recall curve generation for recall levels lower than the base level of the full FOIL program, but has no way to tune for achieving higher recall levels. That would require learning a different set of clauses with other learning parameters.

We plot one point for precision-recall of FOIL programs with citation-based features together with the curves for logistic regression models and note that the recall levels of FOIL programs cannot be improved without re-learning with different parameters. Figures 1 and 2 present precision-recall results for our twelve tasks. The results are very dataset-dependent.

Interestingly, the relative position of the citation-based FOIL precision-recall point to the words-only logistic regression curve determines the relative position of the precision-recall curve of words-citations logistic regression. When words already do a better job in that region adding citations hurts. When, on the other hand, the words are weak predictors in the region compared to citations alone, adding citation features to the words based logistic regression results in improvements. In particular, citation features tend to provide significant improvements in high precision areas. In the high recall area, where the words are "stronger" features, adding citations almost always hurts.

We *speculate* that the degree of connectivity of the communities corresponding to each of the twelve datasets determines whether adding citation-based features helps. As the precision-recall analysis shows, the improvement in perfor-

mance greatly depends on how "strong" the citation-based features are relative to words. The twelve datasets represent communities with different organization. Some are very well defined, e.g. "inductive logic programming", whereas others are rather a collection of several weaker connected communities, e.g. "knowledge discovery". Citation-based features are more useful in well-connected datasets, especially when words alone fail to do a good job. We expect our fully integrated approach to be able to make better choices regarding the inclusion and relative merits of words and citations in different tasks.

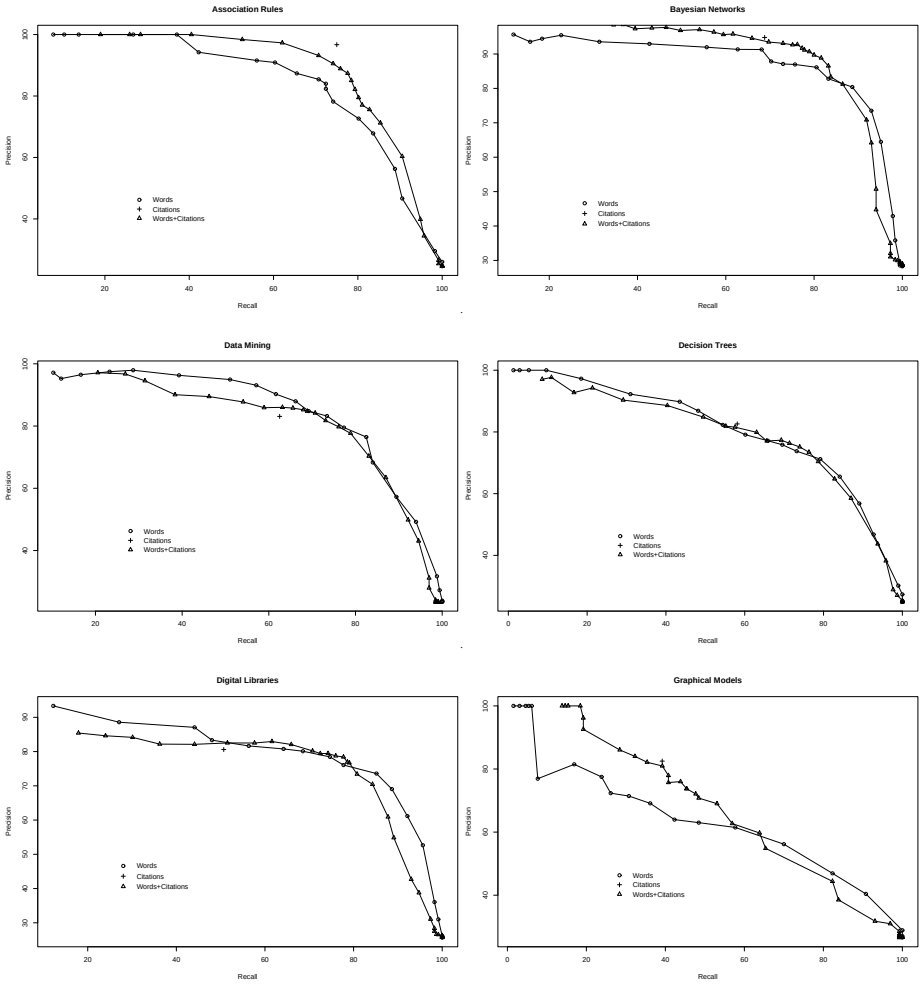

Fig. 1. Precision-Recall (datasets 1-6)

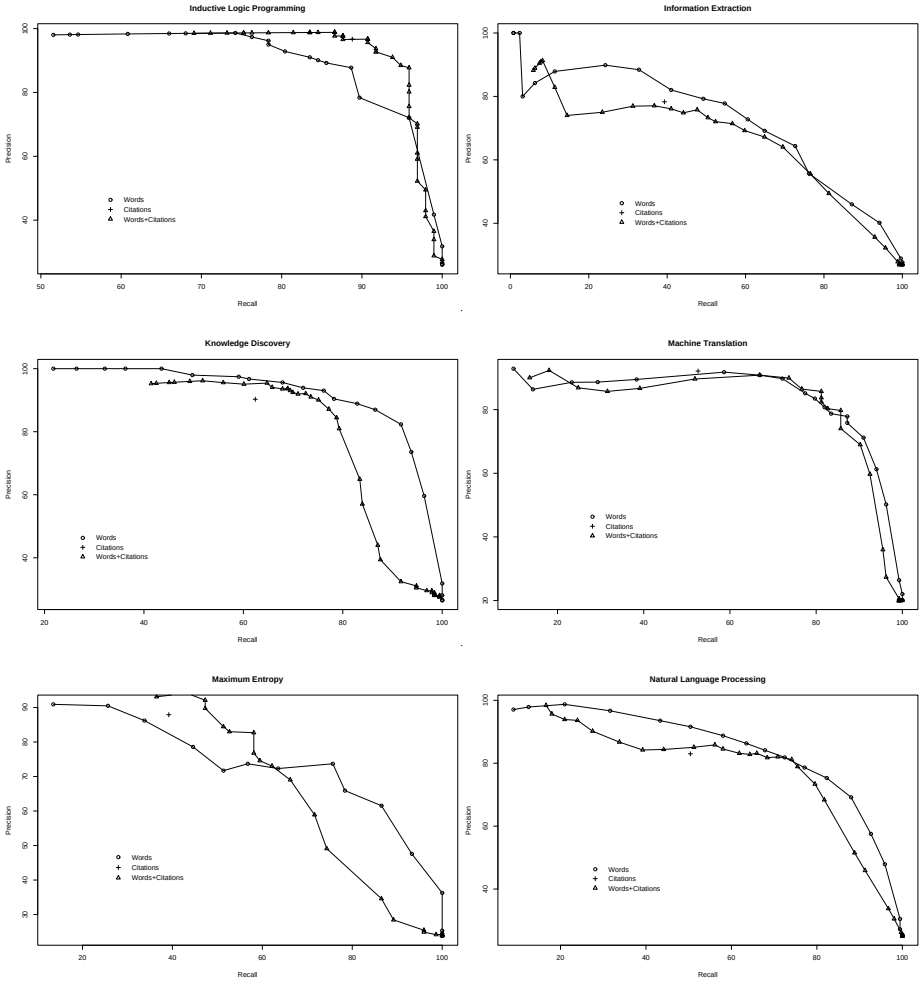

Fig. 2. Precision-Recall (datasets 7-12)

5 Related Work

Representing induced first-order rules as features to be used in propositional learning is known as "propositionalization". Kramer et al. [10] provide a review of the methodology and its applications. The right-hand sides of relational rules can be used as binary features in any modeling tool appropriate for this representation. Propositionalization for linear regression modeling was used by Srinivasan and King [16] to build predictive models in a chemical domain. Decoupling the process assumes the inductive bias of a technique used to construct features. Pure propositionalization is not always applicable when the data is not well suitable for logic-based learning, as in the case of word counts in document classification. Simply flattening relational data before constructing a model also presents problems, Getoor et al. [7] acknowledge this and propose one solution in the context of learning Bayesian networks.

ILP algorithms have been applied to document classification by Cohen [3] to exploit word-order relations in text. Craven et al. [4, 14] propose a technique called *"statistical predicate invention"* to combine statistical and relational learning in the hypertext domain. In "statistical predicate invention", word-based classifications produced by Naive Bayes are included in FOIL search as new predicates. Our approach differs in the "direction" the combination takes place. It has a statistical technique as a modeling component for which the features are supplied by an ILP-based relational structure search. "Statistical predicate invention", on the other hand, preserves the ILP component as the central modeling component and calls statistical modeling from within the inner structure navigation loop to supply new predicates.

Various other techniques have been applied to learning hypertext classifiers. These vary in methodology and information exploited. For example, Chakrabarti et al. [2] use predicted labels of neighboring documents to reinforce classification decisions for a given document. Glover et al. [8] provide an analysis of the utility of text in citing documents for classification. Yang et al. [17] include a more complete discussion of hypertext classification methods as well as a systematic comparison.

6 Discussion and Future Work

We believe that ILP and statistical modeling should be integrated so that structure navigation and regression modeling are a single, integrated process. This should bring the strengths of both methods to bear on the feature generation and selection problem. In the results presented above, we used logistic regression with FOIL-supplied features in a relatively loosely coupled manner. For scientific literature classification, word-based and citation-based features give quite different performances with the different techniques; combining them appropriately gives further improvement, but only in some regimes. We found that using propositionalized citation-based structural features learned by FOIL along with word counts in logistic regression often significantly improves performance when

high precision is required. This, even though FOIL trained on both words and citations fails to learn the same features, and hence fails to get the higher accuracy. Blindly generating features with FOIL and putting them into a regression does not work as well as, for example, generating more features than FOIL would normally generate, and then selecting them in the regression.

We expect that many other tasks can benefit from the approach presented above. For example, the language used in patent descriptions may not be as specific as that in scientific publications, providing room for greater benefit from incorporating citation-based features into classification models. Predicting whether a document will cite another document is another potential application where richer relational structure should help, if implemented in conjunction with regression. Documents are cited based on many criteria, including topic (words), conference or journal, and who the authors of the papers are. All attributes contribute, some in fairly complex ways.

Using clustering or latent class modeling in the Structural Logistic Regression setting should also prove highly beneficial, since one of the problems that is endemic to the use of structural information in document modeling is sparsity. Clusters can generate rich relational structure [5]. For example, a word is a member of one or more word-clusters (topics). Each of these word clusters, in turn, has automatically generated properties such as "most frequently occurring word in cluster". Thus a relation such as `most-frequent-word(main-topic(most-frequent-word(document-231)))` could be learned, as could relations involving sets of frequent words. Interleaving the generation of logic-based rules, the creation of clusters, and the selection of features based on logic and clusters has the potential to produce extremely rich and powerful models.

References

1. Kurt Bollacker, Steve Lawrence, and C. Lee Giles. Discovering relevant scientific literature on the web. *IEEE Intelligent Systems*, 15(2):42–47, March/April 2000.
2. Soumen Chakrabarti, Byron E. Dom, and Piotr Indyk. Enhanced hypertext categorization using hyperlinks. In Laura M. Haas and Ashutosh Tiwary, editors, *Proceedings of the ACM International Conference on Management of Data, SIGMOD-98*, pages 307–318. ACM Press, 1998.
3. William Cohen. Learning to classify English text with ILP methods. In L. De Raedt, editor, *Advances in Inductive Logic Programming*, pages 124–143. IOS Press, 1995.
4. M. Craven and S. Slattery. Relational learning with statistical predicate invention: Better models for hypertext. *Machine Learning*, 43(1/2):97–119, 2001.
5. Dean Foster and Lyle Ungar. A proposal for learning by ontological leaps. In *Proceedings of Snowbird Learning Conference*, Snowbird, Utah, 2002.
6. Eugene Garfield. *Citation indexing: Its theory and application in science, technology, and humanities.* Wiley, New York, 1979. ISBN 089495024X.
7. L. Getoor, N. Friedman, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In Saso Dzeroski and Nada Lavrac, editors, *Relational Data Mining.* Springer-Verlag, 2001.

8. Eric Glover, Kostas Tsioutsiouliklis, Steve Lawrence, David Pennock, and Gary Flake. Using web structure for classifying and describing web pages. In *International World Wide Web Conference*, Honolulu, Hawaii, May 7–11 2002.
9. D.W. Hosmer and S. Lemeshow. *Applied logistic regression.* Wiley, New York, 1989.
10. Stefan Kramer, Nada Lavrac, and Peter Flach. Propositionalization approaches to relational data mining. In Saso Dzeroski and Nada Lavrac, editors, *Relational Data Mining*, pages 262–291. Springer-Verlag, 2001.
11. Steve Lawrence, C. Lee Giles, and Kurt Bollacker. Digital libraries and autonomous citation indexing. *IEEE Computer*, 32(6):67–71, 1999.
12. J.R. Quinlan and R.M. Cameron-Jones. FOIL: A midterm report. In *Proceedings of the 6th European Conference on Machine Learning*, pages 3–20, 1993.
13. J.R. Quinlan and R.M. Cameron-Jones. Induction of logic programs: FOIL and related systems. *New Generation Computing*, 13:287–312, 1995.
14. Sean Slattery and Mark Craven. Combining statistical and relational methods for learning in hypertext domains. In *Proceedings of the 8th International Conference on Inductive Logic Programming, ILP-98*, pages 38–52, 1998.
15. H. Small and B. Griffith. The structure of scientific literatures: Identifying and graphing specialities. *Science Studies*, 4(17):17–40, 1974.
16. A. Srinivasan and R. King. Feature construction with inductive logic programming: A study of quantitative predictions of biological activity aided by structural attributes. *Data Mining and Knowledge Discovery*, 3(1):37–57, 1999.
17. Y. Yang, S. Slattery, and R. Ghani. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*, 18(2), 2002.