In *Proceedings of IEEE International Conference on Data Mining, ICDM-2003.*

Statistical Relational Learning for Document Mining

Alexandrin Popescul
Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104 USA
popescul@cis.upenn.edu

Lyle H. Ungar
Computer and Information Science
University of Pennsylvania
Philadelphia, PA 19104 USA
ungar@cis.upenn.edu

Steve Lawrence*
Google
2400 Bayshore Parkway
Mountain View, CA 94043 USA
lawrence@google.com

David M. Pennock*
Overture Services, Inc.
74 N. Pasadena Ave., 3rd floor
Pasadena, CA 91103 USA
david.pennock@overture.com

Abstract

A major obstacle to fully integrated deployment of many data mining algorithms is the assumption that data sits in a single table, even though most real-world databases have complex relational structures. We propose an integrated approach to statistical modeling from relational databases. We structure the search space based on "refinement graphs", which are widely used in inductive logic programming for learning logic descriptions. The use of statistics allows us to extend the search space to include richer set of features, including many which are not boolean. Search and model selection are integrated into a single process, allowing information criteria native to the statistical model, for example logistic regression, to make feature selection decisions in a step-wise manner. We present experimental results for the task of predicting where scientific papers will be published based on relational data taken from CiteSeer. Our approach results in classification accuracies superior to those achieved when using classical "flat" features. The resulting classifier can be used to recommend where to publish articles.

1. Introduction

Statistical learning techniques play an important role in data mining, however, their standard formulation is almost exclusively limited to a one table domain representation. Such algorithms are presented with a set of candidate features, and a model selection process then makes decisions regarding their inclusion into a model. Thus, the process of feature generation is decoupled from modeling, often being performed manually. This stands as a major obstacle to the fully integrated application of such modeling techniques in real-world practice where data is most often stored in relational form. It is often not obvious which features will be relevant, and the human effort to fully explore the richness of a domain is often too costly. Thus, it is crucial to provide statistical modeling techniques with an integrated functionality to navigate richer data structures to discover potentially new and complex sources of relevant evidence.

We present a form of statistical relational learning which integrates regression with feature generation from relational data. In this paper we use logistic regression, giving a method we call Structural Logistic Regression (SLR). SLR combines the strengths of classical statistical modeling with the high expressivity of features, both boolean and real-valued, automatically generated from a relational database. SLR falls into a family of models proposed in Inductive Logic Programming (ILP) called "upgrades" [23]. Upgrades extend propositional learners to handle relational representation. An upgrade implies that modeling and relational structure search are integrated into a single process dynamically driven by the assumptions and model selection criteria of a propositional learner used. This contrasts with another approach proposed in ILP called "propositionalization" [21]. Propositionalization generally implies a decoupling of relational feature construction and modeling. It has certain disadvantages compared to upgrading, as it is difficult to decide a priori what features will be useful. Upgrading techniques let their learning algorithms select their own features with their own criteria. In large problems it is impossible to "exhaustively" propositionalize, and the fea-

* Work conducted at NEC Laboratories America, Inc., 4 Independence Way, Princeton, NJ 08540 USA.

ture construction should be driven dynamically at the time of learning. An extreme form of propositionalization is generating the full join of a database. This is both impractical and incorrect—the size of the resulting table is prohibitive, and the notion of a training instance is lost, now being represented by multiple rows. Moreover, the entries in the full join table will be atomic attribute values, rather than richer features resulting from complex queries.

We apply SLR to document mining in (hyper)-linked domains. Linked document collections, such as the Web, patent databases or scientific publications are inherently relational, noisy and sparse. The characteristics of such domains form a good match with our method: i) links between documents suggest relational representation and ask for techniques being able to navigate such structures; "flat" file domain representation is inadequate in such domains; ii) the noise in available data sources suggests statistical rather than deterministic approaches, and iii) often extreme sparsity in such domains requires a focused feature generation and their careful selection with a discriminative model, which allows modeling of complex, possibly deep, but local regularities rather than attempting to build a full probabilistic model of the entire domain.

SLR integrates logistic regression with relational feature generation. We formulate the feature generation process as search in the space of relational database queries, based on the top-down search of refinement graphs widely used in ILP [10]. The use of statistics allows modeling of real-valued features, and instead of treating each node of the graph as a logic formula, we treat it as a database query resulting in a table of all satisfying solutions and the introduction of aggregate operators resulting in a richer set of features. We use statistical information criteria during the search to dynamically determine which features are to be included into the model. The language of non-recursive first-order logic formulas has a direct mapping to SQL and relational algebra, which can be used as well for the purposes of our discussion, e.g. as we do in [29]. In large applications SQL should be preferred for efficiency and database connectivity reasons.

We use the data from CiteSeer, an online digital library of computer science papers [24]. CiteSeer contains a rich set of relational data, including citation information, the text of titles, abstracts and documents, author names and affiliations, and conference or journal names, which we represent in relational form (Section 2). We report results for the task of paper classification into conference/journal classes.

The next section introduces the CiteSeer relational domain and defines the learning tasks. The methodology is presented in Section 3. Section 4 presents the experimental results demonstrating that relational features significantly improve classification accuracies. We provide an extended discussion of related work in Section 5. Section 6 concludes the paper with discussion and future work directions.

2. Task and Data

The experimental task explored here is the classification of scientific papers into categories defined by the conference or journal in which they appear. Publication venues vary in size and topic coverage, possibly overlapping, but do focus around a particular scientific endeavor, broader or narrower. This task can be viewed as a variant of community classification. Communities are not only defined by topical commonalities through document content but also by interaction between community members in the form of citations, co-authorship, and publishing in the same venues.

The data for our experiments was taken from CiteSeer [24]. CiteSeer catalogs scientific publications available in full-text on the web in PostScript and PDF formats. It extracts and matches citations to produce a browsable citation graph. Documents in CiteSeer were matched with the DBLP database (`http://dblp.uni-trier.de/`) to extract their publication venues. The domain contains rich sources of information which we represent in relational form. The following schema is used, where capitalized words indicate the type of a corresponding attribute:

$published\_in(Document,\ Venue)$,
$cites(Document,\ Document)$,
$word\_count(Document,\ Word,\ Count)$,
$title\_word(Document,\ Word)$,
$author\_of(Document,\ Person)$.

We define a number of separate binary classification tasks. The choice of binary rather than multi-class tasks is dictated in this paper by the use of (two-class) logistic regression which comes with readily available model selection functionality. A multi-class classifier, if equipped with model selection, can be used for more general multi-class tasks. For each classification task here, we:

- Select a pair of conferences or journals.
- Split the documents of two classes 50/50 into training and test core sets.
- Include documents that are cited by or cite the core documents (citation graph neighborhood).
- Extract relation $cites$ for all core and just added documents in the citation graph neighborhood.
- Exclude from the training background knowledge any reference to the test set core documents.
- For all documents extract the remaining background knowledge: $word\_count$, $title\_word$, $author\_of$, and $published\_in$. Since, in the case of core class documents, the last relation is the actual answer, we allow

it to be used only when relating to the venues of linked documents (Section 4).[1]

3. Methodology

SLR couples two main processes: generation of features from relational data and their selection with statistical model selection criteria. Figure 1 highlights the main components of our learning setting. Relational feature generation is a search problem. It requires formulation of the search in the space of relational database queries. Our structuring of the search space is based on the formulation widely used in inductive logic programming for learning logic descriptions, and extended to include other types of queries as the use of statistics relaxes the necessity of limiting the search space to only boolean values. The process results in a statistical model where each selected feature is the evaluation of a database query encoding a predictive data pattern in a given domain.

Logistic regression is a discriminative model, that is, it models conditional class probabilities without attempting to model marginal distributions of features. Model parameters are learned by maximizing a conditional likelihood function. Regression coefficients linearly combine features in a "logit" transformation resulting in values in the [0,1] interval, interpreted as probabilities of a positive class. More complex models will results in higher likelihood values, but at some point will likely overfit the data, resulting in poor generalization. A number of criteria aiming at striking the balance between optimizing the likelihood of training data and model complexity have been proposed. Among the more widely used are the Akaike Information Criterion (AIC) [1] and the Bayesian Information Criterion (BIC) [33]. These statistics work by penalizing the likelihood by a term that depends on model complexity (i.e. the number of selected features). AIC and BIC differ in this penalty term, which is larger for BIC in all but very small data sets of only several examples, and resulting in smaller models and better generalization performance. A model selection process selects a subset of available predictors with the goal of learning a more generalizable model.

Relational feature generation is a search problem. We use top-down search of refinement graphs [34, 10], and introduce aggregate operators into the search to extend the feature space. The extension is possible when using a statistical modeling which allows modeling of real-valued features in addition to only boolean values used in logic.

The top-down search of refinement graphs starts with the most general rule and specializes it producing more specific ones. The search space is constrained by specifying a language of legal clauses, for example by disallowing negations and recursive definitions, and is structured by imposing a partial ordering on such clauses through the syntactic notion of generality. This defines the refinement graph as a directed acyclic graph, where each node is a clause. The nodes are expanded from the most general to more specific clauses by applying a refinement operator. The refinement operator is a mapping from a clause to a set of its most general specializations. This is achieved by applying two syntactic operations: i) a single variable substitution, or ii) an addition of a new relation to the body of the clause, involving one or more of variables already present, and possibly introducing new variables. We make typing control with meta-schema, that is, we disallow bindings of variables of different types (e.g. $Document$ is never attempted to match with $Word$). Figure 2 presents a fragment of the search space in our domain.

We extend the search space by treating bodies of clauses not as $true/false$ values, but rather as queries resulting in a table of all satisfying solutions. These tables are aggregated to produce scalar numeric values to be used as features in statistical learners. The use of refinement graphs to search over database queries rather than over the first-order formulas retains richer information. In our approach, numeric attributes are always left unbound (substitutions for numeric attributes are disallowed). This avoids certain numeric reasoning limitations known to exist when learning first-order logic rules. Consider a refinement graph node referring to a learning example d:

$$word\_count(d,\ logistic,\ C).$$

Its evaluation will produce a one cell table for each d. Evaluation for all such training examples produces a column of counts of the word "logistic". The query

$$cites(d, D),\ word\_count(D,\ logistic,\ C)$$

will produce, for each training example d, a table of pairs of cited document IDs with their counts of the word "logistic". Averaging the column of counts produces the average count of the word "logistic" in cited documents.

An algebra for aggregations is necessary. Although there is no limit to the number of aggregate operators one may try, for example square root of the sum of column values, logarithm of their product etc., we find a few of them to be particularly useful. We propose the aggregate operators typically used in SQL: $size$, ave, min, max, and $empty$. Aggregations can be applied to a whole table or to individual columns, as appropriate given type restrictions. We use the following notation to denote aggregate query results: $function_{Var}\ [query]$, where $function$ is an aggregate operator, its subscript Var is a variable in the query specifying the aggregated column. If an aggregate operator applies to the whole table rather than an individual column, the subscript is omitted.

[1] Word counts are extracted from the first 5k of document text. The words are normalized with the Porter stemmer. Stop words are removed.

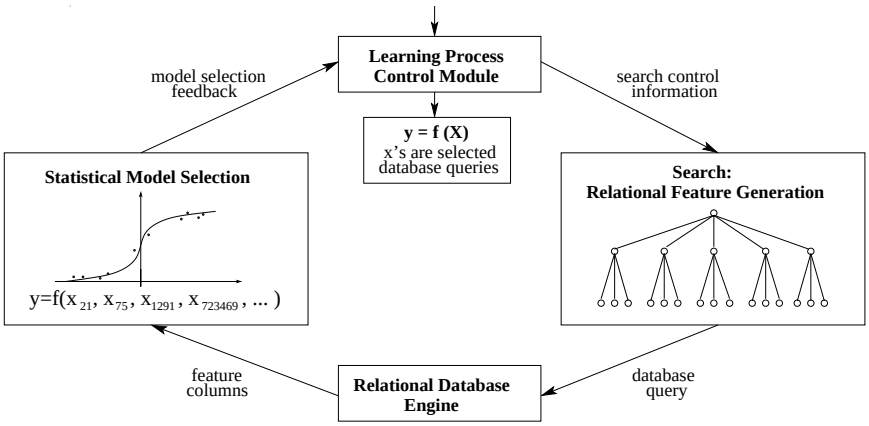

Figure 1. The search in the space of database queries involving one or more relations produces feature candidates one at a time to be considered by the statistical model selection component.

For example, the number of times a training example d is cited is $size[cites(D,\ d)]$. The average count of the word "learning" in documents cited from d is:

$$ave_C\ [cites(d,\ D),\ word\_count(D,\ learning,\ C)].$$

Once the search space is structured, a search strategy should be chosen. Here we use breadth-first search. This will not always be feasible in larger domains, and intelligent search heuristics will be needed. Incorporating statistical modeling into the search, as we do, opens a number of attractive options such as sampling, providing a well understood statistical interpretation of feature usefulness.

4. Results

We demonstrate the use of Structural Logistic Regression (SLR) by predicting publication venues of articles from CiteSeer. We select five binary classification tasks. Four tasks are composed by pairing both KDD and WWW with their two closest conferences in the immediate citation graph neighborhood as measured by the total number of citations connecting the two conferences. SIGMOD and VLDB rank the top two for both KDD and WWW. The fifth task is a pair of AI and ML Journals (Artificial Intelligence, Elsevier Science and Machine Learning, Kluwer respectively). Table 1 contains the sizes of classes and their immediate citation graph neighborhoods. The number of papers analyzed is subject to availability in both CiteSeer and DBLP.

The results reported here are produced by searching the subspace of the search graph over the queries involving one or two relations, where in the latter case one relation is $cites$. At present, we avoid subspaces resulting in a quadratic (or larger) number of features, such as co-authorship or word co-occurrences in a document. The exploration of such larger subspaces as an important future direction (e.g. sampling can be used). We do not consider queries involving the incoming citations to the learning core class documents as we assume that this knowledge is missing for the test examples. Since the relation $published\_in$ duplicates the response class labels, we allow it to participate in the search only as part of more complex queries relating to the venues of cited documents.

Table 1. Sizes of classes and their citation graph neighborhoods.

CLASS	# CORE DOCS	# NEIGHBORS
ARTIF. INTELLIGENCE	431	9,309
KDD	256	7,157
MACHINE LEARNING	284	5,872
SIGMOD	402	7,416
VLDB	484	3,882
WWW	184	1,824

Model selection is performed in two phases: preselection and final model selection. We allow the first phase to be more inclusive and make more rigorous model selection at the final phase. First, the features generated during the search are checked for addition into the model by the AIC. This phase performs a forward-only step-wise selection. A feature is preselected if it improves the AIC by at least 1%. After every 500 search nodes, the model is refreshed. All preselected features participate in the final selection phase with a forward-backward stepwise procedure. A more restrictive BIC statistic is used at this phase. The preselection phase is currently used to remove the ordering bias, which may favor shallow features.

We compare the performance of the resulting models to those trained using only "flat" features. Flat features only involve the data available immediately in a learning exam-

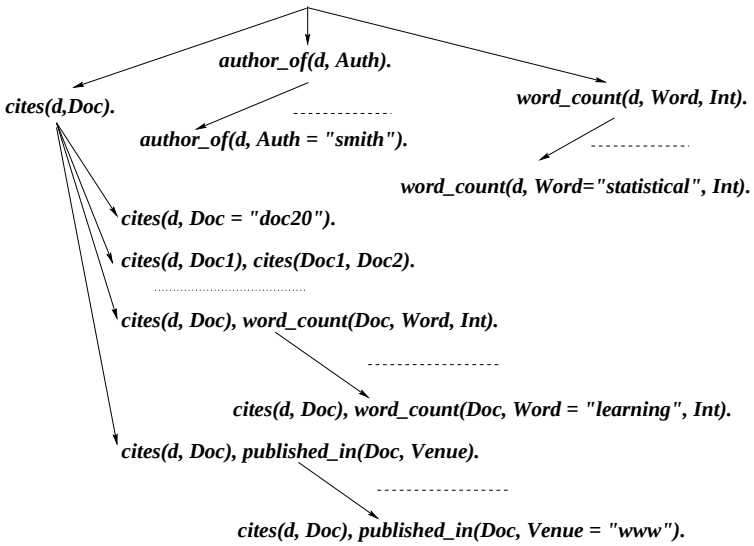

Figure 2. Fragment of the search space. Each node is a database query about a learning example d evaluated to a table of satisfying bindings. Aggregate operators are used to produce scalar features.

ple, that is authorship, text and title words. They do not involve data referring to or contained in other documents. We refer the reader to [30] for evidence supporting logistic regression in comparisons to pure or propositionalized FOIL modeling in this domain. We do not compare to other "flat" components; they can also be used within our approach, if supplied with a model selection mechanism. We focus instead on demonstrating the usefulness of more complex relational features as part of a common feature generation framework. Comparing logistic regression to other propositional models would be attempting to answer a different question. A common conclusion made in large-scale comparison studies, e.g. [25], is that there is no single best algorithm across all different datasets. However, logistic regression is often quite successful.

The models we found had average test set accuracies of 83.5% and 80.2% for relational and flat representations respectively. The 3.3 percentage points improvement is significant at the 99% confidence level based on the standard t-test. The improvement from relational features was achieved in all tasks. Table 2 details the performance in each task. Table 3 summarizes for each task the final number of selected features, the number of preselected features, and the total number of features considered. The total number of features considered is a function of vocabulary selection and of the density of citation graph neighborhoods. We store in the database word counts greater than two for a given document. Authors of only one paper in a data set are not recorded, nor are the non-core class documents linked to the rest of the collection by only one citation.

Table 4 gives examples of some of the highly significant

Table 2. Training and test sets accuracy (selection with BIC).

TASK	TRAIN		TEST	
	REL.	FLAT	REL.	FLAT
WWW - SIGMOD	95.2	90.3	**79.3**	77.5
WWW - VLDB	93.5	90.6	**85.1**	83.2
KDD - SIGMOD	99.1	90.4	**83.3**	79.8
KDD - VLDB	97.8	91.2	**87.7**	83.7
AI - ML	93.8	86.5	**82.2**	76.7

relational features learned. Not all types of relational features were equally useful in terms of how often they were selected into the models. Commonly selected features are based on word counts in cited documents and cited publication venues. Authorship or features involving citations to concrete documents were selected relatively infrequently; their utility would increase when other sources of features are unavailable, for example, when the words are not available or in multi-lingual environments.

5. Related Work and Discussion

A number of approaches "upgrading" propositional learners to relational representations have been proposed in the inductive logic programming (ILP) community. Often, these approaches upgrade learners most suitable to binary attributes, for example, decision trees and association rules [4, 9]. The upgrade of classification and regression trees is proposed in [22]. Reinforcement learning was extended to relational representations [11]. Upgrading usu-

Table 3. Number of selected and generated features.

TASK	IN FINAL MODEL (BIC)		PRESELECTED		TOTAL CONSIDERED	
	REL.	FLAT	REL.	FLAT	REL.	FLAT
WWW - SIGMOD	17	18	588	138	79,878	16,230
WWW - VLDB	15	19	498	105	85,381	16,852
KDD - SIGMOD	20	13	591	122	83,548	16,931
KDD - VLDB	22	15	529	111	76,751	15,685
AI - ML	21	19	657	130	85,260	17,402

Table 4. Examples of selected relational features (p-values are less than 0.05; using BIC).

TASK	FEATURE	SUPPORTS CLASS
WWW - SIGMOD	$not\_empty\ [cites(d,\ D),\ published\_in(D,\ icde)]$	SIGMOD
	$size\ [cites(d,\ D),\ word\_count(D,\ page,\ C)]$	WWW
	$not\_empty\ [cites(d,\ D),\ word\_count(D,\ memori,\ C)]$	SIGMOD
WWW - VLDB	$size\ [cites(d,\ D),\ title\_word(D,\ join)]$	VLDB
	$not\_empty\ [cites(d,\ D),\ published\_in(D,\ www)]$	WWW
	$size\ [cites(d,\ D),\ word\_count(D,\ relat,\ C)]$	VLDB
KDD - SIGMOD	$ave_C\ [cites(d,\ D),\ word\_count(D,\ mine,\ C)]$	KDD
	$not\_empty\ [cites(d, D),\ word\_count(D,\ dbm,\ C)]$	SIGMOD
	$not\_empty\ [cites(d, D),\ cites(doc81863,\ D)]$	KDD
KDD - VLDB	$ave_C\ [cites(d,\ D),\ word\_count(D,\ learn,\ C)]$	KDD
	$not\_empty\ [cites(d,\ D),\ published\_in(D,\ icml)]$	KDD
	$not\_empty\ [cites(d,\ D),\ published\_in(D,\ kdd)]$	KDD
AI - ML	$not\_empty\ [cites(d,\ D),\ author\_of(D,\ schapire)]$	ML
	$ave_C\ [cites(d,\ D),\ word\_count(D,\ exampl,\ C)]$	ML
	$not\_empty\ [cites(d,\ D),\ published\_in(D,\ ijcai)]$	AI

ally implies that generation of relational features and their modeling are tightly coupled and driven by propositional learner's model selection criteria. SLR falls into the category of upgrading methods. Perhaps the approach most similar in spirit to ours is that taken in MACCENT system [8], which uses expected entropy gain from adding binary features to a maximum entropy classifier to direct a beam search over first-order clauses. They determine when to stop adding variables by testing the classifier on a held-out data set. A detailed discussion of upgrading is presented in [23]. ILP provides one way to structure the search space; others can be used [31].

Another approach is "propositionalization" [21], in which, as the term is usually used, features are first constructed from relational representation and then presented to a propositional algorithm. Feature generation is thus fully decoupled from the model used to make predictions. One form of propositionalization is to learn a logic theory with an ILP rule learner and then use it as binary features in a propositional learner. For example, linear regression is used to model features constructed with ILP to build predictive models in a chemical domain [35]. Aggregate operators are an attractive way of extending the set of propositionalized features. For example, aggregates can be used to construct a single table involving aggregate summaries and then using a standard propositional learner on this table [20]. Aggregation in relational learning is discussed in detail in [28].

Decoupling feature construction from modeling, as in propositionalization, however, retains the inductive bias of the technique used to construct features, that is, better models potentially could be built if one allowed the propositional learner itself to select its own features based on its own criteria. First Order Regression System [18] more closely integrates feature construction into regression, but does so using a FOIL-like covering approach for feature construction. Additive models, such as logistic regression, have different criteria for feature usefulness; integrating feature generation and selection into a single loop is advocated in this context in [3, 30]. Coupling feature generation and model selection can also significantly reduce computational cost. By fully integrating a rich set of aggregate operators into the generation and search of the refinement graph, SLR avoids costly generation of features which will not be tested for inclusion in the model.

A number of models have been proposed which combine the expressivity of first-order logic with probabilistic semantics [2, 14, 19, 26, 32, 36]. For example, Stochastic Logic Programs [26] model uncertainty from within the ILP framework by providing logic theories with a probability distribution; Probabilistic Relational Models (PRMs) [14] are a relational upgrade of Bayesian networks. The marriage of richer representations and probability theory yields

extremely powerful formalisms, and inevitably a number of equivalences among them can be observed. In addition to the question of semantic and representational equivalence, it is useful to also consider the differences in how models are built, that is, what objective function is optimized, what algorithm is used to optimize that function, what is done to avoid over-fitting, what simplifying assumptions are made.

A conflict exists between the two goals: i) probabilistically characterizing the entire domain and ii) building a model that addresses a specific question, such as classification or regression modeling of a single response variable. This distinction typically leads to two philosophies: "generative" and "discriminative" modeling. Generative models attempt to model feature distributions, while discriminative models, like SLR, only model the distribution of the response variable conditional on the features, thus vastly reducing the number of parameters which must be estimated. For this reason, our method allows inclusion of arbitrarily complex features without estimating their distribution, which is impossible in large and sparse environments.

PRMs, for example, are generative models of joint probability distribution capturing probabilistic influences between entities and their attributes in a relational domain. PRMs can provide answers to a large number of possible questions about the domain. An important limitation, however, of generative modeling is that often there is not enough data to reliably estimate the entire model. Generative modeling does not allow focusing the search for complex features arbitrarily deep. One can achieve superior performance by focusing only on a particular question, for example, class label prediction, and training models discriminatively to answer that question. Relational Markov Networks (RMNs) [36] address this problem since they are trained discriminatively. In RMNs, however, the structure of a learning domain, which determines which direct interactions are explored, is prespecified by a relational template, which precludes the discovery of deeper and more complex regularities advocated in this paper.

Learning from relational data brings new challenges. Relational correlations invalidate independence assumptions. This can be addressed explicitly by quantifying such relational correlations and learning to control for them [17], or for example, by using random effects structures to model relational dependencies [16]. Here, we address the problem of relational correlations implicitly by generating more complex features automatically. In the presence of a large number of feature candidates, selecting the right features can eliminate the independence violation by making the observations conditionally independent given these features.

Various techniques have been applied to learning hypertext classifiers. For example, predicted labels of neighboring documents can be used to reinforce classification decisions for a given document [5]. An iterative technique based on a Bayesian classifier that uses high confidence inferences to improve class inferences for linked objects at later iterations is proposed in [27]. A technique called Statistical Predicate Invention [7] which combines statistical and relational learning by using Naive Bayes classifications as predicates in FOIL has been applied to hypertext classification. Joint probabilistic models of document content and connectivity have also been used for document classification [6]. The text surrounding hyperlinks pointing to a given document was found to greatly improve text classification accuracy [13], and the so-called "extended anchortext" in citing documents has been used for classification and document description [15].

6. Conclusions and Future Work

We presented Structural Logistic Regression (SLR), an approach for statistical relational learning. It allows learning accurate predictive models from large relational databases. Modeling and feature selection is integrated into the search over the space of database queries generating feature candidates involving complex interactions among objects in a given database.

SLR falls into the category of upgrade methods proposed in inductive logic programming. Upgrades integrate a propositional learner into a search of relational features, where propositional learners feature selection criteria dynamically drive the process. We demonstrate the advantages of coupling a rich SQL-based extension of Horn clauses with classical statistical modeling for document classification. SLR extends beyond standard ILP approaches, allowing generation of richer features, better control of search of the feature space, and more accurate modeling in the presence of noise. On the other hand, SLR differs from relational probabilistic network models, such as PRMs and RMNs, because these network models, while being good at handling uncertainly, have not been used successfully to learn complex new relationships. Our approach is easily extended to other regression models.

Our experimental results show the utility of SLR for document classification in the CiteSeer domain, which includes citation structure, textual evidence, paper authorship and publication venues. Relational features improved classification accuracy in all tasks. The average improvement of 3.3 percentage points over already high accuracies achieved by models using only flat features is statistically significant at the 99% confidence level.

Our approach is designed to scale to large data sets, and thus generates features by searching SQL queries rather than Horn clauses, and uses statistical rather than ad hoc methods for deciding which features to include into the model. SQL encourages the use of a much richer feature space, including many aggregates which produce real val-

ues, rather than the more limited boolean features produced by Horn clauses. SQL is preferable to Prolog for efficiency reasons, as it incorporates many optimizations necessary for scaling to large problems. Also, statistical feature selection allows rigorous determination of what information about current regression models should be used to select the subspaces of the query space to explore. Further information such as sampling from feature subspaces to determine their promise and use of database meta-information will help scale SLR to truly large problems.

We are also working on using clustering to extend the set of relations generating new features. Clusters improve modeling of sparse data, improve scalability, and produce richer representations [12]. New cluster relations can be derived using attributes in other relations. As the schema is expanded, the search space grows rapidly, and intelligent search and feature selection become even more critical.

References

[1] H. Akaike. Information theory and an extension of the maximum likelihood principle. In *2nd Int'l Symposium on Information Theory*, 1973.

[2] C. Anderson, P. Domingos, and D. Weld. Relational Markov models and their application to adaptive web navigation. In *KDD*, 2002.

[3] H. Blockeel and L. Dehaspe. Cumulativity as inductive bias. In *Workshops: Data Mining, Decision Support, Meta-Learning and ILP at PKDD*, 2000.

[4] H. Blockeel and L. D. Raedt. Top-down induction of logical decision trees. *Artificial Intelligence*, 101(1-2), 1998.

[5] S. Chakrabarti, B. E. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *SIGMOD*, 1998.

[6] D. Cohn and T. Hofmann. The missing link - A probabilistic model of document content and hypertext connectivity. In *NIPS*, volume 13. MIT Press, 2001.

[7] M. Craven and S. Slattery. Relational learning with statistical predicate invention: Better models for hypertext. *Machine Learning*, 43(1/2), 2001.

[8] L. Dehaspe. Maximum entropy modeling with clausal constraints. In *ILP*, 1997.

[9] L. Dehaspe and H. Toivonen. Discovery of frequent datalog patterns. *Data Mining and Knowledge Discovery*, 3(1), 1999.

[10] S. Dzeroski and N. Lavrac. An introduction to inductive logic programming. In S. Dzeroski and N. Lavrac, editors, *Relational Data Mining*. Springer-Verlag, 2001.

[11] S. Dzeroski, L. D. Raedt, and H. Blockeel. Relational reinforcement learning. In *ICML*, 1998.

[12] D. Foster and L. Ungar. A proposal for learning by ontological leaps. In *Snowbird Learning Conference*, 2002.

[13] J. Furnkranz. Exploiting structural information for text classification on the WWW. *Intelligent Data Analysis*, 1999.

[14] L. Getoor, N. Friedman, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In S. Dzeroski and N. Lavrac, editors, *Relational Data Mining*. Springer-Verlag, 2001.

[15] E. J. Glover, K. Tsioutsiouliklis, S. Lawrence, D. M. Pennock, and G. Flake. Using web structure for classifying and describing web pages. In *Int'l WWW Conference*, 2002.

[16] P. Hoff. Random effects models for network data. In *National Academy of Sciences: Symposium on Social Network Analysis for National Security*, 2003.

[17] D. Jensen and J. Neville. Linkage and autocorrelation cause feature selection bias in relational learning. In *ICML*, 2002.

[18] A. Karalic and I. Bratko. First order regression. *Machine Learning*, 26, 1997.

[19] K. Kersting and L. D. Raedt. Towards combining inductive logic programming and Bayesian networks. In *ILP*, 2001.

[20] A. J. Knobbe, M. D. Haas, and A. Siebes. Propositionalisation and aggregates. In *PKDD*. 2001.

[21] S. Kramer, N. Lavrac, and P. Flach. Propositionalization approaches to relational data mining. In S. Dzeroski and N. Lavrac, editors, *Relational Data Mining*. Springer-Verlag, 2001.

[22] S. Kramer and G. Widmer. Inducing classification and regression trees in first order logic. In S. Dzeroski and N. Lavrac, editors, *Relational Data Mining*. Springer-Verlag, 2001.

[23] W. V. Laer and L. D. Raedt. How to upgrade propositional learners to first order logic: A case study. In S. Dzeroski and N. Lavrac, editors, *Relational Data Mining*. Springer-Verlag, 2001.

[24] S. Lawrence, C. L. Giles, and K. Bollacker. Digital libraries and autonomous citation indexing. *IEEE Computer*, 32(6), 1999.

[25] T.-S. Lim, W.-Y. Loh, and Y.-S. Shih. A comparison of prediction accuracy, complexity, and training time of thirty-three old and new classification algorithms. *Machine Learning*, 40, 2000.

[26] S. Muggleton. Stochastic logic programs. In *ILP*, 1995.

[27] J. Neville and D. Jensen. Iterative classification in relational data. In *AAAI Workshop on Learning Statistical Models from Relational Data*, 2000.

[28] C. Perlich and F. Provost. Aggregation-based feature invention and relational concept classes. In *KDD*, 2003.

[29] A. Popescul and L. H. Ungar. Statistical relational learning for link prediction. In *IJCAI Workshop on Learning Statistical Models from Relational Data*, 2003.

[30] A. Popescul, L. H. Ungar, S. Lawrence, and D. M. Pennock. Towards structural logistic regression: Combining relational and statistical learning. In *KDD Workshop on Multi-Relational Data Mining*, 2002.

[31] D. Roth and W. Yih. Relational learning via propositional algorithms. In *IJCAI*, 2001.

[32] T. Sato. A statistical learning method for logic programs with distribution semantics. In *ICLP*, 1995.

[33] G. Schwartz. Estimating the dimension of a model. *The Annals of Statistics*, 6(2), 1979.

[34] E. Shapiro. *Algorithmic Program Debugging*. MIT, 1983.

[35] 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), 1999.

[36] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *UAI*, 2002.