Probabilistic Models of Text and Link Structure for Hypertext Classification

Lise Getoor
Computer Science Dept.
Stanford University
Stanford, CA 94305
getoor@cs.stanford.edu

Eran Segal
Computer Science Dept.
Stanford University
Stanford, CA 94305-9010
erans@cs.stanford.edu

Ben Taskar
Computer Science Dept.
Stanford University
Stanford, CA 94305-9010
btaskar@cs.stanford.edu

Daphne Koller
Computer Science Dept.
Stanford University
Stanford, CA 94305
koller@cs.stanford.edu

Abstract

Most text classification methods treat each document as an independent instance. However, in many text domains, documents are linked and the topics of linked documents are correlated. For example, web pages of related topics are often connected by hyperlinks and scientific papers from related fields are commonly linked by citations. We propose a unified probabilistic model for both the textual content and the link structure of a document collection. Our model is based on the recently introduced framework of Probabilistic Relational Models (PRMs), which allows us to capture correlations between linked documents. We show how to learn these models from data and use them efficiently for classification. Since exact methods for classification in these large models are intractable, we utilize belief propagation, an approximate inference algorithm. Belief propagation automatically induces a very natural behavior, where our knowledge about one document helps us classify related ones, which in turn help us classify others. We present preliminary empirical results on a dataset of university web pages.

1 Introduction

The majority of previous work on text classification has made use of “flat” representations, where each document is a data instance whose attributes are the set of words it contains. However, many text domains are much richer in structure, involving multiple documents that are related to each other in complex ways. Examples of such domains are the World Wide Web where web pages are related to each other via hyperlinks and the scientific paper domain where papers are related via citations.

Recently, there has been a growing interest in classification techniques for more richly structured text data sets that make use of the additional link structure information that exists between documents. As a motivating example, consider the task introduced by Craven *et al.* [1998] of classifying web documents as being either a student, faculty, course or project home page. Intuitively, we would like to use our information about one document to help us reach conclusions about other, related documents. For example, we should be able to use the categories of pages to which a web page links to help infer the category of the page.

Several papers have recently proposed algorithms that utilize information from related documents to aid classification. Chakrabarti *et al.* [1998] describe a relaxation labeling algorithm that iteratively reassigns labels based on the current labels the neighboring documents. Neville and Jensen [2000] propose an *iterative classification* algorithm which essentially implements this process. Slattery and Mitchell [2000] propose an application of a similar iterative relaxation scheme to the problem of classifying web pages. This work illustrates that classification accuracy improves by exploiting the relational structure. However, none of these approaches propose a single coherent model of the correlations between different related documents. Hence, they are forced to provide a procedural approach, where the results of different classification steps or algorithms are combined without a general underlying model.

In this paper, we propose a unified framework for modeling and learning relational structure. Our framework allows for inferences, similar to those mentioned above, that propagate via the relational structure that exists over the objects in our domain. The key to our approach is the use of a single probabilistic model that captures the interactions between the objects in our domain. Our work builds on *probabilistic relational models (PRMs)*—a recent development [Koller and Pfeffer, 1998; Poole, 1993]. PRMs extend the standard attribute-based Bayesian network representation to incorporate a much richer relational structure. They allow properties of an entity to depend probabilistically on properties of other *related* entities. The model represents a generic dependence for a *class* of objects, which is then instantiated for particular sets of entities and relations between them. Friedman *et al.* [1999] adapt the machinery for learning Bayesian networks from flat data to the task of learning PRMs from structured relational data.

The basic PRM model takes the relational structure as input; in other words, it is outside the probabilistic model. As many have noted, the relational structure is informative in and of itself. For example, the links from and to a web page are very informative about the type of web page [Craven *et al.*, 1998], and the citation links between papers are very informative about the paper topics [Cohn and Hofmann, 2001]. The knowledge that a certain page is a hub [Kleinberg, 1998] can also be quite informative. For example a directory of student listings is a student hub; this knowledge can help us infer the category of pages pointed to by the hub.

Here, we model the link structure explicitly by modeling the uncertainty over the existence of links between objects in our domain, as introduced in [Getoor *et al.*, 2001]. For example, when classifying web pages, we model the probability of the existence of a hyperlink between all possible pairs of web pages. In addition, we introduce a hidden variable, *Hub*, which not only captures the traditional notion of hub [Kleinberg, 1998], but which also describes the type of hub. For example, in the WebKB domain, a web page may be a student, course, project or faculty hub page. This modeling is precisely that which enables the propagation of influence between objects that are related: a page that points to many student pages is likely to be a student hub; furthermore, a page that is pointed to by a student hub is more likely to be a student page.

We evaluate our method on the task of classification of web pages into a predetermined set of classes from a collection of university web pages. Here, we learn a model over schools in the training set and use it to classify web pages in other schools. The probabilistic inference algorithm we use automatically induces the desired behavior, where our knowledge about one instance helps us classify related ones, which in turn help us classify others. Preliminary experiments show that the relational information provides a significant boost in classification accuracy.

Section 2 describes probabilistic relational models. In Section 3, we propose a probabilistic relational model for the web domain. Section 4 presents a method for learning the models and Section 5 describes how a learned model can be used to make predictions. We explain how relational information in the test set is propagated between instances in Section 6. Finally, Section 7 presents evaluation and results.

2 Probabilistic Relational Models

A *probabilistic relational model (PRM)* specifies a template for a probability distribution over a relational database. The template describes the relational schema for the domain, and the probabilistic dependencies between attributes in the domain. A PRM, together with a particular database of objects and relations, defines a probability distribution over the attributes of the objects and the relations.

Relational Schema A schema for a relational model describes a set of *classes*, $\mathcal{X} = X_1, \ldots, X_n$. Each class is associated with a set of *descriptive attributes* and a set of *reference slots*.[1] The set of descriptive attributes of a class X is denoted $\mathcal{A}(X)$. Attribute A of class X is denoted $X.A$, and its domain of values is denoted $V(X.A)$. We assume here that domains are finite. For example, the Page class might contain a *Category* attribute with a domain of {course, faculty, project, student, other} as well as a set of binary attributes to indicate whether it contains certain words.

The set of reference slots of a class X is denoted $\mathcal{R}(X)$. We use $X.\rho$ to denote the reference slot ρ of X. Each reference slot ρ is typed: the domain type of $\mathrm{Dom}[\rho] = X$ and the range type $\mathrm{Range}[\rho] = Y$, where Y is some class in $\mathcal{X}$. A slot ρ denotes a function from $\mathrm{Dom}[\rho] = X$ to $\mathrm{Range}[\rho] = Y$. For example, we might have a class Link with the reference slots *From-Page* and *To-Page* whose range is the class Page.

It is often useful to distinguish between an *entity* and a *relationship*, as in entity-relationship diagrams. In our language, classes are used to represent both entities and relationships. Thus, entities such as web pages are represented by classes, and a relationship such as Link, which relates web pages to web pages, is also represented as a class, with reference slots to the class Page. We use $\mathcal{X}_{\mathcal{E}}$ to denote the set of classes that represent entities, and $\mathcal{X}_{\mathcal{R}}$ to denote those that represent relationships. The members of classes are called *objects* regardless of whether the class is an entity or relationship class.

The semantics of this language is straightforward. An instantiation $\mathcal{I}$ specifies the set of objects in each class, and the values for each attribute and each reference slots of each object. For example, in a dataset of web pages, an instantiation specifies the set of web pages and hyperlinks between them, along with words they contain.

An instantiation includes the *relational skeleton*, σ_r, which specifies the complete relational structure in the model: the set of objects in all classes, as well as all the relationships that hold between them. In other words, it specifies the set of object in each class X, denoted $\sigma(X)$, and for each object $x \in \sigma(X)$, it specifies the values of all of the reference slots $x.\rho$. In our web page example, the relational skeleton would contain the set of web pages and links between them but not their category or the words they contain.

Probabilistic Model for Attributes A probabilistic relational model Π specifies a probability distributions over all instantiations $\mathcal{I}$ of the relational schema. It consists of the qualitative dependency structure, $\mathcal{S}$, and the parameters associated with it, $\theta_{\mathcal{S}}$. The dependency structure is defined by associating with each attribute $X.A$ a set of *parents* $\mathrm{Pa}(X.A)$.

Each parent of $X.A$ has the form $X.B$ or $X.\tau.B$ where τ is a sequence of reference slots. More precisely, we define a *slot chain* $\rho_1, \ldots, \rho_k$ be a sequence of slots such that for all i, $\mathrm{Range}[\rho_i] = \mathrm{Dom}[\rho_{i+1}]$.

The quantitative part of the PRM specifies the parameterization of the model. Given a set of parents for an attribute, we can define a local probability model by associating with it a *conditional probability distribution (CPD)*. For each attribute we have a CPD that specifies $P(X.A \mid \mathrm{Pa}(X.A))$.

Definition 2.1: A *probabilistic relational model (PRM)* Π for a relational schema $\mathcal{S}$ is defined as follows. For each class $X \in \mathcal{X}$ and each descriptive attribute $A \in \mathcal{A}(X)$, we have:

- a set of *parents* $\mathrm{Pa}(X.A)$, where each parent has the form $X.B$ or $X.\tau.B$.
- a conditional probability distribution that represents $P(X.A \mid \mathrm{Pa}(X.A))$. ▮

For a given skeleton σ, the PRM structure induces an *unrolled* Bayesian network over the random variables $x.A$. For every object $x \in \sigma(X)$, $x.A$ depends probabilistically on parents of the form $x.B$ or $x.\tau.B$. (We will assume that $x.\tau$ is single-valued throughout, although PRMs allow dependence on multi-valued relations as well.) Note that the CPD for $X.A$ is used for each $x.A$ in the unrolled network, and is repeated many times in the network. Thus the same parameters are

[1] There is a direct mapping between our notion of class and the tables in a relational database: descriptive attributes correspond to standard table attributes, and reference slots correspond to foreign keys attributes (key attributes of another table).

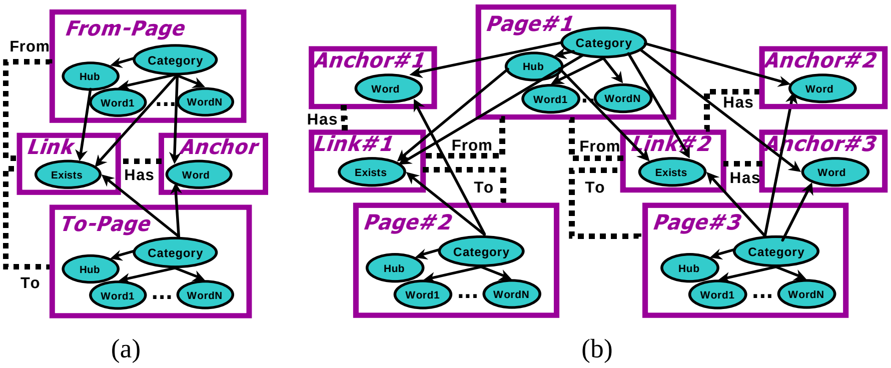

Figure 1: (a) PRM Model for WebKB domain; (b) Fragment of unrolled network for WebKB model.

used in many different contexts in the network. The context is given by the set of parents for each attribute as defined by the CPD together with the relational skeleton.

Structural Uncertainty In the model described in the previous section, all relations between attributes are determined by the relational skeleton σ_r; only the descriptive attributes are uncertain. In this section, we extend our probabilistic model to allow for *structural uncertainty*. Here, we do not treat the relational structure as background knowledge, but choose to model it explicitly within the probabilistic framework. Clearly, there are many ways to represent a probability distribution over the relational structure. In this paper, we use a simple yet natural model: *Existence Uncertainty*.

Suppose we are given only the schema and information about some of the objects in the domain, but we have uncertainty over the links between objects. We can extend our probabilistic model to handle this uncertainty by explicitly modeling the existence of the links themselves.

We begin by introducing the notion of an *entity skeleton*, σ_e. An entity skeleton is less informative than a relational skeleton. It specifies a set of entities $\sigma_e(X)$ only for the classes $X \in \mathcal{X}_{\mathcal{E}}$. In our web page example, the entity skeleton would omit the information about the hyperlinks, and only include information about the set of web pages. We call the entity classes *determined* and the others *undetermined*. We note that relationship classes typically represent many-many relationships; they have at least two reference slots, which refer to determined classes. For example, our Link class would have reference slots *From-Page* and *To-Page* to class Page. While we know the set of web pages, we may be uncertain about which web pages link to each other, and thus we have uncertainty over the existence of the Link objects.

Our basic approach in this model is that we allow objects whose existence is uncertain. These are the objects in the undetermined classes. One way of achieving this effect is by introducing into the model all of the entities that can *potentially* exist in it; with each of them we associate a special binary variable that that tells us whether the entity actually exists or not. Note that this construction is purely conceptual; we never explicitly construct a model containing non-existent objects. In our example above, the domain of the Link class in a given instantiation $\mathcal{I}$ is $\mathcal{I}(\text{Page}) \times \mathcal{I}(\text{Page})$. Each “potential” object $x = \text{Link}(y_f, y_t)$ in this domain is associated with a binary attribute $x.E$ that specifies whether the page y_f did or did not have a link to the page y_t.

The exists attribute for an undetermined class is treated in the same way as a descriptive attribute in our dependency model, in that it can have parents and children, and is associated with a CPD. Our definitions are such that the semantics of the model does not change. For example, the existence of a link between two pages may depend on their categories as well as presence of certain words in those pages.

3 PRMs for the Web

Figure 1(a) shows a PRM for the web page domain. For clarity, the Page class was duplicated in the figure, once as From-Page and once as To-Page. The textual content of each page is described by a simple binomial Naive Bayes type model over words contained in the page (a binomial bag-of-words-model).

Some categories of pages are much more likely to have links to each other (faculty and students) while others are much less likely (course and project). We can model such dependence using the existence uncertainty model described in the previous section. We introduce an attribute Link.*Exists*, and have Link.*Exists* depend on Link.*From-Page.Category* and Link.*To-Page.Category*.

In addition, certain web pages may be *directory* pages. Directory pages point to a large number of web pages of a particular category. For example, a student directory typically points to student web pages. We can model this property of web pages by introducing the attribute *Hub* for Page class. The domain of the *Hub* corresponds to the domain of the *Category*, e.g., {course-hub, faculty-hub, project-hub, student-hub, non-hub}. The existence of a link between a student hub page and a student page is highly probable, while a link from a student hub page to a course page is very unlikely. We can model this dependence by letting Link.*Exists* depend on Link.*From-Page.Hub* as well as on Link.*From-Page.Category* and Link.*To-Page.Category*.

Another important source of information comes from the anchor words contained (underlined) in the hyperlink. For example, a student page with a link containing the word “advisor” is likely to point to a faculty page, while a course page with a link containing the word "instructor" probably links to a faculty page. Note that the category of both the source and destination page is crucial. We can model this dependence by introducing a class Anchor with a reference slot *In-Link* and an attribute *Word*, where *Word* has parents Anchor.*In-Link.From-Page.Category* and Anchor.*In-Link.To-Page.Category*.

Given a particular set of hyperlinked pages, the template is instantiated to produce an "unrolled" Bayesian network. Figure 1(b) shows a fragment of such a network for three web pages. The two existing links from page 1 to page 2 and 3 are shown while non-existing links omitted for clarity (however still play a role in the inference). Also shown are the anchor word for link 1 and two anchor words for link 2. Note that during classification, existence of links and anchor words in the links are used as evidence to infer categories of the web pages. Hence, our unrolled Bayes net has active paths between categories of pages through the v-structures at Link.*Exists* and Anchor.*Word*. These active paths capture exactly the pattern of relational inference we set out to model.

4 Learning the Models

In this paper, we assume that the dependency structure in our models is specified, so learning the models amounts to estimating the parameters. We adapt a Bayesian parameter estimation approach [Heckerman, 1998]. We use a standard Dirichlet prior for the parameters. Conveniently, in this case the CPD of each attribute can be estimated separately. The CPD $P(X.A \mid \mathbf{u})$ depends only on the *sufficient statistics* $N_{X.A}[v, \mathbf{u}]$, that count the number of entities with $x.A = v$ and Pa$(x.A) = \mathbf{u}$. These sufficient statistics can be computed using standard relational database queries.

The extension of parameter estimation to PRMs with existence uncertainty is straightforward. The only new issue is how to compute sufficient statistics that include existence attributes $x.E$ without explicitly adding all non-existent entity into our database. We perform this computation by counting, for each possible instantiation of Pa$(X.E)$, the number of potential objects with that instantiation, and subtracting the actual number of objects x with that parent instantiation.

5 Belief Propagation for Classification

Once we have learned a model, how do we use the model for prediction? Classification in our framework is done by computing the posterior distribution over the unobserved variables given the data and assigning each unobserved variable its most likely value. This requires inference over the unrolled network defined by instantiating a PRM for a particular document collection. We cannot decompose this task into separate inference tasks over the objects in the model, as they are all correlated. In general, the unrolled network can be fairly complex, involving many documents that are linked in various ways. (In our experiments, the networks involve hundreds of thousands of nodes.) Exact inference over these networks is clearly impractical, so we must resort to approximate inference. There are a wide variety of approximation schemes for Bayesian networks. For various reasons (some of which are described below), we chose to use *belief propagation*. Belief Propagation (BP) is a local message passing algorithm introduced by Pearl [1988]. It is guaranteed to converge to the correct marginal probabilities for each node only for singly connected Bayesian networks. However, empirical results [Murphy and Weiss, 1999] show that it often converges in general networks, and when it does, the marginals are a good approximation to the correct posteriors.

We provide a brief outline of one variant of BP, referring to [Murphy and Weiss, 1999] for more details. Consider a Bayesian network over some set of nodes (which in our case would be the variables $x.A$). We first convert the graph into a *family graph*, with a node F_i for each variable X_i in the BN, containing X_i and its parents. Two nodes are connected if they have some variable in common. The CPD of X_i is associated with F_i. Let φ_i represent the factor defined by the CPD; i.e., if F_i contains the variables $X, Y_1, \ldots, Y_k$, then φ_i is a function from the domains of these variables to $[0, 1]$. We also define ψ_i to be a factor over X_i that encompasses our evidence about X_i: $\psi_i(X_i) \equiv 1$ if X_i is not observed. If we observe $X_i = x$, we have that $\psi_i(x) = 1$ and 0 elsewhere. Our posterior distribution is then $\alpha \prod_i \varphi_i \times \prod_i \psi_i$, where α is a normalizing constant.

The belief propagation algorithm is now very simple. At each iteration, all the family nodes simultaneously send message to all others, as follows:

$$m_{ij}(F_i \cap F_j) \leftarrow \alpha \sum_{F_i - F_j} \varphi_i \psi_i \prod_{k \in N(i) - \{j\}} m_{ki}$$

where α is a (different) normalizing constant and $N(i)$ is the set of families that are neighbors of F_i in the family graph. At any point in the algorithm, our marginal distribution about any family F_i is $b_i = \alpha \varphi_i \psi_i \prod_{k \in N(i)} m_{ki}$. This process is repeated until the beliefs converge.

After convergence, the b_i give us approximate marginal distributions over each of the families in the unrolled network. These marginals are then used to predict the class of the documents.

6 Influence propagation over relations

Among the strong motivations for using a relational model is its ability to model dependencies between related instances. Intuitively, we would like to use our information about one object to help us reach conclusions about other, related objects. For example, we should be able to propagate information about the topic of a document p to documents it has links to and documents that link to it. These, in turn, would propagate information to yet other documents.

Recently, several papers have proposed a process along the lines of this "influence propagation" idea. Chakrabarti *et al.* [1998] describe a relaxation labeling algorithm that makes use of the neighboring link information. The algorithm begins with the labeling given by a text-based classifier constructed from the training set. It then uses the estimated class of neighboring documents to update the distribution of the document being classified. They show that even using small neighborhoods around the test document significantly increases accuracy.

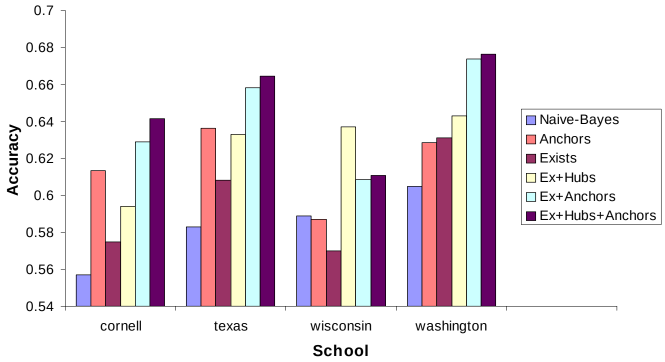

Figure 2: Comparison of accuracy of several models ranging from the simplest model, **Naive-Bayes to the most complex model,Ex+Hubs+Anchors, which incorporates existence uncertainty, hubs and link anchor words. In each case, a model was learned for 3 schools and tested on the remaining school.**

Neville and Jensen [2000] propose a very similar approach. Their *iterative classification* algorithm essentially implements this process exactly. It builds a classifier based on a fully observed relational training set; the classifier uses both base attributes and more relational attributes (e.g., the number of related entities of a given type). It then uses this classifier on a test set where the base attributes are observed, but the class variables are not. Those instances that are classified with high confidence are temporarily labeled with the predicted class; the classification algorithm is then rerun, with the additional information. The process repeats several times. The classification accuracy is shown to improve substantially as the process iterates.

Slattery and Mitchell [2000] propose an iterative algorithm called FOIL-HUBS for the problem of classifying web pages, e.g., as belonging to a university student or not. They note that several pages in the dataset have links to many other pages, most of which were classified as student home pages. Their approach uses recursive predicate rules to identify such a page as a student directory page based on whether the pages it points to are student pages, and conclude that other pages to which it points are also more likely to be student pages. These rules are combined with text-based classifiers in an iterative relaxation scheme. They show that classification accuracy improves by exploiting the relational structure.

Our approach achieves this effect through the probabilistic influences induced by the unrolled Bayesian network over the instances in our domain. For example, in the web domain, our network has a correlation between the class of web pages that link to each other. Thus, our beliefs about the class of one web page will influence our beliefs about the class of its related web pages. In general, probabilistic influence "flows" through active paths in the unrolled network, allowing beliefs about one cluster to influence others to which it is related (directly or indirectly). Moreover, the use of belief propagation implements this effect directly. By propagating a local message from one family to another in the family graph network, the algorithm propagates our beliefs about one variable to other variables to which it is directly connected.

7 Experiments

In this section we describe experimental results on the WebKB dataset [Craven *et al.*, 1998]. The WebKB dataset contains web pages from four different Computer Science departments. We included only pages that have at least one out link; the number of resulting pages for each school are: Cornell (318), Texas (319), Washington (420), and Wisconsin (465). Each page has a category attribute representing the type of web page which is one of $\{$course, faculty, student, project, other$\}$. The text content of the web page is represented using a set of binary attributes that indicate the presence of different words on the page. After stemming, removing stop words and rare words, the dictionary contains around 800 words. Each web page has a hub attribute, which is takes the following values: course-hub, faculty-hub, student-hub, project-hub, nonhub. The original dataset did not contain hub labels. We labeled a page as a hub of a particular category if it pointed to many pages of that category. Note that we hid the hub labels in the test set. Each school had one hub page of each category, except for Washington which does not have a project hub page and Wisconsin which does not have a faculty web page. The data set also describes the links between school web pages; the number of links for each school are: Cornell (923), Texas (1041), Washington (1534) and Wisconsin (1823). In addition, for each link between pages, the dataset specifies the words on the anchor link. We selected top 100 anchor words using mutual information score.

We compared the performance of several models on predicting web page categories. In each case, we learned a model from three schools, and tested the performance of the learned model on the remaining school. Our experiments used Bayesian estimation with a uniform Dirichlet parameter prior with equivalent sample size $\alpha = 2$.

All models we compared can be viewed as a subset of the model in Figure 1(a). Our baseline is a standard binomial Naive Bayes model that uses only words on the page to predict the category of the page. We evaluated the following set of models:

1. **Naive-Bayes**: Our baseline model.

2. **Anchors:** This model uses both words on the page and anchor words on the links to predict the category.
3. **Exists**: This model adds structural uncertainty over the link relationship to the simple baseline model; the parents of Link.*Exists* are Link.*From-Page.Category* and Link.*To-Page.Category*.
4. **Ex+Hubs**: This model extends the **Exists** model with Hubs. In the model Link.*Exists* depends on *Link.From-Page.Hub* in addition to the categories of each of the pages.
5. **Ex+Anchors**: This model extends the **Exists** model with anchor words (but not hubs).
6. **Ex+Hubs+Anchors**: The final model includes existence uncertainty, hubs and anchor words.

Figure 2 compares the accuracy achieved by the different models on each of the schools. The final model, **Ex+Hubs+Anchors**, which incorporates structural uncertainty, hubs and anchor words, consistently outperforms the **Naive-Bayes** model by a significant amount. In addition, it outperforms any of the simpler variants.

Our algorithm was fairly successful at identifying the hubs in the test set. It misclassified 7 out 1522 pages as hubs while recognizing 6 out of the true 14 hubs correctly. The pages mislabeled as hubs often pointed to many pages that had been labeled as Other web pages. However, on further inspection, these hub pages often *were* directories pointing to pages that were likely to be researcher home pages or course home pages and seemed to have been mislabeled in the training set as other. We investigated how much these misclassifications hurt the performance by revealing the labels of the hub attribute in the test data. The improvement in performance was roughly 2%.

8 Discussion and Conclusions

Many real-world domains have a rich relational structure, with complex webs of interacting entities: the web, scientific papers and more. Traditional machine learning algorithms ignore this rich relational structure, flattening it into a set of attribute vectors assumed to be independent. Recently, however, there has been growing interest in learning methods that exploit the relational structure of the domain.

In this paper, we provide a general method for classification in richly structured data with instances and relations. Our approach has coherent probabilistic semantics, allowing us to build on powerful tools for probabilistic reasoning and learning. Our classification algorithm uses a combination of these techniques to provide effective scaling in the number of instances; it can thus be applied to large domains.

Finally, our approach induces a compelling behavior unique to relational settings: Because instances are *not* independent, information about some instances can be used to reach conclusions about others. Our approach is the first to provide a formal framework for this behavior.

References

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

[Cohn and Hofmann, 2001] D. Cohn and T. Hofmann. The missing link: A probabilistic model of document content and hypertext connectivity. In *Proc. NIPS*, 2001. To appear.

[Craven *et al.*, 1998] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the world wide web. In *Proc. AAAI*, 1998.

[Friedman *et al.*, 1999] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *Proc. IJCAI*, 1999.

[Getoor *et al.*, 2001] L. Getoor, N. Friedman, D. Koller, and B. Taskar. Learning probabilistic models of relational structure. In *Proc. ICML*, 2001. To appear.

[Heckerman, 1998] D. Heckerman. A tutorial on learning with Bayesian networks. In M. I. Jordan, editor, *Learning in Graphical Models*. MIT Press, Cambridge, MA, 1998.

[Kleinberg, 1998] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proc. 9th ACM-SIAM Symposium on Discrete Algorithms*, 1998.

[Koller and Pfeffer, 1998] D. Koller and A. Pfeffer. Probabilistic frame-based systems. In *Proc. AAAI*, 1998.

[Murphy and Weiss, 1999] K. Murphy and Y. Weiss. Loopy belief propagation for approximate inference: an empirical study. In *UAI*, 1999.

[Neville and Jensen, 2000] J. Neville and D. Jensen. Iterative classification in relational data. In *Proc. AAAI-2000 Workshop on Learning Statistical Models from Relational Data*, pages 13–20. AAAI Press, 2000.

[Pearl, 1988] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.

[Poole, 1993] D. Poole. Probabilistic Horn abduction and Bayesian networks. *Artificial Intelligence*, 64:81–129, 1993.

[Slattery and Mitchell, 2000] S. Slattery and T. Mitchell. Discovering test set regularities in relational domains. In *Proc. ICML*, 2000.