

AFRL-IF-RS-TR-2005-23
Final Technical Report
January 2005



LEARNING STATISTICAL PATTERNS IN RELATIONAL DATA USING PROBABILISTIC RELATIONAL MODELS

Stanford University

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

**AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE
ROME RESEARCH SITE
ROME, NEW YORK**

STINFO FINAL REPORT

This report has been reviewed by the Air Force Research Laboratory, Information Directorate, Public Affairs Office (IFOIPA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

AFRL-IF-RS-TR-2005-23 has been reviewed and is approved for publication

APPROVED: /s/

RAYMOND A. LIUZZI
Project Engineer

FOR THE DIRECTOR: /s/

JAMES A. COLLINS, Acting Chief
Advanced Computing Division
Information Directorate

REPORT DOCUMENTATION PAGE			<i>Form Approved</i> OMB No. 074-0188	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE JANUARY 2005	3. REPORT TYPE AND DATES COVERED Final Sep 01 – Oct 04	
4. TITLE AND SUBTITLE LEARNING STATISTICAL PATTERNS IN RELATIONAL DATA USING PROBABILISTIC RELATIONAL MODELS			5. FUNDING NUMBERS C - F30602-01-2-0564 PE - 62301E PR - EELD TA - 01 WU - 05	
6. AUTHOR(S) Daphne Koller				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Stanford University 353 Serra Mall Gates Building 1A Stanford California 94305			8. PERFORMING ORGANIZATION REPORT NUMBER N/A	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Research Laboratory/IFED 525 Brooks Road Rome New York 13441-4505			10. SPONSORING / MONITORING AGENCY REPORT NUMBER AFRL-IF-RS-TR-2005-23	
11. SUPPLEMENTARY NOTES AFRL Project Engineer: Raymond A. Liuzzi/IFED/(315) 330-3577/ Raymond.Liuzzi@rl.af.mil				
12a. DISTRIBUTION / AVAILABILITY STATEMENT APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.				12b. DISTRIBUTION CODE
13. ABSTRACT (Maximum 200 Words) This report describes techniques for learning probabilistic models of relational data, and using these models to interpret new relational data. This effort focused on developing undirected probabilistic models for representing and learning graph patterns, learning patterns involving links between objects, learning discriminative models for classification in relational data, developing and labeling two real-world relational data sets - one involving web data and the other a social network - and evaluating the performance of our methods on these data sets, and dealing with distributions that are non-uniform, in that different contexts (time periods, organizations) have statistically different properties. The technology developed under this effort was transitioned and is being used under the Perceptive Assistant Program (PAL) at DARPA.				
14. SUBJECT TERMS Probability, Learning, Network Learning, Artificial Intelligence				15. NUMBER OF PAGES 36
				16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT UNCLASSIFIED	18. SECURITY CLASSIFICATION OF THIS PAGE UNCLASSIFIED	19. SECURITY CLASSIFICATION OF ABSTRACT UNCLASSIFIED	20. LIMITATION OF ABSTRACT UL	

Table of Contents

1	Overview.....	1
2	Relational Markov Networks	1
2.1	Basic Language	1
2.2	Link Prediction.....	2
2.3	Dataset Development and Experimentation.....	2
3	Collective Classification	3
4	Non-Stationary Distributions	3
5	Publications and Presentations	4
5.1	Publications	4
5.2	Presentations	5
6	Transitions.....	6
	Appendix A: Link Prediction in Relational Data.....	7
	Appendix B: Learning Associativ Markoff Records.....	15
	Appendix C: Discriminative Probabilistic Models for Relational Data.....	25

1 Overview

The focus of this project was learning probabilistic models of relational data, and using these models to interpret new relational data. Our work on this project focused on several areas:

1. Developing undirected probabilistic models for representing and learning graph patterns.
2. Learning patterns involving links between objects.
3. Learning discriminative models for classification in relational data.
4. Developing and labeling two real-world relational data set — one involving web data and the other a social network — and evaluating the performance of our methods on these data sets.
5. Dealing with distributions that are non-uniform, in that different contexts (time periods, organizations) have statistically different properties.

The text below elaborates on some of the work that took place along these thrusts. In addition, more information about the work done under this project can be found in the publications that document the work, listed below.

2 Relational Markov Networks

2.1 Basic Language

As one of our key directions, we developed a new class of probabilistic models for relational data based on partially directed and undirected graphical models (chain graphs and Markov networks). This class of models, which we called *relational Markov network (RMN)s*, is particularly well suited to the task of prediction in structured, relational data. These models can incorporate rich information about the attributes of entities and, more importantly, the link graph between entities, for high-precision classification. Furthermore, they address two limitations of the relational Bayesian networks that we proposed in our early work. First, undirected models do not impose the acyclicity constraint that hinders representation of many relational dependencies in directed models. For example, symmetric relations like *Met*(X, Y) or asymmetric relations like *KnowsAbout*(X, Y) that can have cycles present a challenge for directed models. Second, undirected models are well suited for discriminative training, which generally improves classification accuracy. We have developed a system that can learn and reason with such models efficiently on large databases.

We began by experimenting with a publicly available dataset of several web sites of computer science departments at major universities (WebKB). The task consists of identifying the set of student, faculty, courses and research projects at the department from the 1000 to 2000 web pages collected by a web crawl of each site. The entities roughly correspond to web pages and links to hyperlinks between them. Additionally, the web pages themselves are structured, consisting of different sections. This problem presents several interesting challenges typical of natural language and relational domains — text and interconnections of web pages are very heterogeneous as they are authored by many different people. However there are strong relational patterns that can be exploited to identify different entities. For example, pages of faculty members tend to contain a list of students they advise, courses they teach, projects they manage, etc. Students tend to link to their advisor, courses they took or assisted, projects they are involve in, etc.

Using various models that incorporate relational patterns, we reduced the classification error rate from 18% for a very strong approach that uses “flat” data, to 11.5% for our relational approach. This is a significant reduction in error of over 35%. We note that the error reduction relative to our earlier approach using directed probabilistic relational models is even greater. Our experiments on this and other datasets involved reasoning in networks containing over 200 thousand entities connected by over 600 thousand links. The largest experiments take several hours on a 700MHz with 2GB per process machine

2.2 Link Prediction

We then investigated the application of the RMN language to the problem of predicting the existence and the type of relationships between two entities. For example, we want to be able to assert not only that X is a professor and Y a student, but that X is Y ’s advisor. In a terrorist domain, we might want to conclude not only that X is a bank and Y a terrorist organization, but that the relationship between them is that of money laundering.

We developed a suite of subgraph patterns that occur in real-world relational graphs and showed that they can be encoded easily in our RMN language. Such patterns include, for example, transitivity — if X knows Y and Y knows Z then X is more likely to know Z , as well as richer patterns — if X is Y ’s advisor and X teaches course Z then Y is more likely to be the TA for course Z . We also extended our learning and inference algorithms to work on the problem of link classification.

2.3 Dataset Development and Experimentation

To test this algorithm, we constructed a rich dataset that extends the WebKB dataset mentioned above. The new dataset incorporates large web sites of four new schools (Stanford, Berkeley, MIT, and CMU) and uses a more refined ontology of entities that includes students, faculty, staff, research scientists, courses, research projects, research groups, etc. In addition, links between entities are labeled as well: student-advisor, course-instructor, and course-ta (teaching assistant), project-member, group-member, etc. We labeled both hyperlinks, and virtual links, where one person’s name is mentioned on the webpage of another person. In both cases, the observed link may or may not correspond to an actual relationship (e.g., student-advisor). We built several tools that allow us to spider and label such data efficiently. Overall, we hand-labeled 11,000 webpages and 110,000 links.

We also constructed a second data set based on a real-world social network of Stanford students. For each student, we have a set of attributes, such as their hobbies, residence, major, etc. We also have, for each student, the other students they consider to be their friends.

On the university data, we tested different models, including flat classification approaches, and various relational approaches, on the task of predicting the existence and type of link. Overall, the relational models performed much better than the standard “flat” classification approaches, increasing the average classification accuracy from 55% to 60%. On the social network data, we tried a somewhat different task, where some random subset of the links (10%, 25%, or 50%) is observed in the test data, and can be used as evidence for predicting the remaining links. Using just the observed portion of links, we constructed the following features: for each student, the proportion of students in the residence that list him/her and the proportion of students he/she lists; for each pair of students, the proportion of other students they have as common friends. These features are a flat version of relational structure and dependencies

between links, and therefore serve as a good benchmark for comparing against flat approaches. We then compared several models: a “flat” model, which uses these features for predicting links, as well as a feature for each match of a characteristic of the two people involved (e.g., both people are computer science majors or both are freshmen). Our relational model introduces a correlation between each pair of links emanating from each person, allowing an interaction between their existence. Overall, the relational model outperformed the flat model by a statistically significant margin (as measured by a paired t-test).

3 Collective Classification

As a parallel thrust to our development of probabilistic relational languages and associated learning algorithms, we also focused on the fundamental problem of learning models for *collective classification* — classifying an entire set of inter-dependent entities as a whole, rather than classifying each as an independent instance. Although our previous learning algorithms address this task better than previous approaches, the accuracies they achieved were still lower than we would like, especially for domains where the signal-to-noise ratio is very low (i.e., the target class has very low frequency in the general population). We have therefore developed a new approach for learning the parameters of our undirected relational models. Unlike standard approaches, which try to optimize the conditional likelihood of the target labels given the features, our approach focuses explicitly on the classification task. In particular, motivated by ideas from the highly successful flat classification technique of “support vector machines (SVMs)”, our approach tries to maximize the “margin” — the difference in log-probability between the correct label and all other labels.

The key contribution of this approach is that it can apply these ideas to the case of collective classification of an entire set of entities. In this case, the overall set of labels for the set is exponentially large — a set of n entities, each of which has k labels, has a total of k^n total labels. As we show, we can exploit the structure of the probabilistic graphical model to avoid this exponential blowup, allowing the entire learning problem to be formulated as a compact convex quadratic program, making the problem amenable to a variety of standard methods. Another major feature of our method is its ability to use “kernels” — a method that allows very high-dimensional (even infinite-dimensional) feature spaces to be used efficiently. Kernels are one of the factors that contribute to the enormous success of SVMs in many flat classification tasks.

So far, we have applied this method to the task of collectively labeling a set of entities related only by a simple link structure in the form of a sequence. This type of structure is of independent interest, as it can be used to collectively classify a sequence of related events. We have experimented with this approach on the problem of optical character recognition — labeling a sequence of character images that form a word. Our approach shows a relative reduction in error of 45% relative to the state of the art probabilistic models, and a reduction of error of 33% relative to the best flat classifier — SVMs with kernels.

4 Non-Stationary Distributions

In this quarter, our work also took a slightly different direction. A common assumption when performing classification tasks is that both the training and the operation data are drawn IID (Independent and Identically Distributed) from some fixed distribution. In another words, we expect regularities found in the training data to show up in operation data, and vice versa. However, our experience with the university website classification shows that this is often not the

case: different organizations (universities) often exhibit very different patterns. In general, training and operational data can have quite different distributions, depending on factors such as when the data was collected, or where it was collected, or by whom it was collected. As another example, in news articles, the distribution of words in an article is dependent on when it was written. New stories emerge over time, introducing new people names, new place names, etc. Similarly, in identifying terrorist activity from communication data for a new organization, we expect to see new terms that we have rarely or never encountered before in organizations used in our training data. Ignoring the existence of such a phenomenon, results in learning misleading patterns. Moreover, these new terms can be very useful for classification. For example, discovering the code name for a new terrorist operation can help to identify relevant communications.

We introduced an approach to solve this problem in two ways: Firstly, we rely on terms that we have seen before to infer the meanings of new features, and subsequently use these new features for classification. For example, in examining communications data, we might find that a certain new term has been frequently mentioned together with the name of terrorists. As such, we infer that this new term is a terrorist-related term too. Our second way is to learn characteristics of these features useful for classification. For example, we might learn that names of restaurants are more often keywords compared to names of cars, thus this will help us focus our search for useful new keywords.

We tested our approach on two datasets: a news article collection and the university webpage collection described in previous reports. Compared to state of the art approaches that do not take into consideration information from new features, our approach showed a relative reduction in error rate of 56.3% for the news article dataset, and a relative reduction in error rate of 20.7% for the university webpage dataset.

5 Publications and Presentations

5.1 Publications

1. “Max-Margin Markov Nets,” B. Taskar, C. Guestrin, and D. Koller. *Neural Information Processing Systems Conference (NIPS)*, Vancouver, Canada, December 2003. **Winner of the Best Student Paper Award.**
2. “Link Prediction in Relational Data,” B. Taskar, M. F. Wong, P. Abbeel, and D. Koller. *Neural Information Processing Systems Conference (NIPS)*, Vancouver, Canada, December 2003.*
3. “Learning on the Test Data: Leveraging ‘Unseen’ Features,” B. Taskar, M.-F. Wong, and D. Koller. *Twentieth International Conference on Machine Learning (ICML)*, Washington, D.C., August 2003.
4. “Learning Associative Markov Networks,” B. Taskar, V. Chatalbashev, and D. Koller, *Proceedings of the Twenty-First International Conference on Machine Learning (ICML)*, 2004, To appear.*
5. “Discriminative probabilistic models for relational data,” B. Taskar, P. Abbeel, and D. Koller. *Eighteenth Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, Edmonton, Canada, August 2002, pages 485–492.*

* These publications are contained in the appendices beginning on page 7.

5.2 Presentations

The above papers were all presented by the PI or by one of her students at the respective conferences where they appeared. In addition the work performed under this contract was prominently figured in two plenary invited talks given by the PI:

1. **Invited plenary talk** at the *40th Anniversary Meeting of the Association for Computational Linguistics (ACL '02)*, Philadelphia, Pennsylvania, and July 2002. Title: “Probabilistic Models of Relational Data.”
2. **Invited plenary talk:** “Probabilistic Models of Relational Data.” Plenary invited talk joint to the *Twentieth International Conference on Machine Learning (ICML-2003)* and the *Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*, Washington, DC, August 2003.

6 Transitions

The technology developed under this contract was transitioned in two primary ways. First, some of the ideas and algorithms were used by Alphatech as part of the EAGLE TIEs. Second, the collective classification technology developed as part of this project played and is still playing a key role in the work done in other projects. In particular, it was the technical basis for the year 1 deliverable for the Calo project, funded under the Perceptive Assistant that Learns (PAL) program. We are currently applying the same method to the project under a learning seedling, to identify and recognize objects in a 3D scene.

Link Prediction in Relational Data

Ben Taskar Ming-Fai Wong Pieter Abbeel Daphne Koller
 {btaskar, mingfai.wong, abbeel, koller}@cs.stanford.edu
 Stanford University

Abstract

Many real-world domains are *relational* in nature, consisting of a set of objects related to each other in complex ways. This paper focuses on predicting the existence and the type of links between entities in such domains. We apply the *relational Markov network* framework of Taskar *et al.* to define a joint probabilistic model over the entire link graph — entity attributes and links. The application of the RMN algorithm to this task requires the definition of probabilistic patterns over subgraph structures. We apply this method to two new relational datasets, one involving university webpages, and the other a social network. We show that the collective classification approach of RMNs, and the introduction of subgraph patterns over link labels, provide significant improvements in accuracy over flat classification, which attempts to predict each link in isolation.

1 Introduction

Many real world domains are richly structured, involving entities of multiple types that are related to each other through a network of different types of links. Such data poses new challenges to machine learning. One challenge arises from the task of predicting which entities are related to which others and what are the types of these relationships. For example, in a data set consisting of a set of hyperlinked university webpages, we might want to predict not just which page belongs to a professor and which to a student, but also which professor is which student’s advisor. In some cases, the existence of a relationship will be predicted by the presence of a hyperlink between the pages, and we will have only to decide whether the link reflects an advisor-advisee relationship. In other cases, we might have to infer the very existence of a link from indirect evidence, such as a large number of co-authored papers. In a very different application, we might want to predict links representing participation of individuals in certain terrorist activities.

One possible approach to this task is to consider the presence and/or type of the link using only attributes of the potentially linked entities and of the link itself. For example, in our university example, we might try to predict and classify the link using the words on the two webpages, and the anchor words on the link (if present). This approach has the advantage that it reduces to a simple classification task and we can apply standard machine learning techniques. However, it completely ignores a rich source of information that is unique to this task — the graph structure of the link graph. For example, a strong predictor of an advisor-advisee link between a professor and a student is the fact that they jointly participate in several projects. In general, the link graph typically reflects common patterns of interactions between the entities in the domain. Taking these patterns into consideration should allow us to provide a much better prediction for links.

In this paper, we tackle this problem using the *relational Markov network (RMN)* framework of Taskar *et al.* [14]. We use this framework to define a single probabilistic model over the entire link graph, including both object labels (when relevant) and links between

objects. The model parameters are trained discriminatively, to maximize the probability of the (object and) link labels given the known attributes (e.g., the words on the page, hyperlinks). The learned model is then applied, using probabilistic inference, to predict and classify links using any observed attributes and links.

2 Link Prediction

A relational domain is described by a *relational schema*, which specifies a set of object types and attributes for them. In our web example, we have a **Webpage** type, where each page has a binary-valued attribute for each word in the dictionary, denoting whether the page contains the word. It also has an attribute representing the “class” of the webpage, e.g., a professor’s homepage, a student’s homepage, etc.

To address the link prediction problem, we need to make links first-class citizens in our model. Following [5], we introduce into our schema object types that correspond to links between entities. Each link object ℓ is associated with a tuple of entity objects $(o_1, \dots, o_k)$ that participate in the link. For example, a **Hyperlink** link object would be associated with a pair of entities — the linking page, and the linked-to page, which are part of the link definition. We note that link objects may also have other attributes; e.g., a hyperlink object might have attributes for the anchor words on the link.

As our goal is to predict link existence, we must consider links that exist and links that do not. We therefore consider a set of *potential* links between entities. Each potential link is associated with a tuple of entity objects, but it may or may not actually exist. We denote this event using a binary *existence* attribute *Exists*, which is *true* if the link between the associated entities exists and *false* otherwise. In our example, our model may contain a potential link ℓ for each pair of webpages, and the value of the variable $\ell.Exists$ determines whether the link actually exists or not. The link prediction task now reduces to the problem of predicting the existence attributes of these link objects.

An *instantiation* $\mathcal{I}$ specifies the set of entities of each entity type and the values of all attributes for all of the entities. For example, an instantiation of the hypertext schema is a collection of webpages, specifying their labels, the words they contain, and which links between them exist. A partial instantiation specifies the set of objects, and values for some of the attributes. In the link prediction task, we might observe all of the attributes for all of the objects, except for the existence attributes for the links. Our goal is to predict these latter attributes given the rest.

3 Relational Markov Networks

We begin with a brief review of the framework of undirected graphical models or *Markov Networks* [13], and their extension to relational domains presented in [14].

Let $\mathbf{V}$ denote a set of discrete random variables and $\mathbf{v}$ an assignment of values to $\mathbf{V}$. A Markov network for $\mathbf{V}$ defines a joint distribution over $\mathbf{V}$. It consists of an undirected dependency graph, and a set of parameters associated with the graph. For a graph G , a *clique* c is a set of nodes $\mathbf{V}_c$ in G , not necessarily maximal, such that each $V_i, V_j \in \mathbf{V}_c$ are connected by an edge in G . Each clique c is associated with a *clique potential* $\phi_c(\mathbf{V}_c)$, which is a non-negative function defined on the joint domain of $\mathbf{V}_c$. Letting $C(G)$ be the set of cliques, the Markov network defines the distribution $P(\mathbf{v}) = \frac{1}{Z} \prod_{c \in C(G)} \phi_c(\mathbf{v}_c)$, where Z is the standard normalizing *partition function*.

A *relational Markov network (RMN)* [14] specifies the cliques and potentials between attributes of related entities at a template level, so a single model provides a coherent distribution for any collection of instances from the schema. RMNs specify the cliques using the notion of a *relational clique template*, which specify tuples of variables in the instantiation using a relational query language. (See [14] for details.)

For example, if we want to define cliques between the class labels of linked pages, we might define a clique template that applies to all pairs *page1, page2* and *link* of types

Webpage, Webpage and Hyperlink, respectively, such that *link* points from *page1* to *page2*. We then define a potential template that will be used for all pairs of variables *page1.Category* and *page2.Category* for such *page1* and *page2*.

Given a particular instantiation $\mathcal{I}$ of the schema, the RMN $\mathcal{M}$ produces an *unrolled* Markov network over the attributes of entities in $\mathcal{I}$, in the obvious way. The cliques in the unrolled network are determined by the clique templates C . We have one clique for each $c \in C(\mathcal{I})$, and all of these cliques are associated with the same clique potential ϕ_C .

Taskar *et al.* show how the parameters of an RMN over a fixed set of clique templates can be learned from data. In this case, the training data is a single instantiation $\mathcal{I}$, where the same parameters are used multiple times — once for each different entity that uses a feature. A choice of clique potential parameters $\mathbf{w}$ specifies a particular RMN, which induces a probability distribution $P_{\mathbf{w}}$ over the unrolled Markov network.

Gradient descent over $\mathbf{w}$ is used to optimize the conditional likelihood of the target variables given the observed variables in the training set. The gradient involves a term which is the posterior probability of the target variables given the observed, whose computation requires that we run probabilistic inference over the entire unrolled Markov network. In relational domains, this network is typically large and densely connected, making exact inference intractable. Taskar *et al.* therefore propose the use of belief propagation [13, 17].

4 Subgraph Templates in a Link Graph

The structure of link graphs has been widely used to infer importance of documents in scientific publications [4] and hypertext (PageRank [12], Hubs and Authorities [8]). Social networks have been extensively analyzed in their own right in order to quantify trends in social interactions [16]. Link graph structure has also been used to improve document classification [7, 6, 15].

In our experiments, we found that the combination of a relational language with a probabilistic graphical model provides a very flexible framework for modeling complex patterns common in relational graphs. First, as observed by Getoor *et al.* [5], there are often correlations between the attributes of entities and the relations in which they participate. For example, in a social network, people with the same hobby are more likely to be friends.

We can also exploit correlations between the *labels* of entities and the relation type. For example, only students can be teaching assistants in a course. We can easily capture such correlations by introducing cliques that involve these attributes. Importantly, these cliques are informative even when attributes are not observed in the test data. For example, if we have evidence indicating an advisor-advisee relationship, our probability that X is a faculty member increases, and thereby our belief that X participates in a teaching assistant link with some entity Z decreases.

We also found it useful to consider richer subgraph templates over the link graph. One useful type of template is a *similarity* template, where objects that share a certain graph-based property are more likely to have the same label. Consider, for example, a professor X and two other entities Y and Z. If X’s webpage mentions Y and Z in the same context, it is likely that the X-Y relation and the Y-Z relation are of the same type; for example, if Y is Professor X’s advisee, then probably so is Z. Our framework accomodates these patterns easily, by introducing pairwise cliques between the appropriate relation variables.

Another useful type of subgraph template involves *transitivity* patterns, where the presence of an A-B link and of a B-C link increases (or decreases) the likelihood of an A-C link. For example, students often assist in courses taught by their advisor. Note that this type of interaction cannot be accounted for just using pairwise cliques. By introducing cliques over triples of relations, we can capture such patterns as well. We can incorporate even more complicated patterns, but of course we are limited by the ability of belief propagation to scale up as we introduce larger cliques and tighter loops in the Markov network.

We note that our ability to model these more complex graph patterns relies on our use

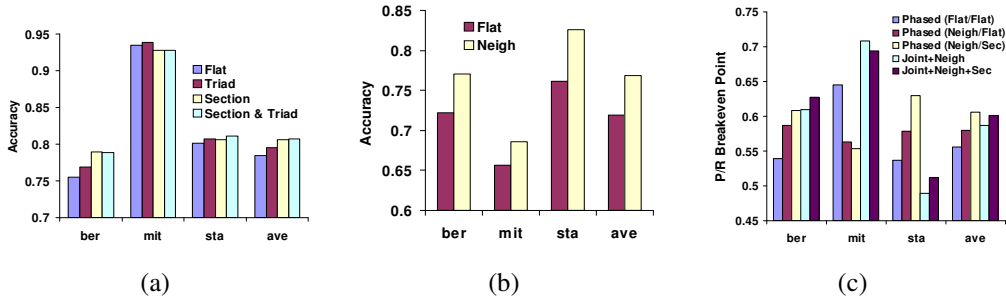


Figure 1: (a) Relation prediction with entity labels given. Relational models on average performed better than the baseline Flat model. (b) Entity label prediction. Relational model Neigh performed significantly better. (c) Relation prediction without entity labels. Relational models performed better most of the time, even though there are schools that some models performed worse.

of an undirected Markov network as our probabilistic model. In contrast, the approach of Getoor *et al.* uses directed graphical models (Bayesian networks and PRMs [9]) to represent a probabilistic model of both relations and attributes. Their approach easily captures the dependence of link existence on attributes of entities. But the constraint that the probabilistic dependency graph be a directed acyclic graph makes it hard to see how we would represent the subgraph patterns described above. For example, for the transitivity pattern, we might consider simply directing the correlation edges between link existence variables arbitrarily. However, it is not clear how we would then parameterize a link existence variable for a link that is involve in multiple triangles. See [15] for further discussion.

5 Experiments on Web Data

We collected and manually labeled a new relational dataset inspired by WebKB [2]. Our dataset consists of Computer Science department webpages from 3 schools: Stanford, Berkeley, and MIT. A total of 2954 of pages are labeled into one of eight categories: faculty, student, research scientist, staff, research group, research project, course and organization (organization refers to any large entity that is not a research group). *Owned pages*, which are owned by an entity but are not the main page for that entity, were manually assigned to that entity. The average distribution of classes across schools is: organization (9%), student (40%), research group (8%), faculty (11%), course (16%), research project (7%), research scientist (5%), and staff (3%).

We established a set of candidate links between entities based on evidence of a relation between them. One type of evidence for a relation is a hyperlink from an entity page or one of its owned pages to the page of another entity. A second type of evidence is a *virtual link*: We assigned a number of aliases to each page using the page title, the anchor text of incoming links, and email addresses of the entity involved. Mentioning an alias of a page on another page constitutes a virtual link. The resulting set of 7161 candidate links were labeled as corresponding to one of five relation types — Advisor (faculty, student), Member (research group/project, student/faculty/research scientist), Teach (faculty/research scientist/staff, course), TA (student, course), Part-Of (research group, research proj) — or “none”, denoting that the link does not correspond to any of these relations.

The observed attributes for each page are the words on the page itself and the “meta-words” on the page — the words in the title, section headings, anchors to the page from other pages. For links, the observed attributes are the anchor text, text just before the link (hyperlink or virtual link), and the heading of the section in which the link appears.

Our task is to predict the relation type, if any, for all the candidate links. We tried two settings for our experiments: with page categories observed (in the test data) and page categories unobserved. For all our experiments, we trained on two schools and tested on

the remaining school.

Observed Entity Labels. We first present results for the setting with observed page categories. Given the page labels, we can rule out many impossible relations; the resulting label breakdown among the candidate links is: none (38%), member (34%), part-of (4%), advisor (11%), teach (9%), TA (5%).

There is a huge range of possible models that one can apply to this task. We selected a set of models that we felt represented some range of patterns that manifested in the data.

Link-Flat is our baseline model, predicting links one at a time using multinomial logistic regression. This is a strong classifier, and its performance is competitive with other classifiers (e.g., support vector machines). The features used by this model are the labels of the two linked pages and the words on the links going from one page and its owned pages to the other page. The number of features is around 1000.

The relational models try to improve upon the baseline model by modeling the interactions between relations and predicting relations jointly. The **Section** model introduces cliques over relations whose links appear consecutively in a section on a page. This model tries to capture the pattern that similarly related entities (e.g., advisees, members of projects) are often listed together on a webpage. This pattern is a type of similarity template, as described in Section 4. The **Triad** model is a type of transitivity template, as discussed in Section 4. Specifically, we introduce cliques over sets of three candidate links that form a triangle in the link graph. The **Section + Triad** model includes the cliques of the two models above.

As shown in Fig. 1(a), both the **Section** and **Triad** models outperform the flat model, and the combined model has an average accuracy gain of 2.26%, or 10.5% relative reduction in error. As we only have three runs (one for each school), we cannot meaningfully analyze the statistical significance of this improvement.

As an example of the interesting inferences made by the models, we found a student-professor pair that was misclassified by the **Flat** model as none (there is only a single hyperlink from the student’s page to the advisor’s) but correctly identified by both the **Section** and **Triad** models. The **Section** model utilizes a paragraph on the student’s webpage describing his research, with a section of links to his research groups and the link to his advisor. Examining the parameters of the **Section** model clique, we found that the model learned that it is likely for people to mention their research groups and advisors in the same section. By capturing this trend, the **Section** model is able to increase the confidence of the student-advisor relation. The **Triad** model corrects the same misclassification in a different way. Using the same example, the **Triad** model makes use of the information that both the student and the teacher belong to the same research group, and the student TAed a class taught by his advisor. It is important to note that none of the other relations are observed in the test data, but rather the model bootstraps its inferences.

Unobserved Entity Labels. When the labels of pages are not known during relations prediction, we cannot rule out possible relations for candidate links based on the labels of participating entities. Thus, we have many more candidate links that do not correspond to any of our relation types (e.g., links between an organization and a student). This makes the existence of relations a very low probability event, with the following breakdown among the potential relations: none (71%), member (16%), part-of (2%), advisor (5%), teach (4%), TA (2%). In addition, when we construct a Markov network in which page labels are not observed, the network is much larger and denser, making the (approximate) inference task much harder. Thus, in addition to models that try to predict page entity and relation labels simultaneously, we also tried a two-phase approach, where we first predict page categories, and then use the predicted labels as features for the model that predicts relations.

For predicting page categories, we compared two models. **Entity-Flat** model is multinomial logistic regression that uses words and “meta-words” from the page and its owned pages in separate “bags” of words. The number of features is roughly 10,000. The **Neighbors** model is a relational model that exploits another type of similarity template: pages

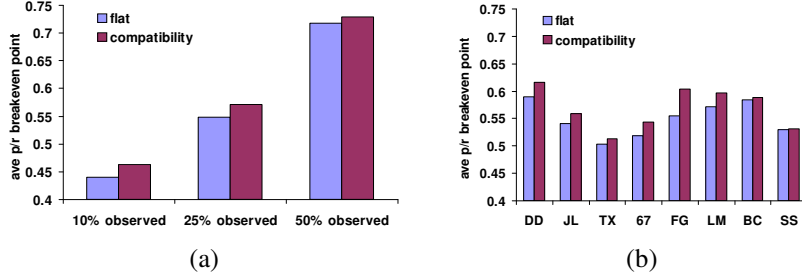


Figure 2: (a) Average precision/recall breakeven point for 10%, 25%, 50% observed links. (b) Average precision/recall breakeven point for each fold of school residences at 25% observed links.

with similar urls often belong to the same category or tightly linked categories (research group/project, professor/course). For each page, two pages with urls closest in edit distance are selected as “neighbors”, and we introduced pairwise cliques between “neighboring” pages. Fig. 1(b) shows that the **Neighbors** model clearly outperforms the **Flat** model across all schools, by an average of 4.9% accuracy gain.

Given the page categories, we can now apply the different models for link classification. Thus, the **Phased (Flat/Flat)** model uses the **Entity-Flat** model to classify the page labels, and then the **Link-Flat** model to classify the candidate links using the resulting entity labels. The **Phased (Neighbors/Flat)** model uses the **Neighbors** model to classify the entity labels, and then the **Link-Flat** model to classify the links. The **Phased (Neighbors/Section)** model uses the **Neighbors** to classify the entity labels and then the **Section** model to classify the links.

We also tried two models that predict page and relation labels simultaneously. The **Joint + Neighbors** model is simply the union of the **Neighbors** model for page categories and the **Flat** model for relation labels given the page categories. The **Joint + Neighbors + Section** model additionally introduces the cliques that appeared in the **Section** model between links that appear consecutively in a section on a page. We train the joint models to predict both page and relation labels simultaneously.

As the proportion of the “none” relation is so large, we use the probability of “none” to define a precision-recall curve. If this probability is less than some threshold, we predict the most likely label (other than none), otherwise we predict the most likely label (including none). As usual, we report results at the precision-recall breakeven point on the test data. Fig. 1(c) show the breakeven points achieved by the different models on the three schools. Relational models, both phased and joint, did better than flat models on the average. However, performance varies from school to school and for both joint and phased models, performance on one of the schools is worse than that of the flat model.

6 Experiments on Social Network Data

The second dataset we used has been collected by a portal website at a large university that hosts an online community for students [1]. Among other services, it allows students to enter information about themselves, create lists of their friends and browse the social network. Personal information includes residence, gender, major and year, as well as favorite sports, music, books, social activities, etc. We focused on the task of predicting the “friendship” links between students from their personal information and a subset of their links. We selected students living in sixteen different residences or dorms and restricted the data to the friendship links only within each residence, eliminating inter-residence links from the data to generate independent training/test splits. Each residence has about 15–25 students and an average student lists about 25% of his or her house-mates as friends.

We used an eight-fold train-test split, where we trained on fourteen residences and tested on two. Predicting links between two students from just personal information alone is a

very difficult task, so we tried a more realistic setting, where some proportion of the links is observed in the test data, and can be used as evidence for predicting the remaining links. We used the following proportions of observed links in the test data: 10%, 25%, and 50%. The observed links were selected at random, and the results we report are averaged over five folds of these random selection trials.

Using just the observed portion of links, we constructed the following flat features: for each student, the proportion of students in the residence that list him/her and the proportion of students he/she lists; for each pair of students, the proportion of other students they have as common friends. The values of the proportions were discretized into four bins. These features capture some of the relational structure and dependencies between links: Students who list (or are listed by) many friends in the observed portion of the links tend to have links in the unobserved portion as well. More importantly, having friends in common increases the likelihood of a link between a pair of students.

The Flat model uses logistic regression with the above features as well as personal information about each user. In addition to individual characteristics of the two people, we also introduced a feature for each match of a characteristic, for example, both people are computer science majors or both are freshmen.

The Compatibility model uses a type of similarity template, introducing cliques between each pair of links emanating from each person. Similarly to the Flat model, these cliques include a feature for each match of the characteristics of the two potential friends. This model captures the tendency of a person to have friends who share many characteristics (even though the person might not possess them). For example, a student may be friends with several CS majors, even though he is not a CS major himself. We also tried models that used transitivity templates, but the approximate inference with 3-cliques often failed to converge or produced erratic results.

Fig. 2(a) compares the average precision/recall breakpoint achieved by the different models at the three different settings of observed links. Fig. 2(b) shows the performance on each of the eight folds containing two residences each. Using a paired t-test, the Compatibility model outperforms Flat with p-values 0.0036, 0.00064 and 0.054 respectively.

7 Discussion and Conclusions

In this paper, we consider the problem of link prediction in relational domains. We focus on the task of collective link classification, where we are simultaneously trying to predict and classify an entire set of links in a link graph. We show that the use of a probabilistic model over link graphs allows us to represent and exploit interesting subgraph patterns in the link graph. Specifically, we have found two types of patterns that seem to be beneficial in several places. Similarity templates relate the classification of links or objects that share a certain graph-based property (e.g., links that share a common endpoint). Transitivity templates relate triples of objects and links organized in a triangle. We show that the use of these patterns significantly improve the classification accuracy over flat models.

Relational Markov networks are not the only method one might consider applying to the link prediction and classification task. We could, for example, build a link predictor that considers other links in the graph by converting graph features into flat features [11], as we did in the social network data. As our experiments show, even with these features, the collective prediction approach work better. Another approach is to use relational classifiers such as variants of *inductive logic programming* [10]. Generally, however, these methods have been applied to the problem of predicting or classifying a single link at a time. It is not clear how well they would extend to the task of simultaneously predicting an entire link graph. Finally, we could apply the directed PRM framework of [5]. However, as shown in [15], the discriminatively trained RMNs perform significantly better than generatively trained PRMs even on the simpler entity classification task. Furthermore, as we discussed, the PRM framework cannot represent (in any natural way) the type of subgraph patterns that seem prevalent in link graph data. Therefore, the RMN framework seems much more

appropriate for this task.

Although the RMN framework worked fairly well on this task, there is significant room for improvement. One of the key problems limiting the applicability of approach is the reliance on belief propagation, which often does not converge in more complex problems. This problem is especially acute in the link prediction problem, where the presence of all potential links leads to densely connected Markov networks with many short loops. This problem can be addressed with heuristics that focus the search on links that are plausible (as we did in a very simple way in the webpage experiments). A more interesting solution would be to develop a more integrated approximate inference / learning algorithm.

Our results use a set of relational patterns that we have discovered to be useful in the domains that we have considered. However, many other rich and interesting patterns are possible. Thus, in the relational setting, even more so than in simpler tasks, the issue of feature construction is critical. It is therefore important to explore the problem of automatic feature induction, as in [3].

Finally, we believe that the problem of modeling link graphs has numerous other applications, including: analyzing communities of people and hierarchical structure of organizations, identifying people or objects that play certain key roles, predicting current and future interactions, and more.

Acknowledgments. This work was supported by ONR Contract F3060-01-2-0564-P00002 under DARPA's EELD program. P. Abbeel was supported by a Siebel Grad. Fellowship.

References

- [1] L. Adamic, O. Buyukkokten, and E. Adar. A social network caught in the web. <http://www.hpl.hp.com/shl/papers/social/>, 2002.
- [2] 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.
- [3] S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(4):380–393, 1997.
- [4] L. Egghe and R. Rousseau. *Introduction to Informetrics*. Elsevier, 1990.
- [5] L. Getoor, N. Friedman, D. Koller, and B. Taskar. Probabilistic models of relational structure. In *Proc. ICML*, 2001.
- [6] L. Getoor, E. Segal, B. Taskar, and D. Koller. Probabilistic models of text and link structure for hypertext classification. In *IJCAI Workshop on Text Learning: Beyond Supervision*, 2001.
- [7] R. Ghani, S. Slattery, and Y. Yang. Hypertext categorization using hyperlink patterns and meta data. In *Proc ICML*, 2001.
- [8] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *JACM*, 46(5):604–632, 1999.
- [9] D. Koller and A. Pfeffer. Probabilistic frame-based systems. In *Proc. AAAI98*, pages 580–587, 1998.
- [10] Nada Lavrač and Saso Džeroski. *Inductive Logic Programming: Techniques and Applications*. Ellis Horwood, 1994.
- [11] J. Neville and D. Jensen. Iterative classification in relational data. In *AAAI Workshop on Learning Statistical Models from Relational Data*, 2000.
- [12] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford University, 1998.
- [13] J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, 1988.
- [14] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proc. UAI*, 2002.
- [15] B. Taskar, E. Segal, and D. Koller. Probabilistic classification and clustering in relational data. In *Proc. IJCAI*, pages 870–876, 2001.
- [16] S. Wasserman and P. Pattison. Logit models and logistic regression for social networks. *Psychometrika*, 61(3):401–425, 1996.
- [17] J. Yedidia, W. Freeman, and Y. Weiss. Generalized belief propagation. In *Proc. NIPS*, 2000.

Learning Associative Markov Networks

Ben Taskar
Vassil Chatalbashev
Daphne Koller

BTASKAR@CS.STANFORD.EDU
VASCO@CS.STANFORD.EDU
KOLLER@CS.STANFORD.EDU

Computer Science Department, Stanford University, Stanford, CA

Abstract

Markov networks are extensively used to model complex sequential, spatial, and relational interactions in fields as diverse as image processing, natural language analysis, and bioinformatics. However, inference and learning in general Markov networks is intractable. In this paper, we focus on learning a large subclass of such models (called *associative Markov networks*) that are tractable or closely approximable. This subclass contains networks of discrete variables with K labels each and clique potentials that favor the same labels for all variables in the clique. Such networks capture the “guilt by association” pattern of reasoning present in many domains, in which connected (“associated”) variables tend to have the same label. Our approach exploits a linear programming relaxation for the task of finding the best joint assignment in such networks, which provides an approximate quadratic program (QP) for the problem of learning a margin-maximizing Markov network. We show that for associative Markov network over binary-valued variables, this approximate QP is guaranteed to return an optimal parameterization for Markov networks of arbitrary topology. For the non-binary case, optimality is not guaranteed, but the relaxation produces good solutions in practice. Experimental results with hypertext and newswire classification show significant advantages over standard approaches.

1. Introduction

Numerous classification methods have been developed for the principal machine learning problem of assigning to a single object one of K labels consistent with its properties. Many classification problems, however, involve sets of related objects whose labels must also be consistent with each other. In hypertext or bibliographic classification, labels of linked and co-cited documents tend to be similar (Chakrabarti et al., 1998; Taskar et al., 2002). In proteomic analysis, lo-

cation and function of proteins that interact are often highly correlated (Vazquez et al., 2003). In image processing, neighboring pixels exhibit local label coherence in denoising, segmentation and stereo correspondence (Besag, 1986; Boykov et al., 1999a).

Markov networks compactly represent complex joint distributions of the label variables by modeling their local interactions. Such models are encoded by a graph, whose nodes represent the different object labels, and whose edges represent direct dependencies between them. For example, a Markov network for the hypertext domain would include a node for each webpage, encoding its label, and an edge between any pair of webpages whose labels are directly correlated (e.g., because one links to the other).

There has been growing interest in training Markov networks for the purpose of collectively classifying sets of related instances. The focus has been on discriminative training, which, given enough data, generally provides significant improvements in classification accuracy over generative training. For example, Markov networks can be trained to maximize the conditional likelihood of the labels given the features of the objects (Lafferty et al., 2001; Taskar et al., 2002). Recently, maximum margin-based training has been shown to additionally boost accuracy over conditional likelihood methods and allow a seamless integration of kernel methods with Markov networks (Taskar et al., 2003a).

The chief computational bottleneck in this task is inference in the underlying network, which is a core subroutine for all methods for training Markov networks. Probabilistic inference is NP-hard in general, and requires exponential time in a broad range of practical Markov network structures, including grid-topology networks (Besag, 1986). One can address the tractability issue by limiting the structure of the underlying network. In some cases, such as the the quad-tree model used for image segmentation (Bouman & Shapiro, 1994), a tractable structure is determined in advance. In other cases (e.g., (Bach & Jordan, 2001)),

Appearing in *Proceedings of the 21st International Conference on Machine Learning*, Banff, Canada, 2004. Copyright 2004 by the first author.

the network structure is learned, subject to the constraint that inference on these networks is tractable. In many cases, however, the topology of the Markov network does not allow tractable inference. In the hypertext domain, the network structure mirrors the hyperlink graph, which is usually highly interconnected, leading to computationally intractable networks.

In this paper, we show that optimal learning is feasible for an important subclass of Markov networks — networks with *attractive potentials*. This subclass, which we call *associative Markov networks (AMNs)*, contains networks of discrete variables with K labels each and arbitrary-size clique potentials with K parameters that favor the same label for all variables in the clique. Such positive interactions capture the “guilt by association” pattern of reasoning present in many domains, in which connected (“associated”) variables tend to have the same label. AMNs are a natural fit for object recognition and segmentation, webpage classification, and many other applications.

Our analysis is based on the maximum margin approach to training Markov networks, presented by Taskar *et al.* (2003a). In this formulation, the learning task is to find the Markov network parameterization that achieves the highest confidence in the target labels. In other words, the goal is to maximize the margin between the target labels and any other label assignment. The inference subtask in this formulation of the learning problem is one of finding the best joint (MAP) assignment to all of the variables in a Markov network. By contrast, other learning tasks (e.g., maximizing the conditional likelihood of the target labels given the features) often require that we compute the posterior probabilities of different label assignments, rather than just the MAP.

The MAP problem can naturally be expressed as an integer programming problem. We show how we can approximate the maximum margin Markov network learning task as a quadratic program that uses a linear program (LP) relaxation of this integer program. This quadratic program can be solved in polynomial time using standard techniques. We show that whenever the MAP LP relaxation is guaranteed to return integer solutions, the approximate max-margin QP provides an optimal solution to the max-margin optimization task. In particular, for associative Markov networks over binary variables ($K = 2$), this linear program provides exact answers. For the non-binary case ($K > 2$), the approximate quadratic program is not guaranteed to be optimal, but our empirical results suggest that the solutions work well in practice. To our knowledge, our method is the first to allow training Markov networks of arbitrary topology.

2. Markov Networks

We restrict attention to networks over discrete variables $\mathbf{Y} = \{Y_1, \dots, Y_N\}$, where each variable corresponds to an object we wish to classify and has K possible labels: $Y_i \in \{1, \dots, K\}$. An assignment of values to $\mathbf{Y}$ is denoted by $\mathbf{y}$. A Markov network for $\mathbf{Y}$ defines a joint distribution over $\{1, \dots, K\}^N$.

A Markov network is defined by an undirected graph over the nodes $\mathbf{Y} = \{Y_1, \dots, Y_N\}$. In general, a Markov network is a set of *cliques* $\mathcal{C}$, where each clique $c \in \mathcal{C}$ is associated with a subset Y_c of $\mathbf{Y}$. The nodes Y_i in a clique c form a fully connected subgraph (a clique) in the Markov network graph. Each clique is accompanied by a *potential* $\phi_c(Y_c)$, which associates a non-negative value with each assignment $\mathbf{y}_c$ to Y_c . The Markov network defines the probability distribution:

$$P_\phi(\mathbf{y}) = \frac{1}{Z} \prod_{c \in \mathcal{C}} \phi_c(\mathbf{y}_c)$$

where Z is the *partition function* given by $Z = \sum_{\mathbf{y}'} \prod_{c \in \mathcal{C}} \phi_c(\mathbf{y}_c')$.

For simplicity of exposition, we focus most of our discussion on *pairwise* Markov networks. We extend our results to higher-order interactions in Sec. 3. A pairwise Markov network is simply a Markov network where all of the cliques involve either a single node or a pair of nodes. Thus, in a pairwise Markov network with edges $E = \{(ij)\}$ ($i < j$), only nodes and edges are associated with potentials $\phi_i(Y_i)$ and $\phi_{ij}(Y_i, Y_j)$. A pairwise Markov net defines the distribution

$$P_\phi(\mathbf{y}) = \frac{1}{Z} \prod_{i=1}^N \phi_i(y_i) \prod_{(ij) \in E} \phi_{ij}(y_i, y_j),$$

where Z is the *partition function* given by $Z = \sum_{\mathbf{y}'} \prod_{i=1}^N \phi_i(y'_i) \prod_{(ij) \in E} \phi_{ij}(y'_i, y'_j)$.

The node and edge potentials are functions of the features of the objects $\mathbf{x}_i \in \mathbb{R}^{d_n}$ and features of the relationships between them $\mathbf{x}_{ij} \in \mathbb{R}^{d_e}$. In hypertext classification, $\mathbf{x}_i$ might be the counts of the words of the document i , while $\mathbf{x}_{ij}$ might be the words surrounding the hyperlink(s) between documents i and j . The simplest model of dependence of the potentials on the features is a log-linear combination: $\log \phi_i(k) = \mathbf{w}_n^k \cdot \mathbf{x}_i$ and $\log \phi_{ij}(k, l) = \mathbf{w}_e^{k,l} \cdot \mathbf{x}_{ij}$, where $\mathbf{w}_n^k$ and $\mathbf{w}_e^{k,l}$ are label-specific row vectors of node and edge parameters, of size d_n and d_e , respectively. Note that this formulation assumes that all of the nodes in the network share the same set of weights, and similarly all of the edges share the same weights.

We represent an assignment $\mathbf{y}$ as a set of $K \cdot N$ indicators $\{y_i^k\}$, where $y_i^k = I(y_i = k)$. With these definitions, the log of conditional probability $\log P_{\mathbf{w}}(\mathbf{y} \mid \mathbf{x})$

is given by:

$$\sum_{i=1}^N \sum_{k=1}^K (\mathbf{w}_n^k \cdot \mathbf{x}_i) y_i^k + \sum_{(ij) \in E} \sum_{k,l=1}^K (\mathbf{w}_e^{k,l} \cdot \mathbf{x}_{ij}) y_i^k y_j^l - \log Z_{\mathbf{w}}(\mathbf{x}).$$

Note that the partition function $Z_{\mathbf{w}}(\mathbf{x})$ above depends on the parameters $\mathbf{w}$ and input features $\mathbf{x}$, but not on the labels y_i 's.

For compactness of notation, we define the node and edge weight vectors $\mathbf{w}_n = (\mathbf{w}_n^1, \dots, \mathbf{w}_n^K)$ and $\mathbf{w}_e = (\mathbf{w}_e^{1,1}, \dots, \mathbf{w}_e^{K,K})$, and let $\mathbf{w} = (\mathbf{w}_n, \mathbf{w}_e)$ be a vector of all the weights, of size $d = Kd_n + K^2d_e$. Also, we define the node and edge labels vectors, $\mathbf{y}_n = (\dots, y_i^1, \dots, y_i^K, \dots)^\top$ and $\mathbf{y}_e = (\dots, y_{ij}^{1,1}, \dots, y_{ij}^{K,K}, \dots)^\top$, where $y_{ij}^{k,l} = y_i^k y_j^l$, and the vector of all labels $\mathbf{y} = (\mathbf{y}_n, \mathbf{y}_e)$ of size $L = KN + K^2|E|$. Finally, we define an appropriate $d \times L$ matrix $\mathbf{X}$ such that

$$\log P_{\mathbf{w}}(\mathbf{y} \mid \mathbf{x}) = \mathbf{w} \mathbf{X} \mathbf{y} - \log Z_{\mathbf{w}}(\mathbf{x}).$$

The matrix $\mathbf{X}$ contains the node feature vectors $\mathbf{x}_i$ and edge feature vectors $\mathbf{x}_{ij}$ repeated multiple times (for each label k or label pair k, l respectively), and padded with zeros appropriately.

A key task in Markov networks is computing the *MAP (maximum a posteriori) assignment* — the assignment $\mathbf{y}$ that maximizes $\log P_{\mathbf{w}}(\mathbf{y} \mid \mathbf{x})$. It is straightforward to formulate the MAP inference task as an integer linear program: The variables are the assignments to the nodes y_i^k and edges $y_{ij}^{k,l}$ which must be in the set $\{0, 1\}$, and satisfy linear normalization and agreement constraints. The optimization criterion is simply the linear function $\mathbf{w} \mathbf{X} \mathbf{y}$, which corresponds to the log of the unnormalized probability of the assignment $\mathbf{y}$.

In certain cases, we can take this integer program, and approximate it as a linear program by relaxing the integrality constraints on y_i^k , with appropriate constraints. For example, Wainwright *et al.* (2002) provides a natural formulation of this form that is guaranteed to produce integral solutions for triangulated graphs.

3. Associative Markov Networks

We now describe one important subclass of problems for which the above relaxation is particularly useful. These networks, which we call *associative Markov networks (AMNs)*, encode situations where related variables tend to have the same value.

Associative interactions arise naturally in the context of image processing, where nearby pixels are likely to have the same label (Besag, 1986; Boykov *et al.*, 1999b). In this setting, a common approach is to use a

generalized Potts model (Potts, 1952), which penalizes assignments that do not have the same label across the edge: $\phi_{ij}(k, l) = \lambda_{ij}$, $\forall k \neq l$ and $\phi_{ij}(k, k) = 1$, where $\lambda_{ij} \leq 1$.

For binary-valued Potts models, Greig *et al.* (1989) show that the MAP problem can be formulated as a min-cut in an appropriately constructed graph. Thus, the MAP problem can be solved exactly for this class of models in polynomial time. For $K > 2$, the MAP problem is NP-hard, but a procedure based on a relaxed linear program guarantees a factor 2 approximation of the optimal solution (Boykov *et al.*, 1999b; Kleinberg & Tardos, 1999). Kleinberg and Tardos (1999) extend the multi-class Potts model to have more general edge potentials, under the constraints that negative log potentials $-\log \phi_{ij}(k, l)$ form a metric on the set of labels. They also provide a solution based on a relaxed LP that has certain approximation guarantees.

More recently, Kolmogorov and Zabih (2002) showed how to optimize energy functions containing binary and ternary interactions using graph cuts, as long as the parameters satisfy a certain regularity condition. Our definition of associative potentials below also satisfies the Kolmogorov and Zabih regularity condition for $K = 2$. However, the structure of our potentials is simpler to describe and extend for the multi-class case. We use a linear programming formulation (instead of min-cut) for the MAP inference, which allows us to use the maximum margin estimation framework, as described below. Note however, that we can also use min-cut to perform exact inference on the learned models for $K = 2$ and also in approximate inference for $K > 2$ as in Boykov *et al.* (1999a).

Our associative potentials extend the Potts model in several ways. Importantly, AMNs allow different labels to have different attraction strength: $\phi_{ij}(k, k) = \lambda_{ij}^k$, where $\lambda_{ij}^k \geq 1$, and $\phi_{ij}(k, l) = 1$, $\forall k \neq l$. This additional flexibility is important in many domains, as different labels can have very diverse affinities. For example, foreground pixels tend to have locally coherent values while background is much more varied.

The linear programming relaxation of the MAP problem for these networks can be written as:

$$\begin{aligned} \max \quad & \sum_{i=1}^N \sum_{k=1}^K (\mathbf{w}_n^k \cdot \mathbf{x}_i) y_i^k + \sum_{(ij) \in E} \sum_{k=1}^K (\mathbf{w}_e^{k,k} \cdot \mathbf{x}_{ij}) y_{ij}^k \quad (1) \\ \text{s.t.} \quad & y_i^k \geq 0, \quad \forall i, k; \quad \sum_k y_i^k = 1, \quad \forall i; \\ & y_{ij}^k \leq y_i^k, \quad y_{ij}^k \leq y_j^k, \quad \forall (ij) \in E, k. \end{aligned}$$

Note that we substitute the constraint $y_{ij}^k = y_i^k \wedge y_j^k$ by two linear constraints $y_{ij}^k \leq y_i^k$ and $y_{ij}^k \leq y_j^k$. This works because the coefficient $\mathbf{w}_e^{k,k} \cdot \mathbf{x}_{ij}$ is non-

negative and we are maximizing the objective function. Hence, at the optimum $y_{ij}^k = \min(y_i^k, y_j^k)$, which is equivalent to $y_{ij}^k = y_i^k \wedge y_j^k$.

In a second important extension, AMNs admit non-pairwise interactions between variables, with potentials over cliques involving m variables $\phi(y_{i1}, \dots, y_{im})$. In this case, the clique potentials are constrained to have the same type of structure as the edge potentials: There are K parameters $\phi(k, \dots, k) = \lambda_{ij}^k$ and the rest of the entries are set to 1. In particular, using this additional expressive power, AMNs allow us to encode the pattern of (soft) transitivity present in many domains. For example, consider the problem of predicting whether two proteins interact (Vazquez et al., 2003); this probability may increase if they *both* interact with another protein. This type of transitivity could be modeled by a ternary clique that has high λ for the assignment with all interactions present.

We can write a linear program for the MAP problem similar to Eq. (1), where we have a variable y_c^k for each clique c and for each label k , which represents the event that all nodes in the clique c have label k :

$$\begin{aligned} \max \quad & \sum_{i=1}^N \sum_{k=1}^K (\mathbf{w}_n^k \cdot \mathbf{x}_i) y_i^k + \sum_{c \in \mathcal{C}} \sum_{k=1}^K (\mathbf{w}_c^k \cdot \mathbf{x}_c) y_c^k \quad (2) \\ \text{s.t.} \quad & y_i^k \geq 0, \quad \forall i, k; \quad \sum_k y_i^k = 1, \quad \forall i; \\ & y_c^k \leq y_i^k, \quad \forall c \in \mathcal{C}, i \in c, k. \end{aligned}$$

It can be shown that in the binary case, the relaxed linear programs Eq. (1) and Eq. (2) are guaranteed to produce an integer solution when a unique solution exists.

Theorem 3.1 *If $K = 2$, for any objective $\mathbf{wX}$, the linear programs in Eq. (1) and Eq. (2) have an integral optimal solution.*

See appendix for the proof. This result states that the MAP problem in binary AMNs is tractable, regardless of network topology or clique size. In the non-binary case ($K > 2$), these LPs can produce fractional solutions and we use a rounding procedure to get an integral solution. In the appendix, we also show that the approximation ratio of the rounding procedure is the inverse of the size of the largest clique (e.g., $\frac{1}{2}$ for pairwise networks). Although artificial examples with fractional solutions can be easily constructed by using symmetry, it seems that in real data such symmetries are often broken. In fact, in all our experiments with $K > 2$ on real data, we never encountered fractional solutions.

4. Max Margin Estimation

We now consider the problem of training the weights $\mathbf{w}$ of a Markov network given a labeled training instance $(\mathbf{x}, \hat{\mathbf{y}})$. For simplicity of exposition, we assume that we have only a single training instance; the extension to the case of multiple instances is entirely straightforward. Note that, in our setting, a single training instance actually contains multiple objects. For example, in the hypertext domain, an instance might be an entire website, containing many inter-linked webpages.

The M³N Framework. The standard approach of learning the weights $\mathbf{w}$ given $(\mathbf{x}, \hat{\mathbf{y}})$ is to maximize the $\log P_{\mathbf{w}}(\hat{\mathbf{y}} \mid \mathbf{x})$, with an additional regularization term, which is usually taken to be the squared-norm of the weights $\mathbf{w}$ (Lafferty et al., 2001). An alternative method, recently proposed by Taskar *et al.* (2003a), is to maximize the margin of confidence in the true label assignment $\hat{\mathbf{y}}$ over any other assignment $\mathbf{y} \neq \hat{\mathbf{y}}$. They show that the margin-maximization criterion provides significant improvements in accuracy over a range of problems. It also allows high-dimensional feature spaces to be utilized by using the kernel trick, as in support vector machines. The maximum margin Markov network (M³N) framework forms the basis for our work, so we begin by reviewing this approach.

As in support vector machines, the goal in an M³N is to maximize our confidence in the true labels $\hat{\mathbf{y}}$ relative to any other possible joint labelling $\mathbf{y}$. Specifically, we define the gain of the true labels $\hat{\mathbf{y}}$ over another possible joint labelling $\mathbf{y}$ as:

$$\log P_{\mathbf{w}}(\hat{\mathbf{y}} \mid \mathbf{x}) - \log P_{\mathbf{w}}(\mathbf{y} \mid \mathbf{x}) = \mathbf{wX}(\hat{\mathbf{y}} - \mathbf{y}).$$

In M³Ns, the desired gain takes into account the number of labels in $\mathbf{y}$ that are misclassified, $\Delta(\hat{\mathbf{y}}, \mathbf{y})$, by scaling linearly with it:

$$\max \quad \gamma \quad \text{s.t.} \quad \mathbf{wX}(\hat{\mathbf{y}} - \mathbf{y}) \geq \gamma \Delta(\hat{\mathbf{y}}, \mathbf{y}); \quad \|\mathbf{w}\|^2 \leq 1.$$

Note that the number of incorrect node labels $\Delta(\hat{\mathbf{y}}, \mathbf{y})$ can also be written as $N - \hat{\mathbf{y}}_n^\top \mathbf{y}_n$. (Whenever $\hat{y}_i$ and y_i agree on some label k , we have that $\hat{y}_i^k = 1$ and $y_i^k = 1$, adding 1 to $\hat{\mathbf{y}}_n^\top \mathbf{y}_n$.) By dividing through by γ and adding a slack variable for non-separable data, we obtain a quadratic program (QP) with exponentially many constraints:

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C\xi \quad (3) \\ \text{s.t.} \quad & \mathbf{wX}(\hat{\mathbf{y}} - \mathbf{y}) \geq N - \hat{\mathbf{y}}_n^\top \mathbf{y}_n - \xi, \quad \forall \mathbf{y} \in \mathcal{Y}. \end{aligned}$$

This QP has a constraint for every possible joint assignment $\mathbf{y}$ to the Markov network variables, resulting in an exponentially-sized QP. Taskar *et al.* show how

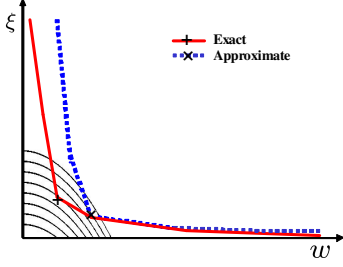


Figure 1. Exact and approximate constraints on the max-margin quadratic program. The solid red line represents the constraints imposed by integer $\mathbf{y}$'s, whereas the dashed blue line represents the stronger constraints imposed by the larger set of fractional $\mathbf{y}$'s. The fractional constraints may coincide with the integer constraints in some cases, and be more stringent in others. The parabolic contours represent the value of the objective function.

structure in the dual of this QP can be exploited to allow an efficient solution when the underlying network has low treewidth.

M³N relaxations.

As an alternative to the approach of Taskar *et al.*, we now derive a more generally applicable approach for exploiting structure and relaxations in max-margin problems. As our first step, we replace the exponential set of linear constraints in the max-margin QP of Eq. (3) with the single equivalent non-linear constraint:

$$\mathbf{w}\mathbf{X}\hat{\mathbf{y}} - N + \xi \geq \max_{\mathbf{y} \in \mathcal{Y}} \mathbf{w}\mathbf{X}\mathbf{y} - \hat{\mathbf{y}}_n^\top \mathbf{y}_n.$$

This non-linear constraint essentially requires that we find the assignment $\mathbf{y}$ to the network variables which has the highest probability relative to the parameterization $\mathbf{w}\mathbf{X} - \hat{\mathbf{y}}_n^\top$. Thus, optimizing the max-margin QP contains the MAP inference task as a component.

As we discussed earlier, we can formulate the MAP problem as an integer program, and then relax it into a linear program. Inserting the relaxed LP into the QP of Eq. (3), we obtain:

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C\xi \\ \text{s.t.} \quad & \mathbf{w}\mathbf{X}\hat{\mathbf{y}} - N + \xi \geq \max_{\mathbf{y} \in \mathcal{Y}'} \mathbf{w}\mathbf{X}\mathbf{y} - \hat{\mathbf{y}}_n^\top \mathbf{y}_n. \end{aligned} \quad (4)$$

where $\mathcal{Y}'$ is the space of all legal fractional values for $\mathbf{y}$. In effect, we obtain a QP with a continuum of constraints, one for every fractional assignment to $\mathbf{y}$.

It follows that, in cases where the relaxed LP is guaranteed to provide integer solutions, the integer and relaxed constraint sets coincide, so that the approximate QP is computing precisely the optimal max-margin solution. In the general case, the linear relaxation strengthens the constraints on $\mathbf{w}$ by potentially adding constraints corresponding to fractional assignments $\mathbf{y}$. Fig. 1 shows how the relaxation of

the max subproblem reduces the feasible space of $\mathbf{w}$ and ξ . Note that for every setting of the weights $\mathbf{w}$ that produces fractional solutions for the LP relaxation, the approximate constraints are tightened because of the additional fractional assignments $\mathbf{y}$. In this case, the fractional MAP solution is better than any integer solution, including $\hat{\mathbf{y}}$, thereby driving up the corresponding slack ξ . By contrast, for weights $\mathbf{w}$ for which the MAP LP is integer-valued, the margin has the standard interpretation as the difference between the probability of $\hat{\mathbf{y}}$ and the MAP $\mathbf{y}$ (according to $\mathbf{w}$). As the objective includes a penalty for the slack variable, intuitively, minimizing the objective tends to drive the weights $\mathbf{w}$ away from the regions where the solutions to the MAP LP are fractional.

While this insight allows us to replace the MAP integer program within the QP with a linear program, the resulting QP does not appear tractable. However, here we can exploit fundamental properties of linear programming duality (Bertsimas & Tsitsiklis, 1997). Assume that our relaxed LP for the inference task has the form:

$$\max_{\mathbf{y}} \quad \mathbf{w}\mathbf{B}\mathbf{y} \quad \text{s.t.} \quad \mathbf{y} \geq 0, \quad \mathbf{A}\mathbf{y} \leq \mathbf{b}. \quad (5)$$

for some polynomial-size $\mathbf{A}, \mathbf{B}, \mathbf{b}$. (For example, Eq. (1) and Eq. (2) can be easily written in this compact form.) The dual of this LP is given by:

$$\min_{\mathbf{z}} \quad \mathbf{b}^\top \mathbf{z} \quad \text{s.t.} \quad \mathbf{z} \geq 0, \quad \mathbf{A}^\top \mathbf{z} \geq (\mathbf{w}\mathbf{B})^\top. \quad (6)$$

When the relaxed LP is feasible and bounded, the value of Eq. (6) provides an upper bound on the primal that achieves the same value as the primal at its minimum. If we substitute Eq. (6) for Eq. (5) in the QP of Eq. (4), we obtain a quadratic program over $\mathbf{w}$, ξ and $\mathbf{z}$ with polynomially many linear constraints:

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C\xi \\ \text{s.t.} \quad & \mathbf{w}\mathbf{X}\hat{\mathbf{y}} - N + \xi \geq \mathbf{b}^\top \mathbf{z}; \\ & \mathbf{z} \geq 0, \quad \mathbf{A}^\top \mathbf{z} \geq (\mathbf{w}\mathbf{B})^\top. \end{aligned} \quad (7)$$

Our ability to perform this transformation is a direct consequence of the connection between the max-margin criterion and the MAP inference problem. The transformation is useful whenever we can solve or approximate MAP using a compact linear program.

5. Max Margin AMNs

The transformation described in the previous section applies to any situation where the MAP problem can be effectively approximated as a linear program. In particular, the LP relaxation of Eq. (1) provides

us with precisely the necessary building block to provide an effective solution for the QP in Eq. (4) for the case of AMNs. As we discussed, the MAP problem is precisely the max subproblem in this QP. In the case of AMNs, this max subproblem can be replaced with the relaxed LP of Eq. (1). In effect, we are replacing the exponential constraint set — one which includes a constraint for every discrete $\mathbf{y}$, with an infinite constraint set — one which includes a constraint for every continuous vector $\mathbf{y}$ in

$$\mathcal{Y}' = \{\mathbf{y} : y_i^k \geq 0; \sum_k y_i^k = 1; y_{ij}^k \leq y_i^k, y_{ij}^k \leq y_j^k\}$$

as defined in Eq. (1).

Stating the AMN restrictions in terms of the parameters $\mathbf{w}$, we require that $\mathbf{w}_e^{k,l} = 0, \forall k \neq l$ and $\mathbf{w}_e^{k,k} \cdot \mathbf{x}_{ij} \geq 0$. To ensure that $\mathbf{w}_e^{k,k} \cdot \mathbf{x}_{ij} \geq 0$, we simply assume (without loss of generality) that $\mathbf{x}_{ij} \geq 0$, and constrain $\mathbf{w}_e^{k,k} \geq 0$. Incorporating this constraint, we obtain our basic AMN QP:

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C\xi \\ \text{s.t.} \quad & \mathbf{w}\mathbf{X}\hat{\mathbf{y}} - N + \xi \geq \max_{\mathbf{y} \in \mathcal{Y}'} \mathbf{w}\mathbf{X}\mathbf{y} - \hat{\mathbf{y}}_n \cdot \mathbf{y}_n; \\ & \mathbf{w}_e \geq 0. \end{aligned} \quad (8)$$

We can now transform this QP as specified in Eq. (7), by taking the dual of the LP used to represent the interior max. Specifically, $\max_{\mathbf{y} \in \mathcal{Y}'} \mathbf{w}\mathbf{X}\mathbf{y} - \hat{\mathbf{y}}_n \cdot \mathbf{y}_n$ is a feasible and bounded linear program in $\mathbf{y}$, with a dual given by:

$$\begin{aligned} \min \quad & \sum_{i=1}^N z_i \\ \text{s.t.} \quad & z_i - \sum_{(ij), (ji) \in E} z_{ij}^k \geq \mathbf{w}_n^k \cdot \mathbf{x}_i - \hat{y}_i^k, \quad \forall i, k; \\ & z_{ij}^k + z_{ji}^k \geq \mathbf{w}_e^{k,k} \cdot \mathbf{x}_{ij}, \quad z_{ij}^k, z_{ji}^k \geq 0, \quad \forall (ij) \in E, k. \end{aligned}$$

In the dual, we have a variable z_i for each normalization constraint in Eq. (1) and variables z_{ij}^k, z_{ji}^k for each of the inequality constraints.

Substituting this dual into Eq. (8), we obtain:

$$\begin{aligned} \min \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C\xi \\ \text{s.t.} \quad & \mathbf{w}\mathbf{X}\hat{\mathbf{y}} - N + \xi \geq \sum_{i=1}^N z_i; \quad \mathbf{w}_e \geq 0; \\ & z_i - \sum_{(ij), (ji) \in E} z_{ij}^k \geq \mathbf{w}_n^k \cdot \mathbf{x}_i - \hat{y}_i^k, \quad \forall i, k; \\ & z_{ij}^k + z_{ji}^k \geq \mathbf{w}_e^{k,k} \cdot \mathbf{x}_{ij}, \quad z_{ij}^k, z_{ji}^k \geq 0, \quad \forall (ij) \in E, k. \end{aligned} \quad (10)$$

For $K = 2$, the LP relaxation is exact, so that Eq. (10) learns *exact* max-margin weights for

Markov networks of *arbitrary* topology. For $K > 2$, the linear relaxation leads to a strengthening of the constraints on $\mathbf{w}$ by potentially adding constraints corresponding to fractional assignments $\mathbf{y}$. Thus, the optimal choice $\mathbf{w}, \xi$ for the original QP may no longer be feasible, leading to a different choice of weights. However, as our experiments show, these weights tend to do well in practice.

The dual of Eq. (10) provides some insight into the structure of the problem:

$$\begin{aligned} \max \quad & \sum_{i=1}^N \sum_{k=1}^K (1 - \hat{y}_i^k) \mu_i^k \\ & - \frac{1}{2} \sum_{k=1}^K \left\| \sum_{i=1}^N \mathbf{x}_i (C\hat{y}_i^k - \mu_i^k) \right\|^2 \\ & - \frac{1}{2} \sum_{k=1}^K \left\| \lambda_e^k + \sum_{(ij) \in E} \mathbf{x}_{ij} (C\hat{y}_{ij}^k - \mu_{ij}^k) \right\|^2 \\ \text{s.t.} \quad & \mu_i^k \geq 0, \quad \forall i, k; \quad \sum_k \mu_i^k = C, \quad \forall i; \\ & \mu_{ij}^k \geq 0, \quad \mu_{ij}^k \leq \mu_i^k, \quad \mu_{ij}^k \leq \mu_j^k, \quad \forall (ij) \in E, k; \\ & \lambda_e \geq 0. \end{aligned} \quad (11)$$

As in the original M³N optimization, the dual variables have an intuitive probabilistic interpretation. In the binary case, the set of the variables μ_i^k, μ_{ij}^k corresponds to marginals of a distribution (normalized to C) over the possible assignments $\mathbf{y}$. (This assertion follows from taking the dual of the original exponential size QP in Eq. (3).) Then the constraints (9) that $\mu_{ij}^k \leq \mu_i^k$ and $\mu_{ij}^k \leq \mu_j^k$ can be explained by the fact that $P(y_i = y_j = k) \leq P(y_i = k)$ and $P(y_i = y_j = k) \leq P(y_j = k)$ for any distribution $P(\mathbf{y})$. For $K > 2$, the set of the variables μ_i^k, μ_{ij}^k may not correspond to a valid distribution.

The primal and dual solution are related by:

$$\mathbf{w}_n^k = \sum_{i=1}^N \mathbf{x}_i (C\hat{y}_i^k - \mu_i^k), \quad (12)$$

$$\mathbf{w}_e^{k,k} = \lambda_e^k + \sum_{(ij) \in E} \mathbf{x}_{ij} (C\hat{y}_{ij}^k - \mu_{ij}^k). \quad (13)$$

One important consequence of these relationships is that the node parameters are all support vector expansions. Thus, the terms in the constraints of the form $\mathbf{w}_n \cdot \mathbf{x}$ can all be expanded in terms of dot products $\mathbf{x}_i^\top \mathbf{x}_j$; the objective ($\|\mathbf{w}\|^2$) can be expanded similarly. Therefore, we can use kernels $K(\mathbf{x}_i, \mathbf{x}_j)$ to define node parameters. Unfortunately, the positivity constraint on the edge potentials, and the resulting λ_e^k dual variable in the expansion of the edge weight, prevent the edge parameters from being kernelized in a similar way.

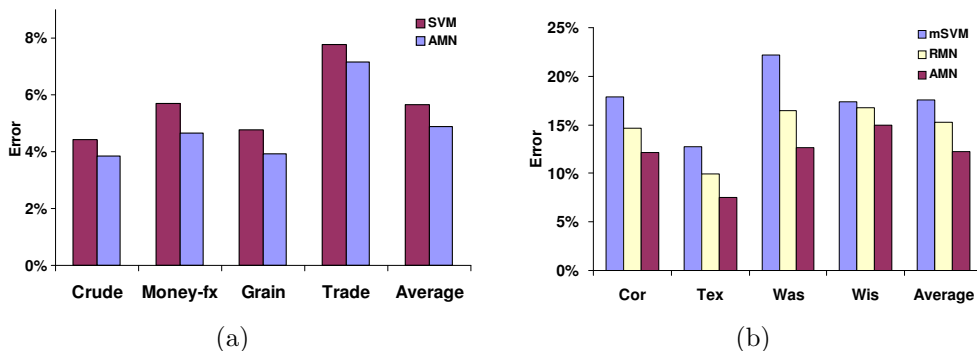


Figure 2. (a) Comparison of test error of SVMs and AMNs on four categories of Reuters articles, averaged over 7-folds; (b) Comparison of test error of SVMs, RMNs and AMNs on four WebKB sites.

6. Experimental Results

We evaluated our approach on two text classification domains, of very different structure.

Reuters. We ran our method on the ModApte set of the Reuters-21578 corpus. We selected four categories containing a substantial number of documents: *crude*, *grain*, *trade*, and *money-fx*. We eliminated documents labeled with more than one category, and represented each document as a bag of words. The resulting dataset contained around 2200 news articles, which were split into seven folds where the articles in each fold occur in the same time period. The reported results were obtained using seven-fold cross-validation with a training set size of ~ 200 documents and a test set size of ~ 2000 documents.

The baseline model is a linear kernel SVM using a bag of words as features. Since we train and test on articles in different time periods, there is an inherent distribution drift between our training and test sets, which hurts the SVM’s performance. For example, there may be words which, in the test set, are highly indicative of a certain label, but are not present in the training set at all since they were very specific to a particular time period (see (Taskar et al., 2003b)).

Our AMN model uses the text similarity of two articles as an indicator of how likely they are to have the same label. The intuition is that two documents that have similar text are likely to share the same label in any time period, so that adding associative edges between them would result in better classification. Such positive correlations are exactly what AMNs represent. In our model, we linked each document to its two closest documents as measured by TF-IDF weighted cosine distance. The TF-IDF score of a term was computed as: $(1 + \log tf) \log \frac{N}{df}$ where tf is the term frequency, N is the number of total documents, and df is the document frequency. The node features were simply the words in the article corresponding to the node. Edge features included the actual TF-IDF weighted cosine distance, as well as the bag of words consisting of union of the words in the linked documents.

We trained both models (SVM and AMN) to predict one category vs. all remaining categories. Fig. 2(a) shows that the AMN model achieves a 13.5% average error reduction over the baseline SVM, with improvement in every category. Applying a paired t-test comparing the AMN and SVM over the 7 folds in each category, *crude*, *trade*, *grain*, *money-fx*, we obtained p-values of 0.004897, 0.017026, 0.012836, 0.000291 respectively. These results indicate that the positive interactions learned by the AMN allow us to correct for some of the distribution drift between the training and test sets.

Hypertext. We tested AMNs on collective hypertext classification, using the variant of the WebKB dataset (Craven et al., 1998) used by Taskar *et al.* (2002). This data set contains web pages from four different Computer Science departments: Cornell, Texas, Washington, and Wisconsin. Each page is labeled as one of *course*, *faculty*, *student*, *project*, *other*. Our goal in this task is to exploit the additional structured information in hypertext using AMNs.

Our flat model is a multiclass linear-kernel SVM predicting categories based on the text content of the webpage. The words are represented as a bag of words. For the AMN model, we used the fact that a webpage’s internal structure can be broken up into disjoint *sections*. For example, a faculty webpage might have one section that discusses research, with a list of links to relevant research projects, another section with links to student webpages, etc. Intuitively, if we have links to two pages in the same section, they are likely have the same topic. As AMNs capture precisely this type of positive correlation, we added edges between pages that appear as hyperlinks in the same section of another page. The node features for the AMN model are the same as for the multiclass SVM.

In performing the experiments we train on the pages from three of the schools in the dataset and test on the remaining one. The results, shown in Fig. 2(b), demonstrate a 30% relative reduction in test error as a result of modeling the positive correlation be-

tween pages in the AMN model. The improvement is present when testing on each of the schools. We also trained the same AMN model using the RMN approach of Taskar *et al.* (2002). In this approach, the Markov network is trained to maximize the conditional log-likelihood, using loopy belief propagation (Yedidia *et al.*, 2000) for computing the posterior probabilities needed for optimization. Due to the high connectivity in the network, the algorithm is not exact, and not guaranteed to converge to the true values for the posterior distribution. In our results, RMNs achieve a worse test error than AMNs. We note that the learned AMN weights never produced fractional solutions when used for inference, which suggests that the optimization successfully avoided problematic parameterizations of the network, even in the case of the non-optimal multi-class relaxation.

7. Conclusion

In this paper, we provide an algorithm for max-margin training of associative Markov networks, a subclass of Markov networks that allows only positive interactions between related variables. Our approach relies on a linear programming relaxation of the MAP problem, which is the key component in the quadratic program associated with the max-margin formulation. We thus provide a polynomial time algorithm which approximately solves the maximum margin estimation problem for any associative Markov network. Importantly, our method is guaranteed to find the optimal (margin-maximizing) solution for all binary-valued AMNs, regardless of the clique size or the connectivity. To our knowledge, this algorithm is the first to provide an effective learning procedure for Markov networks of such general structure.

Our results in the binary case rely on the fact that the LP relaxation of the MAP problem provides exact solutions. In the non-binary case, we are not guaranteed exact solutions, but we can prove constant-factor approximation bounds on the MAP solution returned by the relaxed LP. It would be interesting to see whether these bounds provide us with guarantees on the quality (e.g., the margin) of our learned model.

The class of associative Markov networks appears to cover a large number of interesting applications. We have explored only two such applications in our experimental results, both in the text domain. It would be very interesting to consider other applications, such as image segmentation, extracting protein complexes from protein-protein interaction data, or predicting links in relational data.

However, despite the prevalence of fully associative Markov networks, it is clear that many applications call for repulsive potentials. For example, the

best classification accuracy on the WebKB hypertext data set is obtained in a maximum margin framework (Taskar *et al.*, 2003a), when we allow repulsive potentials on linked webpages (representing, for example, that students tend not to link to pages of students). While clearly we cannot introduce fully general potentials into AMNs without running against the NP-hardness of the general problem, it would be interesting to see whether we can extend the class of networks we can learn effectively.

References

- Bach, F., & Jordan, M. (2001). Thin junction trees. *NIPS*.
- Bertsimas, D., & Tsitsiklis, J. (1997). *Introduction to linear programming*. Athena Scientific.
- Besag, J. E. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society B*, 48.
- Bouman, C., & Shapiro, M. (1994). A multiscale random field model for bayesian image segmentation. *IP*, 3.
- Boykov, Y., Veksler, O., & Zabih, R. (1999a). Fast approximate energy minimization via graph cuts. *ICCV*.
- Boykov, Y., Veksler, O., & Zabih, R. (1999b). Markov random fields with efficient approximations. *CVPR*.
- Chakrabarti, S., Dom, B., & Indyk, P. (1998). Enhanced hypertext categorization using hyperlinks. *SIGMOD*.
- Craven, M., DiPasquo, D., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., & Slattery, S. (1998). Learning to extract symbolic knowledge from the world wide web. *Proc AAAI98* (pp. 509–516).
- Greig, D. M., Porteous, B. T., & Seheult, A. H. (1989). Exact maximum a posteriori estimation for binary images. *J. R. Statist. Soc. B*, 51.
- Kleinberg, J., & Tardos, E. (1999). Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. *FOCS*.
- Kolmogorov, V., & Zabih, R. (2002). What energy functions can be minimized using graph cuts? *PAMI*.
- Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *ICML*.
- Potts, R. B. (1952). Some generalized order-disorder transformations. *Proc. Cambridge Phil. Soc.*, 48.
- Taskar, B., Abbeel, P., & Koller, D. (2002). Discriminative probabilistic models for relational data. *UAI*.
- Taskar, B., Guestrin, C., & Koller, D. (2003a). Max margin markov networks. *Proc. NIPS*.
- Taskar, B., Wong, M., & Koller, D. (2003b). Learning on the test data: Leveraging unseen features. *Proc. ICML*.
- Vazquez, A., Flammini, A., Maritan, A., & Vespignani, A. (2003). Global protein function prediction from protein-protein interaction networks. *Nature Biotechnology*, 6.
- Wainwright, M., Jaakkola, T., & Willsky, A. (2002). Map estimation via agreement on (hyper)trees: Message-passing and linear programming approaches. *Allerton Conference on Communication, Control and Computing*.
- Yedidia, J., Freeman, W., & Weiss, Y. (2000). Generalized belief propagation. *NIPS*.

A. Binary AMNs

Proof (For Theorem 3.1) Consider any fractional, feasible $\mathbf{y}$. We show that we can construct a new feasible assignment $\mathbf{z}$ which increases the objective (or leaves it unchanged) and furthermore has fewer fractional entries.

Since $\theta_c^k \geq 0$, we can assume that $y_c^k = \min_{i \in c} y_i^k$; otherwise we could increase the objective by increasing y_c^k . We construct an assignment $\mathbf{z}$ from $\mathbf{y}$ by leaving integral values unchanged and uniformly shifting fractional values by λ :

$$\begin{aligned} z_i^1 &= y_i^1 - \lambda I(0 < y_i^1 < 1), & z_i^2 &= y_i^2 + \lambda I(0 < y_i^2 < 1), \\ z_c^1 &= y_c^1 - \lambda I(0 < y_c^1 < 1), & z_c^2 &= y_c^2 + \lambda I(0 < y_c^2 < 1), \end{aligned}$$

where $I(\cdot)$ is an indicator function.

Now consider $\lambda^k = \min_{i: y_i^k > 0} y_i^k$. Note that if $\lambda = \lambda^1$ or $\lambda = -\lambda^2$, $\mathbf{z}$ will have at least one more integral z_i^k than $\mathbf{y}$. Thus if we can show that the update results in a feasible and better scoring assignment, we can apply it repeatedly to get an optimal integer solution. To show that $\mathbf{z}$ is feasible, we need $z_i^1 + z_i^2 = 1$, $z_i^k \geq 0$ and $z_c^k = \min_{i \in c} z_i^k$.

First, we show that $z_i^1 + z_i^2 = 1$.

$$\begin{aligned} z_i^1 + z_i^2 &= y_i^1 - \lambda I(0 < y_i^1 < 1) + y_i^2 + \lambda I(0 < y_i^2 < 1) \\ &= y_i^1 + y_i^2 = 1. \end{aligned}$$

Above we used the fact that if y_i^1 is fractional, so is y_i^2 , since $y_i^1 + y_i^2 = 1$.

To show that $z_i^k \geq 0$, we prove $\min_i z_i^k = 0$.

$$\begin{aligned} \min_i z_i^k &= \min_i \left[y_i^k - \left(\min_{i: y_i^k > 0} y_i^k \right) I(0 < y_i^k < 1) \right] \\ &= \min \left(\min_i y_i^k, \min_{i: y_i^k > 0} \left[y_i^k - \min_{i: y_i^k > 0} y_i^k \right] \right) = 0. \end{aligned}$$

Lastly, we show $z_c^k = \min_{i \in c} z_i^k$.

$$\begin{aligned} z_c^1 &= y_c^1 - \lambda I(0 < y_c^1 < 1) \\ &= \left(\min_{i \in c} y_i^1 \right) - \lambda I(0 < \min_{i \in c} y_i^1 < 1) = \min_{i \in c} z_i^1; \\ z_c^2 &= y_c^2 + \lambda I(0 < y_c^2 < 1) \\ &= \left(\min_{i \in c} y_i^2 \right) + \lambda I(0 < \min_{i \in c} y_i^2 < 1) = \min_{i \in c} z_i^2. \end{aligned}$$

We have established that the new $\mathbf{z}$ are feasible, and it remains to show that we can improve the objective. We can show that the change in the objective is always λD for some constant D that depends only on $\mathbf{y}$ and θ . This implies that one of the two cases, $\lambda = \lambda^1$ or $\lambda = -\lambda^2$, will necessarily increase the objective (or

leave it unchanged). The change in the objective is:

$$\begin{aligned} &\sum_{i=1}^N \sum_{k=1,2} \theta_i^k (z_i^k - y_i^k) + \sum_{c \in \mathcal{C}} \sum_{k=1,2} \theta_c^k (z_c^k - y_c^k) \\ &= \lambda \left[\sum_{i=1}^N (D_i^1 - D_i^2) + \sum_{c \in \mathcal{C}} (D_c^1 - D_c^2) \right] = \lambda D \\ D_i^k &= \theta_i^k I(0 < y_i^k < 1), \quad D_c^k = \theta_c^k I(0 < y_c^k < 1). \end{aligned}$$

Hence the new assignment $\mathbf{z}$ is feasible, does not decrease the objective function, and has strictly fewer fractional entries. ■

B. Multi-class AMNs

For $K > 2$, we use the randomized rounding procedure of Kleinberg and Tardos (1999) to produce an integer solution for the linear relaxation, losing at most a factor of $m = \max_{c \in \mathcal{C}} |c|$ in the objective function. The basic idea of the rounding procedure is to treat y_i^k as probabilities and assign labels according to these probabilities in phases. In each phase, we pick a label k , uniformly at random, and a threshold $\alpha \in [0, 1]$ uniformly at random. For each node i which has not yet been assigned a label, we assign the label k if $y_i^k \geq \alpha$. The procedure terminates when all nodes have been assigned a label. Our analysis closely follows that of Tardos (1999).

Lemma B.1 *The probability that a node i is assigned label k by the randomized procedure is y_i^k .*

Proof The probability that an unassigned node is assigned label k during one phase is $\frac{1}{K} y_i^k$, which is proportional to y_i^k . By symmetry, the probability that a node is assigned label k over all phases is exactly y_i^k . ■

Lemma B.2 *The probability that all nodes in a clique c are assigned label k by the procedure is at least $\frac{1}{|c|} y_c^k$.*

Proof For a single phase, the probability that all nodes in a clique c are assigned label k if none of the nodes were previously assigned is $\frac{1}{K} \min_{i \in c} y_i^k = \frac{1}{K} y_c^k$. The probability that *at least one* of the nodes will be assigned label k in a phase is $\frac{1}{K} (\max_{i \in c} y_i^k)$. The probability that *none* of the nodes in the clique will be assigned *any* label in one phase is $1 - \frac{1}{K} \sum_{k=1}^K \max_{i \in c} y_i^k$.

Nodes in the clique c will be assigned label k by the procedure if they are assigned label k in one phase. (They can also be assigned label k as a result of several phases, but we can ignore this possibility for the purposes of the lower bound.) The probability that all the nodes in c will be assigned label k by the procedure

in a single phase is:

$$\begin{aligned} \sum_{j=1}^{\infty} \frac{1}{K} y_c^k \left(1 - \frac{1}{K} \sum_{k=1}^K \max_{i \in c} y_i^k \right)^{j-1} &= \frac{y_c^k}{\sum_{k=1}^K \max_{i \in c} y_i^k} \\ &\geq \frac{y_c^k}{\sum_{k=1}^K \sum_{i \in c} y_i^k} = \frac{y_c^k}{\sum_{i \in c} \sum_{k=1}^K y_i^k} = \frac{y_c^k}{|c|}. \end{aligned}$$

Above, we first used the fact that for $d < 1$, $\sum_{i=0}^{\infty} d^i = \frac{1}{1-d}$, and then upper-bounded the max of the set of positive y_i^k 's by their sum. ■

Theorem B.3 *The expected cost of the assignment found by the randomized procedure given a solution $\mathbf{y}$ to the linear program in Eq. (2) is at least $\sum_{i=1}^N \sum_{k=1}^K \theta_i^k y_i^k + \sum_{c \in \mathcal{C}} \frac{1}{|c|} \sum_{k=1}^K \theta_c^k y_c^k$.*

Proof This is immediate from the previous two lemmas.

The only difference between the expected cost of the rounded solution and the (non-integer) optimal solution is the $\frac{1}{|c|}$ factor in the second term. By picking $m = \max_{c \in \mathcal{C}} |c|$, we have that the rounded solution is at most m times worse than the optimal solution produced by the LP of Eq. (2). ■

We can also derandomize this procedure to get a deterministic algorithm with the same guarantees, using the method of conditional probabilities, similar in spirit to the approach of Kleinberg and Tardos (1999).

Note that the approximation factor of m applies, in fact, only to the clique potentials. Thus, if we compare the log-probability of the optimal MAP solution and the log-probability of the assignment produced by this randomized rounding procedure, the terms corresponding to the log-partition-function and the node potentials are identical. We obtain an additive error (in log-probability space) only for the clique potentials. As node potentials are often larger in magnitude than clique potentials, the fact that we incur no loss proportional to node potentials is likely to lead to smaller errors in practice. Along similar lines, we note that the constant factor approximation is smaller for smaller cliques; again, we observe, the potentials associated with large cliques are typically smaller in magnitude, reducing further the actual error in practice.

Discriminative Probabilistic Models for Relational Data

Ben Taskar

Computer Science Dept.
Stanford University
Stanford, CA 94305
btaskar@cs.stanford.edu

Pieter Abbeel

Computer Science Dept.
Stanford University
Stanford, CA 94305
abbeel@cs.stanford.edu

Daphne Koller

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

Abstract

In many supervised learning tasks, the entities to be labeled are related to each other in complex ways and their labels are not independent. For example, in hypertext classification, the labels of linked pages are highly correlated. A standard approach is to classify each entity independently, ignoring the correlations between them. Recently, Probabilistic Relational Models, a relational version of Bayesian networks, were used to define a joint probabilistic model for a collection of related entities. In this paper, we present an alternative framework that builds on (conditional) Markov networks and addresses two limitations of the previous approach. First, undirected models do not impose the acyclicity constraint that hinders representation of many important relational dependencies in directed models. Second, undirected models are well suited for discriminative training, where we optimize the conditional likelihood of the labels given the features, which generally improves classification accuracy. We show how to train these models effectively, and how to use approximate probabilistic inference over the learned model for collective classification of multiple related entities. We provide experimental results on a webpage classification task, showing that accuracy can be significantly improved by modeling relational dependencies.

1 Introduction

The vast majority of work in statistical classification methods has focused on “flat” data – data consisting of identically-structured entities, typically assumed to be independent and identically distributed (IID). However, many real-world data sets are innately relational: hyperlinked webpages, cross-citations in patents and scientific papers, social networks, medical records, and more. Such data consist of entities of different types, where each entity type is characterized by a different set of attributes. Entities are related to each other via different types of links, and the link structure is an important source of information.

Consider a collection of hypertext documents that we want to classify using some set of labels. Most naively, we can use a bag of words model, classifying each webpage solely using the words that appear on the page. However, hypertext has a very rich structure that this approach loses entirely. One document has hyperlinks to others, typically indicating that their topics are related. Each document also has internal structure, such as a partition into sections; hyperlinks that emanate from the same section of the document are even more likely to point to similar documents. When classifying a collection of documents, these are important cues, that can potentially help us achieve better classification accuracy. Therefore, rather than classifying each document separately, we want to provide a form of *collective classification*, where we simultaneously decide on the class labels of all of the entities together, and thereby can explicitly take advantage of the correlations between the labels of related entities.

We propose the use of a joint probabilistic model for an entire collection of related entities. Following the approach of Lafferty (2001), we base our approach on discriminatively trained undirected graphical models, or *Markov networks* (Pearl 1988). We introduce the framework of *relational Markov network (RMNs)*, which compactly defines a Markov network over a relational data set. The graphical structure of an RMN is based on the relational structure of the domain, and can easily model complex patterns over related entities. For example, we can represent a pattern where two linked documents are likely to have the same topic. We can also capture patterns that involve groups of links: for example, consecutive links in a document tend to refer to documents with the same label. As we show, the use of an undirected graphical model avoids the difficulties of defining a coherent generative model for graph structures in directed models. It thereby allows us tremendous flexibility in representing complex patterns.

Undirected models lend themselves well to discriminative training, where we optimize the conditional likelihood of the labels given the features. Discriminative training, given sufficient data, generally provides significant improvements in classification accuracy over generative training (Vapnik 1995). We provide an effective parameter esti-

mation algorithm for RMNs which uses conjugate gradient combined with approximate probabilistic inference (belief propagation (Pearl 1988)) for estimating the gradient. We also show how to use approximate probabilistic inference over the learned model for collective classification of multiple related entities. We provide experimental results on a webpage classification task, showing significant gains in accuracy arising both from the modeling of relational dependencies and the use of discriminative training.

2 Relational Classification

Consider hypertext as a simple example of a relational domain. A relational domain is defined by a schema, which describes entities, their attributes and relations between them. In our domain, there are two entity types: Doc and Link. If a webpage is represented as a bag of words, Doc would have a set of boolean attributes Doc.HasWord_k indicating whether the word k occurs on the page. It would also have the label attribute Doc.Label , indicating the topic of the page, which takes on a set of categorical values. The Link entity type has two attributes: Link.From and Link.To , both of which refer to Doc entities.

In general, a *schema* specifies of a set of entity types $\mathcal{E} = \{E_1, \dots, E_n\}$. Each type E is associated with three sets of attributes: content attributes $E.X$ (e.g. Doc.HasWord_k), label attributes $E.Y$ (e.g. Doc.Label), and reference attributes $E.R$ (e.g. Link.To). For simplicity, we restrict label and content attributes to take on categorical values. Reference attributes include a special unique key attribute $E.K$ that identifies each entity. Other reference attributes $E.R$ refer to entities of a single type $E' = \text{Range}(E.R)$ and take values in $\text{Domain}(E'.K)$.

An *instantiation* $\mathcal{I}$ of a schema $\mathcal{E}$ specifies the set of entities $\mathcal{I}(E)$ of each entity type $E \in \mathcal{E}$ and the values of all attributes for all of the entities. For example, an instantiation of the hypertext schema is a collection of webpages, specifying their labels, words they contain and links between them. We will use $\mathcal{I}.X$, $\mathcal{I}.Y$ and $\mathcal{I}.R$ to denote the content, label and reference attributes in the instantiation $\mathcal{I}$; $\mathcal{I}.x$, $\mathcal{I}.y$ and $\mathcal{I}.r$ to denote the values of those attributes. The component $\mathcal{I}.r$, which we call an *instantiation skeleton* or *instantiation graph*, specifies the set of entities (nodes) and their reference attributes (edges). A hypertext instantiation graph specifies a set of webpages and links between them, but not their words or labels.

The structure of the instantiation graph has been used extensively to infer their importance in scientific publications (Egghe and Rousseau 1990) and hypertext (Kleinberg 1999). Several recent papers have proposed algorithms that use the link graph to aid classification. Chakrabarti *et al.* (1998) use system-predicted labels of linked documents to iteratively re-label each document in the test set, achieving a significant improvement compared to a baseline of using the text in each document alone. A similar approach was used by Neville and Jensen (2000) in a different domain. Slattery and Mitchell (2000) tried to identify direc-

tory (or hub) pages that commonly list pages of the same topic, and used these pages to improve classification of university webpages. However, none of these approaches provide a coherent model for the correlations between linked webpages. Thus, they apply combinations of classifiers in a procedural way, with no formal justification.

Taskar *et al.* (2001) suggest the use of *probabilistic relational models (PRMs)* for the collective classification task. PRMs (Koller and Pfeffer 1998; Friedman *et al.* 1999) are a relational extension to Bayesian networks (Pearl 1988). A PRM specifies a probability distribution over instantiations consistent with a given instantiation graph by specifying a Bayesian-network-like template-level probabilistic model for each entity type. Given a particular instantiation graph, the PRM induces a large Bayesian network over that instantiation that specifies a joint probability distribution over all attributes of all of the entities. This network reflects the interactions between related instances by allowing us to represent correlations between their attributes.

In our hypertext example, a PRM might use a naive Bayes model for words, with a directed edge between Doc.Label and each attribute Doc.HasWord_k ; each of these attributes would have a *conditional probability distribution* $P(\text{Doc.HasWord}_k \mid \text{Doc.Label})$ associated with it, indicating the probability that word k appears in the document given each of the possible topic labels. More importantly, a PRM can represent the inter-dependencies between topics of linked documents by introducing an edge from Doc.Label to Doc.Label of two documents if there is a link between them. Given a particular instantiation graph containing some set of documents and links, the PRM specifies a Bayesian network over all of the documents in the collection. We would have a probabilistic dependency from each document's label to the words on the document, and a dependency from each document's label to the labels of all of the documents to which it points. Taskar *et al.* show that this approach works well for classifying scientific documents, using both the words in the title and abstract and the citation-link structure.

However the application of this idea to other domains, such as webpages, is problematic since there are many cycles in the link graph, leading to cycles in the induced "Bayesian network", which is therefore not a coherent probabilistic model. Getoor *et al.* (2001) suggest an approach where we do not include direct dependencies between the labels of linked webpages, but rather treat links themselves as random variables. Each two pages have a "potential link", which may or may not exist in the data. The model defines the probability of the link existence as a function of the labels of the two endpoints. In this link existence model, labels have no incoming edges from other labels, and the cyclicity problem disappears. This model, however, has other fundamental limitations. In particular, the resulting Bayesian network has a random variable for each potential link — N^2 variables for collections containing N pages. This quadratic blowup occurs even when the

actual link graph is very sparse. When N is large (e.g., the set of all webpages), a quadratic growth is intractable. Even more problematic are the inherent limitations on the expressive power imposed by the constraint that the directed graph must represent a coherent generative model over graph structures. The link existence model assumes that the presence of different edges is a conditionally independent event. Representing more complex patterns involving correlations between multiple edges is very difficult. For example, if two pages point to the same page, it is more likely that they point to each other as well. Such interactions between many overlapping triples of links do not fit well into the generative framework.

Furthermore, directed models such as Bayesian networks and PRMs are usually trained to optimize the joint probability of the labels and other attributes, while the goal of classification is a discriminative model of labels given the other attributes. The advantage of training a model only to discriminate between labels is that it does not have to trade off between classification accuracy and modeling the joint distribution over non-label attributes. In many cases, discriminatively trained models are more robust to violations of independence assumptions and achieve higher classification accuracy than their generative counterparts.

3 Undirected Models for Classification

As discussed, our approach to the collective classification task is based on the use of undirected graphical models. We begin by reviewing *Markov networks*, a “flat” undirected model. We then discuss how Markov networks can be extended to the relational setting.

Markov networks. We use $\mathbf{V}$ to denote a set of discrete random variables and $\mathbf{v}$ an assignment of values to $\mathbf{V}$. A Markov network for $\mathbf{V}$ defines a joint distribution over $\mathbf{V}$. It consists of a qualitative component, an undirected dependency graph, and a quantitative component, a set of parameters associated with the graph. For a graph G , a *clique* is a set of nodes $\mathbf{V}_c$ in G , not necessarily maximal, such that each $V_i, V_j \in \mathbf{V}_c$ are connected by an edge in G . Note that a single node is also considered a clique.

Definition 1: Let $G = (\mathbf{V}, E)$ be an undirected graph with a set of cliques $C(G)$. Each $c \in C(G)$ is associated with a set of nodes $\mathbf{V}_c$ and a *clique potential* $\phi_c(\mathbf{V}_c)$, which is a non-negative function defined on the joint domain of $\mathbf{V}_c$. Let $\Phi = \{\phi_c(\mathbf{V}_c)\}_{c \in C(G)}$. The Markov net (G, Φ) defines the distribution $P(\mathbf{v}) = \frac{1}{Z} \prod_{c \in C(G)} \phi_c(\mathbf{v}_c)$, where Z is the *partition function* — a normalization constant given by $Z = \sum_{\mathbf{v}'} \prod \phi_c(\mathbf{v}_c')$. ■

Each potential ϕ_c is simply a table of values for each assignment $\mathbf{v}_c$ that defines a “compatibility” between values of variables in the clique. The potential is often represented by a log-linear combination of a small set of indicator functions, or *features*, of the form $f(\mathbf{V}_c) \equiv \delta(\mathbf{V}_c = \mathbf{v}_c)$. In this case, the potential can be more conveniently rep-

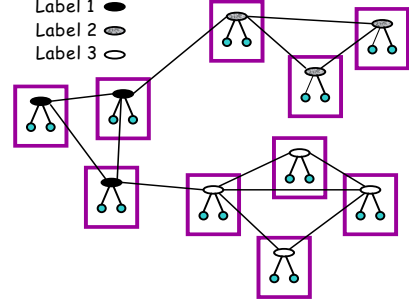


Figure 1: An unrolled Markov net over linked documents. The links follow a common pattern: documents with the same label tend to link to each other more often.

resented in log-linear form:

$$\phi_c(\mathbf{v}_c) = \exp\left\{\sum_i w_i f_i(\mathbf{v}_c)\right\} = \exp\{\mathbf{w}_c \cdot \mathbf{f}_c(\mathbf{v}_c)\}.$$

Hence we can write:

$$\log P(\mathbf{v}) = \sum_c \mathbf{w}_c \cdot \mathbf{f}_c(\mathbf{v}_c) - \log Z = \mathbf{w} \cdot \mathbf{f}(\mathbf{v}) - \log Z$$

where $\mathbf{w}$ and $\mathbf{f}$ are the vectors of all weights and features.

For classification, we are interested in constructing discriminative models using *conditional Markov nets* which are simply Markov networks renormalized to model a conditional distribution.

Definition 2: Let $\mathbf{X}$ be a set of random variables on which we condition and $\mathbf{Y}$ be a set of target (or label) random variables. A *conditional Markov network* is a Markov network (G, Φ) which defines the distribution $P(\mathbf{y} \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \prod_{c \in C(G)} \phi_c(\mathbf{x}_c, \mathbf{y}_c)$, where $Z(\mathbf{x})$ is the partition function, now dependent on $\mathbf{x}$: $Z(\mathbf{x}) = \sum_{\mathbf{y}'} \prod \phi_c(\mathbf{x}_c, \mathbf{y}_c')$. ■

Logistic regression, a well-studied statistical model for classification, can be viewed as the simplest example of a conditional Markov network. In standard form, for $Y = \pm 1$ and $\mathbf{X} \in \{0, 1\}^n$ (or $\mathbf{X} \in \mathbb{R}^n$), $P(y \mid \mathbf{x}) = \frac{1}{Z(\mathbf{x})} \exp\{y \mathbf{w} \cdot \mathbf{x}\}$. Viewing the model as a Markov network, the cliques are simply the edges $c_k = \{X_k, Y\}$ with potentials $\phi_k(x_k, y) = \exp\{y w_k x_k\}$.

Relational Markov Networks. We now extend the framework of Markov networks to the relational setting. A *relational Markov network (RMN)* specifies a conditional distribution over all of the labels of all of the entities in an instantiation given the relational structure and the content attributes. (We provide the definitions directly for the conditional case, as the unconditional case is a special case where the set of content attributes is empty.) Roughly speaking, it specifies the cliques and potentials between attributes of related entities at a template level, so a single model provides a coherent distribution for any collection of instances from the schema.

For example, suppose that pages with the same label tend to link to each other, as in Fig. 1. We can capture this

correlation between labels by introducing, for each link, a clique between the labels of the source and the target page. The potential on the clique will have higher values for assignments that give a common label to the linked pages.

To specify what cliques should be constructed in an instantiation, we will define a notion of a *relational clique template*. A relational clique template specifies tuples of variables in the instantiation by using a relational query language. For our link example, we can write the template as a kind of SQL query:

```
SELECT doc1.Category, doc2.Category
FROM Doc doc1, Doc doc2, Link link
WHERE link.From = doc1.Key and link.To = doc2.Key
```

Note the three clauses that define a query: the FROM clause specifies the cross product of entities to be filtered by the WHERE clause and the SELECT clause picks out the attributes of interest. Our definition of clique templates contains the corresponding three parts.

Definition 3: A *relational clique template* $C = (\mathbf{F}, \mathbf{W}, \mathbf{S})$ consists of three components:

- $\mathbf{F} = \{F_i\}$ — a set of entity variables, where an entity variable F_i is of type $E(F_i)$.
- $\mathbf{W}(\mathbf{F}, \mathbf{R})$ — a boolean formula using conditions of the form $F_i.R_j = F_k.R_l$.
- $\mathbf{F}, \mathbf{S} \subseteq \mathbf{F}, \mathbf{X} \cup \mathbf{F}, \mathbf{Y}$ — a selected subset of content and label attributes in $\mathbf{F}$. ■

For the clique template corresponding to the SQL query above, $\mathbf{F}$ consists of *doc1*, *doc2* and *link* of types Doc, Doc and Link, respectively. $\mathbf{W}(\mathbf{F}, \mathbf{R})$ is *link.From = doc1.Key* $\wedge$ *link.To = doc2.Key* and $\mathbf{F}, \mathbf{S}$ is *doc1.Category* and *doc2.Category*.

A clique template specifies a set of cliques in an instantiation $\mathcal{I}$:

$$C(\mathcal{I}) \equiv \{c = \mathbf{f}, \mathbf{S} : \mathbf{f} \in \mathcal{I}(\mathbf{F}) \wedge \mathbf{W}(\mathbf{f}, \mathbf{r})\},$$

where $\mathbf{f}$ is a tuple of entities $\{f_i\}$ in which each f_i is of type $E(F_i)$; $\mathcal{I}(\mathbf{F}) = \mathcal{I}(E(F_1)) \times \dots \times \mathcal{I}(E(F_n))$ denotes the cross-product of entities in the instantiation; the clause $\mathbf{W}(\mathbf{f}, \mathbf{r})$ ensures that the entities are related to each other in specified ways; and finally, $\mathbf{f}, \mathbf{S}$ selects the appropriate attributes of the entities. Note that the clique template does not specify the nature of the interaction between the attributes; that is determined by the clique potentials, which will be associated with the template.

This definition of a clique template is very flexible, as the WHERE clause of a template can be an arbitrary predicate. It allows modeling complex relational patterns on the instantiation graphs. To continue our webpage example, consider another common pattern in hypertext: links in a webpage tend to point to pages of the same category. This pattern can be expressed by the following template:

```
SELECT doc1.Category, doc2.Category
FROM Doc doc1, Doc doc2, Link link1, Link link2
WHERE link1.From = link2.From and link1.To = doc1.Key
and link2.To = doc2.Key and not doc1.Key = doc2.Key
```

Depending on the expressive power of our template definition language, we may be able to construct very complex templates that select entire subgraph structures of an instantiation. We can easily represent patterns involving three (or more) interconnected documents without worrying about the acyclicity constraint imposed by directed models. Since the clique templates do not explicitly depend on the identities of entities, the same template can select subgraphs whose structure is fairly different. The RMN allows us to associate the same clique potential parameters with all of the subgraphs satisfying the template, thereby allowing generalization over a wide range of different structures.

Definition 4: A *Relational Markov network (RMN)* $\mathcal{M} = (\mathbf{C}, \Phi)$ specifies a set of clique templates $\mathbf{C}$ and corresponding potentials $\Phi = \{\phi_C\}_{C \in \mathbf{C}}$ to define a conditional distribution:

$$P(\mathcal{I}, \mathbf{y} \mid \mathcal{I}, \mathbf{x}, \mathcal{I}, \mathbf{r}) = \frac{1}{Z(\mathcal{I}, \mathbf{x}, \mathcal{I}, \mathbf{r})} \prod_{C \in \mathbf{C}} \prod_{c \in C(\mathcal{I})} \phi_C(\mathcal{I}, \mathbf{x}_c, \mathcal{I}, \mathbf{y}_c)$$

where $Z(\mathcal{I}, \mathbf{x}, \mathcal{I}, \mathbf{r})$ is the normalizing partition function: $Z(\mathcal{I}, \mathbf{x}, \mathcal{I}, \mathbf{r}) = \sum_{\mathcal{I}, \mathbf{y}'} \prod_{C \in \mathbf{C}} \prod_{c \in C(\mathcal{I})} \phi_C(\mathcal{I}, \mathbf{x}_c, \mathcal{I}, \mathbf{y}'_c)$ ■

Using the log-linear representation of potentials, $\phi_C(\mathbf{V}_C) = \exp\{\mathbf{w}_C \cdot \mathbf{f}_C(\mathbf{V}_C)\}$, we can write

$$\begin{aligned} \log P(\mathcal{I}, \mathbf{y} \mid \mathcal{I}, \mathbf{x}, \mathcal{I}, \mathbf{r}) &= \sum_{C \in \mathbf{C}} \sum_{c \in C(\mathcal{I})} \mathbf{w}_C \cdot \mathbf{f}_C(\mathcal{I}, \mathbf{x}_c, \mathcal{I}, \mathbf{y}_c) - \log Z(\mathcal{I}, \mathbf{x}, \mathcal{I}, \mathbf{r}) \\ &= \sum_{C \in \mathbf{C}} \mathbf{w}_C \cdot \mathbf{f}_C(\mathcal{I}, \mathbf{x}, \mathcal{I}, \mathbf{y}, \mathcal{I}, \mathbf{r}) - \log Z(\mathcal{I}, \mathbf{x}, \mathcal{I}, \mathbf{r}) \\ &= \mathbf{w} \cdot \mathbf{f}(\mathcal{I}, \mathbf{x}, \mathcal{I}, \mathbf{y}, \mathcal{I}, \mathbf{r}) - \log Z(\mathcal{I}, \mathbf{x}, \mathcal{I}, \mathbf{r}) \end{aligned}$$

where

$$\mathbf{f}_C(\mathcal{I}, \mathbf{x}, \mathcal{I}, \mathbf{y}, \mathcal{I}, \mathbf{r}) = \sum_{c \in C(\mathcal{I})} \mathbf{f}_C(\mathcal{I}, \mathbf{x}_c, \mathcal{I}, \mathbf{y}_c)$$

is the sum over all appearances of the template $C(\mathcal{I})$ in the instantiation, and $\mathbf{f}$ is the vector of all $\mathbf{f}_C$.

Given a particular instantiation $\mathcal{I}$ of the schema, the RMN $\mathcal{M}$ produces an *unrolled* Markov network over the attributes of entities in $\mathcal{I}$. The cliques in the unrolled network are determined by the clique templates $\mathbf{C}$. We have one clique for each $c \in C(\mathcal{I})$, and all of these cliques are associated with the same clique potential ϕ_C . In our webpage example, an RMN with the link feature described above would define a Markov net in which, for every link between two pages, there is an edge between the labels of these pages. Fig. 1 illustrates a simple instance of this unrolled Markov network.

4 Learning the Models

In this paper, we focus on the case where the clique templates are given; our task is to estimate the clique potentials, or feature weights. Thus, assume that we are given a

set of clique templates $\mathbf{C}$ which partially specify our (relational) Markov network, and our task is to compute the weights $\mathbf{w}$ for the potentials Φ . In the learning task, we are given some training set D where both the content attributes and the labels are observed. Any particular setting for $\mathbf{w}$ fully specifies a probability distribution $P_{\mathbf{w}}$ over D , so we can use the *likelihood* as our objective function, and attempt to find the weight setting that maximizes the likelihood (ML) of the labels given other attributes. However, to help avoid overfitting, we assume a “shrinkage” prior over the weights (a zero-mean Gaussian), and use maximum a posteriori (MAP) estimation. More precisely, we assume that different parameters are a priori independent and define $p(w_i) = \frac{1}{\sqrt{2\pi}\sigma^2} \exp\{-w_i^2/2\sigma^2\}$.

Both the ML and MAP objective functions are concave and there are many methods available for maximizing them. Our experience is that conjugate gradient and even simple gradient perform very well for logistic regression (Minka 2000) and relational Markov nets.

Learning Markov Networks. We first consider discriminative MAP training in the flat setting. In this case D is simply a set of IID instances; let d index over all labeled training data D . The discriminative likelihood of the data is $\prod_d P_{\mathbf{w}}(y_d | \mathbf{x}_d)$. We introduce the parameter prior, and maximize the log of the resulting MAP objective function:

$$L(\mathbf{w}, D) = \sum_{d \in D} (\mathbf{w} \cdot \mathbf{f}(\mathbf{x}_d, y_d) - \log Z(\mathbf{x}_d)) - \frac{\mathbf{w} \cdot \mathbf{w}}{2\sigma^2} + C.$$

The gradient of the objective function is computed as:

$$\nabla L(\mathbf{w}, D) = \sum_{d \in D} (\mathbf{f}(\mathbf{x}_d, y_d) - E_{P_{\mathbf{w}}}[\mathbf{f}(\mathbf{x}_d, Y_d)]) - \frac{\mathbf{w}}{\sigma^2}.$$

The last term is the shrinking effect of the prior and the other two terms are the difference between the expected feature counts and the empirical feature counts, where the expectation is taken relative to $P_{\mathbf{w}}$:

$$E_{P_{\mathbf{w}}}[\mathbf{f}(\mathbf{x}_d, Y_d)] = \sum_{y'} \mathbf{f}(\mathbf{x}_d, y'_d) P_{\mathbf{w}}(y'_d | \mathbf{x}_d).$$

Thus, ignoring the effect of the prior, the gradient is zero when empirical and expected feature counts are equal.¹ The prior term gives the smoothing we expect from the prior: small weights are preferred in order to reduce overfitting. Note that the sum over y' is just over the possible categorizations for one data sample every time.

Learning RMNs. The analysis for the relational setting is very similar. Now, our data set D is actually a single instantiation $\mathcal{I}$, where the same parameters are used multiple times — once for each different entity that uses a feature. A particular choice of parameters $\mathbf{w}$ specifies a particular

RMN, which induces a probability distribution $P_{\mathbf{w}}$ over the unrolled Markov network. The product of the likelihood of $\mathcal{I}$ and the parameter prior define our objective function, whose gradient $\nabla L(\mathbf{w}, \mathcal{I})$ again consists of the empirical feature counts minus the expected features counts and a smoothing term due to the prior:

$$\mathbf{f}(\mathcal{I}, \mathbf{y}, \mathcal{I}, \mathbf{x}, \mathcal{I}, \mathbf{r}) - E_{\mathbf{w}}[\mathbf{f}(\mathcal{I}, \mathbf{Y}, \mathcal{I}, \mathbf{x}, \mathcal{I}, \mathbf{r})] - \frac{\mathbf{w}}{\sigma^2}$$

where the expectation $E_{P_{\mathbf{w}}}[\mathbf{f}(\mathcal{I}, \mathbf{Y}, \mathcal{I}, \mathbf{x}, \mathcal{I}, \mathbf{r})]$ is

$$\sum_{\mathcal{I}, \mathbf{y}'} \mathbf{f}(\mathcal{I}, \mathbf{y}', \mathcal{I}, \mathbf{x}, \mathcal{I}, \mathbf{r}) P_{\mathbf{w}}(\mathcal{I}, \mathbf{y}' | \mathcal{I}, \mathbf{x}, \mathcal{I}, \mathbf{r}).$$

This last formula reveals a key difference between the relational and the flat case: the sum over $\mathcal{I}, \mathbf{y}'$ involves the exponential number of assignments to all the label attributes in the instantiation. In the flat case, the probability decomposes as a product of probabilities for individual data instances, so we can compute the expected feature count for each instance separately. In the relational case, these labels are correlated — indeed, this correlation was our main goal in defining this model. Hence, we need to compute the expectation over the joint assignments to all the entities together. Computing these expectations over an exponentially large set is the expensive step in calculating the gradient. It requires that we run inference on the unrolled Markov network.

Inference in Markov Networks. The inference task in our conditional Markov networks is to compute the posterior distribution over the label variables in the instantiation given the content variables. Exact algorithms for inference in graphical models can execute this process efficiently for specific graph topologies such as sequences, trees and other low treewidth graphs. However, the networks resulting from domains such as our hypertext classification task are very large (in our experiments, they contain tens of thousands of nodes) and densely connected. Exact inference is completely intractable in these cases.

We therefore resort to approximate inference. There is a wide variety of approximation schemes for Markov networks. We chose to use *belief propagation* for its simplicity and relative efficiency and accuracy. 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 Markov networks. However, recent analysis (Yedidia *et al.* 2000) provides some theoretical justification. Empirical results (Murphy *et al.* 1999) show that it often converges in general networks, and when it does, the marginals are a good approximation to the correct posteriors. As our results in Section 5 show, this approach works well in our domain. We refer the reader to Yedidia *et al.* for a detailed description of the BP algorithm.

5 Experiments

We tried out our framework on the *WebKB* dataset (Craven *et al.* 1998), which is an instance of our hypertext exam-

¹The solution of maximum likelihood estimation with log-linear models is actually also the solution to the dual problem of maximum entropy estimation with constraints that empirical and expected feature counts must be equal (Della Pietra *et al.* 1997).

ple. The data set contains webpages from four different Computer Science departments: Cornell, Texas, Washington and Wisconsin. Each page has a label attribute, representing the type of webpage which is one of *course*, *faculty*, *student*, *project* or *other*. The data set is problematic in that the category *other* is a grab-bag of pages of many different types. The number of pages classified as *other* is quite large, so that a baseline algorithm that simply always selected *other* as the label would get an average accuracy of 75%. We could restrict attention to just the pages with the four other labels, but in a relational classification setting, the deleted webpages might be useful in terms of their interactions with other webpages. Hence, we compromised by eliminating all *other* pages with fewer than three outlinks, making the number of *other* pages commensurate with the other categories.² For each page, we have access to the entire html of the page and the links to other pages. Our goal is to collectively classify webpages into one of these five categories. In all of our experiments, we learn a model from three schools and test the performance of the learned model on the remaining school, thus evaluating the generalization performance of the different models.

Unfortunately, we cannot directly compare our accuracy results with previous work because different papers use different subsets of the data and different training/test splits. However, we compare to standard text classifiers such as Naive Bayes, Logistic Regression, and Support Vector Machines, which have been demonstrated to be successful on this data set (Joachims 1999).

Flat Models. The simplest approach we tried predicts the categories based on just the text content on the webpage. The text of the webpage is represented using a set of binary attributes that indicate the presence of different words on the page. We found that stemming and feature selection did not provide much benefit and simply pruned words that appeared in fewer than three documents in each of the three schools in the training data. We also experimented with incorporating meta-data: words appearing in the title of the page, in anchors of links to the page and in the last header before a link to the page (Yang *et al.* 2002). Note that meta-data, although mostly originating from pages linking into the considered page, are easily incorporated as features, i.e. the resulting classification task is still flat feature-based classification. Our first experimental setup compares three well-known text classifiers — Naive Bayes, linear support vector machines³ (Svm), and logistic regression (Logistic) — using words and meta-words. The results, shown in Fig. 2(a), show that the two discriminative approaches outperform Naive Bayes. Logistic and Svm give very similar

²The resulting category distribution is: course (237), faculty (148), other (332), research-project (82) and student (542). The number of remaining pages for each school are: Cornell (280), Texas (292), Washington (315) and Wisconsin (454). The number of links for each school are: Cornell (574), Texas (574), Washington (728) and Wisconsin (1614).

³We trained one-against-others Svm for each category and during testing, picked the category with the largest margin.

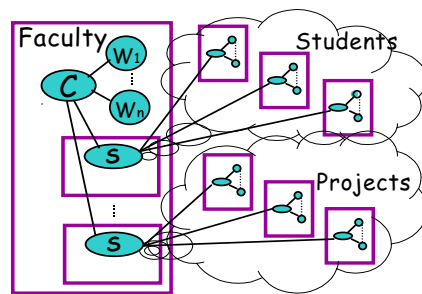


Figure 3: An illustration of the Section model.

results. The average error over the 4 schools was reduced by around 4% by introducing the meta-data attributes.

Relational Models. Incorporating meta-data gives a significant improvement, but we can take additional advantage of the correlation in labels of related pages by classifying them collectively. We want to capture these correlations in our model and use them for transmitting information between linked pages to provide more accurate classification. We experimented with several relational models. Recall that logistic regression is simply a flat conditional Markov network. All of our relational Markov networks use a logistic regression model locally for each page.

Our first model captures direct correlations between labels of linked pages. These correlations are very common in our data: courses and research projects almost never link to each other; faculty rarely link to each other; students have links to all categories but mostly courses. The Link model, shown in Fig. 1, captures this correlation through links: in addition to the local bag of words and meta-data attributes, we introduce a relational clique template over the labels of two pages that are linked.

A second relational model uses the insight that a webpage often has internal structure that allows it to be broken up into *sections*. For example, a faculty webpage might have one section that discusses research, with a list of links to all of the projects of the faculty member, a second section might contain links to the courses taught by the faculty member, and a third to his advisees. This pattern is illustrated in Fig. 3. We can view a section of a webpage as a fine-grained version of Kleinberg’s hub (Kleinberg 1999) (a page that contains a lot of links to pages of particular category). Intuitively, if we have links to two pages in the same section, they are likely to be on similar topics. To take advantage of this trend, we need to enrich our schema with a new relation *Section*, with attributes *Key*, *Doc* (document in which it appears), and *Category*. We also need to add the attribute *Section* to *Link* to refer to the section it appears in. In the RMN, we have two new relational clique templates. The first contains the label of a section and the label of the page it is on:

```
SELECT doc.Category, sec.Category
FROM Doc doc, Section sec
WHERE sec.Doc = doc.Key
```

The second clique template involves the label of the section

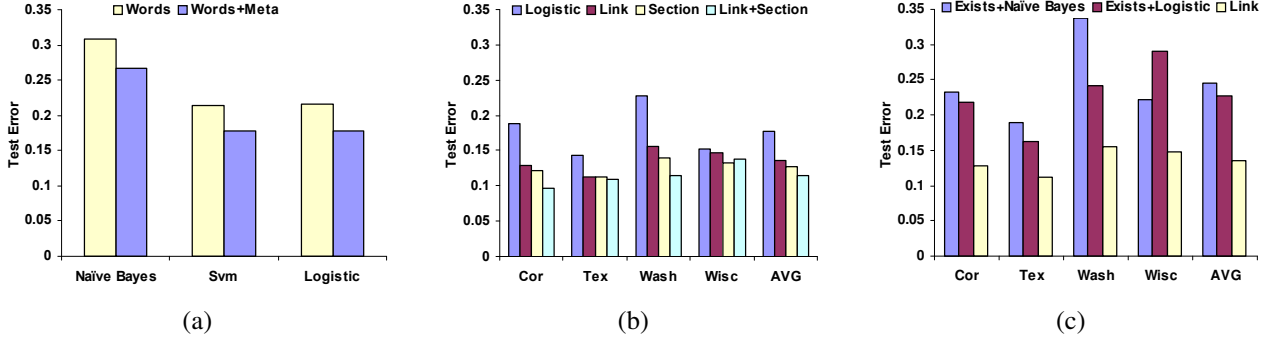


Figure 2: (a) Comparison of Naive Bayes, Svm, and Logistic on WebKB, with and without meta-data features. (Only averages over the 4 schools are shown here.) (b) Flat versus collective classification on WebKB: flat logistic regression with meta-data, and three different relational models: Link, Section, and a combined Section+Link. (c) Comparison of generative and discriminative relational models. Exists+NaiveBayes is completely generative. Exists+Logistic is generative in the links, but locally discriminative in the page labels given the local features (words, meta-words). The Link model is completely discriminative.

containing the link and the label of the target page.

```
SELECT sec.Category, doc.Category
FROM Section sec, Link link, Doc doc
WHERE link.Sec = sec.Key and link.To = doc.Key
```

The original dataset did not contain section labels, so we introduced them using the following simple procedure. We defined a section as a sequence of three or more links that have the same path to the root in the html parse tree. In the training set, a section is labeled with the most frequent category of its links. There is a sixth category *none*, assigned when the two most frequent categories of the links are less than a factor of 2 apart. In the entire data set, the breakdown of labels for the sections we found is: *course* (40), *faculty* (24), *other* (187), *research.project* (11), *student* (71) and *none* (17). Note that these labels are hidden in the test data, so the learning algorithm now also has to learn to predict section labels. Although not our final aim, correct prediction of section labels is very helpful. Words appearing in the last header before the section are used to better predict the section label by introducing a clique over these words and section labels.

We compared the performance of Link, Section and Section+Link (a combined model which uses both types of cliques) on the task of predicting webpage labels, relative to the baseline of flat logistic regression with meta-data. Our experiments used MAP estimation with a Gaussian prior on the feature weights with standard deviation of 0.3. Fig. 2(b) compares the average error achieved by the different models on the four schools, training on three and testing on the fourth. We see that incorporating any type of relational information consistently gives significant improvement over the baseline model. The Link model incorporates more relational interactions, but each is a weaker indicator. The Section model ignores links outside of coherent sections, but each of the links it includes is a very strong indicator. In general, we see that the Section models performs slightly better. The joint model is able to combine bene-

fits from both and generally outperforms all of the other models. The only exception is for the task of classifying the Wisconsin data. In this case, the joint Section+Link model contains many links, as well as some large tightly connected loops, so belief propagation did not converge for a subset of nodes. Hence, the results of the inference, which was stopped at a fixed arbitrary number of iterations, were highly variable and resulted in lower accuracy.

Discriminative vs Generative. Our last experiment illustrates the benefits of discriminative training in relational classification. We compared three models. The Exists+Naive Bayes model is a completely generative model proposed by Getoor *et al.* (2001). At each page, a naive Bayes model generates the words on a page given the page label. A separate generative model specifies a probability over the existence of links between pages conditioned on both pages' labels. We can also consider an alternative Exists+Logistic model that uses a discriminative model for the connection between page label and words — i.e. uses logistic regression for the conditional probability distribution of page label given words. This model has equivalent expressive power to the naive Bayes model but is discriminatively rather than generatively trained. Finally, the Link model is a fully discriminative (undirected) variant we have presented earlier, which uses a discriminative model for the label given both words and link existence. The results, shown in Fig. 2(c), show that discriminative training provides a significant improvement in accuracy: the Link model outperforms Exists+Logistic which in turn outperforms Exists+Naive Bayes.

As illustrated in Table 1, the gain in accuracy comes at some cost in training time: for the generative models, parameter estimation is closed form while the discriminative models are trained using conjugate gradient, where each iteration requires inference over the unrolled RMN. On the other hand, both types of models require inference when the model is used on new data; the generative model con-

	Links	Links+Section	Exists+NB
Training	1530	6060	1
Testing	7	10	100

Table 1: Average train/test running times (seconds). All runs were done on a 700Mhz Pentium III. Training times are averaged over four runs on three schools each. Testing times are averaged over four runs on one school each.

structs a much larger, fully-connected network, resulting in significantly longer testing times. We also note that the situation changes if some of the data is unobserved in the training set. In this case, generative training also require an iterative procedure (such as EM) where each iteration uses the significantly more expressive inference.

6 Discussion and Conclusions

In this paper, we propose a new approach for classification in relational domains. Our approach provides a coherent probabilistic foundation for the process of collective classification, where we want to classify multiple entities, exploiting the interactions between their labels. We have shown that we can exploit a very rich set of relational patterns in classification, significantly improving the classification accuracy over standard flat classification.

In some cases, we can incorporate relational features into standard flat classification. For example, when classifying papers into topics, it is possible to simply view the presence of particular citations as atomic features. However, this approach is limited in cases where some or even all of the relational features that occur in the test data are not observed in the training data. In our WebKB example, there is no overlap between the webpages in the different schools, so we cannot learn anything from the training data about the significance of a hyperlink to/from a particular webpage in the test data. Incorporating basic features (e.g., words) from the related entities can aid in classification, but cannot exploit the strong correlation between the *labels* of related entities that RMNs capture.

Our results in this paper are only a first step towards understanding the power of relational classification. On the technical side, we can gain significant power from introducing hidden variables (that are not observed even in the training data), such as the degree to which a webpage is an authority (Kleinberg 1999). Furthermore, as we discussed, there are many other types of relational patterns that we can exploit. We can also naturally extend the proposed models to predict relations between entities, for example, advisor-advisee, instructor-course or project-member.

Hypertext is the most easily available source of structured data, however, RMNs are generally applicable to any relational domain. In particular, social networks provide extensive information about interactions among people and organizations. RMNs offer a principled method for learning to predict communities of and hierarchical structure between people and organizations based on both the local at-

tributes and the patterns of static and dynamic interaction. Given the wealth of possible patterns, it is particularly interesting to explore the problem of inducing them automatically. We intend to explore this topic in future work.

Acknowledgments. This work was supported by ONR Contract F3060-01-2-0564-P00002 under DARPA's EELD program. P. Abbeel was also supported by a Siebel Graduate Fellowship.

References

- S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *Proc. of ACM SIGMOD98*, pages 307–318, 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 AAAI98*, pages 509–516, 1998.
- S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(4):380–393, 1997.
- L. Egghe and R. Rousseau. *Introduction to Informetrics*. Elsevier, 1990.
- N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *Proc. IJCAI99*, pages 1300–1309, Stockholm, Sweden, 1999.
- L. Getoor, E. Segal, B. Taskar, and D. Koller. Probabilistic models of text and link structure for hypertext classification. In *Proc. IJCAI01 Workshop on Text Learning: Beyond Supervision*, Seattle, Wash., 2001.
- T. Joachims. Transductive inference for text classification using support vector machines. In *Proc. ICML99*, pages 200–209. Morgan Kaufmann Publishers, San Francisco, US, 1999.
- J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46(5):604–632, 1999.
- D. Koller and A. Pfeffer. Probabilistic frame-based systems. In *Proc. AAAI98*, pages 580–587, Madison, Wisc., 1998.
- J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proc. ICML01*, 2001.
- T. Minka. Algorithms for maximum-likelihood logistic regression. <http://lib.stat.cmu.edu/minka/papers/logreg.html>, 2000.
- K. P. Murphy, Y. Weiss, and M. I. Jordan. Loopy belief propagation for approximate inference: an empirical study. In *Proc. UAI99*, pages 467–475, 1999.
- 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, 2000.
- J. Pearl. *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann, San Francisco, 1988.
- S. Slattery and T. Mitchell. Discovering test set regularities in relational domains. In *Proc. ICML00*, pages 895–902, 2000.
- B. Taskar, E. Segal, and D. Koller. Probabilistic classification and clustering in relational data. In *Proc. IJCAI01*, pages 870–876, Seattle, Wash., 2001.
- V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, New York, 1995.
- Y. Yang, S. Slattery, and R. Ghani. A study of approaches to hypertext categorization. *Journal of Intelligent Information Systems*, 18(2), 2002.
- J. Yedidia, W. Freeman, and Y. Weiss. Generalized belief propagation. In *NIPS*, pages 689–695, 2000.