Simple Models and Classification in Networked Data*

Sofus A. Macskassy SMACSKAS@STERN.NYU.EDU
Foster Provost FPROVOST@STERN.NYU.EDU
NYU Stern School of Business, 44 W. 4th Street, New York, NY 10012

Abstract

When entities are linked by explicit relations, classification methods that take advantage of the network can perform substantially better than methods that ignore the network. This paper argues that studies of relational classification in networked data should include simple network-only methods as baselines for comparison, in addition to the non-relational baselines that generally are used. In particular, comparing more complex algorithms with algorithms that only consider the network (and not the features of the entities) allows one to factor out the contribution of the network structure itself to the predictive power of the model. We examine several simple methods for network-only classification on previously used relational data sets, and show that they can perform remarkably well. The results demonstrate that the inclusion of network-only classifiers can shed new light on studies of relational learners.

1. Motivation

In recent years, we have seen considerable advances in algorithms for relational learning, especially statistically based algorithms. These algorithms have been developed in a variety of different research fields and problem settings. Generally, these algorithms consider not only the features of the entities to be classified, but the relations to and the features of linked entities. Observed improvements in generalization performance demonstrate that taking advantage of relational information in addition to attribute-value information can improve performance—sometimes substantially.

In this paper,[1] we argue that complex relational methods should be compared not only to non-relational methods, but also to simple relational methods. Comparison to simple relational methods is essential for understanding the source(s) of improved performance. In particular, for relational learning in networked data (which we discuss in detail below), simple "network-only" classifiers must be used as baselines to assess how much classification accuracy can be attributed simply to the nodes' positions in the network, and not to more complex modeling.

[0]Macskassy, S.A. and Provost, F.J., "Simple Models and Classification in Networked Data," CeDER Working Paper 03-04, Stern School of Business, New York University, NY, NY 10012. Jan 2004.

[1]Some of this paper's argument and results appeared in a workshop last year (Macskassy & Provost, 2003).

In the traditional, non-relational learning setting, the entities to be classified are (assumed to be) i.i.d., so they can safely be considered independently when modeling. Relational data have dependencies between entities, violating the assumption of independence: the classification of an entity may depend on information about entities to which it is related, either directly or through arbitrarily long chains of relations.

In this paper, we consider entities that are connected in a single network. Some entities' classifications are known and some are to be estimated. The labeled entities may be used as training data and also are available for use during estimation. This scenario represents many real-world classification tasks, especially those involving social networks. For example, in fraud detection entities to be classified as being fraudulent or legitimate are intertwined with those for whom classifications are known. In counterterrorism and law enforcement, suspicious people may interact with known bad guys. Some networked data are a by-product of social networks, rather than directly representing the network itself. For example, networks of web pages are built by people and organizations that are interconnected; when classifying web pages, some classifications may be known and some may need to be estimated.

Prior studies have shown that relational learning algorithms can improve classification in networked data, by taking into account not only the features of the entities to be classified, but also the relations to and features of linked entities. There are many applicable relational learning algorithms (Emde & Wettschereck, 1996; Flach & Lachiche, 1999; Dzeroski & Lavrac, 2001); two examples of their use in networked data are (Taskar et al., 2001; Neville et al., 2003). However, it has not been standard practice for studies of learning from networked data to compare to classification methods that ignore all feature information, using only the links between entities and any known class labels.

The main contribution of this paper is the argument and demonstration that for classification in networked data, simple, network-only methods must be considered as baselines against which more complex methods are compared. Studies showing the power of relational learning in networked data require a control for the classification power of the network itself. We present several straightforward methods for network-only classification. We show that these simple network-only classifiers can perform re-

markably well on some data sets by comparing their performance with published results using complex relational learning algorithms. The inclusion of the simple models can shed new light on the performance of relational learners. Finally we point out some limitations of our study—a main one being that even though they perform remarkably in our study, the simple models still may fall short of a complete control for the predictive power of the network.

2. Relational Data and Networked Data

We begin by clarifying the type of relational classification task that is the focus of this paper. A relational classification task must contain: (1) entities—*e.g.*, people, movies, atoms, web-pages, etc., (2) relations between entities—*e.g.*, directs(person,movie), bound-to(atom,atom), links-to(web-page,web-page), and (3) a set of target entities—*e.g.*, person, molecule, web-page, etc. The notion of a relation is vague; we simply assume that relations between entities have been identified intrinsically by the domain structure, by a domain expert, or by some other means. We will describe relational data using graph terminology, where nodes in the graph represent the entities in the domain and the edges between nodes represent the relations between the entities.

2.1. Single-Entity Graphs

Single-entity graphs represent relational tasks where the target entities are not interconnected by relationship paths in the graph. Each target entity is isolated from other target entities. Sample problems that belong to this category include:

Medical diagnosis:
Entities: patients, medical tests.
Target entities: patients.
Relations: test-performed(patient,medical-test).
Classification problem: diagnose whether a patient has a particular disease.

Carcinogenic molecules:
Entities: atoms.
Target entities: molecules made up of all atoms in a single graph.
Relations: bound-to(atom,atom).
Classification problem: is the molecule carcinogenic?

Single-entity graph tasks can be further subcategorized based on whether the target entity is a node in the graph or is a non-trivial subgraph. For single-entity, node-target (SENT) problems, the graph often forms a tree with the target entity (e.g., patient) at the root of the tree, and other entities (e.g., various medical tests) as descendents. For single-entity, subgraph-target (SEST) problems the target subgraph often is the entire graph, to be characterized by its internal structure.

2.2. Multi-Entity Graphs

A different sort of relational classification task has target entities interconnected to some degree (although there may be disconnected graph components). Sample problems that belong to this category include:

- **IMDb (Internet Movie Database):**
  Entities: movies, people (actors, directors, producers), production companies.
  Target entities: movies.
  Relations: acted-in, produced-by, directed-by.
  Classification problem: "predict" blockbusters.

- **Research paper citations:**
  Entities: research papers, authors.
  Target entities: authors or papers.
  Relations: author-of, cited-by.
  Classification problem: identify the type of research paper or identify the author of a paper.

- **Web-pages/Web-sites:**
  Entities: web-pages.
  Target entities: web-pages.
  Relations: links-to.
  Classification problem: classify web pages or web sites by type.

Multi-entity graph tasks can be further subdivided based on whether entities with known labels interconnect with entities for which labels are to be estimated. For multi-entity, separate-data (MESD) tasks, entities with known labels and entities to be estimated are in separate (disconnected) graphs. MESD tasks lend themselves naturally to training/testing divisions.

For multi-entity, networked-data (MEND) tasks, entities for which labels are to be estimated can be interconnected with entities with known data. For MEND tasks, there is not a natural notion of separation into training and test sets. In real MEND applications (such as those mentioned in the introduction) it is likely that labeled entities would be used both for training and as background knowledge during testing. It is on this last category, MEND or simply "networked data," that this paper focuses.

2.3. The Power of the Network

Generally, when assessing the modeling power of relational learning algorithms on networked data, it is essential to control for the classification power inherent in the structure of the network given a set of known labels. We argue that a straightforward (albeit incomplete) way to provide such a control is to compare to simple classifiers that only use the network and the known labels. Consider a leave-one-out classification setting on a graph that contains two disconnected components, each of which has homogenous class membership. In this extreme example, the structure of the network alone allows perfect classification.

Furthermore, for MEND tasks it is possible for information about known labels to propagate through the network from known to unknown entities. Note that for both MEND and MESD data, it is possible to propagate information about *estimations* through the network, if actual labels are not available. Such "collective classification" has been receiving increasing attention in machine learning research (Pearl, 1998; Chakrabarti et al., 1998; Murphy et al., 1999; Neville & Jensen, 2000; Taskar et al., 2001) and is the basis for the simple models we describe next.

3. Simple Relational Models

To demonstrate our thesis, we implemented three simple, network-only methods. All three take advantage of a first-order Markov assumption on the network: only a node's local neighborhood is necessary for classification. The first two, closely related, methods add an assumption of homophily—that similar entities are more likely to be interconnected—and simple belief propagation (Pearl, 1998) in cases where neighbors' labels are not known. The third method is a simplified variant of a Markov Random Field (Dobrushin, 1968; Geman & Geman, 1984; Winkler, 2003).

3.1. Relational Neighbor (RN) Classifiers

Our first classifier estimates class probabilities based on two assumptions in addition to the Markov assumption: (1) some entities' class labels are known within the same MEND structure, and (2) the entities exhibit homophily—i.e., linked entities have a propensity to belong to the same class (Blau, 1977; McPherson et al., 2001). A homophily-based classifier is an important baseline, because homophily is ubiquitous in social networks and is the basis for social theories (Blau, 1977; McPherson et al., 2001). Homophily was one of the first characteristics noted by early social network researchers (Almack, 1922; Bott, 1928; Richardson, 1940; Loomis, 1946; Lazarsfeld & Merton, 1954), and holds for a wide variety of different relationships (McPherson et al., 2001). It seems reasonable to conjecture that homophily will also be present in other sorts of networks, especially networks of artifacts.

Definition. Given an entity e and a set D_e of entities linked to e, the relational-neighbor (RN) classifier estimates $P(c|e)$, the probability that an entity e belongs to class c, as the (weighted) proportion of entities in D_e that belong to class c.

$$P(c|e) = \frac{1}{Z} \sum_{\{e_j \in D_e | \text{label}(e_j)=c\}} w(e, e_j), \qquad (1)$$

where $Z = \sum_{e_i \in D_e} w(e, e_i)$, and $w(e, e_i)$ is the weight of the link[2] between entities e and e_i. Entities in D_e that are not of the same type as e are ignored. If D_e is empty or has no entities with known class labels, then the RN will estimate e based on the class prior (of the known labels).

RN only takes the local neighborhood of a target node into account. This may be a poor approximation to "the power of the network" if many of the entities in D_e are unknown. More of the network can be taken into account by allowing class information to propagate through the network, a technique that has been used successfully before—e.g., iterative classification (Neville & Jensen, 2000), relaxation labeling (Chakrabarti et al., 1998) and belief propagation (Pearl, 1998).

Definition. The iterative relational-neighbor classifier (RN*) iteratively classifies networked entities using the RN classifier in its inner loop. We define RN^i as the model at iteration i, where RN^0 defines what is initially known and RN^1 is equivalent to RN. At iteration i, RN^i uses the labels given by RN$^{(i-1)}$ to predict class-membership of unknown instances. In the case where the class-membership probability of a neighboring entity, $e_j \in D_e$, is a prediction from RN* and was not initially known, the class with the highest probability score is used. An entity e will be classified as unknown if the (weighted) majority is unknown.[3] For the experiments in paper, RN* stops when no unknown entities are left or when no new entities can be labeled (as could be the case when there are isolated components with no known labels).

All RN results in this paper are based on RN*.

3.2. Probabilistic Relational Neighbor (pRN) Classifiers

RN* propagates class labels only when certainty reaches a critical level. An alternative is to propagate estimates of the probability of class membership, such that there is an estimate for all nodes at all times. This allows the incorporation of the marginal class distribution as a prior, and also would allow the incorporation of Bayesian priors or estimation by other learning algorithms (Neville & Jensen, 2000) (neither of which we consider further in this paper).

Definition. The probabilistic relational-neighbor classifier (pRN) estimates $P(c|e)$ as the (weighted) mean of the class-membership probabilities of the entities in D_e.

$$P(c|e) = \frac{1}{Z} \sum_{e_j \in D_e} w(e, e_j) * P(c|e_j), \qquad (2)$$

where D_e, Z and $w(e, e_j)$ are defined as before. Entities whose class labels are not known are assigned a prior (for this paper: the marginal class frequency).

[2] Note that the notion of "linked to" is domain dependent and, as we will show, even for the same domain different definitions can lead to (very) different performance.

[3] In cases where too little propagation takes place, because of too much weight from unknown labels, the need for a majority of weight from a known class can be weakened. This was not necessary for the cases presented in this paper.

Definition. The iterative probabilistic relational-neighbor classifier (pRN*) is similar to RN*, except it uses pRN in its inner loop and all initially unknown instances have their probabilities continuously updated. Unlike RN*, pRN* updates class-probabilities of *all* initially unknown entities at every iteration. Because of the loopy nature of the propagation, there is no guarantee of convergence, though in all our test cases the probabilities seem to be converging. Propagation stops based on a maximum number of iterations as well as a convergence stopping criterion.

All pRN results in this paper will be based on pRN*.

3.3. Network-only Markov Random Field Classifiers (noMRF)

Methods based on Markov Random Fields (MRFs) (Dobrushin, 1968; Geman & Geman, 1984; Winkler, 2003) also utilize the structure of the network and (potentially) known class labels. They do not rely on a homophily assumption, but rather learn how different configurations of neighbors' classes affect a target entity's class. To our knowledge, all classification methods based on MRFs algorithms need to be provided with initial estimations of the entities' classes. These initial estimations are based either on an exogenous initial estimation (e.g., an original pixel value in an image) or based on a separate (learned) classifier that uses only attributes of the instance (e.g., the text of a web page (Chakrabarti et al., 1998)).

We implemented a simple network-only Markov Random Field (noMRF) model, based on the relaxation labeling algorithm described by Chakrabarti (Chakrabarti et al., 1998), with some notable differences. For a network-only classifier, using initial estimates based on attributes would not be proper. We therefore use the same scheme as for pRN—initialize unknown labels to the class prior. We use the same inner Bayesian classifier as in the original work, without the attribute-specific probabilities:[4]

$$P(c|e) = P(D_e|c) \cdot P(c), \tag{3}$$

where

$$P(D_e|c) = \Pi_{e_j \in D_e} P(\text{label}(e_j)|c)^{w(e,e_j)}. \tag{4}$$

Note that the original work differentiated between incoming and outgoing links, whereas we do not.

This classifier assumes all neighbor labels are known. When a subset of neighbors are unknown, we use their current class estimations to predict $P(c|e)$. This updates the class estimation of e, and thus would influence e's neighbor estimations. We therefore iterate until the estimations converge. More formally:

$$P(c|\Delta_K)^{(n+1)} = \sum_{D_e^U \in \Omega_e^U} P(c|D_e^U, D_e^K) \cdot P(D_e^U|\Delta_K)^{(n)}, \tag{5}$$

where

$$P(D_e^U|\Delta_K) = \Pi_{e_j \in D_e^U} P(\text{label}(e_j)|\Delta_K). \tag{6}$$

where D_e^U is the set of neighbors of e whose labels are unknown, D_e^K are the set of neighbors whose labels are known, Δ_K is everything that is known in the network and Ω_e^U is the set of all possible class labelings of D_e^U. There are $m^{|D_e^U|}$ of these, where m is the number of possible classes. This can quickly become intractable as the number of unknown neighbors grows and an approximate solution is needed. Gibbs sampling (Geman & Geman, 1984) is rapidly becoming the method of choice. An alternative method, proposed by the Chakrabarti et al., is based on the k-shortest-path algorithm (KSP). We adopt this idea, partially to stay as close to the original method as possible. However, we base our summation on the Viterbi algorithm (Forney, 1973) rather than KSP. This is the only other difference between our algorithm and the original algorithm. Note that the use of the Viterbi algorithm is possible due to the independence among neighbors. Use of the Viterbi algorithm rather than KSP further makes our summation over Ω_e^U exact rather than approximate.

4. Case Studies

We now assess the classification performance of these simple, network-only classifiers in three case studies. Each study is based on data that were used in a published relational-learning research paper. The purpose of these case studies is to demonstrate that the power of the network indeed must be controlled for, because a simple, network-only model performs remarkably well. The existence of published results allows us to calibrate how well the simple models perform. In each case, we replicated the experimental setup used in prior work as faithfully as we could; we indicate our deviations. In the third case study, the results are not directly comparable to the prior results (for reasons we explain), but nevertheless support our thesis. We emphasize that we are taking the results out of context for the purpose of our demonstration; the reader should consult the original papers before drawing any conclusions about the research contained therein.

4.1. IMDb

We used the Internet Movie Database (IMDb)[5] data set to build models predicting successful movies based on box-office receipts (Jensen & Neville, 2002a). Following the work of Neville et al. (Neville et al., 2003), we focus on movies released in the United States between 1996 and 2001 with the goal of predicting whether a movie "will be" a blockbuster (the opening weekend box-office receipts exceed $2 million) (Neville et al., 2003). Using the database from the authors of this study we extracted "blockbuster" classifications. However, we could not recreate the com-

[4] The original classifier was defined as: $P(c|e) = P(D_e|c) \cdot P(\tau_e|e) \cdot P(c)$, with τ_e being the text of document-entity e.

[5] `http://www.imdb.com`

link-type	$\mathbf{AUC}_{\mathrm{RN}*}$	$\mathbf{AUC}_{\mathrm{pRN}*}$	$\mathbf{AUC}_{\mathrm{noMRF}}$
actor	0.766	0.734	0.704
director	0.658	0.483	0.595
producer	0.850	0.705	0.755
prodco	0.862	0.822	0.836

Table 1. AUCs of RN*, pRN* and noMRF using only 1 link type.

link-type	$\mathbf{AUC}_{\mathrm{RN}*}$	$\mathbf{AUC}_{\mathrm{pRN}*}$	$\mathbf{AUC}_{\mathrm{noMRF}}$
prod+prodco	0.884	0.841	0.853
dir+prod+prodco	0.885	0.841	0.854

Table 2. AUCs of RN*, pRN* and noMRF using a forward-selection feature-based search to combine multiple link types.

plete graph as described in the original work, which used 1364 movies (45% of those being blockbusters). We instead used a data set obtained from the IMDb web-site. We identified 1441 movies released between 1996 and 2001 that we were able to link up with a "blockbuster" classification in the original database. In our version of the data, 615 of the 1441 movies (42.6%) were classified as "blockbuster."

Links between movies are through various other entities (actors, studios, production companies, etc.), and we consider the links to be typed by the entity through which they pass (e.g., $\mathrm{RN}_{\mathrm{producer}}$). Based on a suggestion from David Jensen, we consider four types of links: {`actor, director, producer, production company`[6]}.

We used 10-fold cross-validation to generate predictions for all training examples. It is then straightforward to generate an ROC curve—using the class-membership probabilities produced by RN—and the area under the ROC curve (AUC) by pooling these predictions and sorting the prediction scores for the primary class ("blockbuster," in our case) (Fawcett, 2003). In order to account for variance, we ran the 10-fold cross-validation 10 times, each time with a different random partition. Table 1 shows the mean AUCs for the simple network-only classifiers on each of the four link types (the standard deviations all are less than 0.013).

Using Relational Probability Trees (RPTs) and Relational Bayesian Classifiers (RBCs), the prior work reported AUCs of 0.82 and 0.85, respectively, using eight attributes of related entities, such as the most prevalent genre of the movie's studio. Observe that in comparison, the network-only classifiers can perform very well, supporting our argument that such classifiers must be considered as baselines against which to compare more complex relational learning methods.

For this case study, the selection of links to use is an important problem which we do not claim to solve. However, it may be possible to perform even better by considering more than one link type. To test this, we ran a simple forward feature-selection search: for each remaining (unused) link-type/feature, add it to the current best performer—starting with prodco, the best performer from Table 1—and keep the combination that reported the best performance; keep adding one feature at a time until it stops improving the performance.[7] Using this methodology, we end up with the AUCs presented in Table 2.[8]

noMRF generally performs worse than RN*. This suggests that the homophily bias is appropriate, which concurs with prior observations of considerable relational autocorrelation in this domain (Jensen & Neville, 2002b). Apparently, noMRF can not take advantage of its increased flexibility, either because sufficient signal is not present, or because there is not enough training data to capture it. To our knowledge no better results have been reported on this relational learning problem.

4.2. CoRA

This case study uses the CoRA data set (McCallum et al., 2000), a data set of computer science research papers, which includes the full citation graph as well as labels for the topic of each paper (and potentially sub- and sub-sub-topics).[9] Following a prior study (Taskar et al., 2001), we focused on 4240 papers within the machine learning topic with the classification task of predicting a paper's sub-topic (there are seven). We used all 4007 unique authors that we could identify in this subset. Thus, our graph differed from the prior study (Taskar et al., 2001) in which they report using 4187 papers and 1454 authors.

Papers can be linked in one of two ways: using a common author, or using a citation. We (somewhat arbitrarily) assign the weight of a relation as the sum of the number of authors two papers have in common and the number of citations that link them to each other. This latter number ordinarily would only be zero or one unless the two papers cite each other.

Using essentially the same methodology as Taskar et al., we varied the proportion of papers for which the class is initially known from 10% to 60%. We varied in 5% increments; we performed a 10–fold cross-validation at

[6] We shorten 'production company' to 'prodco' when describing our results below.

[7] The relational structure of the data complicates things, making it unclear what it means to use only the training set to perform the feature-selection. We circumvented this problem by using all the data in this study. These feature-selection results therefore are optimistic.

[8] We also, separately, performed a brute-force analysis of all possible combinations of link types. For this study, the AUCs reported in Table 2 were the best results.

[9] These labels were assigned by a naive Bayes classifier (McCallum et al., 2000).

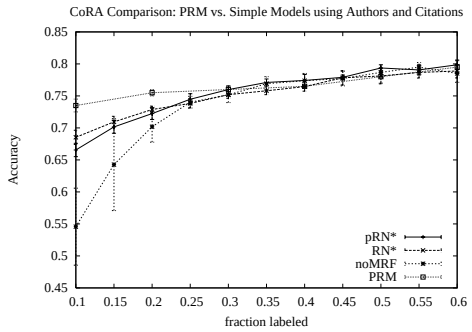

Figure 1. CoRA Comparison

each setting (the previous study performed a 5–fold cross-validation, varying the training set in 10% increments). Figure 1 shows the classification accuracy of RN*, pRN*, noMRF, and Taskar et al.'s Probabilistic Relational Model, which used the text of the page in addition to the author and citation links.[10] RN* took 3–5 iterations to converge, decreasing as we increased the labels that were known initially. We see RN* is competitive with the PRM, matching its performance once 30% of the labels are known initially. For this task, the performance of pRN* is statistically indistinguishable from that of RN*.[11] noMRF performed similarly, but worse for very small amounts of labeled data, with higher variance (recall that noMRF builds a model mapping neighborhood members to entity classifications).

These results again support our argument that simple, network-only models must be considered as baselines for comparing complex relational learners in networked data. If we assume that the results presented here and those of the prior study are directly comparable, much of the performance of the PRM might be explained by the power of the network. The PRM only has an advantage over the simple, network-only methods for very small amounts of labeled data (in which case the estimations from the text apparently add value).

4.3. WebKB

The last case study we present is based on the data set collected by the WebKB Project (Craven et al., 1998).[12] It consists of a set of web pages from four computer science departments, with each page manually labeled into the categories: course, department, faculty, person, project, staff, student or other. This data set includes clearly defined `link-to` relations between pages. Following a prior study, we will classify whether a page belongs to a student (Neville et al., 2003). As with the prior study, we extracted the pages that have at least one incoming and one outgoing link, and kept remaining ("background") pages that either link to a page or are linked to by a page in this subset of pages. This resulted in a data set of 920 pages and 3036 background pages, giving us a total of 3956 pages. This differs from the prior work which had 910 extracted pages and a total of 3877 pages, including the background pages (Neville, 2003).

We create a preliminary edge (*p-edge*) between two pages if one page contains a hyperlink to the other. We weight these p-edges by summing the number of hyperlinks from one page to the other and vice versa (ignoring directionality).

We define "neighbors" for this task as pages 2 p-edges away, based on the prior observation that a student is more likely to have a hyperlink to her advisor or a group/project page rather than to one of her peers (Craven et al., 1998). Therefore it is more likely that student pages have intermediaries in common than direct links. Such 2-hop relations are not unique to this domain; for example, for fraud detection in telecommunications, bandits are often two hops away from previously identified bandits (they call the same numbers). We weight each relation by multiplying the p-edge weights composing the linkage (e.g., if a student page has 2 p-edges to a group page, and a fellow student has 3 p-edges to the same group page, then the weight along that path between those 2 students would be 6). This weight represents how many possible ways two pages could reach each other.

Using the 10-by-10-fold cross-validation methodology described above for the IMDB study, we used the 920 pages identified earlier to create the training/testing folds. For all pages we allowed paths to any background page. For this data set, RN* and pRN* perform very well in an absolute sense: they have mean AUCs of 0.949 and 0.946 respectively, both with standard deviations of 0.001. That means that there is a 95% probability that a randomly selected student page will get a higher RN* score than a randomly selected non-student page. This is the case even though the data comprise four disconnected MEND subnetworks.

Unlike the previous two case studies, these results can not be compared directly to those of prior work, because prior studies have treated these data as comprising a MESD task. Specifically, prior results have been based on a 4-fold cross-validation methodology in which one university's web site is used as a holdout set while the remaining three are used for training.

Nevertheless, results using simple network-only classifiers can provide interesting insight. Let us consider the four sites as separate networks. Observing for each network the relationship between the number of labeled data within a network and the classification performance of the simple classifiers gives a view of the difficulty of the problem.

[10] The PRM values were approximated from the graphs in the original paper.

[11] For the studies in this paper, pRN* generally performed worse or only comparably to RN*. In virtually all tests presented in this paper, when only a small fraction ($\leq 30\%$) of data was initially labeled, RN* performed better than pRN*, though they often had similar performance when $\geq 75\%$ of the data was labeled.

[12] We use the WebKB-ILP-98 data set.

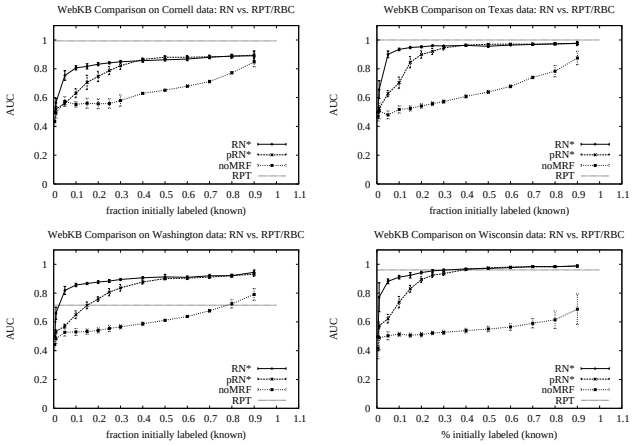

Figure 2. Comparison of RN*, pRN*, noMRF and RPT on the four universities in the WebKB data set. The horizontal lines are for the RPTs trained on the other three universities.

Specifically, it shows how much data would have to be labeled for the power of the network alone to yield a particular accuracy. In many domains the amount of labeling could be translated directly to a cost, which would indicate the value of the MESD modeling.

To demonstrate, considering each university as a separate data set we performed a study similar to that of the CoRA data set above: we randomly pick $l\%$ of the pages and label them. The remaining labels are unknown. Running our methods, we calculate their resulting AUCs. We do this 10 times, each time randomly picking $l\%$ pages to label, giving us an average and standard deviation for the AUCs for a given l. Doing this for $l \in \{\frac{1}{2}, 1, 5, 10, 15, 20, 25, 30, 40, \ldots, 90\}$, we graph the resulting average AUCs as l increases. These results can be compared to the AUCs reported in a MESD study. Figure 2 shows, for each university, the resulting graphs.

Using RPTs, the prior MESD study shows AUCs ranging from 0.716 to 1.0 for RPTs (Neville, 2003). These MESD learners used a set of ten attributes on related entities, such as the URL path and host, as well as structural attributes such as the number of in-links and out-links of each page. (Of course, it does not make sense for them to use labels or entity ids.)

In all cases RN* was able to get close to its best performance even when labeling only 5% of the data. pRN* showed similar behavior, though it needed 30% of the data to be labeled in order to reach close to its best performance. In all cases, though less so on the Cornell data set, RN* is competitive with RPT even having seen only 5% of the data. In fact, RN* was able to outperform RPT on the Washington data set, having seen only 5% of the data.

Some side notes about the different network-only classifiers: pRN* was worse than RN* when few ($< 30\%$) initial labels were known, but was able to "catch up" as the number of known labels increased. Finally, we see that noMRF performed far worse than all methods except when almost nothing was known ($\leq 5\%$). Only on the Washington data was it finally able to beat RPT after having seen 90% of the labels. We will return to this below.

Regarding our main argument, treating WebKB as a MEND classification task, the simple network-only methods again perform remarkably well. Furthermore, they perform well even with only small amounts of labeled data, which provides an interesting complementary view to a MESD classification study.

5. Discussion and Limitations

All three case studies show clearly that network-only classifiers can classify remarkably well. In particular, the simplest—RN*—performs remarkably well, often even with very few labeled data. The other network-only methods never do better and often do worse. RN is biased strongly by the homophily assumption, which should result in good performance when the assumption holds, even for few known labels if the propagation is effective.

On the other hand, basing classification on an assumption of homophily will result in poor or even pathological performance if the assumption does not hold or is violated more seriously. Let us return to the WebKB case study. If we define links simply as the p-edges themselves (hyperlinks weighted by frequency), we get a graph that does not exhibit homophily. Indeed, using a leave-one-out cross-validation (which allows as much labeled data as possible) RN* achieved an AUC of 0.352—classification performance much worse than random guessing!

In these case studies, homophily gave RN* an advantage too large for the lower-bias noMRF to overcome. However, consider again the p-edge version of the WebKB task. noMRF does not exhibit the worse-than-random pathology of RN. In fact, it is able to learn patterns of neighbor labels moderately well (classifying student web pages with AUC $= 0.706$).

In exchange for its lower bias noMRF should exhibit higher variance, which is likely to require more data to overcome. Therefore, it should be expected that when homophily is present to some degree, RN will perform relatively better for smaller amounts of data. This is suggested by our results, but we have not yet seen clear cases of the curves crossing.

What about pRN versus RN? On good-guy/bad-guy networks from a terrorist simulator, we have observed pRN to dominate RN. We believe it is related to the large imbalance in the classes (for which the voting-based RN fails), but we have not looked carefully into the conditions for the applicability of each.

An important limitation of this work is that we randomly choose training data to be labeled. It is likely in real net-

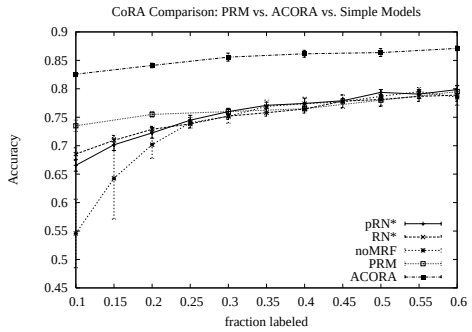

Figure 3. CoRA Comparison

work data that the data for which labels are available are interdependent. For example, you may know all the members of one terrorist cell and none from another. This may dilute the power of network-based methods. If other attributes are available more uniformly, then studies such as this may artificially favor network-only methods, especially simple ones, over attribute-based methods.

We use a particularly strong notion of "network-only" classifiers. Comprehensive studies should look at other notions as well. For example, in MEND data the exact identifiers of known entities can be used directly in classification and learning. For the other three scenarios (SENT, SEST and MESD), this does not make sense: the same entities will not appear in "test" data, so learning programs must generalize based on characteristics of the entities. For example, on the CoRA task Perlich's ACORA (Perlich, 2003) builds a logistic-regression based model, where features are constructed by looking at distributions of the actual paper ids (rather than just the labels), but still ignoring the text. As Figure 3 shows, when given only 5% of the data, ACORA already outperforms all the other methods even when they are given 60% of the data. This suggests a "next tier" of network-only baseline methods.

Finally, with respect to the main argument of our paper, even considering only links and class labels, none of these methods completely controls for the power of the network. They all make a (first-order) Markov assumption–i.e., that "the power of the network" can be reduced to "the power of the neighborhood." Moreover, they use relatively simple techniques even within the neighborhood. Nevertheless, the case studies have shown that very simple models based on the neighborhood alone (perhaps with influence propagated from the rest of the network) can be remarkably powerful. Furthermore, from the point of view of conducting relational learning studies on MEND data, these methods are straightforward enough for anyone to include as baselines.

Acknowledgments

David Jensen made many helpful suggestions, including pointing us to the WebKB data set and suggesting ways to do well on it. We thank Ben Taskar and Andrew McCallum for providing us with versions of the Cora data set. Jennifer Neville was instrumental in our recreating the test environments and setup for the IMDb and WebKB studies. Thanks to Claudia Perlich for many helpful discussions.

References

Almack, J. C. (1922). The Influence of Intelligence on the Selection of Associates. *Sch. Soc., 16*, 529–530.

Blau, P. M. (1977). *Inequality and Heterogeneity: A Primitive Theory of Social Structure*. New York: Free Press.

Bott, H. (1928). Observation of Play Activites in a Nursery School. *Genet. Psychol. Monogr., 4*, 44–88.

Chakrabarti, S., Dom, B., & Indyk, P. (1998). Enhanced Hypertext Categorization Using Hyperlinks. *SIGMOD*.

Craven, M., Freitag, D., McCallum, A., Mitchell, T., Nigam, K., & Quek, C. Y. (1998). Learning to Extract Symbolic Knowledge from the World Wide Web. *15th Conference of the American Association for Artificial Intelligence*.

Dobrushin, R. L. (1968). The Description of a Random Field by Means of Conditional Probabilities and Conditions of its Regularity. *Theory of Probability and its Application, 13*, 197–224.

Dzeroski, S., & Lavrac, N. (2001). *Relational Data Mining*. Berlin; New York: Springer.

Emde, W., & Wettschereck, D. (1996). Relational Instance-Based Learning. *Proceedings 13th International Conference on Machine Learning* (pp. 122–130). Morgan Kaufmann.

Fawcett, T. (2003). *ROC Graphs: Notes and Practical Considerations for Data Mining Researchers*. Technical Report (HPL-2003-4). HP Labs.

Flach, P. A., & Lachiche, N. (1999). 1BC: A First-Order Bayesian Classifier. *Ninth International Workshop on Inductive Logic Programming (ILP'99)* (pp. 92–103). Springer-Verlag.

Forney, G. D. (1973). The Viterbi algorithm. *IEEE, 61*, 268–278.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Trans. PAMI, 6*, 721–741.

Jensen, D., & Neville, J. (2002a). Data Mining in Social Networks. *National Academy of Sciences workshop on Dynamic Social Network Modeling and Analysis*.

Jensen, D., & Neville, J. (2002b). Linkage and Autocorrelation Cause Feature Selection Bias in Relational Learning. *Nineteenth International Conference on Machine Learning (ICML2002)*.

Lazarsfeld, P., & Merton, R. K. (1954). Friendship as a Social Process: A Substantive and Methodological Analysis. In M. Berger, T. Abel and C. H. Page (Eds.), *Freedom and control in modern society*, 18–66. Van Nostrand.

Loomis, C. P. (1946). Political and Occupational Cleavages in a Hanoverian Village. *Sociometry, 9*, 316–3333.

Macskassy, S. A., & Provost, F. J. (2003). A Simple Relational Classifier. *Proceedings of the Second Workshop on Multi-Relational Data Mining (MRDM-2003) at KDD-2003*.

McCallum, A., Nigam, K., Rennie, J., & Seymore, K. (2000). Automating the Construction of Internet Portals with Machine Learning. *Information Retrieval, 3*, 127–163.

McPherson, M., Smith-Lovin, L., & Cook, J. M. (2001). Birds of a Feather: Homophily in Social Networks. *Annual Review of Sociology*, *27*, 415–444.

Murphy, K., Weiss, Y., & Jordan, M. I. (1999). Loopy Belief-propagation for Approximate Inference: An Empirical Study. *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence (UAI)*. Morgan Kaufmann.

Neville, J. (2003). Personal communication.

Neville, J., & Jensen, D. (2000). Iterative Classification in Relational Data. *AAAI Workshop on Learning Statistical Models from Relational Data* (pp. 13–20).

Neville, J., Jensen, D., Friedland, L., & Hay, M. (2003). Learning Relational Probability Trees. *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD-2003)*.

Pearl, J. (1998). *Probabilistic Reasoning in Intelligent Systems*. Morgan Kaufmann.

Perlich, C. (2003). Citation-Based Document Classification. *Workshop on Information Technology and Systems (WITS)*.

Richardson, H. M. (1940). Community of Values as a Factor in Friendsheps of College and Adult Women. *Journal of Social Psychology*, *11*, 303–312.

Taskar, B., Segal, E., & Koller, D. (2001). Probabilistic Classification and Clustering in Relational Data. *17th International Joint Conference on Artificial Intelligence* (pp. 870–878).

Winkler, G. (2003). *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods*. Springer-Verlag. 2nd edition.