A brief survey of machine learning methods for classification in networked data and an application to suspicion scoring

Sofus A. Macskassy SOFMAC@FETCH.COM
Fetch Technologies, Inc, 2041 Rosecrans Ave, El Segundo, CA 90245 USA

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

Abstract

This paper surveys work from the field of machine learning on the problem of within-network learning and inference. To give motivation and context to the rest of the survey, we start by presenting some (published) applications of within-network inference. After a brief formulation of this problem and a discussion of probabilistic inference in arbitrary networks, we survey machine learning work applied to networked data, along with some important predecessors—mostly from the statistics and pattern recognition literature. We then describe an application of within-network inference in the domain of suspicion scoring in social networks. We close the paper with pointers to toolkits and benchmark data sets used in machine learning research on classification in network data. We hope that such a survey will be a useful resource to workshop participants, and perhaps will be complemented by others.

1. Introduction

This paper[1] briefly surveys work from the field of machine learning. We concentrate on methods published in the machine learning literature, as well as methods from other fields that have had considerable impact on the machine learning literature.

Networked data are the special case of relational data where entities are interconnected, such as web-pages or research papers (connected through citations). We focus on *within-network* inference, for which training entities are connected directly to entities whose classifications (*labels*) are to be estimated. This is in contrast to *across-network* inference: learning from one network and applying the learned models to a separate, presumably similar network (Craven et al., 1998; Lu & Getoor, 2003). For within-network inference, networked data have several unique characteristics that both complicate and provide leverage to learning and inference.

Although the network may contain disconnected components, generally there is not a clean separation between the entities for which class membership is known and the entities for which estimations of class membership are to be made. The data are patently not i.i.d., which introduces bias to learning and inference procedures (Jensen & Neville, 2002b). The usual careful separation of data into training and test sets is difficult, and more importantly, thinking in terms of separating training and test sets obscures an important facet of the data. Entities with known classifications can serve two roles. They act first as training data and subsequently as background knowledge during inference. Relatedly, within-network inference allows models to use specific node identifiers to aid inference (Perlich & Provost, 2006).

Network data generally allow *collective inference*, meaning that various interrelated values can be inferred simultaneously. For example, inference in Markov random fields (Besag, 1974) uses estimates of a node's neighbor's labels to influence the estimation of the nodes labels—and vice versa. Within-network inference complicates such procedures by pinning certain values, but again also offers opportunities such as the application of network-flow algorithms to inference. More generally, network data allow the use of the features of a node's neighbors, although that must be done with care to avoid greatly increasing estimation variance (and thereby error) (Jensen et al., 2004).

For this survey we consider methods for classification and class-probability estimation on arbitrary graphs, and not on regular grids. To give motivation and context to the rest of the survey, we start by presenting some (published) applications of within-network inference. Section 3 presents a brief formulation of the within-network problem and discusses probabilistic inference in arbitrary networks. Then, in section 4, we survey machine learning work applied to networked data, along with some important predecessors—mostly from the statistics and pattern recognition literature. We then describe in Section 5 an application for within-network classification: suspicion scoring in social

[1] More detail can be found in three papers that this paper summarizes and excerpts (Macskassy & Provost, 2004; Macskassy & Provost, 2005a; Macskassy & Provost, 2005b).

Appearing in *Workshop on Statistical Network Analysis at the 23^{rd} International Conference on Machine Learning*, Pittsburgh, PA, 2006.

networks. We close the paper with pointers to toolkits and benchmark data sets used in machine learning research on classification in network data.

2. Applications

Many real-world problems exhibit opportunities for within-network classification. A natural application domain is the analysis of linked documents, for example by citations of one sort or another. Such networks of documents are built by people and organizations that are interconnected; when classifying, some classifications may be known and some may need to be estimated (Chakrabarti et al., 1998; Taskar et al., 2001; Neville et al., 2003a; Lu & Getoor, 2003; Macskassy & Provost, 2004). Chakrabarti et al. (1998) classify patents, and various authors classify scientific research papers (e.g., (Taskar et al., 2001)) and web pages (e.g. (Neville et al., 2003a)).

Science also gives us opportunities. The original work on classification in networked data arose in part from statistical physics (for example, consider the Ising model). Segal et al. (2003a; 2003b) present the application of Markov random fields (Dobrushin, 1968; Besag, 1974; Geman & Geman, 1984), and more specifically, relational Markov networks (Taskar et al., 2002), to the discovery of molecular *pathways*, i.e., sets of genes that coordinate to achieve a particular task, from networks formed from protein interactions. Computational linguistics has provided network classification problems, such as the segmentation and labeling of text (e.g., part-of-speech tagging (Lafferty et al., 2001)).

Business has seen a few direct applications of network-based techniques. In fraud detection entities to be classified as being fraudulent or legitimate are intertwined with those for which classifications are known. Fawcett and Provost (1997) discuss and experiment with so-called "state-of-the-art" fraud detection techniques, which look at indirect (two-hop) connections in the call network to prior fraudulent accounts. Cortes et al. (2001) explicitly represent and reason with accounts' local network neighborhoods, for detecting telecommunications fraud. For making product recommendations, Huang et al. (2004) define a neighborhood of similar people for collaborative filtering and use graph-based message passing to draw inferences. Domingos and Richardson (2001) describe how a MRF-based technique could be used to estimate the best candidates for viral marketing. Hill et al. (2006) show strong evidence that statistical, network-based marketing techniques can perform substantially (several times) better than traditional targeted marketing based on demographics and prior purchase data.

With the exception of collaborative filtering, these business applications share the characteristic that data are available on actual social networks, as defined by direct communications. Whether or not hard data are easily available, such networks play an important role in counterterrorism and law enforcement: suspicious people may interact with known 'bad' people. Macskassy and Provost (2005a) present a network classification approach for suspicion scoring in surveillance networks, which we describe below (cf., (Galstyan & Cohen, 2005)). Neville et al. (2005) identified potential SEC violators from data provided by the National Association of Securities Dealers (NASD). Their models, performed as well as or better than the handcrafted models currently in use at NASD.

Some other domains do not quite qualify as applications, but are interesting nevertheless. The Internet Movie Database contains rich relational and networked data such as actors, movies, producers, directors, etc. Jensen and Neville (2002a) predict whether a movie will make more than $2 million dollars in its opening weekend, and others followed (Neville et al., 2003a; Macskassy & Provost, 2003; Macskassy & Provost, 2004). Bernstein et al. (2003) and Macskassy and Provost (2003; 2004) have addressed the problem of categorizing what industry sector a given company is in using financial news to link companies based on whether they were mentioned in the same news story.

Finally, network classification approaches have seen elegant application on a problem that initially does not present itself as a network classification problem. The problem of "transductive" inference (Vapnik, 1998b) is faced when there is a set of labeled data plus a set of data for which classifications must be made. Data points can be linked into a network based on any similarity measure. In this scenario, any classification problem may be seen as a (within-) network classification problem, depending on how the different sorts of data present themselves.

3. Problem Formulation: Within-Network Classification

Traditionally, machine learning methods have treated entities as being independent, which makes it possible to infer class membership on an entity-by-entity basis. With networked data, the class membership of one entity may have an influence on the class membership of a related entity. Furthermore, entities not directly linked may be related by chains of links, which suggests that it may be beneficial to infer the class memberships of all entities simultaneously. Collective inferencing in relational data (Jensen et al., 2004) makes simultaneous statistical judgments regarding the values of an attribute or attributes for multiple entities in a graph G for which some attribute values are not known.

> Given graph $G = (\mathbf{V}, \mathbf{E}, \mathbf{X}, \mathbf{Y})$, where $\mathbf{X}_i$ is an attribute vector and Y_i is a label variable for vertex $v_i \in \mathbf{V}$, and given known values y_i for some subset of vertices $\mathbf{V}^K$, within-network *collective inferencing* is the process of simultaneously inferring the values of Y_i for the remaining vertices, $\mathbf{V}^U = \mathbf{V}–\mathbf{V}^K$, or a probability distribution over those values.

As a shorthand, we will use $\mathbf{Y}^K$ to denote the set (vector) of class values for $\mathbf{V}^K$, and similarly for $\mathbf{Y}^U$. Then,

$G^K = (\mathbf{V}, \mathbf{E}, \mathbf{X}, \mathbf{Y}^K)$ denotes everything that is known about the graph (we do not consider the possibility of unknown edges). Edge $e_{ij} \in \mathbf{E}$ represents the edge between vertices v_i and v_j, and w_{ij} represents the edge weight.

Rather than estimating the full joint probability distribution $P(\mathbf{Y}^U|G^K)$, relational learning often enhances tractability by making a Markov assumption:

$$P(y_i|G) = P(y_i|\mathcal{N}_i), \tag{1}$$

where $\mathcal{N}_i$ is the set of "neighbors" of vertex v_i such that $P(y_i|\mathcal{N}_i)$ is independent of $G - \mathcal{N}_i$ (i.e., $P(y_i|\mathcal{N}_i) = P(y_i|G)$).

Given $\mathcal{N}_i$, a relational model can be used to estimate y_i. Note that $\mathcal{N}_i^U$ $(= \mathcal{N}_i \cap \mathbf{V}^U)$—the set of neighbors of v_i whose label values are not known—could be non-empty. Therefore, a simple application of the relational model may be insufficient. However, the relational model may be used to estimate the labels of $\mathcal{N}_i^U = \mathcal{N}_i - N_i^K$. Further, just as estimates for the labels of $\mathcal{N}_i^U$ influence the estimate of y_i, the estimate of y_i also influences the estimate of the labels of $\mathcal{N}_i^U$. Thus, in order to simultaneously estimate $\mathbf{Y}^U$ various collective methods have been introduced, which we discuss below.

Many of the algorithms developed for within-network classification are heuristic methods without a formal probabilistic semantics (others are heuristic methods with a formal probabilistic semantics). Nevertheless, let us suppose that at inference time we are presented with a probability distribution structured as a graphical model—the network. In general, there are various inference tasks we might be interested in undertaking (Pearl, 1988). We focus primarily on within-network, univariate classification: the computation of the marginal probability of class membership of a particular node (i.e., the variable represented by the node taking on a particular value), conditioned on knowledge of the class membership of certain other nodes in the network. We also discuss methods for the related problem of computing the maximum a posteriori (MAP) joint labeling for $\mathbf{V}$ or only $\mathbf{V}^U$.

For the sort of graphs we expect to encounter in the aforementioned applications, such probabilistic inference is quite difficult. As discussed by Wainwright and Jordan (2003), the naive method of marginalizing by summing over all configurations of the remaining variables is intractable even for graphs of modest size; for binary classification with around 400 unknown nodes, the summation involves more terms than atoms in the visible universe. Inference via belief propagation (Pearl, 1988) is applicable only as a heuristic approximation, because directed versions of many network classification graphs will contain cycles.

The alternative to heuristic ("loopy") belief propagation is the junction-tree algorithm (Cowell et al., 1999), which provides exact solutions for arbitrary graphs. Unfortunately, the computational complexity of the junction-tree algorithm is exponential in the "treewidth" of the junction tree formed by the graph (Wainwright & Jordan, 2003). Since the treewidth is one less than the size of the largest clique, and the junction tree is formed by triangulating the original graph, the complexity is likely to be prohibitive for graphs such as social networks, which can have dense local connectivity and long cycles.

4. Approaches and Techniques

4.1. Local, Relational, and Collective Inference

A large set of approaches to the problem of network classification could be viewed as "node centric," in the sense that they can be viewed as focusing on a single node at a time. For a couple reasons, which we elaborate presently, it is useful to divide such systems into three components. One component, the *relational classifier*, addresses the question: given a node and the node's neighborhood, how should a classification or a class-probability estimate be produced? For example, the relational classifier might combine local features and the labels of neighbors using a naive Bayes model (Chakrabarti et al., 1998) or a logistic regression (Lu & Getoor, 2003). A second component addresses the problem of collective inference: what should we do when a classification depends on a neighbor's classification, and vice versa. Finally, most such methods require initial ("prior") estimates of the values for $P(\mathbf{y}^U|G^K)$. The priors could be Bayesian subjective priors (Savage, 1954), or they could be estimated from data. A common estimation method is to employ a non-relational learner, using available "local" attributes of v_i to estimate y_i (e.g., as done by Besag (1986)).

Viewing network classification approaches through this decomposition is useful for two main reasons. First, it provides a way of describing certain approaches that highlights the similarities and differences among them. Secondly, it expands the small set of methods to a design space of methods, since components can be mixed and matched in ways different from those specified by their creators. In fact, some novel combination may well perform better than the original; there has been little systematic experimentation along these lines (Macskassy & Provost, 2004).

Local and relational classifiers can be drawn from the vast space of classifiers introduced over the decades in machine learning, statistics, pattern recognition, etc., and treated in great detail elsewhere. Collective inference has received much less attention in all these fields, and therefore warrants additional introduction.

Collective inference has its roots mainly in pattern recognition and statistical physics. Markov Random Fields (MRFs) (Dobrushin, 1968; Besag, 1974; Geman & Geman, 1984) have been used extensively for univariate network classification for vision and image restoration. Introductions to MRFs fill textbooks (Winkler, 2003); for our purposes, it is important to point out that they are the basis both directly and indirectly for many network classification approaches. MRFs are used to estimate the

joint probability of a set of nodes based on their immediate neighborhoods under the first-order Markov assumption that $P(x_i|\mathbf{X}/x_i) = P(x_i|\mathcal{N}_i)$, where $\mathbf{X}/x_i$ means all nodes in $\mathbf{X}$ except x_i and $\mathcal{N}_i$ is a neighborhood function returning the neighbors of x_i. In a typical image application, nodes in the network are pixels and the labels are image properties such as whether a pixel is part of a vertical or horizontal border.

Because of the obvious interdependencies among the nodes in an MRF, computing the joint probability of assignments of labels to the nodes ("configurations") requires collective inference. Gibbs sampling (Geman & Geman, 1984) was developed for this purpose in the problem setting of restoring degraded images. Geman and Geman enforce that the Gibbs sampler settles to a final state by using a simulated annealing approach where the temperature is dropped slowly until nodes no longer change state. We omit here the details of Gibbs sampling, discussing it in more detail elsewhere (Macskassy & Provost, 2004).

Besag (1986) noted two problems with Gibbs sampling that are particularly relevant for machine learning applications of network classification. First, the most common use of Gibbs sampling in vision was not to compute the final marginal posteriors, as required by many "scoring" applications, but rather to get final MAP classifications. Second, Gibbs sampling can be very time consuming, especially for large networks (not to mention the problems detecting convergence in the first place). With his Iterated Conditional Modes (ICM) algorithm, Besag introduced the notion of *iterative classification* for scene reconstruction. ICM was presented as being efficient and particularly well suited to maximum marginal classification by node (pixel), as opposed to maximum joint classification over all the nodes (the scene).

Two other, closely related, collective inference techniques are relaxation labeling (RL) (Rosenfeld et al., 1976; Hummel & Zucker, 1983) and (loopy) belief propagation (Pearl, 1988), introduced above. RL was originally proposed as a class of parallel iterative numerical procedures which use contextual constraints to reduce ambiguities in image analysis.

Graph-cut techniques recently have been used in vision research as an alternative to using Gibbs sampling (Boykov et al., 2001); these in essence are collective inference procedures, and are the basis of a collection of modern machine learning techniques. However, they do not quite fit in the node-centric framework, so we treat them separately below.

4.2. Node-centric Approaches

Now, we can describe the essence of several prior systems simply by listing how they solve the problems of local classification, relational classification, and collective inference.

Chakrabarti et al. (1998) studied classifying web-pages based on the text and (possibly inferred) class labels of neighboring pages. Their system paired naive Bayes local and relational classifiers, with relaxation labeling for collective inference. In their experiments, performing network classification using the web-pages' link structure substantially improved classification over using only the local (text) information. Specifically, considering the text of the neighbors generally hurt performance, whereas using only the (inferred) class labels improved performance.

The ICM of Besag (1986) is a node-centric approach where the local and relational classifiers are domain-dependent probabilistic models (based on local attributes and a MRF), and iterative classification is used for collective inference (described above). Neville and Jensen (2000) applied naive Bayes classifiers, and used a simulated annealing approach in their iterative classification collective inference procedure, where a label for a given node was kept only if the relational classifier was confident about the label at a given threshold, otherwise the label would be set to `null`. By slowly lowering this threshold, the system was eventually able to label all nodes.

Also applying iterative classification for collective inference, Lu and Getoor (2003) investigated network classification applied to linked documents (web pages and published manuscripts with an accompanying citation graph). Similarly to the work of Chakrabarti et al. (1998), Lu and Getoor (2003) use the (local) text of the document as well as neighbor labels. More specifically, their "link-based" relational classifier is a logistic regression model applied to a vector of aggregations of properties of the sets of neighbor labels linked with different types of links (in-links, out-links, co-links). Various aggregates were considered, such as the mode (the value of the most often occurring neighbor class), a binary vector with a value of 1 at cell i if there was a neighbor whose class label was c_i, and a count vector where cell i contained the number of neighbors belonging to class c_i. In their experiments, the count model performed best.

Node-centric methods also have been introduced for simple, univariate within-network classification (Bernstein et al., 2002; Bernstein et al., 2003; Macskassy & Provost, 2003). Macskassy and Provost (2003; 2004) investigated a simple univariate classifier, the weighted-vote relational neighbor (wvRN). Node priors were instantiated simply by the marginal class frequency in the training data. Relational classification was achieved by a simple weighted average of the estimated class membership scores ("probabilities") of the node's neighbors. Collective inference was performed via a relaxation labeling method similar to that used by Chakrabarti et al. (1998). This simple model performed surprisingly well in large-scale empirical studies on a variety of data sets. The authors argue that the network structure and known labels alone often allow this simple, univariate, classifier to be at least a strong baseline and more often than one might think a strong classifier in its own right.

This node-centric framework is presented in detail by Macskassy and Provost (2004) as the basis for NetKit-SRL, a

Network Learning Toolkit for Statistical Relational Learning. NetKit implements the framework with modular mix-and-match components. As discussed above, having such a modular system opens up the design space for network classification by allowing new combinations to be composed simply. Macskassy and Provost (2004) present an in-depth univariate study of various methods for network-only classification.

4.3. Graph Cut Methods and the Relationship to Transductive Inference

One complication to *within-network* classification is that in the same network the to-be-classified nodes are intermixed with nodes for which the labels are known. Much prior work on network learning and classification assumes that all the nodes in the network must be classified, perhaps having learned something from a separate, related network. Pinning the values of certain nodes intuitively should be advantageous, since it gives to the classification procedure clear points of reference.

This complication is addressed directly by several lines of recent work, by Blum and Chawla (2001), by Joachims (2003), by Zhu et al. (2003), and by Blum et al. (2004). In this work, the setting is not initially one of network classification. Rather, these techniques are designed to address semi-supervised learning in a transductive setting (Vapnik, 1998a), but their methods may have direct application to certain instances of univariate network-classification.[2] Specifically, they consider data sets where labels are given for a subset of cases, and classifications are desired for a subset of the rest. They connect the data into a weighted network, by adding edges (in various ways) based on similarity between cases.

In the computer vision literature, Greig et al. (1989) point out that finding the minimum energy configuration of a MRF, the partition of the nodes that maximizes self-consistency under the constraint that the configuration be consistent with the known labels, is equivalent to finding a minimum cut of the graph. Blum and Chawla (2001) follow this idea and subsequent work by Kleinberg and Tardos (1999) connecting the classification problem to the problem of computing minimum cuts. They investigate how to define weighted edges for a transductive classification problem such that polynomial-time mincut algorithms give optimal solutions to objective functions of interest. For example, they show elegantly how forms of leave-one-out-cross-validation error (on the predicted labels) can be minimized for various nearest-neighbor algorithms, including a weighted-voting algorithm. This procedure corresponds to optimizing the consistency of the predictions in particular ways—as Blum and Chawla put it, optimizing the "happiness" of the classification algorithm. Of course, optimizing the consistency of the labeling may not be ideal, for example in the case of highly unbalanced class frequency. Joachims (2003) points out that the need to preprocess the graph to avoid degenerate cuts (e.g, cutting off the one positive example) stems from the basic objective: the minimum of the sum of cut-through edge weights depends directly on the sizes of the cut sets. He proposes to normalize for the cut size, and introduces a solution based on ratiocut optimization constrained by the known labels, following the unconstrained procedure proposed by Dhillon (2001) for unsupervised learning.

Subsequently, Blum et al. (2004) point out that the min-cut solution corresponds to the most probable joint labeling of the graph (taking an MRF perspective), whereas as discussed earlier we often would like a per-node class-probability estimation. Unfortunately, in the case we are considering—when some node labels are known in a general graph—there is no known algorithm for determining these estimates. They also point out several other drawbacks, including that there may be many minimum cuts for a graph (from which mincut algorithms choose rather arbitrarily), and that the min-cut approach does not yield a measure of confidence on the classifications. Blum et al. address these drawbacks by repeatedly adding artificial noise to the edge weights in the induced graph. They then can compute fractional labels for each node corresponding to the frequency of labeling by the various min-cut instances. As mentioned above, this method (and the following) was intended to be applied to an induced graph, which can be designed specifically for the application. Blum et al. point out that min-cut approaches are appropriate for graphs that have at least some small, balanced cuts (whether or not these correspond to the labeled data), and furthermore their method discards highly unbalanced cuts. So, it is not clear whether it would apply to network classification problems such as fraud detection in transaction networks.

In the experiments of Blum et al., their randomized mincut method empirically does not perform as well as the method introduced by Zhu et al. (2003). In the same transductive setting, Zhu et al. treat the induced network as a Gaussian field (a random field with soft node labels) constrained such that the labeled nodes maintain their values. The value of the energy function is the weighted average of the function's value at the neighboring points. The result is a principled, independent development of the weighted-voting relational neighbor classifier (Macskassy & Provost, 2004; Macskassy & Provost, 2003) discussed above.[3] The energy function then can be normalized based on desired class priors ("class mass normalization"). Notably, they also discuss various physical interpretations, including random walks, electric networks, and spectral graph theory. For example, extending the random walk interpretation to a telecommunications network including legitimate and fraudulent accounts, consider starting at an account of interest and walking randomly through the call graph based

[2] All these references induce graphs over initially non-graph data by creating edges between examples.

[3] Experiments show these two procedures to yield almost identical generalization performance, albeit the matrix-based procedure of Zhu et al. is much slower than the iterative wvRN.

on the link weights; the node score is the probability that the walk hits a known fraudulent account before hitting a known legitimate account.

4.4. Using Node Identifiers

As mentioned in the introduction, another unique aspect of within-network classification is that *node identifiers*, symbols for individual nodes, can be used in learning and inference. For example, for suspicion scoring in social networks, the fact that someone met with a particular individual may be quite telling. Very little work has incorporated identifiers, because of the obvious difficulty of modeling with very high cardinality categorical attributes. Fawcett and Provost (1997) and Cortes et al. (2001) describe the use of identifiers (telephone numbers) for fraud detection. To our knowledge, Perlich and Provost (2006) provide the only comprehensive treatment of this topic, which could benefit from further research.

4.5. Beyond Univariate Classification

Several recent methods apply to learning in networked data, beyond the homogeneous, univariate case on which most of this survey focuses. Conditional Random Fields (CRFs) (Lafferty et al., 2001) are random fields where the probability of a node's label is conditioned not only on the labels of neighbors (as in MRFs), but also on the entire observed attribute data $\mathbf{X}$. CRFs were applied to part-of-speech (POS) tagging in text, where the nodes in the graphs represented the words in the sentence, connected by their word order. The labels to be predicted were POS-tags and the attribute of a node was the word it represents. The neighborhood of a word comprised the words on either side of it.

Relational Bayesian Networks (RBNs, a.k.a. Probabilistic Relational Models) (Koller & Pfeffer, 1998; Friedman et al., 1999; Taskar et al., 2001) extend Bayesian networks (BNs (Pearl, 1988)) by taking advantage of the fact that a variable used in one instantiation of a BN may refer to the exact same variable in another BN. For example, the grade of a student depends upon his professor, and this professor is the same for all students in the class. Therefore, rather than building one BN and using it in isolation for each entity, RBNs directly link shared variables, thereby generating one big network of connected entities for which collective inferencing can be performed. For within-network classification RBNs were applied by Taskar et al. (2001) to various domains, including a data set of published manuscripts linked by authors and citations. Loopy belief propagation was used to perform the collective inferencing. The study showed that the PRM performed better than a non-relational naive Bayes classifier and that using both author and citation information in conjunction with the text of the paper worked better than using only author or citation information in conjunction with the text.

Relational Dependency Networks (RDNs) (Neville & Jensen, 2003; Neville & Jensen, 2004), extend dependency networks (Heckerman et al., 2000) in much the same way that RBNs extend Bayesian Networks. RDNs have been used successfully on a bibliometrics data set, a movie data set and a linked web-page data set, where they were shown to perform much better than a relational probability tree (RPT) (Neville et al., 2003a) using no collective inferencing. Gibbs sampling was used to perform collective inferencing. Similarly, Relational Markov Networks (RMNs) (Taskar et al., 2002) extend Markov Networks (Pearl, 1988). The clique potential functions used are based on functional templates, each of which is a (learned, class-conditional) probability function based on a user-specified set of relations. Taskar et al. (2002) applied RMNs to a set of web-pages and showed that they performed better than other non-relational learners as well as naive Bayes and logistic regression when used with the same relations as the RMN. Loopy belief propagation was used to perform collective inferencing.

Associative Markov Networks (AMNs) (Taskar et al., 2004) is another extension of Markov Networks, where auto-correlation of labels is explicitly taken into account. AMNs extend the *generalized Potts model* (Potts, 1952) to allow different labels to have different penalties. Exact inferencing in AMNs is formulated as a quadratic programming problem for binary classification and can be relaxed for multi-class problems.

Neville and Jensen (2005) proposed using latent group models (LGMs) to specify joint models of attributes, links, and *groups*. Shared membership in groups such as communities with shared interests is an important reason for similarity among interconnected nodes. Inference can be more effective if these groups are modeled explicitly, as shown by Neville and Jensen.

These methods for network classification use only a few of the many relational learning techniques. There are many more, for example from the rich literature of inductive logic programming (ILP) (*e.g.* (Flach & Lachiche, 1999; Raedt et al., 2001; Dzeroski & Lavrac, 2001; Kramer et al., 2001; Richardson & Domingos, 2006)), or based on using relational database joins to generate features (e.g. (Perlich & Provost, 2003; Popescul & Ungar, 2003; Perlich & Provost, 2006)).

5. Application: Suspicion Scoring in Social Networks

One problem of current interest for within-network classification is suspicion scoring: ranking people in a social network by their estimated likelihood of being malicious (Macskassy & Provost, 2005a; Macskassy & Provost, 2005b). According to a recent Time magazine cover story (summarized by Tumulty (2006, May 22)), the National Security Agency has aimed to create a huge database of all telephone calls made within the U.S. for the purpose of identifying potential terrorists. Polling shows that the citizenry generally approves. These data provide an absolutely huge, explicit social network. Assigning a variable such as a maliciousness index defines a within-network (univari-

ate[4]) classification problem. The problem is in principle directly analogous to the network approaches to fraud detection described earlier, and is similar to suspicion scoring for other law enforcement applications.

In this section, we describe an application of suspicion scoring in networks of people (entities), linked by communications, meetings, or other associations (e.g., being in the same vicinity at the same time). We demonstrate within-network classification using the weighted-vote relational neighbor classifier, which explicitly assumes that the (social) network exhibits the simple-yet-ubiquitous principle of homophily (Blau, 1977; McPherson et al., 2001)—social science research has shown repeatedly that a person is more likely to associate with people who share similar interests or characteristics. We extend the techniques described above for fraud detection by propagating suspicion through the association network using relaxation labeling (Rosenfeld et al., 1976; Hummel & Zucker, 1983).

Furthermore, we highlight an interesting twist to the problem of network classification. The aforementioned NSA program notwithstanding, it often is costly to acquire knowledge of the links between individuals—especially between malicious individuals. Costs correspond to surveillance, subpoenas to gather data, political fallout, and so on. However, gathering data in a focused manner may improve classification substantially. So, an important problem associated with network classification is: if we have a budget for acquiring data on network linkage, how should we spend it?

5.1. Case Study

Using data from a Department of Defense simulator (described below), we evaluate whether a straightforward network-classification method, the weighted-vote Relational Neighbor algorithm (described above) can rank entities accurately by suspicion: Are the highest-scoring entities predominantly malicious? The case study assesses: (1) the initial rankings, (2) the improvement as more information is acquired, and (3) how sensitive the rankings are to the initial "knowledge" of maliciousness (i.e., how much noise/error can be tolerated).

For collective inferencing wvRN uses relaxation labeling. This requires initial suspicion estimates to bootstrap the inference. We initialize suspicion scores to 1 for initially "known" malicious entities (and freeze them) and 0.01 for the rest. If we had had knowledge of truly benign entities, we would have initialized (and frozen) those scores to 0.

5.2. Data Acquisition

As mentioned above, it is possible to augment the association network at a cost. Specifically, the application setting stipulates that there are secondary data that can be queried at a cost. Queries of the form "give me all associations for entity v" have uniform cost, and return links to other known entities plus, potentially, links to currently unknown associates. The acquisition strategy used for this case study is shown in Table 1; it incrementally acquires additional information—associations and possibly unknown associates—about the currently most-suspicious entities.

1. Acquire information on all entities initially known to be malicious.
2. Generate suspicion scores for all entities with unknown scores.
3. Get information on the top k entities not yet queried ($k = 50$ for this case study).
4. Generate new suspicion scores.
5. Repeat steps 3 and 4 until some stopping criterion is met (in this study: either when all entities in the data set have been queried against or when we have run at least 25 iterations *and* have queried against all entities who are only connected to "known" malicious entities.[a])

[a]This is different from the original study in (Macskassy & Provost, 2005a), where we always stopped after the 25^{th} iteration. See Section 5.3.

Table 1. Data Acquisition algorithm.

5.3. Score-based Evaluation

Notice that the data acquisition methodology outlined in Table 1 keeps querying for more data until all entities that are connected only to "known" malicious entities have been queried against. This is due to a scoring anomaly that was identified during an evaluation of an earlier methodology (Macskassy & Provost, 2005a), which was rectified in a later study (Macskassy & Provost, 2005b). We refer the reader to this later paper for further details.

In the original study, the system is evaluated using two metrics. The first is the Area Under the ROC Curve (AUC), which is equivalent to the Mann-Whitney-Wilcoxon statistic. The AUC computes the probability that a randomly selected malicious entity would be given a higher suspicion score than a randomly selected benign entity.

The second metric, which we use in this study, is the fraction of the top-100 highest-scoring entities that truly are malicious. This evaluates the system under the assumption that an intelligence analyst will consider only the highest scoring entities for possible further investigation. For the analyst's processing capacity we chose 100 cases fairly arbitrarily (based on brief discussions with domain experts). For this evaluation, we disregard scores that are maximal-by-construction; specifically, we remove from consideration entities for which the secondary data have not yet been queried.

5.4. Data

There are many varieties of intelligence data—no single comparison of classified and synthetic data will be com-

[4]The data provided by the telephone companies reportedly are only the call detail records, not any information on the communicating parties.

	World Parameters		Primary Data			
Data set	Size	Number malicious	Size	True malicious	False malicious	Noise
5057	1011	274	497	101	0	0.000
5069	9907	2893	7374	520	0	0.000
5058	100301	7350	40316	886	189	0.176
5046	13236	1484	4212	143	52	0.267
5056	1022	300	510	99	218	0.688
5050	1008	316	692	103	336	0.765

Table 2. Characteristics of synthetic data sets, sorted and "grouped" by noise. In the World Parameters, Size refers to the number of entities in the true synthetic world and Number malicious refers to the total number of truly malicious entities. In the Primary Data, Size refers to the number of entities known initially; True malicious refers the number of entities tagged as "malicious" who truly were malicious in the world and False malicious refers to the number of entities falsely tagged as "malicious". The error rate (Noise) of these labelings ranges from none (0) to very high as seen in the bottom group of data sets. Note that step 1 in Table 1 above must query the secondary database for information on all "false malicious" as well as all "true malicious" entities.

prehensive. The data we use for this case study were generated by a flexible simulator as part of a DoD program to assess the feasibility of large-scale information systems to help identify terrorists. The synthetic data generated by this simulator are moderately sized examples of structured data representing terrorists and benign entities who are conducting activities over an extended period of time. The data are contained wholly in a single data source (although costly secondary access is simulated) and are self-consistent, neither of which is reliably true of classified data. However, the data do replicate a range of noisy and poorly observed activities, and the entities are intentionally obscured to simulate lack of knowledge, obfuscation, poor data-entry practices, multiple identities, etc. Nonetheless, we do not claim that the data fully replicate the limitations of actual data collection, aggregation and enrichment that the intelligence community routinely experiences. The data sets we use were generated for the purposes of DoD program evaluation. We did not create data sets ourselves for this study.

One run of the simulator generates three databases:

1. "primary" data that are known ex ante. These often are sparse and may contain partial (or no) information on any particular entity or group;

2. "secondary" data consisting of information that only can be acquired by querying (theoretically at a cost) to get information on a particular entity;

3. "truth" data, for evaluation, consisting of what really happened in the world.

The first two databases together reflect what possibly can be observed. They are potentially corrupt and contain only a subset of the complete truth. Further, the data never give hard evidence that an entity is benign, and therefore we only "know" about some malicious entities—those who are known to belong to one or more terrorist groups. Sometimes this knowledge is wrong. We evaluate the suspicion scoring on 6 data sets whose characteristics are shown in Table 2.[5]

[5]We evaluated 17 data sets in the original study. These 6 are a representative sample of the results reported there.

5.5. Results

In order to address how noise affects performance, we group the data sets into three categories: no noise (5057,5062), low to moderate noise (5046, 5058), and very high noise (5050,5056).

For an analyst, knowing that the system can achieve an AUC of 0.9 does not necessarily mean that the system is useful. Although the system, in general, will rank suspicious entities higher, when considering 10000 entities, the top 100 could potentially be primarily benign with the next 900 being primarily malicious. Albeit unusual, this would achieve a relatively high AUC, but not be very useful for analysts who only have time to look at a select few entities.

Usually for rankings such as suspicion scorings, the density of entities of interest is highest at the very top of the list, especially if the scores are estimated probabilities of membership in a class (e.g., malicious entity), and so the top of the list contains the entities with the highest estimated probabilities of being suspicious. For a given AUC value, how dense one expects the top of a ranking to be depends primarily on the marginal probability of entities of interest in the data (this and related issues are treated in detail elsewhere (Provost & Fawcett, 2001)). An AUC of 0.9 may have 99 truly malicious entities in the top 100 highest-suspicion entities, or it may have 10. In either case, the system may be useful to an analyst, depending on the application and how the ranking will be used (e.g., as a primary basis for action versus as an alternative source of evidence to augment existing practices[6]).

To illustrate the effectiveness of the scorings for these data sets for a particular threshold, we analyze the number of truly malicious entities at the top of the suspicion rankings. If we look across the 17 data sets, we can ask how many truly malicious entities are among the top 100 most suspicious.

Table 3 shows the results for the 6 data sets, grouped by their noise level. The evaluation starts at Iteration 3 because

[6]For example, White and Fournelle (2005) show that an early version of our suspicion scores are an effective prefilter for their CADRE analysis system for link discovery.

		Iteration						
	Data	3	4	5	10	15	20	Last
No	5057	19	21	63	100	100	100	100
Noise	5069	25	60	95	100	100	100	100
Moderate	5046	8	22	43	87	98	100	100
Noise	5058	6	29	51	97	96	94	91
High	5050	37	33	37	10	12	-	12
Noise	5056	28	27	42	16	28	-	32

Table 3. How many truly malicious entities were in the top 100 after iterations 3,4,5,10,15,20 and after the last iteration. We start at iteration 3 because that is the first iteration in which we have scores for queried entities that were not initially "known". Iteration 3 therefore only contains 50 entities. Further, all high noise data sets were so small that they stopped at iteration 16.

Iterations 1 and 2 do not yet contain scores on entities that have been queried and are not part of the "known" entities. Iteration 3 is therefore based only on the 50 entities which have been queried after the initial querying of all "known" entities. Iteration 4 contains 100 queried entities and so on.

The table shows quantitatively that in the no-noise group, we get very high density already at iteration 5, and perfection at iteration 10. We see a similar behavior for the moderate data sets, where iteration 10 already has very good accuracy for the of the data sets, and nearly perfect accuracy on both data sets by the last iteration. Finally, we see in the very large noise group that by the final iteration the system has very few malicious entities in the top 100. This may be expected for relational-neighbor inference, since more than half of the "known" malicious entities actually are benign.

5.6. Limitations

The approach we described here has notable limitations. We have assumed substantial prior knowledge: of entities, of links between them, of maliciousness. We have shown some robustness to the knowledge of maliciousness, but have not systematically explored robustness along other dimensions. Moreover, collecting the data to build such a network is a considerable effort, and it would make sense to consider network construction in tandem with the system that would make use of the network.

Relatedly, we have considered network-based suspicion scoring in isolation. In reality, network-based scoring would be one source of evidence, combined for example with "profile"-based scoring. We conducted a preliminary investigation into augmenting the scoring by setting initial priors based on uncertain-but-better-than-random knowledge (as from a profiling system). We found that priors had little-to-no effect due to the algorithm's dominance by the scores propagated from the static labels (Macskassy & Provost, 2005a). This is a problem which can affect many collective inference techniques. In retrospect, it appears necessary to integrate closely the use of profiling information with the network scoring (cf. (Macskassy & Provost, 2004)). This issue is likely to affect many collective inference techniques and needs to receive more attention.

5.7. Final Remarks

We described and evaluated a guilt-by-association system for generating suspicion scores based on entities' known associates. The system is notable for several reasons. First, it is able to generate remarkably good rankings even when very few entities are known to be malicious. Second, it can be relatively robust even to moderate noise in these prior labels. Third, it works remarkably well considering that it only uses prior labels and the network, but no profiling. Finally, it can be used as a data gathering tool not only to perform focused data acquisition of suspicious entities, but also to further improve its ranking—and in the process it often learns about suspicious entities that were not initially in the database.

6. Toolkits and benchmark data sets

6.1. Toolkits

We are aware of three publically available relational learning systems, all of which can be used for classification of networked data.

Alchemy[7] is a software package providing a series of algorithms for statistical relational learning and probabilistic logic inference, based on the Markov logic representation (Richardson & Domingos, 2006).

NetKit-SRL[8] is a toolkit especially designed for network learning (Macskassy & Provost, 2004). It is fully integrated with the Weka machine learning toolkit,[9] and has a plug-and-play architecture allowing a user to easily select any non-relational classifier, relational classifier and inference method for network classification.

Proximity[10] is an open-source system for relational knowledge discovery including methods such as the Relational Bayes Classifier (Neville et al., 2003b), Relational Probability Trees (Neville & Jensen, 2003) and Relational Dependency Networks (Neville & Jensen, 2004).

6.2. Benchmark data sets

Three publically available benchmark networked data sets have often been used for machine learning research

The *WebKB* data set (Craven et al., 1998)[11] consists of sets of web pages from four computer science departments, with each page manually labeled into 7 categories: course, department, faculty, project, staff, student or other.

The *CoRA*[12] data set (McCallum et al., 2000) comprises computer science research papers. It includes the full citation graph as well as labels for the topic of each paper (and potentially sub- and sub-sub-topics).

[7] http://www.cs.washington.edu/ai/alchemy/

[8] http://www.research.rutgers.edu/~sofmac/NetKit.html

[9] http://www.cs.waikato.ac.nz/~ml/weka

[10] http://kdl.cs.umass.edu/proximity/

[11] The WebKB-ILP-98 data set is the one commonly used. It is available at http://www.cs.cmu.edu/~webkb/

[12] Available at http://www.cs.umass.edu/~mccallum/code-data.ht

Networked data from the *Internet Movie Database* (IMDb)[13] have been used to build models predicting movie success based on box-office receipts (Jensen & Neville, 2002a). This problem was also addressed by Macskassy and Provost (2004).

References

Bernstein, A., Clearwater, S., Hill, S., Perlich, C., & Provost, F. (2002). Discovering knowledge from relational data extracted from business news. *Proceedings of the Multi-Relational Data Mining Workshop (MRDM) at the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Bernstein, A., Clearwater, S., & Provost, F. (2003). The relational vector-space model and industry classification. *Proceedings of the Learning Statistical Models from Relational Data Workshop (SRL) at the 19th International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 8–18).

Besag, J. (1974). Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society*, *36*, 192–236.

Besag, J. (1986). On the statistical analysis of dirty pictures. *Journal of the Royal Statistical Society*, *48*, 259–302.

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

Blum, A., & Chawla, S. (2001). Learning from labeled and unlabeled data using graph mincuts. *Proceedings of the International Conference on Machine Learning (ICML)* (pp. 19–26).

Blum, A., Lafferty, J., Reddy, R., & Rwebangira, M. R. (2004). Semi-supervised learning using randomized mincuts. *Proceedings of the 21st International Conference on Machine Learning (ICML)*. Banff, Canada.

Boykov, Y., Veksler, O., & Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intellingence (PAMI)*, *23*.

Chakrabarti, S., Dom, B., & Indyk, P. (1998). Enhanced hypertext categorization using hyperlinks. *ACM SIGMOD International Conference on Management of Data*.

Cortes, C., Pregibon, D., & Volinsky, C. T. (2001). Communities of interest. *Proceedings of Intelligent Data Analysis (IDA)*.

Cowell, R. G., Dawid, A. P., Lauritzen, S. L., & Spiegelhalter, D. J. (1999). *Probabilistic Networks and Expert Systems*. Springer.

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*.

Dhillon, I. S. (2001). Co-clustering documents and words using bipartite spectral graph partitioning. *Knowledge Discovery and Data Mining* (pp. 269–274).

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.

Domingos, P., & Richardson, M. (2001). Mining the network value of customers. *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 57–66). CA: ACM Press.

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

Fawcett, T., & Provost, F. (1997). Adaptive fraud detection. *Data Mining and Knowledge Discovery*, *3*, 291–316.

Flach, P. A., & Lachiche, N. (1999). 1BC: A first-order Bayesian classifier. *Proceedings of the Ninth International Workshop on Inductive Logic Programming (ILP)* (pp. 92–103). Springer-Verlag.

Friedman, N., Getoor, L., Koller, D., & Pfeffer, A. (1999). Learning probabilistic relational models. *Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*. Stockholm, Sweden.

Galstyan, A., & Cohen, P. (2005). Is guilt by association a bad thing? *Proceedings of the Intelligence Analysis Conference (IA)*.

Geman, S., & Geman, D. (1984). Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intellingence (PAMI)*, *6*, 721–741.

Greig, D., Porteous, B., & Seheult, A. (1989). Exact maximum a posteriori estimation for binary images. *J. Royal Statistical Soc.*, *51*, 271–279.

Heckerman, D., Chickering, D. M., Meek, C., Rounthwaite, R., & Kadie, C. (2000). Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research (JMLR)*, *1*, 49–75.

Hill, S., Provost, F., & Volinsky, C. (2006). Viral marketing: Identifying likely adopters via consumer networks. *Statistical Science*. (to appear).

Huang, Z., Chen, H., & Zeng, D. (2004). Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems (TOIS)*, *22*.

Hummel, R. A., & Zucker, S. W. (1983). On the foundations of relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *5*, 267–287.

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. *Proceedings of the 19th International Conference on Machine Learning (ICML)*.

Jensen, D., Neville, J., & Gallagher, B. (2004). Why collective inference improves relational classification. *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Joachims, T. (2003). Transductive learning via spectral graph partitioning. *Proceedings of the International Conference on Machine Learning (ICML)*.

Kleinberg, J., & Tardos, E. (1999). Approximation algorithms for classification problems with pairwise relations: metric labeling and Markov random fields. *IEEE Symposium on Foundations of Computer Science* (pp. 14–23).

Koller, D., & Pfeffer, A. (1998). Probabilistic frame-based systems. *AAAI/IAAI* (pp. 580–587).

Kramer, S., Lavrac, N., & Flach, P. (2001). Propositionalization approaches to relational data mining. In S. Dzeroski and N. Lavrac (Eds.), *Relational data mining*, 262–291. Springer-Verlag.

Lafferty, J., McCallum, A., & Pereira, F. (2001). Conditional random fields: probabilistic models for segmenting and labeling sequence data. *Proceedings of the 18th International Conference on Machine Learning (ICML)*.

Lu, Q., & Getoor, L. (2003). Link-based classification. *Proceedings of the 20th International Conference on Machine Learning (ICML)*.

Macskassy, S. A., & Provost, F. (2003). A simple relational classifier. *Proceedings of the Multi-Relational Data Mining Workshop (MRDM) at the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

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

Macskassy, S. A., & Provost, F. (2004). *Classification in networked data: A toolkit and a univariate case study*. Technical Report CeDER Working Paper 04-08, Stern School of Business, New York University, 2004.

Stern School of Business, New York University.

Macskassy, S. A., & Provost, F. (2005a). Suspicion scoring based on guilt-by-association, collective inference, and focused data access. *International Conference on Intelligence Analysis*.

Macskassy, S. A., & Provost, F. (2005b). Suspicion scoring of entities based on guilt-by-association, collective inference, and focused data access. *Annual Conference of the North American Association for Computational Social and Organizational Science (NAACSOS)*.

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.

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. (2003). Collective classification with relational dependency networks. *Proceedings of the Second Workshop on Multi-Relational Data Mining (MRDM-2003) at KDD-2003*.

Neville, J., & Jensen, D. (2004). Dependency networks for relational data. *Proceedings of the Fourth IEEE International Conference in Data Mining (ICDM)*.

Neville, J., & Jensen, D. (2005). Leveraging relational autocorrelation with latent group models. *Proceedings of the IEEE International Conference on Data Mining (ICDM)*.

Neville, J., Jensen, D., Friedland, L., & Hay, M. (2003a). Learning relational probability trees. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Neville, J., Jensen, D., & Gallagher, B. (2003b). Simple estimators for relational Bayesian classifiers. *Proceedings of the Third IEEE International Conference on Data Mining (ICDM-2003)*.

Neville, J., Özgür Şimşek, Jensen, D., Komoroske, J., Palmer, K., & Goldberg, H. (2005). Using relational knowledge discovery to prevent securities fraud. *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Pearl, J. (1988). *Probabilistic reasoning in intelligent systems*. Morgan Kaufmann.

Perlich, C., & Provost, F. (2003). Aggregation-based feature invention and relational concept classes. *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Perlich, C., & Provost, F. (2006). Distribution-based aggregation for relational learning with identifier attributes. *Machine Learning*, *62*, 65–105.

Popescul, A., & Ungar, L. H. (2003). Statistical relational learning for link prediction. *Proceedings of the Learning Statistical Models from Relational Data Workshop (SRL) at the 19th International Joint Conference on Artificial Intelligence (IJCAI)*.

Potts, R. B. (1952). Some generalized order-disorder transformations. *Proceedings of the Cambridge Philosophic Society*, *48*, 106–109.

Provost, F., & Fawcett, T. (2001). Robust classification for imprecise environments. *Machine Learning*, *42*, 203–231.

Raedt, L. D., Blockeel, H., Dehaspe, L., & Laer, W. V. (2001). Three companions for data mining in first order logic. In S. Dzeroski and N. Lavrac (Eds.), *Relational data mining*, 105–139. Springer-Verlag.

Richardson, M., & Domingos, P. (2006). Markov logic networks. *Machine Learning*, *62*, 107–136.

Rosenfeld, A., Hummel, R., & Zucker, S. (1976). Scene labeling by relaxation operations. *IEEE Transactions on Systems, Man and Cybernetics*, *6*, 420–433.

Savage, L. J. (1954). *The foundations of statistics*. John Wiley and Sons.

Segal, E., Wang, H., & Koller, D. (2003a). Discovering molecular pathways from protein interaction and gene expression data. *Bioinformatics*, *19*, I264–I272.

Segal, E., Yelensky, R., & Koller, D. (2003b). Genome-wide discovery of transcriptional modules from DNA sequence and gene expression. *Bioinformatics*, *19*, I273–I282.

Taskar, B., Abbeel, P., & Koller, D. (2002). Discriminative probabilistic models for relational data. *Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI)*. Edmonton, Canada.

Taskar, B., Chatalbashev, V., & Koller, D. (2004). Learning associative Markov networks. *Proceedings of the 21st International Conference on Machine Learning (ICML)*. Banff, Canada.

Taskar, B., Segal, E., & Koller, D. (2001). Probabilistic classification and clustering in relational data. *Proceedings of the 17th International Joint Conference on Artificial Intelligence (IJCAI)* (pp. 870–878).

Tumulty, K. (2006, May 22). Inside Bush's secret spy net [electronic version]. *Time*, *167*. Retrieved May 25, 2006, from `http://www.time.com/time/archive/preview/0,10987,1194021,00.html`.

Vapnik, V. N. (1998a). *Statistical learning theory*. John Wiley, NY.

Vapnik, V. N. (1998b). The support vector method of function estimation. In J. Suykens and J. Vandewalle (Eds.), *Nonlinear Modeling: Advanced Black-Box techniques*, 55–86. Kluwer, Boston.

Wainwright, M. J., & Jordan, M. I. (2003). *Graphical models, exponential families, and variational inference*. Technical Report 649, University of California, Berkeley, 2003.

White, J. V., & Fournelle, C. G. (2005). Threat detection for improved link discovery. *International Conference on Intelligence Analysis*.

Winkler, G. (2003). *Image analysis, random fields and Markov chain Monte Carlo methods*. Springer-Verlag. 2nd edition.

Zhu, X., Ghahramani, Z., & Lafferty, J. (2003). Semi-supervised learning using Gaussian fields and harmonic functions. *Proceedings of the 12th International Conference on Machine Learning*. Washingon DC, United States.