Appears in *the North American Association for Computational Social and Organizational Science (NAACSOS) Conference*, 2005

NetKit-SRL: A Toolkit for Network Learning and Inference

and its use for classification of networked data

Sofus A. Macskassy
New York University
44 W. 4th Street
New York, NY 10012
smacskas@stern.nyu.edu

Foster Provost
New York University
44 W. 4th Street
New York, NY 10012
fprovost@stern.nyu.edu

Abstract

This paper describes NetKit-SRL, or NetKit for short, a toolkit for learning from and classifying networked data. The toolkit is open-source and publicly available. It is modular and built for ease of plug-and-play—such that it is easy to add new modules and have them interact with other existing modules. Currently available NetKit modules are focused on "batch" within-network learning and classification: given a partially labeled network, where all nodes and edges are already known to exist, estimate the class membership probability of the unlabeled nodes in the network. NetKit has been used in various network domains such as websites, citation graphs, movies and social networks.

Contact:
Sofus A. Macskassy
Stern School of Business
Department of Information, Operations & Management Sciences
New York University
New York, NY 10012-1126, USA

Tel: 1-212-998-0584
Fax: 1-212-995-4228
Email: smacskas@stern.nyu.edu

Keywords: relational learning, network learning, machine learning collective inference, guilt-by-association

Acknowledgments

This research was sponsored in part by the Air Force Research Laboratory, Air Force Materiel Command, USAF, under Agreement number F30602-01-2-0585. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of AFRL or the U.S. Government. This work was also funded in part by a grant from the New York Software Industry Association.

NetKit-SRL: A Toolkit for Network Learning and Inference

Sofus A. Macskassy and Foster Provost

1 Introduction

This paper[1] describes NetKit, a network learning toolkit, and its use for classification of entities in *networked* data, one type of relational data. NetKit enables studies of techniques for statistical relational learning and inference with networked data (Macskassy & Provost, 2004).

Relational classifier induction algorithms, and associated inference procedures, have been developed in a variety of different research fields and problem settings (cf. (Dzeroski & Lavrac, 2001)). 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 (e.g. (Taskar et al., 2001; Jensen et al., 2004)).

Networked data are the special case of relational data where entities are interconnected, such as social networks, webpages or research papers (connected through citations). With a few exceptions (e.g., (Chakrabarti et al., 1998), (Taskar et al., 2001)), recent machine learning research on classification with networked data has focused on *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).

Here we focus on *within-network* inference. In this case, networked data have the unique characteristic that training entities and entities whose labels are to be estimated are interconnected. This setup has a property that is not normally found, or used, in network learning: entities with known classifications can serve two roles. They act first as training data and subsequently as background knowledge during inference (Provost et al., 2003).

Many real-world problems, especially those involving social networks, exhibit opportunities for within-network classification. For example, in fraud detection entities to be classified as being fraudulent or legitimate are intertwined with those for which classifications are known. In counter-terrorism and law enforcement, suspicious people may interact with known 'bad' people. Some networked data are by-products of social networks, rather than directly representing the networks themselves. For example, networks of webpages are built by people and organizations that are interconnected; when classifying webpages, some classifications (henceforth, *labels*) may be known and some may need to be estimated.

NetKit is based on a general network learning framework. Starting with prior published work (Chakrabarti et al., 1998; Macskassy & Provost, 2003; Lu & Getoor, 2003), we have abstracted the described algorithms and methodologies into a modular framework consisting of three main modules. NetKit is an instantiation of this framework. It is written in Java 1.5 and is available as open source. NetKit is important for several reasons. It encompasses several previously published but unavailable network classification/learning systems, which are realized by choosing particular instantiations for the different components. More importantly, NetKit will facilitate research on classification and learning in networked data by providing publicly available algorithms implemented on a common platform, much as toolkits like MLC++ (Kohavi et al., 1994) and Weka (Witten & Frank, 2000) have made machine learning algorithms for feature-vector data readily available.

The remainder of the paper is outlined as follows: Section 2 first describes our setup for network learning, including the learning framework, prior work, and the toolkit. Section 3 describes one specific case study, followed in Section 4 by discussion and limitations. We finish in Section 5 with concluding remarks.

2 Network Learning

2.1 Network Learning Framework

The general network learning framework is based on three components, a non-relational model, a relational model, and a collective inference component. These were derived from the abstraction of prior work in network learning. Each of the three components has many possible instantiations.

2.1.1 Non-relational model

The **Non-relational ("local") model** consists of a (learned) model, which uses only local information—namely information about (attributes of) the entities whose target variable is to be estimated. The local models can be used to generate priors that comprise the initial state for the relational learning and collective inference components. They also

[1] This paper summarizes prior work (Macskassy & Provost, 2004).

can be used as one source of evidence during collective inference. These models typically are produced by traditional machine learning methods.

2.1.2 Relational model

In contrast to the non-relational component, the relational model makes use of the relations in the network as well as the values of attributes of related entities, possibly through long chains of relations. Relational models also may use local attributes of the entities.

2.1.3 Collective inferencing

In network classification one common goal is to maximize $P(\mathbf{x}|G^K)$, where $\mathbf{x}$ are the labels to be estimated and G^K is everything that is known in the network.[2] When entities are interconnected, the labels may not be independent, and hence we may need to estimate all labels simultaneously or "collectively." The collective inferencing component determines how the unknown values are estimated together, possibly influencing each other.

2.2 Prior Work

For machine learning research on networked data, the watershed paper of Chakrabarti et al. (1998) studied classifying webpages based on the text and (possibly inferred) class labels of neighboring pages, using relaxation labeling paired with naive Bayes local and relational classifiers. In their experiments, using the link structure substantially improved classification over using the local (text) information alone. Further, considering the text of the neighbors generally hurt performance (based on the methods they used), whereas using only the (inferred) class labels improved performance. More recently, Lu & Getoor (2003) investigated network classification applied to linked documents (webpages and published manuscripts with an accompanying citation graph). They used the text of the document as well as a relational classifier.

Univariate within-network classification has been considered previously (Bernstein et al., 2002; Macskassy & Provost, 2003). Using business news, Bernstein et al. (2003) linked companies if they co-occurred in a news story. They demonstrated the effectiveness of various vector-space techniques for network classification of companies into industry sectors. Other domains such as webpages, movies and citation graphs have also been considered for univariate within-network classification; Macskassy & Provost (2003) investigated how well the univariate classification performed as varying amounts of data initially were labeled.

Markov Random Fields (MRFs) have been used extensively for univariate network classification for vision and image restoration. Nodes in the network are pixels in an image and the labels are image-related such as whether a pixel is part of a vertical or horizontal border (Geman & Geman, 1984; Winkler, 2003). One popular method to compute the MRF joint probability is Gibbs sampling (Geman & Geman, 1984). The most common use of Gibbs sampling in vision is not to compute the final posteriors as we do in NetKit, but rather to get final classifications. Graph-cut techniques recently have been used in vision research as an alternative to using Gibbs sampling (Boykov et al., 2001), iteratively changing the labelings of many nodes at once by solving a min-cut/max-flow problem based on the current labelings.

Several recent methods apply to learning in networked data, beyond the homogeneous, univariate case treated in this paper. Conditional Random Fields (CRFs) (Lafferty et al., 2001) are an extension of MRFs where labels are conditioned not only on the labels of neighbors, but also on the attributes of the node itself and the attributes of the neighborhood nodes. There has been a considerable amount of work studying Probabilistic Relational Models, such as Relational Bayesian Networks (RBNs)[3] (Koller & Pfeffer, 1998; Taskar et al., 2001), Relational Dependency Networks (RDNs) (Neville & Jensen, 2004), and Relational Markov Networks (RMNs) (Taskar et al., 2002).

The above systems use only a few of the many relational learning techniques proposed in the literature. There are many more, for example from the rich literature of inductive logic programming (ILP) (*e.g.* (Flach & Lachiche, 1999; Dzeroski & Lavrac, 2001; Kramer et al., 2001; Domingos & Richardson, 2004)), or based on using relational database joins to generate relational features (e.g. (Perlich & Provost, 2003; Popescul & Ungar, 2003)). These techniques could be the basis for additional relational model components in NetKit.

2.3 Instantiating prior work

Certain network classification procedures from prior work can be instantiated with particular choices of the components in this framework. For example, using a naive Bayes classifier as the local model, a naive Bayes Markov Random

---

[2] Alternative goals include estimating the joint distribution over these labels, or estimating the marginal posterior distributions for the labels of particular nodes.

[3] These originally were called Probabilistic Relational Models (PRMs). PRM now typically is used as a more general term which includes other models such as Relational Dependency Networks and Relational Markov Networks.

Input: G^K, V^U, $\text{RC}_{\text{type}}, \text{LC}_{\text{type}}, \text{CI}_{\text{type}}$
Induce a local classification model, LC, of type LC_{type}, using G^K Induce a relational classification model, RC, of type RC_{type}, using G^K Initially, estimate $x \in V^U$ using LC. Apply collective inferencing of type CI_{type}, using RC as the relational model and LC as the local model.
Output: Final estimates for $x_i \in V^U$

Table 1: High-level pseudo code for the main core of the Network Learning Toolkit. G^K is the network and everything that is known about it. V^U are the vertices in the graph whose labels (x_i) are to be estimated.

Field classifier for the relational model, and relaxation labeling for the inferencing method forms the system used by Chakrabarti et al. (1998). Using logistic regression for the local and relational models, and iterative classification for the inferencing method produces Lu & Getoor's (2003) link-based classifier. Using class priors for the local model, a (weighted) majority vote of neighboring classes for the relational model, and relaxation labeling for the inference method forms Macskassy & Provost's (2003) relational neighbor classifier. NetKit is able to instantiate all of these learning systems, and more.

2.4 Network Learning Toolkit (NetKit)

NetKit is designed to accommodate the interchange of components and the introduction of new components. Any local model can be paired with any relational model, which can then be combined with any collective inference method. NetKit's core routine is simple and is outlined in Table 1.

The current version of NetKit, while able to read in heterogeneous graphs, only supports classification in graphs consisting of homogeneous nodes. Following the framework outlined above, NetKit has three main modules that each can be instantiated into one of several possible methods. These are outlined below.

2.4.1 Local classifier

This module returns a model which uses only attributes of a node to estimate its class label.
Currently available instantiations: NetKit has two native local 'dummy' models: uniform-prior and class-prior. It also has a Weka-wrapper, allowing the use of any Weka classifier (Witten & Frank, 2000).[4] Lastly, it allows reading in priors from a file, allowing the use of any external classifier which produces estimates of class membership probability.

2.4.2 Relational classifier

This module returns a model which uses not only the local attributes of a node but also attributes of related nodes, including their (estimated) class membership.
Currently available instantiations: NetKit has native univariate models: the weighted-vote relational neighbor (wvRN), the class-distributional relational neighbor (cdRN) and a network-only multinomial Bayes classifier with Markov Random Field estimation (cf. (Chakrabarti et al., 1998)). It also has a Weka-wrapper, allowing the use of any Weka classifier. With the addition of various aggregations of neighbor-attributes, the Weka classifiers function as relational classifiers. For example, aggregating neighbor class values into feature vectors and using logistic regression instantiates Lu and Getoor's (2003) Link-based Classifier.

2.4.3 Collective inferencing

This module applies collective inferencing in order to (approximately) maximize the joint probability of the labels of all nodes in the graph whose labels were initially unknown.
Currently available instantiations: Relaxation labeling (Chakrabarti et al., 1998), Iterative Classification (Lu & Getoor, 2003), Gibbs Sampling (Geman & Geman, 1984).

3 Univariate Case Study

NetKit has been used in a variety of domains. We here describe one case study where NetKit was used to conduct an in-depth case study of *within-network* classification based on 12 machine learning benchmark data sets from 4 domains (Macskassy & Provost, 2004).[5] The case study focused on the simple but important special case of univariate network classification, for which the only information available is the structure of class linkage in the network (i.e., only links and some class labels are available).

[4] We use version 3.4.2. Weka is available at `http://www.cs.waikato.ac.nz/~ml/weka/`

[5] A different case study involving social networks and counter-terrorism is presented at this conference (Macskassy & Provost, 2005).

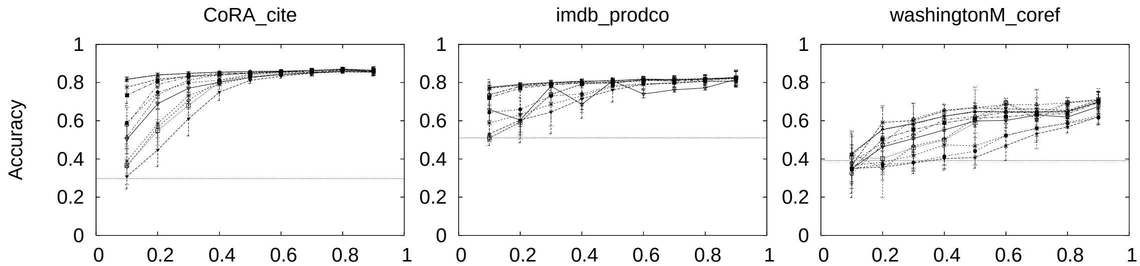

Figure 1: Overall classification accuracies for the 12 network classifiers for 3 selected data sets. Horizontal lines represent predicting the most prevalent class. The horizontal axis plots the fraction of a network's nodes for which the class label is known ex ante. When few labels are known (left end) there is a large variation in performance. Data sets are tagged based on the edge-type used, where 'prodco' is short for 'production company', and 'M' in the WebKB data sets represents the 'multi-class' classification.

3.1 Data sets used

The study made use of 12 data sets from 4 domains.

Movies [1 data set]: We used networked data from the Internet Movie Database (IMDb)[6] with the goal of estimating whether the opening weekend box-office receipts "will" exceed \$2 million (Neville et al., 2003). The data set contains 1169 movies, 572 of which are high-revenue. Movies were linked if they shared a production company (see Section 3.3 for a discussion on how to pick edges).

Citation graph [1 data set]: The CoRA data set (McCallum et al., 2000) comprises computer science research papers. We focused on 3583 machine learning papers with the classification task of predicting a paper's sub-topic (of which there are seven).

Websites [8 data sets]: The WebKB Project (Craven et al., 1998) data consists of sets of webpages from four computer science departments, with each page manually labeled into 7 categories.[7] As with other work, we link pages through co-references rather than the direct links (Neville et al., 2003; Lu & Getoor, 2003). We treat each computer science website separately, and consider two classification problems: the multi-class problem and a binary student/not-student problem. The data set sizes ranged from 346 to 434 pages.

Industry Classification [2 data sets]: This domain involves classifying public companies by industry sector. Companies are linked via co-occurrence in text documents. We use two different data sets, representing different sources. The first data set consists of 22,170 business news stories collected from Yahoo! between 4/1/1999 and 8/4/1999 (Fawcett & Provost, 1999). The second data set consists of 35,318 prnewswire press releases gathered from April 1, 2003 through September 30, 2003 (Bernstein et al., 2003).

3.2 Results

We ran NetKit on each data set using 12 possible pair-wise combinations of 4 relational classifiers and 3 collective inferencing methods (the local classifiers were not appropriate to use here because the study only contained class labels).

The case study was three-pronged—(1) analyze whether the network itself (when partially labeled) carried enough information to yield good predictive performance, (2) study sensitivity to the amount of initially labeled data, and (3) perform a component-analysis to see which components (by themselves, and in combination) performed well. We also investigated how to pick "good" types of edges which would yield good classification performance—the results are presented in Section 3.3.

Figure 1 shows the the performance of the 12 learning systems on 3 of the 12 data sets. These results are explained fully elsewhere (Macskassy & Provost, 2004). As we can see, the methods performed better than predicting the majority class when 90% of the network was labeled—this was true across all data sets; more interestingly, we found that a simple model (wvRN) based on the principle of homophily (Blau, 1977; McPherson et al., 2001) (cf., assortativity (Newman, 2003)) paired with relaxation labeling (RL) performed exceedingly well across all domains.[8] In fact, as we decreased the amount of initially labeled nodes from 90% down to 10%, this component pair was significantly better than the others. Further, when compared to results reported in prior work using more complex methods, we see that this pairwise combination performs comparably to these more complex models.

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

[7] We use the WebKB-ILP-98 data.

[8] This combination is similar to Hopfield Networks (Hopfield, 1982) and Boltzman Machines (Ackley et al., 1985).

These results demonstrate clearly that simple network-classification models perform remarkably well—well enough that they should be used regularly as baseline classifiers for studies of relational learning for networked data.

3.3 Selecting Edges

Creating a graph with a single type of edge from a problem where various possible links exist is a representation engineering problem reminiscent of the selection of a small set of useful features for traditional classification.[9] We investigated the effect of the choice of edges on the overall performance of the wvRN-RL combination on the three domains where multiple types of edges were possible:

1. **cora**: Link entities either through citations, authors or both.
2. **imdb**: We considered four types of edges as suggested by David Jensen: actors, directors, producers and production companies. We also considered combining them all into one type of homogeneous edge.
3. **WebKB**: We initially used co-references based on prior work. We could also use direct hyperlinks, or both.

We tested three heuristic methods to select edges. The first two methods are based on assortativity (Newman, 2003): (1) selecting the edge-type with the highest (edge-based) assortativity, or (2) select the edge-type with the highest node-based assortativity (Macskassy & Provost, 2004). The third heuristic method was a leave-one-out estimation, where we would choose the edge-type with the best performance. All of these methods used only initially known labels. The node-based assortativity was shown to be the best predictor to identify the edge-type that would result in the best performance.

4 Discussion and Limitations

The study showed the use of NetKit on data sets where network-only methods (i.e., methods not using any attributes other than the class label) were able to perform very well. Pushing the idea of the "power of the network", we should potentially also look at identifying the individual nodes and using those for classification and learning—for example, being linked to Mohammed Atta may be informative (Perlich & Provost, 2004).

Another noteworthy limitation of the work thus far is that we have ignored the inherent complexity of real networked data, such as heterogeneity of nodes and edges, directed edges, as well as attributes of the nodes. Each of these introduces complexities but also opens up opportunities for modeling. For example, how much information is there in the attributes of nodes versus just the link structure? Are they complementary, and how do we combine them such that one does not inappropriately dominate the other?

One important practical limitation is that in this work we randomly chose the nodes to be labeled. It is likely that the data for which labels are available are interdependent—e.g., we might know all people within one social group but none from another group. The methods used in this study would likely not be able to identify people from such unknown group. There are, however, network-only clustering methods that might help in such a scenario (cf. (Newman, 2004)).

5 Conclusion

We described NetKit, a network learning toolkit, and its use for various network classification tasks. We showed that simple methods performed very well across all these tasks, even when we only had very few initial labels. We described an edge-selection methodology to identify the edge-types that would yield the best performance, when constructing a homogeneous network for analysis. NetKit is freely available and open-source. We hope that it can be useful for social and organizational research.

References

Ackley, D. H., Hinton, G. E., & Sejnowski, T. J. (1985). A learning algorithm for Boltzmann machines. *Cognitive Science, 9*, 147–169.

Bernstein, A., Clearwater, S., Hill, S., Perlich, C., & Provost, F. (2002). Discovering Knowledge from Relational Data Extracted from Business News. *Proceedings of the KDD-2002 Workshop on Multi-Relational Data Mining (MRDM-2002)*.

Bernstein, A., Clearwater, S., & Provost, F. (2003). The Relational Vector-space Model and Industry Classification. *IJCAI 2003 Workshop on Learning Statistical Models from Relational Data (SRL-2003)* (pp. 8–18).

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

[9] We required a single edge type for our homogeneous case study; it is reasonable to conjecture that even if heterogeneous links are allowed, a small set of good links would be preferable. For example, a link-based classifier produces a feature vector representation with multiple positions per link type.

Boykov, Y., Veksler, O., & Zabih, R. (2001). Fast Approximate Energy Minimization via Graph Cuts. *IEEE Transactions on Pattern Analysis and Machine Intellingence (PAMI)*, *23*, 1222–1239.

Chakrabarti, S., Dom, B., & Indyk, P. (1998). Enhanced Hypertext Categorization Using Hyperlinks. *Proceedings of the ACM SIGMOD International Conference on Management of Data* (pp. 307–318).

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* (pp. 509–516).

Domingos, P., & Richardson, M. (2004). Markov Logic: A Unifying Framework for Statistical Relational Learning. *Proceedings of the ICML-2004 Workshop on Statistical Relational Learning and its Connections to Other Fields (pp. 49-54)*. Banff, Canada.

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

Fawcett, T., & Provost, F. (1999). Activity monitoring: Noticing interesting changes in behavior. *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining* (pp. 63–62).

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.

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.

Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *National Academy of Sciences*, *79*, 2554–2558.

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* (pp. 593–598).

Kohavi, R., John, G., Long, R., Manley, D., & Pfleger, K. (1994). MLC++:a Machine Learning Library in C++. *Tools with Artificial Intelligence*.

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. *Eighteenth International Conference on Machine Learning* (pp. 282–289).

Lu, Q., & Getoor, L. (2003). Link-Based Classification. *International Conference on Machine Learning, ICML-2003* (pp. 496–503).

Macskassy, S. A., & Provost, F. (2003). A Simple Relational Classifier. *Proceedings of the Second Workshop on Multi-Relational Data Mining (MRDM-2003) at KDD-2003* (pp. 64–76).

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.

Macskassy, S. A., & Provost, F. (2005). 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. (2004). Dependency Networks for Relational Data. *Proceedings of the Fourth IEEE International Conference in Data Mining (ICDML)* (pp. 170–177).

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)* (pp. 625–630).

Newman, M. E. J. (2003). Mixing patterns in networks. *Phys. Rev. E*, *67*. 026126.

Newman, M. E. J. (2004). Fast algorithm for detecting community structure in networks. *Phys. Rev. E*, *69*. 066133.

Perlich, C., & Provost, F. (2003). Aggregation-based feature invention and relational concept classes. *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 167–176).

Perlich, C., & Provost, F. (2004). *ACORA: Distribution-based Aggregation for Relational Learning from Identifier Attributes*. Technical Report CeDER Working Paper CeDER-04-04. Stern School of Business, New York University.

Popescul, A., & Ungar, L. H. (2003). Statistical Relational Learning for Link Prediction. *Workshop on Learning Statistical Models from Relational Data at IJCAI-2003*.

Provost, F., Perlich, C., & Macskassy, S. A. (2003). Relational Learning Problems and Simple Models. *IJCAI 2003 Workshop on Learning Statistical Models from Relational Data (SRL-2003)* (pp. 116–120).

Taskar, B., Abbeel, P., & Koller, D. (2002). Discriminative Probabilistic Models for Relational Data. *Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI02)* (pp. 895–902). Edmonton, Canada.

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.

Witten, I. H., & Frank, E. (2000). In *Data Mining: Practical machine learning tools with Java implementations*. San Francisco: Morgan Kaufmann.