A Shrinkage Approach for Modeling Non-Stationary Relational Autocorrelation

Pelin Angin
Department of Computer Science
Purdue University
pangin@cs.purdue.edu

Jennifer Neville
Department of Computer Science and Statistics
Purdue University
neville@cs.purdue.edu

ABSTRACT

Recent research has shown that *collective classification* in relational data often exhibit significant performance gains over conventional approaches that classify instances individually. This is primarily due to the presence of autocorrelation in relational datasets, which means that the class label of related entities are correlated and inferences about one instance can be used to improve inferences about linked instances. Statistical relational learning techniques exploit relational autocorrelation by modeling *global* autocorrelation dependencies under the assumption that the level of autocorrelation is stationary throughout the dataset. To date, there has been no work examining the appropriateness of this stationarity assumption. In this paper, we examine two real-world datasets and show that there is significant variance in the autocorrelation dependencies throughout the relational data graphs. To account for this, we develop a technique for modeling non-stationary autocorrelation in relational data. We compare to two baseline techniques which model either the local or the global autocorrelation dependencies in isolation and show that a shrinkage model results in significantly improved model accuracy.

1. INTRODUCTION

Relational data offer unique opportunities to boost model accuracy and to improve decision-making quality if the learning algorithms can effectively model the additional information the relationships provide. The power of relational data lies in combining intrinsic information about objects in isolation with information about related objects and the connections among those objects. For example, relational information is often central to the task of fraud detection because fraud and malfeasance are social phenomena, communicated and encouraged by the presence of other individuals who also wish to commit fraud (e.g., [4]).

In particular, the presence of *autocorrelation* provides a strong motivation for using relational techniques for learning and inference. Autocorrelation is a statistical dependency between the values of the same variable on related entities, which is a nearly ubiquitous characteristic of relational datasets. For example, pairs of brokers working at the same branch are more likely to share the same fraud status than randomly selected pairs of brokers.

Recent research has advocated the utility of modeling autocorrelation in relational domains [21, 2, 9]. The presence of autocorrelation offers a unique opportunity to improve model performance because inferences about one object can be used to improve inferences about related objects. For example, when broker fraud status exhibits autocorrelation, if we know one broker is involved in fraudulent activity, then his coworkers have increased likelihood of being engaged in misconduct as well. Indeed, recent work in relational domains has shown that *collective inference* over an entire dataset can result in more accurate predictions than conditional inference for each instance independently [3, 14, 22, 17, 8] and that the gains over conditional models increase as autocorrelation increases [7].

There have been a number of approaches to modeling autocorrelation and the success of each approach depends on the characteristics of the target application. When there is overlap between the dataset used for learning and the dataset where the model will be applied (i.e., some instances appear in both datasets), then models can reason about *local* autocorrelation dependencies by incorporating the *identity* of instances in the data [18]. For example, in relational data with a temporal ordering, we can learn a model on all data up to time t, then apply the model to the same dataset at time $t + x$, inferring class values for the instances that appear after t. On the other hand, when the training set and test set are either disjoint or largely separable, then models must generalize about the autocorrelation dependencies in the training set. This is achieved by reasoning about *global* autocorrelation dependencies and inferring the labels in the test set *collectively* [21, 15, 19]. For example, a single website could be used to learn a model of page topics, then the model can be applied to predict the topics of pages in other (separate) websites.

One limitation of models that represent and reason with global autocorrelation is that the methods assume the autocorrelation dependencies are *stationary* throughout the relational data graph. To date, there has been no work examining the appropriateness of this assumption. We conjecture that many real-world relational datasets will exhibit

The 2nd SNA-KDD Workshop '08 (SNA-KDD'08), August 24, 2008 , Las Vegas, Nevada , USA .
Copyright 2008 ACM 978-1-59593-848-0...$5.00.

significant variability in the autocorrelation dependencies throughout the dataset and thus violate the assumption of stationarity. The variability could be due to a number of factors, including an underlying latent community structure with varying group properties or an association between the graph topology and autocorrelation dependence. For example, different research communities may have varying levels of cohesiveness and thus cite papers on other topics with varying degrees. When the autocorrelation varies significantly throughout a dataset, it may be more accurate to model the dependencies *locally* rather than *globally.* In this case, identity-based approaches may more accurately capture local variations in autocorrelation. However, identity-based approaches, by definition, do not generalize about dependencies but focus on the characteristics of a single instance in the data. This limits their applicability to situations when there is significant overlap between the training and test sets, because they can only reason about the identity of instances that appeared in the training set. When there is insufficient information about an instance in the training set, a *shrinkage* approach, which backs off to the global estimate, may be better able to exploit the full range of autocorrelation in the data.

In this work, we outline and investigate an approach to exploiting non-stationary autocorrelation dependencies in relational data. Our approach combines local and global dependencies in a shrinkage model. We evaluate our method on two real-world relational datasets and one synthetic dataset, comparing to local-only and global-only methods and show that the shrinkage model achieves significant higher accuracy over all three datasets.

2. RELATIONAL AUTOCORRELATION

Relational autocorrelation refers to a statistical dependency between values of the same variable on related objects. More formally, we define relational autocorrelation with respect to an attributed graph $G = (V, E)$, where each node $v \in V$ represents an object and each edge $e \in E$ represents a binary relation. Autocorrelation is measured for a set of instance pairs P_R related through paths of length l in a set of edges E_R: $P_R = \{(v_i, v_j) : e_{ik_1}, e_{k_1k_2}, ..., e_{k_lj} \in E_R\}$, where $E_R = \{e_{ij}\} \subseteq E$. It is the correlation between the values of a variable X on the instance pairs $(v_i.x, v_j.x)$ such that $(v_i, v_j) \in P_R$.

A number of widely occurring phenomena give rise to autocorrelation dependencies. A hidden condition or event, whose influence is correlated among instances that are closely located in time or space, can result in autocorrelated observations [13, 1]. Social phenomena, including social influence [10], diffusion [5], and homophily [12], can also cause autocorrelated observations through their influence on social interactions that govern the data generation process. Relational data often record information about people (e.g., organizational structure, email transactions) or about artifacts created by people (e.g., citation networks, World Wide Web) so it is likely that these social phenomena will contribute to autocorrelated observations in relational domains.

Autocorrelation is a nearly ubiquitous characteristic of relational datasets. For example, recent analysis of relational datasets has reported autocorrelation in the topics of hyperlinked web pages [3], the topics of coreferent scientific papers [22], and the industry categorization of corporations that co-occur in news stories [2].

When there are dependencies among the class labels of related instances, relational models can exploit those dependencies in two ways. The first technique takes a local approach to modeling autocorrelation. More specifically, the probability distribution for the target class label (Y) of an instance i can be conditioned not only on the attributes (X) of i in isolation, but also on the *identity* and observed attributes of instances $\{1, ..., r\}$ related to i:[1]

$$p(y^i|\mathbf{x}^i, \{\mathbf{x}^1, ..., \mathbf{x}^r\}, \{id^1, ..., id^r\})$$
$$= p(y^i|x_1^i, ..., x_m^i, x_1^1, ..., x_m^1, \ ...\ x_1^r, ..., x_m^r, id^1, ..., id^r)$$

This approach assumes that there is sufficient information to estimate $P(Y|ID)$ for all instances in the data. However, it is unlikely that we can accurately estimate this probability for all instances—for instances that are only in the test set there is no information to estimate this probability distribution and even for instances in the training data if they only link to a few labeled instances our estimate will have high variance.

A second approach models autocorrelation by including related class labels as dependent variables in the model. More specifically, the probability distribution for the target class label (Y) of an instance i can be conditioned not only on the attributes (X) of i in isolation, but also on the attributes and the class labels of instances $\{1, ..., r\}$ related to i:

$$p(y^i|\mathbf{x}^i, \{\mathbf{x}^1, ..., \mathbf{x}^r\}, \{y^1, ..., y^r\})$$
$$= p(y^i|x_1^i, ..., x_m^i, x_1^1, ..., x_m^1, \ ...\ x_1^r, ..., x_m^r, y^1, ..., y^r)$$

Depending on the overlap between training and test sets, some of the values of $\mathbf{y}^R$ may be unknown when inferring the value for y^i. *Collective inference* techniques jointly infer the unknown y values in a set of interrelated instances in this case. Collective inference approaches typically model the autocorrelation at a global level (e.g., $P(Y|Y^r)$.

One key assumption of global autocorrelation models is that the autocorrelation dependencies do not vary significantly throughout the data. We have empirically investigated the validity of this assumption on two real-world datasets. The first dataset is Cora, a database of computer science research papers extracted automatically from the web using machine learning techniques [11]. We investigated the variation of autocorrelation dependencies in the set of 4,330 machine-learning papers in the following way. We first calculated the autocorrelation among the topics of pairs of cited papers using Pearson's corrected contingency coefficient [20]. The global autocorrelation is 0.824. Then we used snowball sampling [6] to partition the citation graph into sets of roughly equal-sized subgraphs and calculated the autocorrelation within each subgraph.

Figure 1 graphs the distribution of autocorrelation across the subgraphs, for snowball samples of varying size. The solid

[1] Here we use superscripts to refer to instances and subscripts to refer to attributes.

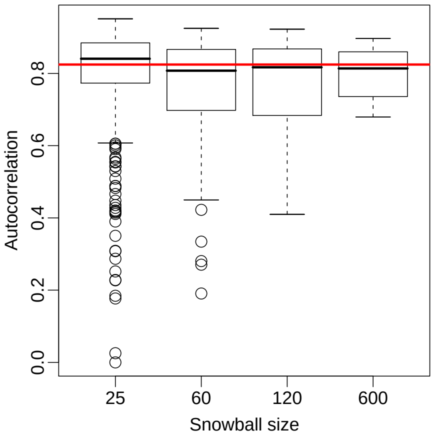


Figure 1: Autocorrelation variation in Cora.

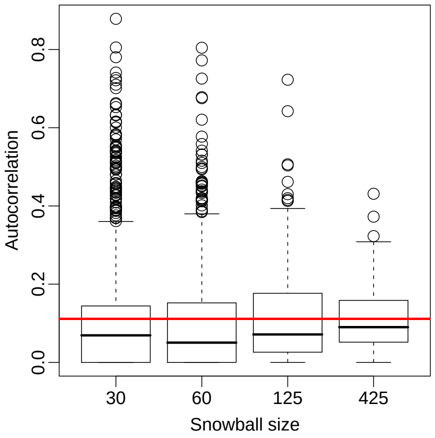


Figure 2: Autocorrelation variation in IMDB.

red line is the global level of autocorrelation. At the smallest snowball size (25), the variation of local autocorrelation is clear. More than 25% of the subgraphs have autocorrelation significantly higher than the global level. In addition, there are a non-trivial number of subgraphs that have significantly lower levels of autocorrelation.

The second dataset is the Internet Movie Database (IMDB; www.imdb.com). Our sample consists of movies released between 1980-2006, where movies are linked if they share a common actor. Figure 2 graphs the distribution of autocorrelation between the *isblockbuster* attribute of related movies, which indicates whether a movie has total earnings of more than $30,000,000 in box office receipts. The same procedure described above was used to generate subgraphs of different sizes. We observe that non-stationarity in autocorrelation is also apparent in this dataset. The global autocorrelation (0.112) is low, but more than 30% of the subgraphs have significantly higher local values of autocorrelation at a snowball size of 30.

These empirical findings motivate our approach to combining information from both the local and global level. If there is sufficient information to estimate autocorrelation locally, this will likely result in more accurate models since the estimates will capture the variability throughout the data.

3. SHRINKAGE MODELS

Here we present two classification models accounting for non-stationary autocorrelation in relational data. The models are designed for domains with overlapping training and test sets, as they use identity-based *local* dependencies as a significant component. For each classification task, a subset of the nodes in the network is treated as *unlabeled*, determined by a random labeling scheme. Only the *labeled* instances are taken into account while learning the models.

The basic model employs a simple naive Bayes classification scheme, where the probability of class is conditioned on the characteristics of the nodes related to that instance (which we call the *neighbor* nodes) and each neighbor is conditionally independent given the class. The probability that a node i has class label y is:

$$p(y^i|N(i)) \propto p(y) \cdot \prod_{j \in N(i)} p(y|j)$$

where $N(i)$ represents the set of *neighbor* nodes of i, $p(y)$ is the prior probability of class label y, and $p(y|j)$ is the conditional probability of y given that a node is linked to instance j.

Our *shrinkage* method uses a combination of the global and local dependencies between class labels of related instances to classify nodes in a network. Let j be a *neighbor* of the node to be classified and $I_y(k)$ be an indicator function, which returns 1 if node k has class label y and 0 otherwise. Then the local component of the probability that a node i has class label y, conditioned on its neighbor node j is:

$$p_L(y|j) = \frac{\sum_{k \in N(j)} I_y(k)}{|N(j)|}$$

Let Y be the set of all possible class labels and $G_{y^l y^m}$ be the set of linked node pairs in the whole network, where the class label of the first node in the pair is y^l and that of the other is y^m. Then the global component of the probability that the node i has class label y, conditioned on its neighbor node j with class label y^j is:

$$p_G(y|j) = \frac{|G_{yy^j}|}{\sum_{y' \in Y} |G_{y'y^j}|}$$

The *shrinkage* method gives the probability that node i has class label y, conditioned on its neighbor j as:

$$p(y|j) = \alpha \cdot p_L(y|j) + (1 - \alpha) \cdot p_G(y|j)$$

The weight α, which takes a value in the range [0,1], determines how much to *back off* to the global level of autocorrelation when the local autocorrelation alone will not suffice to make an accurate prediction. Setting this parameter to 0 gives the *global* baseline model, where we make use of the global autocorrelation factor alone to predict class labels of nodes. When local autocorrelation around a neighbor of the node to be classified is believed to provide enough information to make an accurate prediction, this parameter is set to 1, giving the *local* baseline model.

As α is the weight we assign to the local autocorrelation component, we would like its value to be high when there is sufficient number of labeled instances at the local surrounding of the neighbor being considered. Different rules can be used for setting this parameter. Our initial approach for the *shrinkage* model employs a three-branch rule based on the number of labeled instances linked to a neighbor node. Our second approach uses a two-branch rule with an α determined by cross-validation.

3.1 Three-Branch Model

The *three-branch rule* for the *shrinkage* method involves a manual setting of α values based on the number of labeled neighbors of the neighboring node j. Two values, 5 and 10, are used as the thresholds for different branches of the rule. The rule for setting α in this method is:

$$\alpha = \begin{cases} 0.1 & \text{if } |N(j)| < 5 \\ 0.5 & \text{if } 5 \leq |N(j)| < 10 \\ 0.9 & \text{if } |N(j)| \geq 10 \end{cases}$$

This rule has the effect of relying mainly on the global level of autocorrelation without completely ignoring the local level, when there are relatively few labeled instances in the local neighborhood. The choice of these two threshold was determined from our background knowledge on the number of data points needed to estimate parameters accurately. Although it is an adhoc rule, the experimental results in Section 4 show that it captures local variations in autocorrelation quite well.

3.2 Two-Branch Model

In addition to the *three-branch* model, we also considered a two-branch rule with a single α parameter set by cross-validation:

$$\alpha = \begin{cases} 1 - \alpha' & \text{if } |N(j)| < 5 \\ \alpha' & \text{if } 5 \geq |N(j)| \end{cases}$$

To determine the value of α' to use in a classification task, we perform cross-validation experiments on the set of labeled instances in the network, with a range of different α's. Then we average across the values of α' giving the best performance in each fold of the cross-validation to get the value to use in the classification task. The details of the cross-validation procedure for each dataset used in the experimental evaluation are provided in Section 4.

4. EXPERIMENTAL EVALUATION

Experimental evaluation of the proposed models was done on two real-world datasets and a synthetic dataset. The performance metric used in all evaluations is the area under the ROC curve (AUC). For the two-branch model , six different values (0.99, 0.9, 0.8, 0.7, 0.6 and 0.5) were considered for α' during cross-validation.

4.1 Cora Dataset

The Cora dataset is a collection of computer science research papers as described in Section 2. The task for the Cora dataset was to classify research papers published between 1994-1998, in the *Machine Learning* field, into seven subtopics based on their citation relations. We used temporal sampling, where the training and test sets were based on the *publication year* attribute of the papers in the dataset. For each year t, the task was to predict the topics of the papers published in t, where the models were learned on papers published before t. Papers published after t were not considered in the classification task.

For the *two-branch* model, we used the following procedure to determine the value of α', which would be used in the model for year t:

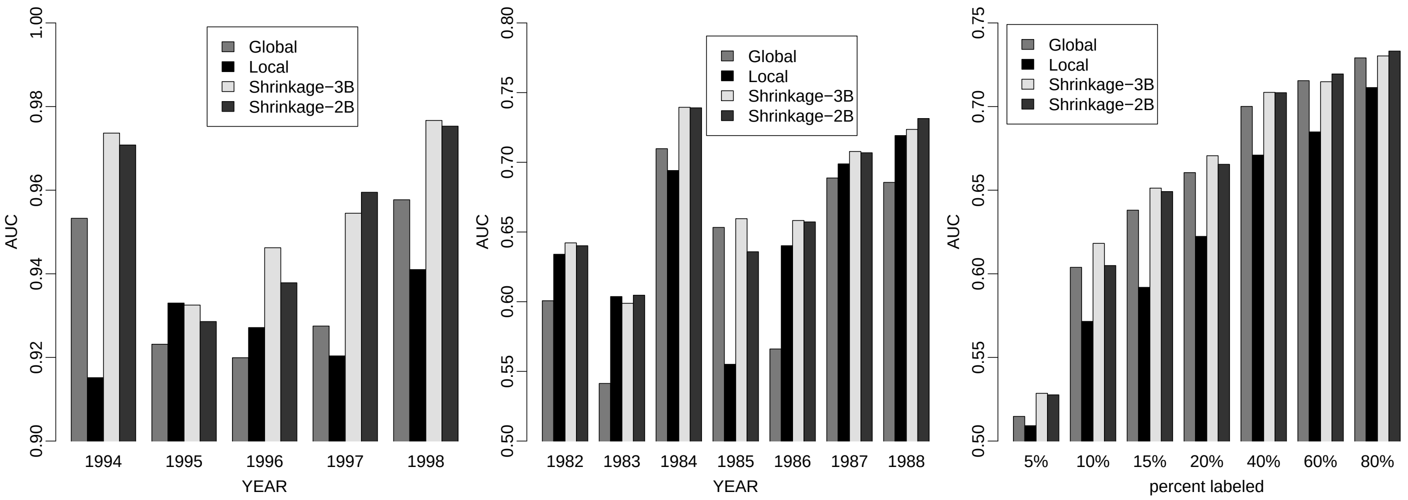
```
$S_{\alpha'} = \emptyset$
Repeat five times:
  • Randomly select 50% of the papers in year $t-1$, mark them as unlabeled and use as the test set; use the remaining data as the training set.
  • For every value of $\alpha'$ in the set {0.99, 0.9, 0.8, 0.7, 0.6, 0.5}:
      – Estimate the two-branch model on the training set using $\alpha'$.
      – Apply the model to the test set.
      – Calculate AUC.
  • Add $\alpha'$ with highest AUC to $S_{\alpha'}$
$\alpha'$ = average of values in $S_{\alpha'}$.
```

Figure 3(a) provides a comparison of the performances of the local, global and shrinkage approaches in the Cora classification task (*Shrinkage-2B* stands for the shrinkage method with the two-branch rule, and *Shrinkage-3B* is that with the three-branch rule). For the AUC calculations, the *Neural Networks* category, which has the greatest number of publications, was considered as the *positive* category and all the others were considered *negative.* As seen in the figure, both shrinkage models achieve better accuracy than the *global* baseline model for all years. The shrinkage models also outperform the *local* baseline model in all years except 1995, where they have similar performance.

We performed one-tailed, paired t-tests to measure the significance of the difference between the AUC values we achieve with the different classification methods. A p-value of 0.05 or less is generally considered as an indication of significant difference between the performances of two algorithms. Both *shrinkage* models give p-values much smaller than 0.05 when they are compared with the baseline models. This suggests that the proposed *shrinkage* model provides significantly better performance than the previously studied *global* and *local* models for classification.

4.2 IMDB Dataset

The IMDB dataset consists of movies released between 1980-2006 and related entities such as directors of movies, actors casting in these movies and studios making the movies. Each movie in the dataset has the binary class label *isblockbuster*,



(a) Cora classification

(b) IMDB classification

(c) Synthetic dataset classification

Figure 3: Experimental results

which indicates whether the total earnings of the movie was greater than \$30,000,000 (inflation adjusted). We considered the movies released in the U.S. in the years 1982-1988 and again sampled temporally by year. For the experiments, we a the version of the dataset where two movies are linked if they have at least one actor in common. As in the Cora classification task, the samle for each year t was made to consist of movies released before or in t. For the IMDB dataset we use the correlation between the class labels of movies released in year t and all past movies they are linked to for the global autocorrelation component, instead of considering all labeled node pairs in the network. For the *two-branch* model, we used the same procedure as we did for the Cora dataset to determine the value of α', where the labeling was based on the release years of movies.

Figure 3(b) shows the accuracies obtained for the IMDB classification task using the different classification methods discussed. As seen in the figure, the *shrinkage* model with the three-branch rule always performs better than the *global* baseline model, which is also valid for the two-branch rule with the exception of 1985. The *shrinkage* approaches also provides better accuracy than the *local* baseline model for all years except 1983, where they have similar AUC values.

We performed one-tailed, paired t-tests to measure the significance of the difference between the AUC values achieved with different classification methods, as we did for the Cora dataset. Both shrinkage models are significantly better than the local and gloabl baseline models ($p < 0.05$) with the exception of *Shrinkage-3B* vs. *local*, which is only significant at a $p = 0.06$ level. Once again, the proposed models' significant improvements in classification accuracy were confirmed with the t-test results.

4.3 Synthetic Data

Data for these last set of experiments were generated with a latent group model framework [16], where a single attribute on the objects was used to determine the class label of the object. Variance in autocorrelation was explicitly introduced during the data generation process. We used five different class label probabilities (0.5, 0.6, 0.7, 0.8 and 0.9) to generate the class labels of instances in the same group, producing varying levels of autocorrelation throughout the graph. The dataset consists of 6500 objects linked through undirected edges, each of which has a binary class label. For the experiments, 20% of all nodes were selected randomly to form the test set in all cases. The labeled instances were chosen randomly from the remaining 80%. We considered labeled proportions of 5%, 10%, 15%, 20%, 40%, 60% and 80% to investigate the performance of the models at different levels of labeled training data. For the *two-branch* model, we used a cross-validation procedure similar to that used for the real-world datasets.

Figure 3(c) provides a comparison of the performances (for different percentages of labeled data) of the *shrinkage*, *local* and *global* models in this classification task. As seen in this figure, the shrinkage model gives better accuracy than both the local-only and global-only approaches for all levels of random labelings of the data. This figure also allows us to see that classification accuracy follows an increasing trend with increasing amount of labeled data regardless of the model used, which demonstrates the importance of having sufficient labeled nodes in relational classification tasks. The most significant improvement resulting from increasing amounts of labeled data is observed in the *local* baseline model, justifying more effective use of local information in the presence of sufficiently labeled local neighborhoods. The significance of the difference in the performances of the discussed algorithms was most clearly seen in this dataset, with p-values smaller than 0.005 for all t-tests performed.

5. CONCLUSIONS

In this paper, we proposed two classification models accounting for non-stationary autocorrelation in relational data. While previous research on collective inference models for relational data has assumed that autocorrelation is stationary throughout the data graph, we have demonstrated that this assumption does not hold in at least two real-world datasets. The models that we propose combine *local* and *global* dependencies between attributes and/or class labels of related instances to make accurate predictions.

We evaluated the performance of the proposed shrinkage models on one synthetic and two real-world datasets and compared with two baseline models considering either *global* or *local* dependencies alone. The results indicate that our *shrinkage* models, which *back off* from local to global autocorrelation as necessary, allows significant improvements in prediction accuracy over both of the baseline classification models. We achieve this improvement with a very simple model that adjusts weights of local and global autocorrelation factors based on the amount of information (labeled instances) available at the local level. The success of the shrinkage model is due to its focus on the local autocorrelation when there is enough local information for accurate estimation. Obviously, *backing off* to the global autocorrelation when the information provided at the local level is not sufficient is also contributes to the success of the model, as having sufficient training data is important for achieving good prediction performance with any classification model.

We provided empirical evalution on two real-world relational datasets, but the models we propose can be used for classification tasks in any relational domain due to their simplicity and generality. In addition, the shrinkage approach could easily be incorporated into other statistical relational models that use global autocorrelation and collective inference. The *shrinkage* model we propose is likely to be a useful tool for social network analysis, yielding better performance than the baseline models discussed, in the presence of non-stationary autocorrelation.

Acknowledgments

This material is based on research sponsored by DARPA, AFRL, and IARPA under grants HR0011-07-1-0018 and FA8750-07-2-0158. The views and conclusion 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 DARPA, AFRL, IARPA, or the U.S. Government.

6. REFERENCES

[1] L. Anselin. *Spatial Econometrics: Methods and Models.* Kluwer Academic Publisher, The Netherlands, 1998.

[2] A. Bernstein, S. Clearwater, and F. Provost. The relational vector-space model and industry classification. In *Proceedings of the IJCAI-2003 Workshop on Learning Statistical Models from Relational Data*, pages 8–18, 2003.

[3] S. Chakrabarti, B. Dom, and P. Indyk. Enhanced hypertext categorization using hyperlinks. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pages 307–318, 1998.

[4] C. Cortes, D. Pregibon, and C. Volinsky. Communities of interest. In *Proceedings of the 4th International Symposium of Intelligent Data Analysis*, pages 105–114, 2001.

[5] P. Doreian. Network autocorrelation models: Problems and prospects. In *Spatial Statistics: Past, Present, and Future*, chapter Monograph 12, pages pp. 369–389. Ann Arbor Institute of Mathematical Geography, 1990.

[6] L. Goodman. Snowball sampling. *Annals of Mathematical Statistics*, 32:148–170, 1961.

[7] D. Jensen, J. Neville, and B. Gallagher. Why collective inference improves relational classification. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 593–598, 2004.

[8] Q. Lu and L. Getoor. Link-based classification. In *Proceedings of the 20th International Conference on Machine Learning*, pages 496–503, 2003.

[9] S. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. Technical Report CeDER-04-08, Stern School of Business, New York University, 2004.

[10] P. Marsden and N. Friedkin. Network studies of social influence. *Sociological Methods and Research*, 22(1):127–151, 1993.

[11] A. McCallum, K. Nigam, J. Rennie, and K. Seymore. A machine learning approach to building domain-specific search engines. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pages 662–667, 1999.

[12] M. McPherson, L. Smith-Lovin, and J. Cook. Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27:415–445, 2001.

[13] T. Mirer. *Economic Statistics and Econometrics.* Macmillan Publishing Co, New York, 1983.

[14] J. Neville and D. Jensen. Iterative classification in relational data. In *Proceedings of the Workshop on Statistical Relational Learning, 17th National Conference on Artificial Intelligence*, pages 42–49, 2000.

[15] J. Neville and D. Jensen. Dependency networks for relational data. In *Proceedings of the 4th IEEE International Conference on Data Mining*, pages 170–177, 2004.

[16] J. Neville and D. Jensen. Leveraging relational autocorrelation with latent group models. In *Proceedings of the 4th Multi-Relational Data Mining Workshop, KDD2005*, 2005.

[17] J. Neville, D. Jensen, and B. Gallagher. Simple estimators for relational Bayesian classifers. In *Proceedings of the 3rd IEEE International Conference on Data Mining*, pages 609–612, 2003.

[18] C. Perlich and F. Provost. Acora: Distribution-based aggregation for relational learning from identifier attributes. *Machine Learning*, 62(1/2):65–105, 2006.

[19] M. Richardson and P. Domingos. Markov logic networks. *Machine Learning*, 62:107–136, 2006.

[20] L. Sachs. *Applied Statistics.* Springer-Verlag, 1992.

[21] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pages 485–492, 2002.

[22] B. Taskar, E. Segal, and D. Koller. Probabilistic classification and clustering in relational data. In *Proceedings of the 17th International Joint Conference on Artificial Intelligence*, pages 870–878, 2001.