

Leveraging Label-Independent Features for Classification in Sparsely Labeled Networks: An Empirical Study

Brian Gallagher

Lawrence Livermore National Laboratory
P.O. Box 808, L-560, Livermore, CA 94551

bgallagher@llnl.gov

Tina Eliassi-Rad

Lawrence Livermore National Laboratory
P.O. Box 808, L-560, Livermore, CA 94551

eliassi@llnl.gov

ABSTRACT

We address the problem of *within-network classification* in sparsely labeled networks. Recent work has demonstrated success with *statistical relational learning* (SRL) and *semi-supervised learning* (SSL) on such problems. However, both approaches rely on the availability of labeled nodes to infer the values of missing labels. When few labels are available, the performance of these approaches can degrade. In addition, many such approaches are sensitive to the specific set of nodes labeled. So, although average performance may be acceptable, the performance on a specific task may not. We explore a complimentary approach to within-network classification, based on the use of *label-independent (LI) features* – i.e., features not influenced by the values of class labels. While previous work has made some use of LI features, the effects of these features on classification performance have not been extensively studied. Here, we present an empirical study in order to better understand these effects. Through experiments on several real-world data sets, we show that the use of LI features produces classifiers that are less sensitive to specific label assignments and can lead to performance improvements of over 40% for both SRL and SSL based classifiers. We also examine the relative utility of individual LI features and show that, in many cases, it is a combination of a few diverse network-structural characteristics that is most informative.

Categories and Subject Descriptors

H.2.8 [Database Management]: Database Applications – *Data Mining*; I.2.6 [Artificial Intelligence]: Learning; I.5.1 [Pattern Recognition]: Models – *Statistical*; I.5.2 [Pattern Recognition]: Design Methodology – *Feature evaluation and selection*.

General Terms

Algorithms, Performance, Experimentation.

Keywords

Statistical relational learning, semi-supervised learning, social network analysis, feature extraction, collective classification.

Copyright 2008 Association for Computing Machinery. ACM acknowledges that this contribution was authored or co-authored by an employee, contractor or affiliate of the U.S. Government. As such, the Government retains a nonexclusive, royalty-free right to publish or reproduce this article, or to allow others to do so, for Government purposes only.

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.

1. INTRODUCTION

In this paper, we address the problem of within-network classification. We are given a network in which some of the nodes are “labeled” and others are “unlabeled.” Our goal is to assign the correct labels to the unlabeled nodes from among a set of possible class labels (i.e., to “classify” them). For example, we may wish to identify cell phone users as either ‘fraudulent’ or ‘legitimate.’

Cell phone fraud is an example of an application where networks are often very sparsely labeled. We may have a handful of known fraudsters and a handful of known legitimate users, but for the vast majority of users, we do not know the correct label. For such applications, it is reasonable to expect that we may have access to labels for fewer than 10%, 5%, or even 1% of the nodes. In addition, cell phone networks are generally anonymized. That is, nodes in these networks often contain no attributes besides class labels that could be used to identify them. It is this kind of sparsely labeled, anonymized network that is the focus of this work. Put another way, our work focuses on *univariate within-network classification in sparsely labeled networks*.

Relational classifiers have been shown to perform well on network classification tasks because of their ability to make use of dependencies between class labels (or attributes) of related nodes [17]. However, because of their dependence on attributes, the performance of relational classifiers can substantially degrade when a large proportion of neighboring instances are also unlabeled. In many cases, *collective classification* provides a solution to this problem, by enabling the simultaneous classification of a number of related instances [15]. However, previous work has shown that the performance of collective classification can also degrade when there are too few labels available, eventually to the point where classifiers perform better without it [13].

In this paper, we explore another source of information present in networks that is independent of the available node labels. Such information can be represented using what we call *label-independent (LI) features*. The main contribution of this paper is an in-depth examination of the effects of label-independent features on within-network classification. In particular, we address the following questions:

1. *Can LI features make up for a lack of information due to sparsely labeled data?* Answer: Yes.
2. *Can LI features provide information above and beyond that provided by the class labels?* Answer: Yes.
3. *How do LI features improve classification performance?* Answer: Because they are less sensitive to the specific labeling as-

signed to a graph, classifiers that use label-independent features produce more consistent results across prediction tasks.

4. *Which LI features are the most useful?* Answer: A combination of a few diverse network-structural characteristics (node and link counts and betweenness) is the most informative.

Section 2 covers related work. Section 3 describes our approach for modeling label-independent characteristics of networks. Sections 4 and 5, respectively, present our experimental design and results. We conclude the paper in Section 6.

2. RELATED WORK

In recent years, there has been a great deal of work on statistical relational learning (SRL) [5, 7, 8, 11, 13]. All SRL techniques make use of label-dependent relational information. Some use label-independent information as well. Relational Probability Trees [10] use degree-based features. Singh et al. [16] use structural properties to prune networks for attribute prediction. Neville and Jensen [12] use spectral clustering to group instances based on link structure and use these groups in learning classifiers.

There has also been extensive work on overcoming label sparsity through techniques for label propagation. This work falls into two research areas: (1) collective classification (CC) [2, 6, 8, 9, 13, 15, 17] and (2) graph-based semi-supervised learning (SSL) [18, 19]. Previous work confirms our observation that the performance of CC can suffer when labeled data is very sparse [13]. Other research shows that commonly used approaches to collective classification and semi-supervised learning are essentially equivalent. In particular, Macskassy and Provost [8] compare the SSL Gaussian Random Field (GRF) model [18] to a SRL weighted-vote relational neighbor (wvRN) model that uses relaxation labeling for CC (wvRN+RL) and conclude that the two models are nearly identical in terms of accuracy, but GRF can produce better probability rankings. Our results with wvRN+RL are consistent with this conclusion. The "ghost edge" approach of Gallagher et al. [4] combines aspects of both SRL and SSL, and compares favorably with both wvRN+RL and GRF.

3. Label-Dependent vs. Independent Features

Relational classifiers leverage link structure to improve performance. Most frequently, links are used to incorporate attribute information from neighboring nodes. However, link structure can also be used to extract structural statistics of a node (e.g., the number of adjacent links). We can divide relational features into two categories: label-dependent and label-independent.

Label-dependent (LD) features use both structure and attributes (or labels) of nodes in the network. The most commonly used LD features are aggregations of the class labels of nodes one link away (e.g., the number of neighbors with the class label 'fraudulent'). LD features are the basis for incorporating relational information in many SRL classifiers. *Label-independent* (LI) features use network structure, but no attributes or labels (e.g., the number of neighboring nodes). The essential difference between LD and LI features is that LD features change as the class-label assignments in a network change, whereas LI features have the same value regardless of the values of class labels.

3.1 Extracting Label-Independent Features

We consider four LI features on nodes: (1) the number of neighboring nodes, (2) the number of incident links, (3) between-

ness centrality, and (4) clustering coefficient. Features 1 and 2, respectively, are node-based and link-based measures of degree. Note that in multigraphs, these two are different. Betweenness centrality measures how "central" a node is in a network, based on the number of shortest paths that pass through it. Clustering coefficient measures neighborhood strength, based on how connected a node's neighbors are to one another. For details, we refer the reader to a study by Mark Newman [14].

The success of network-structural characteristics as predictors of class relies on two assumptions: (1) members of different classes play different roles in a network and (2) these roles can be differentiated by structural characteristics. Assumption 2 is met in many cases. For instance, popular nodes can be identified by degree and "important" nodes can be identified by centrality measures. Whether assumption 1 is met depends on the class label. Suppose that executives tend to be more popular and central than an average employee in a company's communication network and employees with a particular job title tend to have similar popularity and centrality, regardless of department. Then, we would expect structural features to be more useful for identifying executives than members of a particular department.

4. EXPERIMENTAL DESIGN

4.1 Classifiers

On each classification task, we ran ten individual classifiers: four variations of a link-based classifier [7], four variations of a relational neighbor classifier [8], and two variations of the Gaussian Random Field classifier [18]. We describe each below.

nLB is the network-only link-based classifier [7]. It uses logistic regression to model a node's class given the class of neighboring nodes. A node's neighborhood is summarized by link-weighted counts of neighboring nodes for each class.

nLB LI is composed of two logistic regression models: (1) nLB and (2) logLI, which uses the four LI features. The nLB LI classifier calculates the probability of each class as:

$$P(C) = w \cdot P_{nLB}(C) + (1 - w) \cdot P_{logLI}(C) \quad (1)$$

where w is calculated based on the individual performance of nLB and logLI over 10-fold cross validation on the training data. We calculate area under the ROC curve (AUC) for each fold and then obtain an average AUC score for each classifier, AUC_{LD} and AUC_{LI} . We then set w as follows:

$$w = \frac{AUC_{LD}}{AUC_{LD} + AUC_{LI}} \quad (2)$$

nLB+ICA uses the nLB classifier, but performs collective classification using the ICA algorithm described in Section 4.2.

nLB LI+ICA uses the nLB LI classifier, but performs collective classification using the ICA algorithm described in Section 4.2.

wvRN is the weighted-vote relational neighbor classifier [8]. It is a simple non-learning classifier. Given a node i and a set of neighboring nodes, N , the wvRN classifier calculates the probability of each class for node i as:

$$P(C_i = c | N) = \frac{1}{L_i} \sum_{j \in N} \begin{cases} w_{i,j} & \text{if } C_j = c \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where $w_{i,j}$ is the number of links between nodes i and j and L_j is the number of links connecting node i to labeled nodes. When node i has no labeled neighbors, we use the prior probabilities observed in the training data.

wvRNLI combines the LI features with wvRN in the same way that nLBLI does with nLB.

wvRN+ICA uses the wvRN classifier, but performs collective classification using the ICA algorithm described in Section 4.2.

wvRNLI+ICA uses wvRNLI, but performs collective classification using the ICA algorithm described in Section 4.2.

GRF is the semi-supervised Gaussian Random Field approach of Zhu et al. [18]. We made one modification to accommodate disconnected graphs. Zhu computes the graph Laplacian as $L = D - cW$, where $c=1$. We set $c=0.9$ to ensure that L is diagonally dominant and thus invertible. We observed no substantial impact on performance in connected graphs due to this change.

GRFLI combines the LI features with GRF as nLBLI does with nLB. We also tried the approach of Zhu et al. [18]: attach a “dongle” node to each unlabeled node and assign it a label using the external LI classifier. The transition probability from node i to its dongle is η and all other transitions from i are discounted by $1-\eta$. This approach did not yield any improvements. So, we use the weighted sum approach (i.e., Equation 1) for consistency.

4.2 Collective Classification

To perform collective classification, we use the iterative classification algorithm (ICA) [8], up to 1000 iterations. We chose ICA because (1) it is simple, (2) it performs well on a variety of tasks, and (3) it tends to converge more quickly than other approaches. We also performed experiments using relaxation labeling (RL) [8]. Our results are consistent with previous research showing that the accuracy of wvRN+RL is nearly identical to GRF, but GRF produces higher AUC values [8]. We omit these results due to the similarity to GRF. For a comparison of wvRN+RL and GRF on several of the same tasks used here, see Gallagher et al. [4]. Overall, ICA slightly outperforms RL for the nLB classifier.

Several of our data sets have large amounts of unlabeled data since ground truth is simply not available. There are two reasonable approaches to collective classification (CC) here: (1) perform CC over the entire graph and (2) perform CC over the core set of nodes only (i.e., nodes with known labels). Approach 2 outperforms 1 in almost all cases, despite disconnecting the network in some cases. Therefore, we report results for approach 2.

4.3 Experimental Methodology

Each data set has a set of core nodes for which we know the true class labels. Several data sets have additional nodes for which there is no ground truth available. Classifiers have access to the entire graph for both training and testing. However, we hide labels for 10% – 90% of the core nodes. Classifiers are trained on all labeled core nodes and evaluated on all unlabeled core nodes.

For each proportion labeled we run 30 trials. For each trial, we choose a class-stratified random sample containing $100 \times (1.0 - \text{proportion labeled})\%$ of the core nodes as a test set and the remaining core nodes as a training set. Note that a single node will necessarily appear in multiple test sets. However, we carefully

choose test sets to ensure that each node in a data set occurs in the same number of test sets over the course of our experiments; and therefore, carries the same weight in the overall evaluation. Labels are kept on training nodes and removed from test nodes. We use identical train/test splits for each classifier. For more on experimental methodologies for relational classification, see Gallagher and Eliassi-Rad [3].

We use the area under the ROC curve (AUC) to compare classifiers because it is more discriminating than accuracy. Since most of our tasks have a large class imbalance (see Table 1), accuracy cannot adequately differentiate between classifiers.

Table 1: Summary of data sets and prediction tasks.

Data Set	Sample	Task	V	L	E	P(+)
Political Books	Full	Neutral?	105	105	441	0.12
Enron	Time	Executive?	9K	1.6K	50K	0.02
Reality Mining	BFS	Student?	1K	84	32K	0.62
Reality Mining	BFS	In Study?	1K	1K	32K	0.08
HEP-TH	BFS	Diff Geometry?	3K	284	36K	0.06

4.4 Data Sets

We present results on four real-world data sets:¹ political book purchases, Enron emails, Reality Mining (RM) cell phone calls, and physics publications (HEP-TH) from arXiv. Our five tasks are to identify neutral political books, Enron executives, Reality Mining students and study participants, and HEP-TH papers with the topic “Differential Geometry.” Table 1 summarizes the prediction tasks. The *Sample* column describes the method used to obtain a data sample for our experiments: use the entire set (*full*), use a time-slice (*time*), or sample a continuous subgraph via breadth-first search (*BFS*). The *Task* column indicates the class label we try to predict. The $|V|$, $|L|$, and $|E|$ columns indicate counts of total nodes, labeled nodes, and total edges in each network. The $P(+)$ column indicates the proportion of labeled nodes that have the positive class label (e.g., 12% of the political books are neutral). For Enron, Reality Mining student, and HEP-TH, we have labels for only a subset of nodes (i.e., the “core” nodes) and can only train and test our classifiers on these nodes. However, unlabeled nodes and their connections to labeled nodes may still be exploited to calculate LI features of the labeled nodes.

5. EXPERIMENTAL RESULTS

In this section, we discuss our results. We assessed significance using paired t-tests (p -values ≤ 0.05 are considered significant).

5.1 Effects of Learning Label Dependencies

Supervised learning approaches, like nLB, use labeled nodes as training data to build a dependency model over neighboring class labels. The non-learning wvRN and GRF assume that class labels of neighbors tend to be the same. GRF performs well on the Enron and RM student tasks (Figure 1), which have high label consistency between neighbors. On the RM study task, where neighboring labels are inversely correlated, wvRN and GRF perform poorly, whereas nLB can learn the correct dependencies.

¹ More information is available at: <http://www.orgnet.com/divided2.html> (political books), <http://www.cs.cmu.edu/~enron/> (Enron), <http://reality.media.mit.edu/> (Reality Mining), and <http://kdl.cs.umass.edu/data/hepth/hepth-info.html> (HEP-TH).

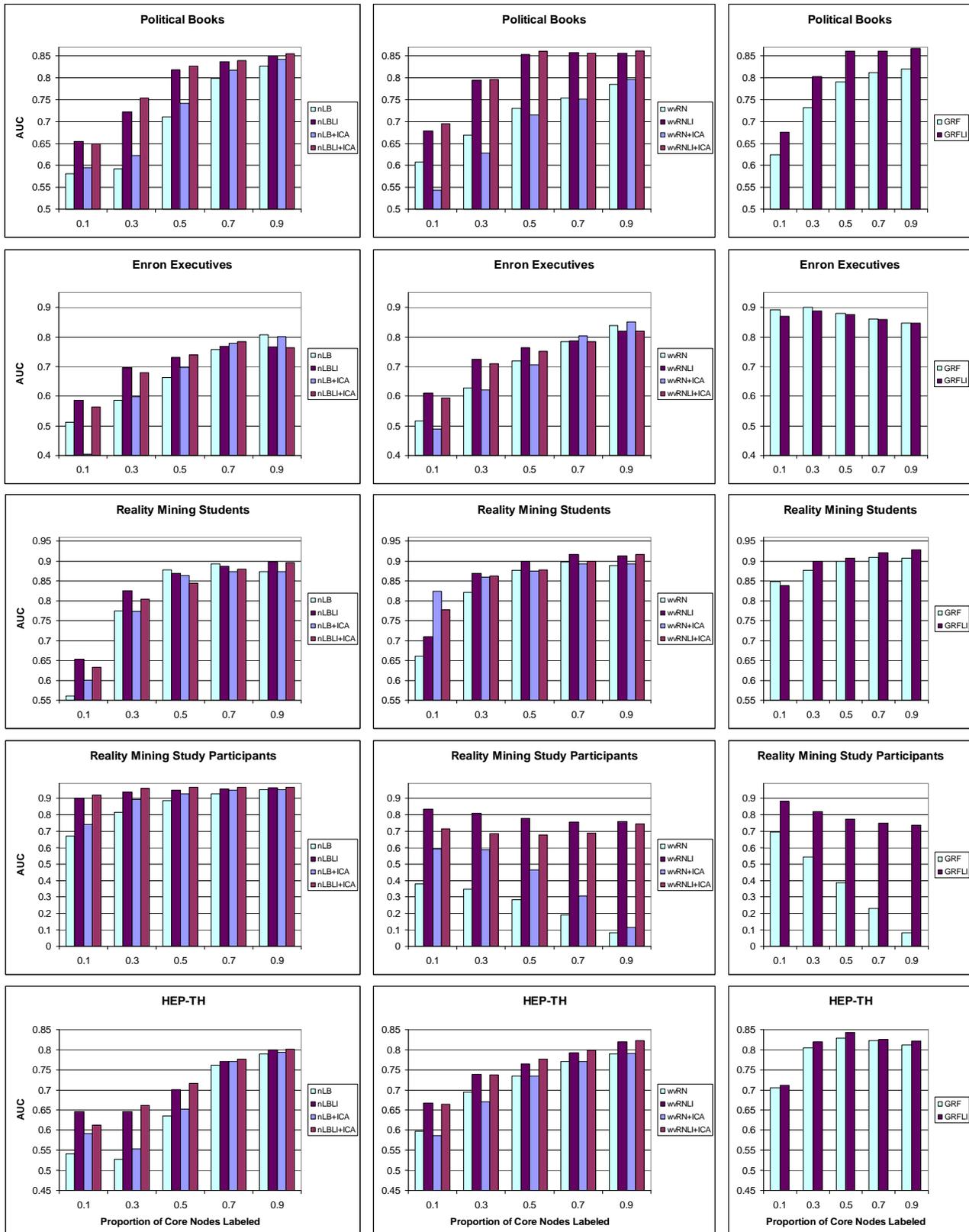


Figure 1: Classification results on political books, Enron, Reality Mining, and HEP-TH data sets. The classifiers are: *nLB*, *nLB+ICA*, *nLB+ICA+ICA*, *nLB+ICA+ICA+ICA*, *wvRN*, *wvRN+ICA*, *wvRN+ICA+ICA*, *wvRN+ICA+ICA+ICA*, *GRF*, and *GRFLI*. Results using relaxation labeling (RL) are omitted. *GRF* slightly outperforms *wvRN+RL*. *nLB+ICA* slightly outperforms *nLB+RL* overall. See Sections 4.1-2 for details.

5.2 Effects of Label-Independent Features

Figure 1 illustrates several effects of LI features. In general, the performance of the LI classifiers degrades more slowly than that of the corresponding base classifiers as fewer nodes are labeled. At $\leq 50\%$ labeled, the LI features produce a significant improvement in 36 of 45 cases. The exceptions mainly occur for GRF on Enron, RM Student, and HEP-TH, where (in most cases) we have a statistical tie. In general, the information provided by the LI features is able to make up, at least in part, for information lost due to missing labels. Note that there are three separate effects that lower performance as the number of labels decreases. (1) Fewer labels available for inference lead to lower quality LD features at inference time, but do not impact the quality of LI features. (2) Fewer labels at training time mean that (labeled) training examples have fewer labeled neighbors. This impacts the quality of the LD features available at training time and the quality of the resulting model. LI features are not affected. (3) Fewer labels mean less training data. This impacts model quality for both LD and LI features. Note that wvRN and GRF are affected only by 1, since they do not rely on training data.

In general, the LI models outperform the corresponding base models, leading to significant improvements in 49 out of 75 cases across all proportions of labeled data. There is only one case where the use of LI features significantly degrades performance: using *GRF* on the Enron task at ≤ 0.3 labeled. The *GRF* classifier does so well on this task that the LI features simply add complexity without additional predictive information. However, the degradation here is small compared to gains on other tasks.

Another effect demonstrated in Figure 1 is the interaction between LI features and label propagation (i.e., ICA or GRF). In several cases, combining the two significantly outperforms either on its own (e.g., GRFLI on political books and the RM tasks). However, the benefit is not consistent across all tasks.

The improved performance due to LI features on several tasks at 90% labeled (i.e., political books, both RM tasks) suggests that LI features can provide information above and beyond that provided by class labels. Political books and RM study are the only data sets fully labeled to begin with. This indicates that LI features may have more general applicability beyond sparsely labeled data.

Figure 2 shows the sensitivity of classifiers to the specific nodes that are initially labeled. For each classifier and task, we measure variance in AUC across 30 trials. For each trial, a different 50% of nodes is labeled. ICA has very little impact on the sensitivity of nLB to labeling changes. However, the LI features decrease the labeling sensitivity of nLB dramatically for all but one data set. The results for wvRN are qualitatively similar. LI features also decrease sensitivity for GRF in most cases. Since GRF has low sensitivity to begin with, the improvements are less dramatic. The observed reduction in label sensitivity is not surprising since LI features do not rely on class labels. However, it suggests that LI features make classifiers more stable. So, even in cases where average classifier performance does not increase, we expect an increase in the worst case due to the use of LI features.

5.3 Effects of Specific LI Features

To understand which LI features contribute to the observed performance gains, we re-ran our experiments using subsets of the LI features. We used logistic regression with different combinations

of the four LI features: each alone (4), leave one out (4), degree-based features (1), non-degree-based features (1), and all features (1). This yields 11 classifiers. We present results for 50% of nodes labeled. Results for other proportions labeled are similar.

Figure 3 shows AUC using each LI feature alone vs. all features together. This demonstrates the utility of each feature in the absence of any other information. Figure 4 shows the increase in AUC due to adding the specified feature to a classifier that already has access to all other LI features. The y-axis is the AUC of a classifier that uses all LI features minus the AUC of a classifier that uses all except the specified feature. This demonstrates the power of each feature when combined with the others.

All features appear to be useful for some tasks. Clustering coefficient is the least useful overall, improving AUC slightly on two tasks and degrading AUC slightly on three. For all tasks, a combination of at least three features yields the best results. Interestingly, features that perform poorly on their own can be combined to produce good results. On the RM student task, node count,

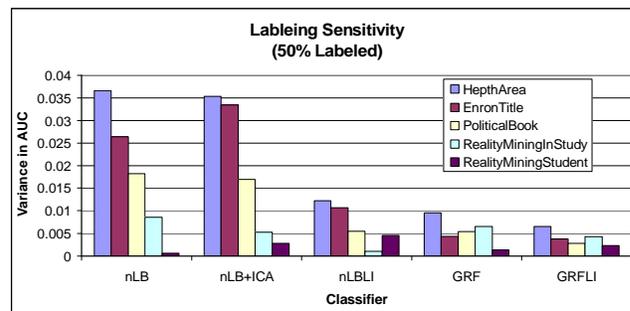


Figure 2: Sensitivity of classifiers to specific assignments of 50% known labels across data sets.

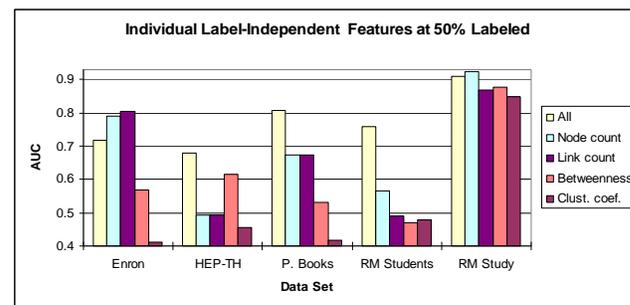


Figure 3: Performance of LI features in isolation.

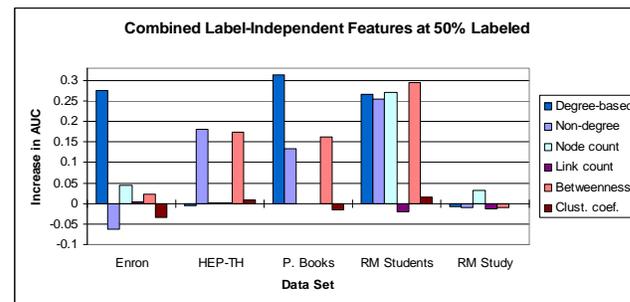


Figure 4: Performance of LI features in combination. Degree-based features are node and link count. Non-degree features are betweenness and clustering coefficient.

betweenness, and clustering coefficient produce AUCs of 0.57, 0.49, and 0.48 alone, respectively. When combined, these three produce an AUC of 0.78. Betweenness, which performs below random (AUC < 0.5) on its own, provides a boost of 0.32 AUC to a classifier using node count and clustering coefficient.

For most tasks, performance improves due to using all four LI features. On Enron, however, clustering coefficient appears to mislead the classifier to the point where it is better to use either node or link count individually than to use all features. This is one case where we might benefit from a more selective classifier. Figure 5 compares logistic regression with a random forest classifier [1], both using the same four LI features. As expected, random forest is better able to make use of the informative features without being misled by the uninformative ones.

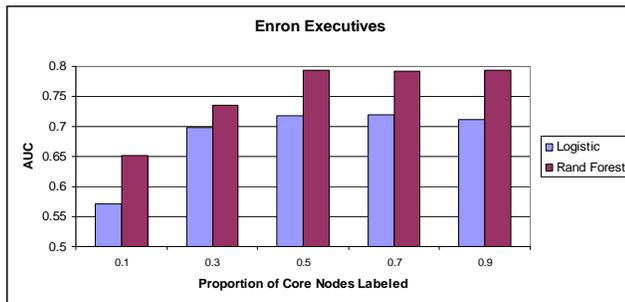


Figure 5: Comparison of logistic regression and random forest classifiers with all four LI features.

6. CONCLUSION

We have examined the utility of label-independent features in the context of within-network classification. Our experiments reveal a number of interesting findings: (1) LI features can make up for large amounts of missing class labels; (2) LI features can provide information above and beyond that provided by class labels alone; (3) the effectiveness of LI features is due, at least in part, to their consistency and their stabilizing effect on network classifiers; (4) no single label-independent feature dominates, and there is generally a benefit to combining a few diverse LI features. Lastly, we observed a benefit to combining LI features with label propagation, although the benefit is not consistent across tasks.

7. ACKNOWLEDGMENTS

We would like to thank Luke McDowell for his insightful comments. This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract No. W-7405-ENG-48 and No. DE-AC52-07NA27344 (LLNL-CONF-405028).

8. REFERENCES

- [1] L. Breiman, "Random forests," *Machine Learning*, 45(1), 2001, pp. 5-32.
- [2] S. Chakrabarti, B. Dom, and P. Indyk, "Enhanced hypertext categorization using hyperlinks," In *Proc. of ACM SIGMOD Int'l Conf. on Management of Data*, 1998, pp. 307-318.
- [3] B. Gallagher and T. Eliassi-Rad, "An examination of experimental methodology for classifiers of relational data," In *Proc. of the 7th IEEE Int'l Conf. on Data Mining Workshops*, 2007, pp. 411-416.
- [4] B. Gallagher, H. Tong, T. Eliassi-Rad, C. Faloutsos, "Using ghost edges for classification in sparsely labeled networks," In *Proc. of the 14th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, 2008.
- [5] L. Getoor, N. Friedman, D. Koller, and B. Taskar, "Learning probabilistic models of link structure," *Journal of Machine Learning Research*, 3, 2002, pp. 679-707.
- [6] D. Jensen, J. Neville, and B. Gallagher, "Why collective inference improves relational classification," In *Proc. of the 10th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, 2004, pp. 593-598.
- [7] Q. Lu and L. Getoor, "Link-based classification," In *Proc. of the 20th Int'l Conf. on Machine Learning*, 2003, pp. 496-503.
- [8] S. Macskassy and F. Provost, "Classification in networked data: a toolkit and a univariate case study," *Journal of Machine Learning Research*, 8, 2007, pp. 935-983.
- [9] L. McDowell, K.M. Gupta, D.W. Aha, "Cautious inference in collective classification," In *Proc. of the 22nd AAAI Conference on Artificial Intelligence*, 2007, pp. 596-601.
- [10] J. Neville, D. Jensen, L. Friedland, and M. Hay, "Learning relational probability trees," In *Proc. of the 9th ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining*, 2003, pp. 625-630.
- [11] J. Neville, D. Jensen, and B. Gallagher, "Simple estimators for relational Bayesian classifiers," In *Proc. of the 3rd IEEE Int'l Conf. on Data Mining*, 2003, pp. 609-612.
- [12] J. Neville and D. Jensen, "Leveraging relational autocorrelation with latent group models," In *Proc. of the 5th IEEE Int'l Conf. on Data Mining*, 2005, pp. 322-329.
- [13] J. Neville and D. Jensen, "Relational dependency networks," *Journal of Machine Learning Research*, 8, 2007, pp. 653-692.
- [14] M.E.J. Newman, "The structure and function of complex networks," *SIAM Review*, 45, 2003, pp. 167-256.
- [15] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine, Special Issue on AI and Networks*, forthcoming Fall 2008.
- [16] L. Singh, L. Getoor, and L. Licamele, "Pruning social networks using structural properties and descriptive attributes," In *Proc. of the 5th IEEE Int'l Conf. on Data Mining*, 2005, pp. 773-776.
- [17] B. Taskar, P. Abbeel, and D. Koller, "Discriminative probabilistic models for relational data," In *Proc. of the 18th Conf. on Uncertainty in AI (UAI)*, 2002, pp. 485-492.
- [18] X. Zhu, Z. Ghahramani, and J. Lafferty, "Semi-supervised learning using Gaussian fields and harmonic functions," In *Proc. of the 20th Int'l Conf. on Machine Learning (ICML)*, 2003, pp. 912-919.
- [19] X. Zhu, "Semi-supervised learning literature survey," UW-Madison Technical Report CS-TR-1530, December 2007, http://pages.cs.wisc.edu/~jerryzhu/pub/ssl_survey.pdf.