Combining Semi-supervised Learning and Relational Resampling for Active Learning in Network Domains

Ankit Kuwadekar AKUWADEK@PURDUE.EDU
Jennifer Neville NEVILLE@CS.PURDUE.EDU
Computer Science Department, Purdue University, West Lafayette, IN 47907 USA

Abstract

Recent work in statistical relational learning has demonstrated the effectiveness of network-based classification methods, which exploit relational dependencies among instances to improve predictions. These methods have been applied in a broad range of domains, from bioinformatics to fraud detection. Although labeled training examples can be costly to acquire in these domains, there has been little work focusing on *active learning* techniques that can identify the most beneficial (network) instances to label for learning. Past work has mainly focused on learning with a fixed set of labeled nodes—either from a fully labeled or partially labeled network. Recent work on *active inference* has demonstrated that labeling efforts can be used to improve collective inference, based on the network topology (Bilgic & Getoor, 2008; Macskassy, 2009). However, it is more difficult to incorporate network characteristics into the learning process. Two primary issues are: (1) how to learn an accurate model from a partially-labeled network, and (2) how to get an accurate assessment of uncertainty from the model. In this work, we address these issues by combining semi-supervised learning with relational resampling and a utility metric based on network variance. We evaluate our approach on synthetic and real-world networks and show that it results in faster learning compared to several alternative baselines.

Appearing in *Proceedings of the Budgeted Learning Workshop, 27th International Conference on Machine Learning*, Haifa, Israel, 2010.

1. Introduction

Recent work in the area of statistical relational learning has demonstrated that using network information to learn joint models, for *collective inference*, can often result in significant performance gains (Getoor & Taskar, 2007). These models have been broadly applied in a wide range of domains, including bioinformatics, fraud detection, and social network analysis. In many of these applications, the observed improvement in classification accuracy is primarily due to the methods' ability to identify and exploit dependencies among instances.

This past work in relational learning has mainly focused on learning models from a fixed set of labeled training nodes—either with a fully labeled or partially labeled network. The implicit assumption has been that labeled training data is either cheap to obtain, or that it is not possible to obtain additional labels for training. However, in many real-world domains, while it may be cheap to acquire the network topology (e.g., email network in an organization), it may be costly to acquire node labels for training (e.g., assessment of which employees are involved in fraud). In these situations, *active learning* methods could be used to learn accurate models while minimizing labeling costs.

However, in network domains where instances are dependent, the utility of labeling an instance may depend on more than just the properties of the instance itself. Indeed, recent work on *active inference* has shown that selectively querying for node labels based on network connectivity can significantly improve the collective inference process (Bilgic & Getoor, 2008; Macskassy, 2009). Despite these findings, there has been little work focusing on how network dependencies can impact the *learning* process other than the preliminary work of Bilgic & Getoor (2009).

There are two main challenges to incorporating network characteristics into an active learning process for relational domains. First, it is difficult to learn accu-

rate models from partially-labeled networks where the number of labels changes over the learning process. If learning methods ignore the unlabeled portion of the network, then as the amount of labeled data increases, the structure of the network can change significantly (e.g., node degree increases). Second, it is difficult to estimate prediction uncertainty for an instance in network domains where collective inference methods are used. This is both because there are dependencies among the inferences of neighbors, and because it is difficult to tease apart the uncertainty due to learning and uncertainty due to collective inference.

In this paper, we present a Relational Active Learning (RAL) method that addresses these issues by combining semi-supervised learning with relational resampling and a utility metric based on network variance. Our approach explores the example space more effectively by considering the nature of an instance's immediate neighborhood instead of favoring only high degree nodes in expected high-density regions. We evaluate our approach on across-network classification tasks, with both synthetic and real-world datasets, and show that it results in faster learning compared to several alternative algorithms.

2. Background and Related Work

2.1. Active Learning

Active learning is a learning strategy for domains where it is costly to acquire labeled training examples, but unlabeled examples are plentiful. The methods typically economize the number of unlabeled data that needs to be labeled in order to learn an accurate model. The objective is to selectively label examples that will result in the greatest reduction in the model's generalization error (Saar-Tsechansky & Provost, 2004).

The majority of active learning techniques use a utility-based approach, where instances are chosen for labeling based on a calculation of their expected contribution (i.e., utility) to the accuracy of the learned model. Generally, a utility-based active learning technique starts with a pool of labeled examples (typically empty: $L = \emptyset$), a pool of unlabeled examples (typically the dataset: $UL = D$), and an inducer I (e.g., an SVM learner). Then the algorithms proceed as follows:

- Compute the *utility* $u(i)$ for each unlabeled example $i \in UL$.
- Based on the utility scores (e.g., maximizing), choose one or more unlabeled examples to label (e.g., $L = L \cup \{i\}$).
- Apply I to learn a new model from L, repeat.

Most active learning methods differ chiefly in their choice of utility function $u(.)$. For example, Tong & Koller (2002) use the expected reduction in version space, while Roy & Mccallum (2001) use the expected reduction in future error. Seung et al. (1992) use a committee of classifiers and define the utility to be the amount of disagreement between members of this committees. Saar-Tsechansky & Provost (2004) subsample the training data, learn a model on each subsample, and define the utility of an instance as the variance of the predictions of the set of classifiers.

Although there is a broad set of methods for active learning that have been successfully applied to high-cost or resource-constrained domains, the majority of this work has focused on independent and identically distributed (i.i.d.) data. As a result, the utility measures are calculated for each example *independently*. In relational and network domains, where the i.i.d. assumption does not hold, the benefit of labeling an instance can go beyond the instance itself, as it may improve the predictions about neighbors in the network. Thus, active learning methods need to consider the network structure in the utility calculation.

In addition, many active learning techniques involve training the classifier only using the labeled training set (L), while passive (i.e., non-active) semi-supervised learning has been shown to improve performance, by training the classifier with both the labeled and the unlabeled data ($L \cup UL$) (Towell, 1995; Blum & Mitchell, 1998; Goldman & Zhou, 1998; Zhang & Oles, 2000). As mentioned above, in relational contexts, the connectivity of the instances provides additional information to the model, which may indicate a potential benefit to considering the unlabeled data during learning. Indeed, recent work on *semi-supervised relational learning* has been shown to improve learning in a passive context, particularly when there are only a moderate number of labeled examples in the network (Xiang & Neville, 2008).

In this paper, we will outline and investigate a novel active learning method for network domains that combines semi-supervised learning with a network based utility measure.

2.2. Relational Label Acquisition Methods

Recent work in the statistical relational learning community, has just begun to explore the idea of active labeling for both learning and collective inference. Research on *active inference* has focused on acquiring labels that will improve the accuracy of collective inference, by considering properties of the network structure (Bilgic & Getoor, 2008; Macskassy, 2009). No-

tably, these methods only query for labels during the inference process—either the model is learned from a fixed set of labels (Bilgic & Getoor, 2008) or the dependencies in the data are not learned (Macskassy, 2009).

Recently, Bilgic & Getoor (2009) developed an active learning method to exploit basic link-based relationships in network data. Their method uses a Nave Bayes classifier combined with probabilistic uncertainty (entropy) as the base utility function. The *network utility* of an instance is a weighted sum of the utility of the instance itself and the utility of its neighbors. Although their method incorporates the network structure into the utility measure, the method doesn't use the unlabeled data while training the classifier—the model is learned by dividing the example-space into an independent training and dependent test sets (the links between the training and test set are used for inference). This process of ignoring the unlabeled part of the data during learning can change the underlying network structure between learning and inference, which may degrade the accuracy of the learned model.

3. Relational Active Learning (RAL)

In this section, we outline our approach to active learning for relational data in detail. The discussion above points to the larger goal of this work, which is to generalize active learning for network domains and to unite both active and semi-supervised (passive) learning into one approach. To achieve our goal, we introduce a novel learning method that combines semi-supervised learning with a network based utility measure.

The primary differences between a conventional active learning approach and the *Relational Active Learning* (RAL) method are the following:

- The inducer I is a relational learning algorithm.
- The utility function $u(.)$ considers the network structure when calculating the benefit of labeling each instance.

In this work, we consider the relational dependency network (RDN) model (Neville & Jensen, 2007) as the inducer I. To apply the RDN in a semi-supervised setting we use the pseudolikelihood EM method of Xiang & Neville (2008). Then we will propose a novel network-based utility measure for $u(.)$. We describe each of these in detail below.

3.1. General RAL Algorithm

In this work, we consider *across-network* relational learning tasks, where we learn a model from a (partially) labeled training network G_{tr}, and apply the model for collective inference on a separate (i.e., disjoint) testing network G_{te}. The input to the RAL algorithm is $G_{tr} = (V_{tr}, E_{tr})$, which initially contains no labeled examples (i.e., $L = \emptyset$, $UL = V_{tr}$), and an RDN inducer I. Then, given a fixed labeling budget B (which limits the number of instances we can label), the algorithm proceeds as follows:

1. WHILE $B \geq 0$:
   (a) Using $L \cup UL$, apply inducer I to learn an ensemble of m models $\mathbf{M} = \{M_1, ..., M_m\}$.
   (b) Using $\mathbf{M}$, compute the utility $u(i)$ for each unlabeled example $i \in UL$.
   (c) Randomly select k examples in proportion to their utility scores (i.e., $p_i = u(i)/\sum_j u(j)$).
   (d) Add the selected examples $\mathbf{S}_k$ to the label set $L = L \cup \mathbf{S}_k$, $UL = UL - \mathbf{S}_k$.
   (e) $B = B - k$.
2. Apply inducer I to learn a model M with $L \cup UL$, return M.

Therefore the two main components of the algorithm are the method by which we use semi-supervised learning to learn an ensemble of models (step 1a) and the utility measures (step 1b). We discuss each of these in detail below.

3.2. Semi-supervised Ensemble Learning

Our RAL algorithm takes a similar approach as the Bootstrap LV approach of Saar-Tsechansky & Provost (2004). The Bootstrap LV method uses *resampling* from the label set L to create multiple training sets $\mathbf{L}' = \{L'_1, ..., L'_m\}$ for learning. The inducer I is applied to each training set in $\mathbf{L}'$ to learn an *ensemble* of models $\mathbf{M} = \{M_1, ..., M_m\}$. Then the ensemble is applied to the examples in UL, resulting in a set of predictions for each example. These sets of predictions can be used to calculate the prediction variance for each instance, and then the utility measure ranks the instances in descending order by variance.

There are two key challenges to developing a similar approach for network datasets. First, we need a method to sample *with replacement* from a network, while adequately preserving the link structure of the network. To this end, we apply a recently developed method for resampling relational data (Eldardiry & Neville, 2008) (described below).

The second challenge is to get an accurate estimate of prediction variance, given the unique characteristics of network datasets. If we ignore the unlabeled part

of the network during learning, this can significantly change the structure of the training examples (e.g., there is much lower degree at first). This may bias the models—due to the fact that the labeled (training) examples may appear to be drawn from a different distribution than the unlabeled examples. This is particularly difficult in collective inference settings, where we are trying to *learn* the dependencies among neighboring nodes in the network. Semi-supervised learning is one approach to offset this issue, since both labeled and unlabeled examples will be used during learning. For this purpose, we will apply another recently developed semi-supervised relational learning method, which uses pseudolikelihood EM (PLEM) for estimation in RDNs (Xiang & Neville, 2008).

Relational Resampling

The Relational Subgraph Resampling (RSR) method (Eldardiry & Neville, 2008) uses a subgraph sampling approach to preserve the local relational dependencies while generating a pseudosample with sufficient global variance. It has been shown to result in significantly higher accuracy, compared to an i.i.d. resampling approach, when applied to estimate the variance of feature scores in network datasets (Eldardiry & Neville, 2008).

The first phase of the algorithm selects subgraphs based on snowball sampling. It repeatedly selects a subgraph of size b via breadth-first search from a randomly selected seed node. The second phase links up the selected subgraphs. The aim is to preserve the local relational dependencies among instances in each subgraph while randomizing the dependencies across the set of selected subgraphs, in order to generate a pseudosample with sufficient global variance.

Due to the varied link structure of relational data, there will be a large number of nodes on the periphery of the selected subgraphs. If the peripheral nodes are missing a significant portion of their neighbors, this could bias the properties of the sample. To deal with this issue, the RSR algorithm links up the peripheral nodes in the selected subgraphs, while attempting to maintain the global graph properties and attribute dependencies of the original data. More specifically, the relational autocorrelation is maintained by maximizing attribute similarity between nodes as they are linked, while the link structure is maintained by considering the neighborhood similarity when linking nodes.

Semi-supervised RDNs

RDNs typically use pseudolikelihood estimation to learn a model from a fully labeled network. For each data instance x_i, pseudolikelihood models use a local conditional probability distribution (CPD) to represent the conditional probability of the label value v_{x_i}, given its linked nodes, i.e. $P(v_{x_i}|P_a(x_i))$. However, the local CPDs are not required to factor the full joint distribution. Instead of maximizing likelihood during learning, we maximize the following pseudolikelihood: $PL(X;\theta) = \prod_{x_i \epsilon X} p(v_{x_i}|P_a(x_i);\theta)$.

The pseudolikelihood EM (PLEM) approach to learning RDNs learns a joint model of labeled and unlabeled data in the network (Xiang & Neville, 2008) . It has been shown that when there is moderate number of labeled examples, the PLEM approach achieves significantly higher accuracy than other within-network relational learning techniques. The implication of this for active learning is that the model will be more stable and it will produce more accurate estimates of variance in partially-labeled networks.

In conventional expectation maximization (EM) approaches to semi-supervised learning, the full data likelihood $P(X|Z,\theta)$ is considered, where Z is the set of unlabeled data. EM consists of two alternating steps:

- **E-Step:** Evaluate $p(Z|\theta^{old})$
- **M-step:** Update the estimator: $\theta^{\text{new}} = \arg\max_\theta \sum_Z p(Z|X,\theta^{old}) \log p(X,Z|\theta)$

In PLEM, this update equation is rewritten using the pseudolikelihood of the complete data (X,Z) rather than the full likelihood:

$$\theta^{\text{new}} = \arg\max_\theta \sum_Z p(Z|X,\theta^{\text{old}}) \sum_{x_i \epsilon X} \log p(v_{x_i}|P_a(x_i);\theta)$$

The complete data pseudolikelihood is defined as the product of CPDs of the observed instance labels, but conditioned on *all* related instances (i.e., $\text{Pa}(x_i)$ contains both the labeled and unlabeled related instances of x_i). Therefore, the M-step can be interpreted as a *collective learning* method. By contrast, in a disjoint learning approach, we only perform parameter estimation once, in which each CPD factor of the pseudolikelihood function is conditioned only on the labeled related instances, and hence the MPLE may be biased when only a few related instances are labeled.

3.3. Network-Based Utility Scores

To understand how different ways of calculating utility affect performance, we will outline and compare five utility metrics for network datasets.

Random

This is a baseline measure, which selects nodes uniformly at random from UL to label: $u_{RAND}(i) = \frac{1}{|UL|}$.

Degree

This measure is included to measure the effectiveness of labeling based on graph structure alone. We define the utility to be the degree of a node: $u_{DEG}(i) = deg(i)$. Thus, the nodes with maximum degree are more likely to be selected for labeling.

Local Variance

The local variance approach is similar to the utility measures that are widely used in current active learning algorithms (for i.i.d. data). We simply calculate the prediction variance for each instance *independently*, over the set of ensemble models: $u_{IND}(i) = Var([P_1(i), P_2(i), ..., P_m(i)])$, where $P_j(i) = p_{M_j}(y_i = +)$ refers to the predicted (marginal) probability that model M_j associates with instance i belonging to class $+$ (assuming a binary classification task).

Relational Variance

Based of the local estimates of variance, we can define a relational variance measure that simply sums the variance of the node i and all its neighbors: $u_{REL}(i) = u_{IND}(i) + \sum_{j \in \mathcal{N}_i} u_{IND}(j)$.

To approximate the method of Bilgic & Getoor (2009), we define a link-based utility measure that first selects a degree d_s randomly based on u_{DEG}, then from among the instances with degree d_s, chooses the instance that maximizes u_{REL} to label. We refer to this process as u_{LB} for *link-based* selection. Instances are chosen first probabilistically in proportion to their degree, and then the selection is refined to maximize relational variance.

Weighted Density Disagreement

For our approach, we propose a novel network-based utility measure, based on the idea that the most valuable unlabeled examples have the following properties:

- They lie in high-density (unlabeled) regions.
- Their predictions disagree the least with their immediate neighborhood.
- Their predictions disagree the most with the mean prediction of the ensemble.

We define the *weighted density disagreement* (WDD) utility measure as the product of the divergence of the instance from the overall mean predictions of the ensemble, and the sum of the disagreement between the predictions of the instance with those of its neighbors. To compute the WDD utility of an unlabeled instance, we use the following method:

1. Compute the prediction for each instance i over the ensemble of m models.
2. Let $P_j(i) = p_{M_j}(y_i = +)$ refer to the predicted (marginal) probability that model M_j associates with instance i belonging to class $+$.
3. Let $P_j(UL) = \frac{1}{|UL|} \sum_{i \in UL} P_j(i)$ refer to the average probability that model M_j predicts for all instances in the unlabeled set UL.
4. To represent the certainty of our prediction for an unlabeled example i, we use the Kullback-Leibler (KL) divergence between the ensemble predictions for i and the average predictions for unlabeled instances: $v(i) = KL[P(i)||P(UL)] = \sum_{j \in m} P_j(i) \times \log \frac{P_j(i)}{P_j(UL)}$.
5. We then define WDD for i as follows:

$$u_{WDD}(i) = v(i) \times \sum_{j \in \mathcal{N}_i} e^{-KL[P(i)||P(j)]}$$

Our WDD measure has two parts. The first part, $v(i)$, considers the ensemble of predictions for a node i in isolation. We measure how much the predictions *diverge* from the current average predictions for the unlabeled nodes in the graph, which means nodes with highly confident predictions (e.g., average probabilities close to 0 or 1) will maximize WDD. The second part of the measure considers the neighbors of node i. In order to maximize WDD, the neighbors predictions should by close to the set of predictions for node i (e.g., disagree least). The overall result is that WDD will favor nodes with highly confident predictions, which also lie in a neighborhood with many similar predictions. We note that this approach is in contrast to many other utility metrics for i.i.d. settings, which favor nodes with high uncertainty. In relational settings, where inferences are propagated throughout the network during learning, it is more beneficial to label nodes with highly consistent neighborhoods, as it will improve *both* learning and inference.

4. Experimental Evaluation

We evaluated our RAL algorithm on both synthetic and real networks, comparing our proposed approach (RAL-WDD) with several other competing baseline

methods. The experiments are intended to evaluate the benefit of (1) the network-based utility measure, and (2) the semi-supervised and resampling approach to variance estimation.

4.1. Data Sets

The synthetic datasets are generated with the latent group model described in the work of Neville & Jensen (2005). The model uses a hidden group structure to generate network data with autocorrelation. For this work, we generated graphs with 250 nodes, in groups with an average size of 25 nodes. Each node has one binary class label and three other boolean attributes. The class label has an autocorrelation level of 0.5.

The real world dataset is drawn from the Adolescent Health (AddHealth) data, which consists of survey information from middle and high schools, collected in 1994-1995. The survey questions queried for the students social networks along with myriad behavioral/academic attributes. In this paper, we consider the social networks of schools with similar autocorrelation and link patterns. The classification task is to predict whether the student has ever smoked, based on the behavior of their friends in the social network. For the experiments, we selected three similar schools, with sizes ranging from 300-500 nodes, average degree of 7-8, and autocorrelation in the range [0.25,0.35].

4.2. Methodology

We evaluate the RAL algorithm in an across-network classification setting, where we learn the model on a partially labeled training network, and apply the learned model to another network (drawn from the same distribution) for prediction. At each step of the active learning phase we chose $k = 30$ examples to label. We resampled 10 times to learn an ensemble of $m = 10$ models at each iteration. So the size of the training set increases from 30, 60, ..., 150. At each iteration, we evaluate performance of the learned model on a disjoint test set. For this evaluation, we measure area under the ROC curve (AUC) while allowing the model to see the true labels of neighbors. This gives us a measure of the *ceiling* performance of the collective inference model—which we use to focus on the quality of the learned model, rather than the collective inference process.

For the synthetic data experiments, we generated ten different synthetic datasets and used five of them for training, five for testing. For each train/test pair, we ran the experiment five times to control for random variation in labeling choices. The reported results are averaged over the 5X5=25 trials. For the AddHealth data, we repeatedly selected one school network as the training set for learning and then applied the learned model to the remaining two school networks for evaluation. For each train/test pair, we ran the experiment 25 times to control for variation, thus the reported results are averages of 3X25=75 trials.

We compare our proposed approach (RAL-WDD) with several other competing baseline methods. To evaluate the benefit of the network-based utility measure, we compare the following:

- **RAL-RAND:** This approach randomly chooses instances to label; we include it as a baseline.
- **RAL-DEG:** This approach uses only network structure for choosing instances to label; we include it to illustrate the utility of network structure without a measure of instance uncertainty.
- **RAL-IND:** This approach uses only local estimates of variance for choosing instances to label (as formulated in the Bootstrap LV (Saar-Tsechansky & Provost, 2004)); we include it to illustrate the utility of variance without a measure of network structure.
- **RAL-LB:** This approach uses a network-based measure of variance, that approximately emulates link-based active learning as introduced by Bilgic & Getoor (2009); we include it to illustrate the utility of network-based measures that do not incorporate the unlabeled instances in a semi-supervised approach.
- **RAL-WDD:** This is our proposed approach, which uses the weighted disagreement of a node with its neighbors to incorporate uncertainty and network structure.

Next, to assess the benefit of using a semi-supervised approach to variance estimation, we compare RAL-WDD to versions of the method without semi-supervised learning. In the *disjoint learning* (DL) approach, we ignore the unlabeled part of the network and simply learn RDNs from the label set alone (i.e., L). Although, we measured performance with all the various utility metrics, for clarity we only include the results from the best two measures: **DL-LB** and **DL-WDD**.

4.3. Results

Figures 1-2 plot the results on the various active learning approaches on the synthetic datasets. Figures 3-4 plot the results on the AddHealth datasets.

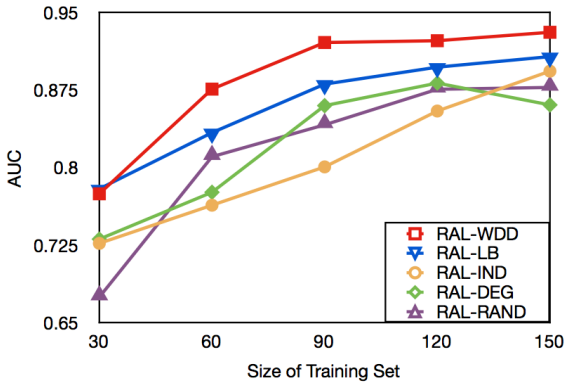

Figure 1. RAL performance on synthetic data.

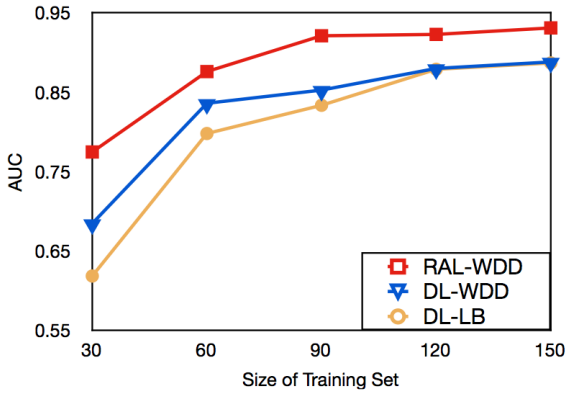

Figure 2. RAL-WDD compared to disjoint learning approaches, on synthetic data.

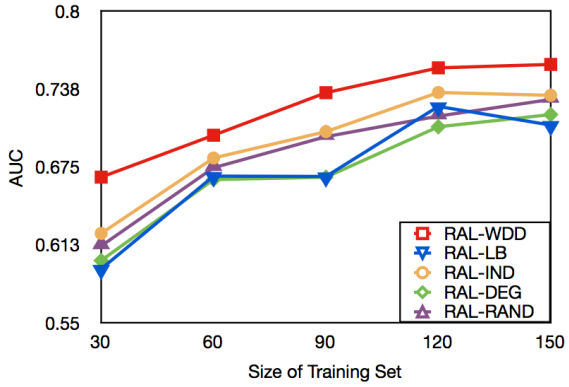

Figure 3. RAL performance on AddHealth data.

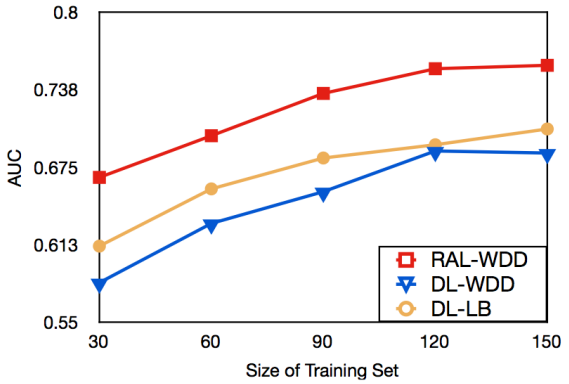

Figure 4. RAL-WDD compared to disjoint learning approaches, on AddHealth data.

Figures 1 and 3 present the evaluation of the RAL algorithm with the various utility measures on the synthetic and real datasets, respectively. From these results it is clear that RAL-WDD outperforms the other measures, by learning a more accurate model with fewer labeled nodes. In particular, we note that RAL-WDD outperforms RAL-IND by a large margin, which indicates the benefit of considering relational information in the active learning process. Notably, RAL-DEG does not perform better than RAL-RAND, which indicates that a measure based on network topology alone is not an effective approach to active learning. In addition, RAL-WDD outperforms RAL-LB on both types of data, but by a particularly large margin in the AddHealth data. This indicates that density disagreement is a more effective measure of neighborhood information, than a sum over the uncertainty of the nodes.

Figures 2 and 4 present the evaluation of the RAL algorithm compared to active learning approaches that do not use the unlabeled part of the network (i.e., disjoint learning). In both the synthetic and real datasets, RAL-WDD significantly outperforms DL-WDD. This illustrates the benefit of considering the unlabeled data during active learning. Since the same utility measure is used in both methods, the improvement in performance is due to the more accurate assessment of the variance of predictions across the models, which comes from the semi-supervised approach to learning. For a baseline comparison, we also present the accuracy of the DL-LB approach. This is quite similar to the method of Bilgic & Getoor (2009), which does not use semi-supervised learning. Again, we can see that the RAL-WDD outperforms this approach on both datasets.

We should note that the synthetic datasets have higher autocorrelation than the AddHealth dataset. This is why the overall performance of the models on the AddHealth data is lower (95% vs. 75% AUC). In Figure 3, we can see that RAL-LB performs slightly worse than RAL-IND. This indicates that relational variance estimation may be more useful when autocorrelation is higher. However, when autocorrelation is low, network-based measures are still useful—as indicated

by the improvement of the WDD measure, which considers the disagreement of a node's predictions with that of its neighbors.

5. Discussion and Conclusion

In this paper, we outlined a novel approach to active learning in relational and collective inference domains, where we combine a network-based utility measure with semi-supervised learning and relational resampling. Our experimental results show that RAL-WDD results in significantly faster learning (i.e., higher accuracy with fewer labeled examples) across a range of conditions. The key idea in our approach is to consider the nature of the neighborhood—both in selecting instances whose predictions are similar to its neighbors, and in estimating the models with semi-supervised learning.

Our experimental results illustrate the benefit of considering the similarity of an instance's predictions to that of its immediate neighborhood, when identifying instances to actively label. This network-based WDD approach results in significantly higher accuracy compared to i.i.d. active learning approaches that consider the uncertainty of an instance in isolation. Our WDD measure is also better than choosing instances that lie in high-density regions and have high degree.

The WDD measure favors nodes that have (1) highly confident predictions (i.e., diverge from the overall average predictions), and (2) neighbors with similar predictions. This is opposite to many utility metrics used in i.i.d. settings, which favor nodes with high uncertainty. Although we do not report the results here, our initial experiments using a metric that favored highly uncertain nodes resulted in poor performance. We conjecture that this is due to the use of semi-supervised relational learning, which propagates inferences during learning and may be unduly biased if nodes with less consistent neighborhoods are labeled initially. In future work, we will explore this effect in more depth, using bias-variance analysis (Neville & Jensen, 2008) to determine whether the performance improvements are due to reduction in learning or inference error.

The RAL-WDD approach can be applied to wider variety of scenarios than previous work on active learning in relational domains. The method can be applied to within-network or across-network classification tasks, and handle both unlabeled and partially-labeled networks. In future work, we will evaluate our method on additional real-world network datasets, in both a within-network and across-network scenario, comparing to Macskassy (2009) and Bilgic & Getoor (2009) more explicitly.

Acknowledgments

We thank Hoda Eldardiry and Rongjing Xiang for their help with the RSR and PLEM code. This research is supported by DARPA under contract number(s) NBCH1080005.

References

Bilgic, M. and Getoor, L. Effective label acquisition for collective classification. In *KDD'08*, 2008.

Bilgic, M. and Getoor, L. Link-based active learning. In *NIPS Workshop on Analyzing Networks and Learning with Graphs*, 2009.

Blum, A. and Mitchell, T. Combining labeled and unlabeled data with co-training. In *COLT'98*, 1998.

Eldardiry, H. and Neville, J. A resampling technique for relational data graphs. In *Proceedings of the 2nd SNA Workshop, KDD'08*, 2008.

Getoor, L. and Taskar, B. (eds.). *Introduction to Statistical Relational Learning*. MIT Press, 2007.

Goldman, S. and Zhou, Y. Enhancing supervised learning with unlabeled data. In *ICML'08*, 1998.

Macskassy, S. A. Using graph-based metrics with empirical risk minimization to speed up active learning on networked data. In *KDD"09*, 2009.

Neville, J. and Jensen, D. Leveraging relational autocorrelation with latent group models. In *ICDM'05*, 2005.

Neville, J. and Jensen, D. Relational dependency networks. *The Journal of Machine Learning Research*, 8, 2007.

Neville, J. and Jensen, D. A bias-variance decomposition for collective inference models. *Machine Learning Journal*, 2008.

Roy, N. and Mccallum, A. Toward optimal active learning through sampling estimation of error reduction. In *ICML'01*, 2001.

Saar-Tsechansky, M. and Provost, F. Active sampling for class probability estimation and ranking. *Machine Learning*, 54(2):153–178, 2004.

Seung, H. S., Opper, M., and Sompolinsky, H. Query by committee. In *COLT'92*, 1992.

Tong, S. and Koller, D. Support vector machine active learning with applications to text classification. *The Journal of Machine Learning Research*, 2:45–66, 2002.

Towell, G. Using unlabeled data for supervised learning. In *NIPS*, 1995.

Xiang, R. and Neville, J. Pseudolikelihood em for within-network relational learning. In *ICDM'08*, 2008.

Zhang, T. and Oles, F. The value of unlabeled data for classification problems. In *ICML'00*, 2000.