Pseudolikelihood EM for Within-Network Relational Learning

Rongjing Xiang
Department of Computer Science
Purdue University
rxiang@cs.purdue.edu

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

ABSTRACT

In this work, we study the problem of *within-network* relational learning and inference, where models are learned on a partially labeled relational dataset and then are applied to predict the classes of unlabeled instance in the same graph. Recent work in statistical relational learning has considered three alternative approaches for this setting: independent learning with independent inference, independent learning with collective inference, and collective learning with collective inference. Here *independent* refers to techniques that ignore the unlabeled data and *collective* refers to techniques that jointly consider the labeled and unlabeled data. Models from each of these categories has been employed previously in different settings, but to our knowledge there has been no systematic investigation comparing models from the three categories. In this paper, we develop a novel pseudolikelihood EM method that facilitates *collective learning* and *collective inference* on partially labeled networks. We then compare this method to competing methods from the other categories on both synthetic and real-world data. We show that there is a region of performance, when there is a moderate number of labeled examples, where the pseudolikelihood EM approach achieves significantly higher accuracy. However, when there are few, or many, labeled examples, all three approaches achieve similar performance—indicating that a simple approach to independent learning and inference could be the best choice in many situations.

1. INTRODUCTION

Recent work in statistical relational learning has focused on two different types of modeling tasks. The first, which we will call *across-network learning*, aims to generalize across relational data graphs—models are generally learned on a fully-labeled training graph that is disjoint from the test graph. The second, which we will call *within-network learning*, aims to generalize within a single relational data graph—models are generally learned on partially labeled subgraphs and then applied to the unlabeled part of the graph.

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.

In this paper, we focus on the second task of within-network learning. In many real world applications, relational learning tasks fall naturally into the within-network classification setting. For example, in the task of research paper classification, new papers to be classified usually have citation links to papers in the past whose topics are known. Similarly, in fraud detection, brokers whose fraud status is yet to be determined might associate with other brokers who have already been identified as fraudulent or not.

Many recently developed relational learning techniques have implicitly assumed that the training data are fully labeled. These methods have typically been applied in across-networks settings (e.g., a model to predict web page topic is learned from a set of fully labeled websites and then applied to predict the topics of webpages in a separate website). However, when relational learning techniques are applied to within-network settings the data are not fully labeled. A common approach to dealing with this is to ignore or break the the relational ties between labeled and unlabeled instances. For example, a model learned to predict movie success at time t, would consider all movies (and associated actors, directors and producers) made before t for learning, but the movies made at time t would be ignored. We call this approach *independent* learning, because even though there are links between the labeled training data ($t' > t$) and the unlabeled test set (t), the unlabeled data are not incorporated into the learning phase.

Although previous work in relational learning has typically ignored unlabeled data during learning, many approaches incorporate both labeled and unlabeled data during inference. These *collective inference* techniques jointly infer the class labels of unlabeled instances based on links to both labeled and unlabeled data (see e.g., [9]). Collective inference models have been shown to produce more accurate predictions than *independent inference* models, which ignore links to unlabeled instances and use conditional inference for each instance independently [7]. The success of collective inference models is due to the presence of autocorrelation among the class labels of related instances [6]. When relational data exhibit autocorrelation, inferences about one instance can be used to improve inferences about other related instances.

In within-network settings, when autocorrelation is present, there will be correlation between the labeled and unlabeled instances in the dataset. *Collective learning* techniques, which incorporate the unlabeled data into the learning phase,

may be able to more fully exploit the autocorrelation dependencies in the data by modeling dependencies throughout the data graph (and not just the labeled subgraph). This aspect of within-network learning, has not been fully explored in previous work. One exception is the work of Taskar et al. [17], in which collective learning is studied in the context of probabilistic relational models (PRMs). However, a limitation of their approach is that the models cannot represent arbitrary autocorrelation dependencies due to acyclicity constraints [11]. In the present work, we develop an alternative approach to within-network collective learning. We propose a pseudolikelihood expectation maximization (PL-EM) algorithm to facilitate simultaneous collective learning and collective inference. Expectation maximization (EM) is a well-studied algorithm for learning with incomplete data [2]; psuedolikelihood models have previously seen successful applications in relational learning [11]. The combination of these two makes possible a practically tractable approximation to maximum likelihood learning when there are both labeled and unlabeled instances in the network.

The rest of this paper is organized as follows. We start by characterizing different learning and inference approaches to within-network tasks. Then we formulate the PL-EM algorithm for within-network relational learning, and outline the specifics of collective learning and collective inference. We present experimental results on both synthetic data and real-world data. Three methods are considered: independent learning with independent inference, independent learning with collective inference, and collective learning with collective inference (i.e., the PL-EM approach). We evaluate these methods at different sizes of labeled instances, and show that the PL-EM approach performs substantially better than the other two when the number of labeled instance is moderate (i.e., not sparse or abundant).

2. WITHIN-NETWORK RELATIONAL LEARNING

An within-network relational learning task can be specified by a single data graph in which part of the instances are labeled and the task is to infer the labels of the remaining instances. An example is shown in Figure 1(a). A statistical relational classifier learns a model from the data graph, and applys it to infer the labels of the unlabeled data. Existing learning and inference approaches can be divided into different categories based on how they make use of the labeled and unlabeled data in the graph.

A classifier performs *independent learning* if it only uses labeled data during the learning phase. A number of simple relational classifiers fall into this category, e.g., probability relational trees (RPT) [12], relational bayes classifier (RBC) [13], structural logistic regression [15]) and ACORA [14]. For example, in Figure 1(b), the links to unlabeled instances in the data graph are removed (i.e., the instance are ignored), and the conditional models are learned only from labeled related instances.

Alternatively, a classifier performs *collective learning* if it uses both labeled and unlabeled data during the learning phase. The work of Taskar et al. [17], which uses EM for transductive learning of PRMs fall in this category. Figure 1(c) depicts a way of collective learning, the classifier attempts to learn a conditional model from each labeled instances given all its labeled and unlabeled neighbors.

Similarly, inference techniques for relational data generally fall into two types: *independent inference* and *collective inference*. As illustrated by Figure 1(d) and 1(e), an independent inference approach breaks the relational ties among unlabeled instances, while a collective inference approach makes use of both links between labeled and unlabeled instances, and links among unlabeled instances. Most simple relational classifiers (e.g., RPT, RBC, structural logistic regression, and ACORA) perform independent inference by applying the learned model to each unlabeled instance independently (i.e., inferring the label of each unlabeled instance given the observed attributes of neighboring instances). More sophisticated relational classifiers employ collective inference to jointly exploit the autocorrelation among unlabeled instances. Examples include probabilistic relational models [4], relational Markov networks [16] and relational dependency networks [11].

3. ALGORITHMS

In this section, we discuss our approach to within-network relational learning in detail. We begin with the relational probability tree algorithm, which on its own can be used for independent learning and independent inference. Then we describe psuedolikelihood models that use RPTs for independent learning and collective inference. Finally we formulate the PL-EM algorithm, which provides a unified framework for collective learning and collective inference.

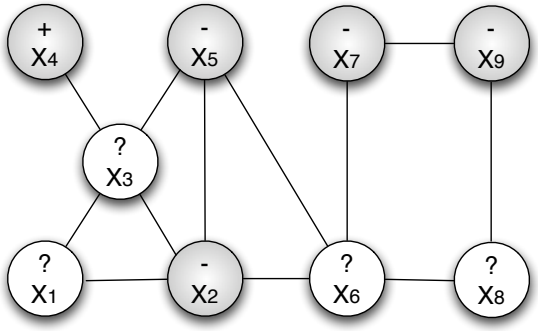
3.1 Independent learning and inference

In this work, the relational probability tree (RPT) [12] is used as the baseline relational classifier for independent learning with independent inference. The RPT algorithm extends the well-known decision tree classifier to a relational setting. For each data instance, it dynamically aggregates the values of attributes on heterogeneous sets of neighboring nodes to create homogeneous relational feature values. The RPT learning algorithm performs feature selection via chi-square statistics, and adjusts for biases towards particular features due to degree disparity and autocorrelation in relational data [6].

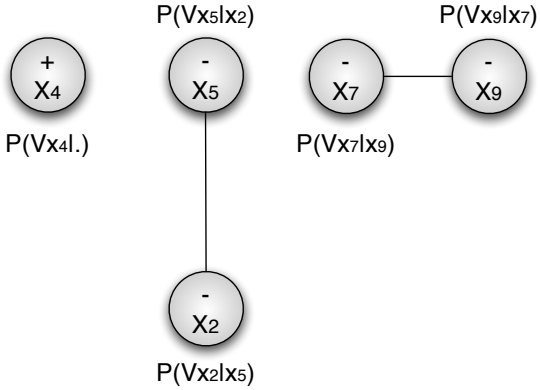
3.2 Independent learning and collective inference

We also use the relational probability tree (RPT) [12] as the base relational classifier for our independent learning and collective inference approach. The RPT has been used in previous work on relational dependency networks (RDNs) for collective inference [11]. RDNs use independent learning with pseudolikelihood and collective inference using Gibbs sampling to recover the full joint distribution. In this work, we also use pseudolikelihood estimation for learning, but we use a mean-field approximation for inference.

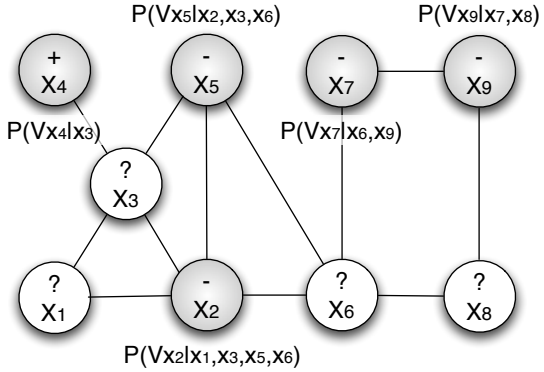
Maximum pseudolikelihood estimation (MPLE) is an approximate parameter estimation technique originally proposed in spatial statistics [1]. Previous work on RDNs has demonstrated that pseudolikelihood models are effective for relational domains due to their capability of representing



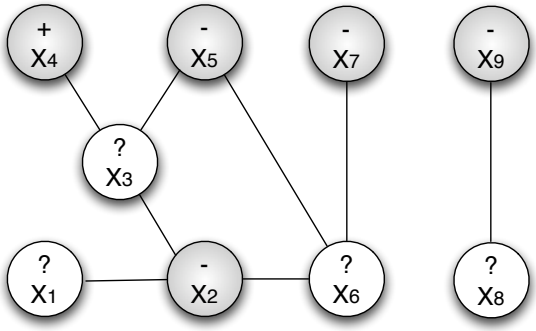
(a) data graph



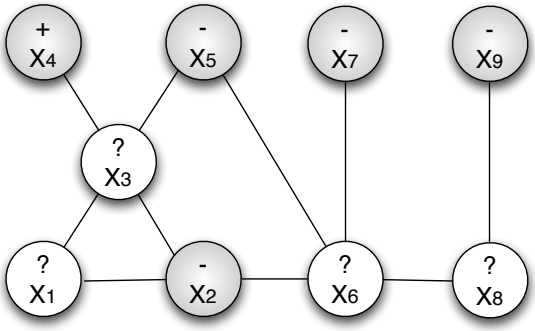
(b) independent learning



(c) collective learning



(d) independent inference



(e) collective inference

Figure 1: Types of within-network relational learning and inference.

cyclic autocorrelation while affording efficient learning techniques [11].

For each data instance x_i, psuedolikelihood 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., $P(v_{x_i}|Pa(x_i))$. However, the local CPDs are not required to factor the full joint distribution. More precisely, instead of maximizing likelihood during learning, we maximize the following pseudolikelihood:

$$PL(X;\theta) = \prod_{x_i \in X} p(v_{x_i}|\mathrm{Pa}(x_i);\theta) \quad (1)$$

The parents of x_i may include other attributes on the same object, class labels on the related objects and other attributes on the related objects.

Psuedolikelihood is an approximate representation of the true joint distribution. By applying MPLE, we only need to perform maximum likelihood estimation for each local CPD independently. In comparison with approximate parameter estimation algorithms for exact representations, such as loopy belief propagation for relational Markov networks [16], pseudolikelihood models substantially reduce the computational cost of learning, since the time complexity is just the product of the data size and the complexity of the base learner. Previous work has shown that MPLE is asymptotically consistent with the maximum likelihood estimator [3], and is reasonably accurate in a number of real-world applications [5, 11].

Any simple relational learner can be used as the base learner to estimate the local CPDs (e.g., RPTs, RBCs, structural logistic regression); such base classifiers need not be the same for each node in the graph. In practice, based on assumptions of homogeneity across the network, objects of the same type are assumed to be exchangeable and thus share the same parameters. Therefore, a sufficient number of data points are available for each relational classifier to learn a good model. For this work, we use RPTs as the base learner. Strictly speaking, RPTs are not maximum likelihood estimators. However, previous studies [11] have shown that within the pseudolikelihood framework, using RPTs as the local conditional model usually results in better empirical performance than other MLE learners such as the RBC [13].

During inference, a unique joint probability distribution over unlabeled instances must be recovered by applying the learned pseudolikelihood estimator. Since the graph represents complex dependencies among instances, performing exact inference to get a joint probability distribution over all unlabeled instance is generally intractable.

In this work, we propose to use a mean-field approximation for collective inference. The mean-field method was first developed in statistical mechanics as a deterministic approximation approach for regular lattice models; it was latter successfully applied to many other applications and to approximate inference for graphical models in general. The major advantage of mean-field approximation is its efficiency as well as relative accuracy. The mean-field method uses a fully factorized distribution to approximate the true posterior. Formally, let the true posterior be $p(Z|X,\theta)$, where Z is the set of unobserved variables and X denotes the set of observed variables. The approximating distribution q is factorized as follows:

$$q(Z) = \prod_i q_i(Z_i) \quad (2)$$

By minimizing the Kullback-Leibler divergence between q and p, the optimal solution of q can be obtained with an iterative updating scheme [8]:

$$\begin{aligned} \log q_i^*(Z_i) &= \sum_{Z \setminus Z_i} \prod_{j \neq i} q_j \log p(X,Z) + C \\ &= E_{q,j\neq i}[\log p(X,Z)] + C \end{aligned} \quad (3)$$

(3) shows that each factor q_i is iteratively updated by considering the joint distribution over all hidden and visible variables and then taking the expectation with respect to all of the other factors q_j for $j \neq i$.

From an algorithmic perspective, the mean-field procedure is an intuitively sensible method for collective inference. The probabilities of instance labels are updated iteratively; when approximating the probability of one label, other instances are considered to be constant and the mean estimates of their labels are used to calculate the posterior probability distribution of the label in question.

To facilitate this mean-field collective inference approach, we modified the RPT algorithm so that it estimates the probability of the instance label in question given the current mean probabilities of other instance labels. This estimation procedure corresponds to applying the learned RPT model with probabilistic attributes, which can be implemented by dividing one test instance into several pseudo-instances, and putting weights on each of them according to the appropriate probabilities.

3.3 Collective learning and inference

For our collective learning and collective inference approach, we extend our pseudolikelihood models to use EM and jointly infer the class values of unlabeled instances while learning the parameters and structure of the model.

The primary goal of maximum likelihood learning is to optimize the likelihood function over labeled node set X, i.e., find an optimal estimator $\hat{\theta}$ such that

$$\hat{\theta} = \arg\max_{\theta} P(X|\theta) \quad (4)$$

However, given a within-network setting, where there are unknown values coupled with observed values in the likelihood function, directly optimizing $p(X|\theta)$ is intractable. Therefore, we resort to an EM treatment, in which we consider the full data likelihood $p(X,Z|\theta)$, where Z denotes the unlabeled data.

- **E-step**: evaluate $p(Z|\theta^{old})$;
- **M-step**: update the estimator:

$$\theta^{new} = \arg\max_{\theta} \sum_{Z} p(Z|X,\theta^{old}) \log p(X,Z|\theta) \quad (5)$$

In the M-step, for the sake of tractability, we substitute likelihood of the complete data with pseudo-likelihood as discussed in Section 3.2, i.e., using the following equation to approximate (5):

$$\theta^{new} = \arg\max_{\theta} \sum_{Z} p(Z|X, \theta^{old}) \sum_{x_i \in X} \log p(x_i|\text{Pa}(x_i); \theta) \quad (6)$$

In (6), 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 independent 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 tends to be highly biased when only a few related instances are labeled.

The PL-EM based collective learning algorithm can be described as follows.

---

1. Learn initial local classifier using only labeled instances.
2. For each PL-EM iteration:
   E-step: apply local relational classifiers to unlabeled data, use current expected values for neighbor labels; obtain new probability estimates for missing labels;
   M-step: re-train local models with updated label probabilities of unlabeled instances.

In the general PL-EM algorithm, there is no restriction on the choice of local relational classifiers and the inference algorithm. For this work, we use mean-field approximation for collective inference to update label probabilities in the E-step, and train RPTs to obtain new parameter estimates in the M-step. For the M-step, we modify the RPT learning algorithm to use probabilistic inputs. The implementation simply enumerates all possible attribute values, and divides each training instance into pseudo-instances with the different values and corresponding weights.

4. EXPERIMENTS

In this section, we present empirical evaluation of the following three methods for within-network relational learning.

- *Independent learning, independent inference (IL-II)*: learning on the labeled subgraph of the original datagraph, and applying the learned model independently on each test instance.
- *Independent learning, collective inference (IL-CI)*: learning on the labeled subgraph of the original datagraph, and applying the learned model with mean-field collective inference.
- *Collective learning, collective inference (CL-CI)*: learning and inference on the entire datagraph with the proposed pseudolikelihood-EM algorithm.

We evaluate the three methods in within-network classification tasks on two datasets: a synthetic dataset with autocorrelation, and the Cora dataset, on which the citation link structure has previously been shown useful for paper topic classification.

Note that we do not investigate a *collective learning, independent inference* approach, because the use of collective learning generally implies the use of collective inference as well (since the class labels of unlabeled instances need to be inferred before learning). An approach that uses independent inference within a collective learning context is unlikely to improve performance, since the independently inferred class labels would not be used to improve the classification of related instances.

4.1 Synthetic Data

The synthetic dataset is generated with latent group model, described in the work of Neville and Jensen [10]. The number of objects in a graph is 250, the size of each group is 25; each object has one binary class label and three other boolean attributes; the class label has an autocorrelation level of 0.5. We experiment with labeled node selections of size 15, 30, 50, 70, 100, 150. Furthermore, labeled subsets were generated from three different labeling mechanisms (which emulate possible real-world data collecting procedures).

- Random sampling: a subset of nodes are sampled uniformly from the whole graph for labeling;
- Degree sampling: the nodes with maximum degree are selected for labeling;
- Snowball sampling: first an initial node is sampled randomly, then its neighboring nodes are selected, and the selection keeps expanding in a breath-first manner.

The experimental results are shown in Figure 2. All results are averaged over 10 randomly generated datasets. Clearly, in all cases, when there are moderate number of labeled nodes (i.e., above 10%), PL-EM method outperforms the baseline algorithm significantly, and the collective inference method also performs better than the baseline, but not as well as PL-EM. This result confirms that by further exploiting autocorrelation in labeled and unlabeled data, there is a unique opportunity to improve both learning and inference. When there is only very few labeled instances (i.e., below 10%), however, performing collective learning and collective inference can reduce model accuracy, which indicates that information propagation could be problematic if there is too much uncertainty in the data. When the number of labeled instances becomes larger, the improvement over the baseline approach tends to decrease, which is simply because using the labeled data alone is sufficient for learning and inference.

Moreover, regarding different labeling schemes, the results suggest that choosing large-degree instances to label could be economic if the labeling budget is small (i.e., only less than 30% instances can be labeled); otherwise random sampling is simply the best way to select instances among the three. When the number of labeled instances is large, the baseline model in the degree sampling case performs significantly worse than that in the random sampling case, which

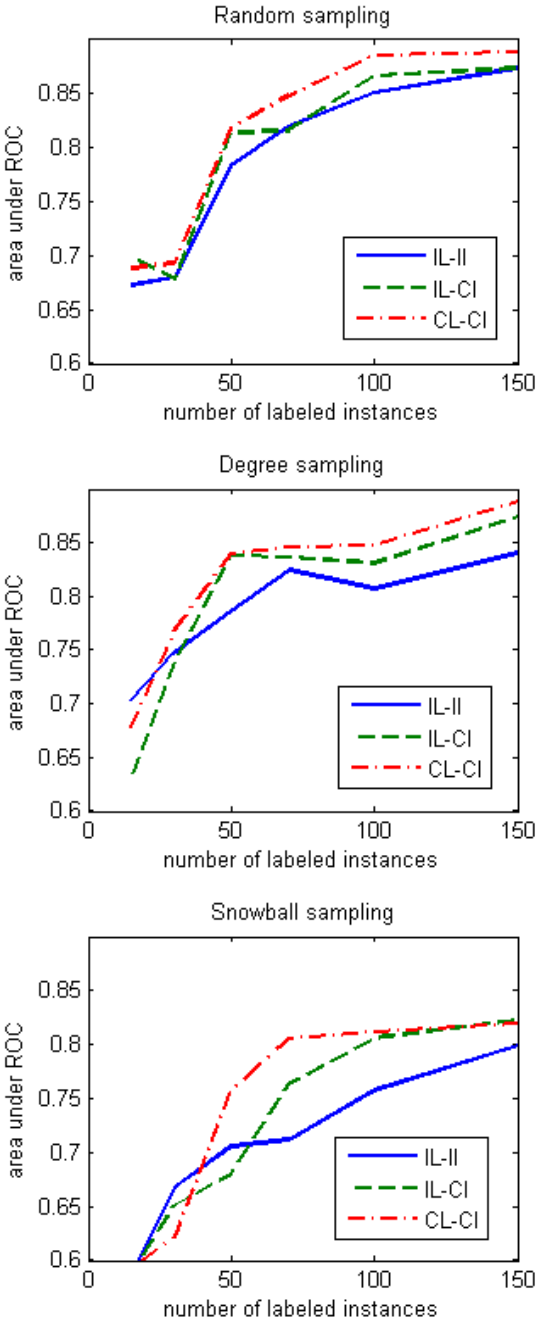


Figure 2: Results on synthetic data

might imply that autocorrelation is not well represented by the labeled instances with large degree. The snowball sampling case is much worse than the other two, suggesting that this labeling method tends to highly bias model learning and inference. However, the PL-EM improvement over the other two models is more dramatic in the snowball sampling case (compared to degree or random sampling). This suggests that the PL-EM approach will be even more useful when the labeled instances are clustered in the graph.

Finally, we notice that the PL-EM procedure usually converges very quickly (after 1 or 2 iterations). Figure 3 shows two example curves of AUC values against PL-EM iterations (in the random sampling case).

4.2 Cora Dataset

The Cora dataset is a collection of computer science papers. We sample partial machine learning papers on three topic (neural networks, probabilistic models and theory) published between 1993 and 1998. The size of the subset is 944, among which 452 papers are on neural networks. This gives a balanced classification task. We apply the models to predict whether a paper's topic is neural networks. The citation network among these papers forms a graph for within-network relational learning. Besides link structures, we extract attributes for each paper including author ranks and publication venue information. Nodes in the graph are randomly selected for labeling.

The experimental results at different labeled instance sizes are shown in Figure 4. For each size, we test the methods on 5 random samples and report the average AUC values. The PL-EM approach again performs the best when there is a moderate number of labeled instances. The improvement here, however, is mainly due to collective learning since the collective inference method alone is even slightly worse than the baseline. The failure of the collective inference in this case is likely to be caused by the ill behavior of mean-field approximation, which might have got stuck at non-maximum points. When labeled instances are sparse or sufficient, the baseline model tends to perform equally well as the PL-EM model, which indicates that the simple independent learning approach could be acceptable in these situations.

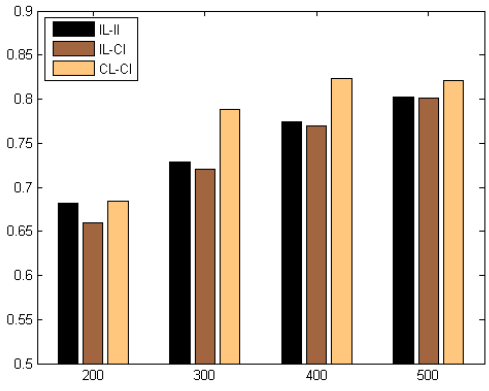


Figure 4: Results on Cora dataset

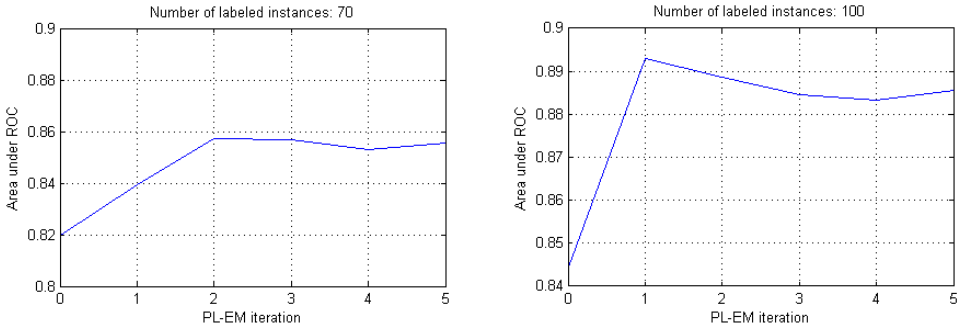


Figure 3: PL-EM convergence

5. RELATED WORK

The task of within-network relational learning studied in this paper is a semi-supervised learning problem [18]. Previous work in this area of semi-supervised learning has primarily focused on i.i.d. data. More specifically, the graph-based semi-supervised learning approaches construct conceptual links among i.i.d. instances to encode similarities, while in our setting the graph represents physical connections or associations among instances. However, a number of algorithms based on graph cut and label propagation have been proposed to improve learning on partially-labeled graphs constructed from i.i.d. data [19]. These methods of learning with graphs might also be effective for within-network relational learning. We will compare the PL-EM approach with graph-based semi-supervised learning methods in our future work.

Although semi-supervised learning for i.i.d. data has received a lot of attention from the machine learning community, the issue of learning with labeled and unlabeled data in a relational setting has not been fully explored. In the recent work of Taskar et al. [17], semi-supervised learning for relational data is considered in the context of probabilistic relational models (PRMs). However, PRMs cannot represent arbitrary cyclic autocorrelation dependencies, which limits their modeling power for many relational contexts. By contrast, the simple and efficient PL-EM approach could serve as a general framework to fully exploit autocorrelation among labeled and unlabeled data for widespread relational learning tasks.

Previous research on relational learning has shown the utility of collective inference models and their ability to improve classification in relational domains [17, 16, 11, 9]. Our work is most closely related to relational dependency networks (RDNs) [11], which are also pseudolikelihood models that performs collective inference during classification. However, RDNs employ Gibbs sampling, a stochastic approximation procedure, for collective inference. The convergence of our mean-field approximation is usually substantially faster than Gibbs sampling, but there is no theoretical guarantee that the mean-field can always converge to a local maximum. In future work, we will conduct empirical evaluation to compare these two approaches in pseudolikelihood models for relational learning.

6. CONCLUSIONS

In this work, we developed a pseudolikelihood-EM algorithm for within-network relational learning. Our algorithm combines collective learning and collective inference in a formal way, and provides a unified framework which allows the model to fully exploit dependencies between both labeled and unlabeled data. In the E-step of PL-EM, we use the labeled data for collective inference. In the M-step, we optimize the complete data pseudolikelihood by incorporating the unlabeled data, and collective learning is achieved in this way. The empirical evaluation on both synthetic data and real-world data has demonstrated the effectiveness of our PL-EM algorithm.

There are several future directions to improve this work. First, we will perform further empirical studies, testing our algorithm on more real datasets. Since many real-world tasks fall into the within-network learning setting, our approach is applicable to a number of existing relational datasets. Second, we will compare our algorithm to graph-based semi-supervised learning methods. The connection between i.i.d. semi-supervised learning and relational learning via graph-based methods is worth exploring and might improve our understanding of both methods. Third, since the mean-field approximation can be ill-behaviored in certain circumstances, we will explore the performance of different deterministic and stochastic approximation methods for collective inference in relational graphs.

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.

7. REFERENCES

[1] J. Besag. Statistical analysis of non-lattice data. *The Statistician*, 24(3):179–195, 1975.

[2] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *J. Royal Stat. Soc.*, 39:1–38, 1977.

[3] S. Geman and C. Graffine. Markov random field image models and their applications to computer vision. In *Proceedings of the 1986 International Congress of*

Mathematicians, page 1496Ű1517, 1987.

[4] L. Getoor, N. Friedman, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *Relational Data Mining*. 2001.

[5] D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. M. Kadie. Dependency networks for collaborative filtering and data visualization. In *UAI '00: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence*, pages 264–273, 2000.

[6] D. Jensen and J. Neville. Linkage and autocorrelation cause feature selection bias in relational learning. In *ICML '02: Proceedings of the Nineteenth International Conference on Machine Learning*, pages 259–266, 2002.

[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] M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Mach. Learn.*, 37(2):183–233, 1999.

[9] S. A. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. *J. Mach. Learn. Res.*, 8:935–983, 2007.

[10] J. Neville and D. Jensen. Leveraging relational autocorrelation with latent group models. In *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 322–329, 2005.

[11] J. Neville and D. Jensen. Relational dependency networks. *J. Mach. Learn. Res.*, 8:653–692, 2007.

[12] J. Neville, D. Jensen, L. Friedland, and M. Hay. Learning relational probability trees. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 625–630, 2003.

[13] 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.

[14] C. Perlich and F. Provost. Aggregation-based feature invention and relational concept classes. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 167–176, New York, NY, USA, 2003. ACM.

[15] A. Popescul, L. H. Ungar, S. Lawrence, and D. M. Pennock. Towards structural logistic regression: Combining relational and statistical learning. In *Proceedings of the Workshop on Multi-Relational Data Mining (MRDM-2002) at KDD-2002*, pages 130–141, Edmonton, Canada, 2002.

[16] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *UAI '02, Proceedings of the 18th Conference in Uncertainty in Artificial Intelligence*, pages 485–492, 2002.

[17] B. Taskar, E. Segal, and D. Koller. Probabilistic classification and clustering in relational data. In *Proceeding of IJCAI-01, 17th International Joint Conference on Artificial Intelligence*, pages 870–878, 2001.

[18] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.

[19] X. Zhu. *Semi-supervised learning with graphs*. PhD thesis, Carnegie Mellon University, 2005. Chair-John Lafferty and Chair-Ronald Rosenfeld.