Understanding Propagation Error and Its Effect on Collective Classification

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

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

ABSTRACT

Recent empirical evaluation of statistical relational models for collective classification has shown that performance can vary based on the amount of class label information that is available for use during inference. In this paper, we further demonstrate that relative performance of estimated models using different learning techniques may change as the amount of test set labels varies. We reason about the cause of this phenomenon, and characterize the high propagation effect of collective inference using maximum pseudolikelihood estimation (MPLE) that is responsible for the significantly different performance in different regimes of labeled proportions. This points to a previously unidentified consideration in the development of learning algorithms for probabilistic relational models. We formally study the propagation error in collective inference with MPLE, which leads to a quantitative characterization that can be used to predict the confidence in local propagation using MPLE models. We then propose a mixture modeling approach to achieve a good trade-off between high propagation and low propagation models. Empirical evaluation on synthetic and real-world dataset show that our proposed method can achieve comparable, or superior, results to both MPLE and low propagation models across a number of settings.

1. INTRODUCTION

Collective classification with probabilistic relational models has received much attention lately, due to the abundance of relational and network domains that exhibit correlation among the class labels of related instances (e.g., friends in a social network are like to have similar political views). In statistical relational learning, recent work has focused on *learning* the joint distribution of relational dependencies in a labeled training graph (e.g., social network) and then applying the learned model to *collectively* infer the unknown class labels in another, disjoint (test) graph [6, 18, 16, 15].

While probabilistic models are able to represent complex dependencies in the data, they are also difficult to estimate. A number of learning algorithms for probabilistic relational models have been developed (see e.g. [15, 10, 9, 8]), among which the most representative two approaches are maximimum likelihood estimation (MLE) and maximum pseudo-likelihood estimation (MPLE). In i.i.d. domains, MPLE can be viewed as an approximation to MLE since it converges to MLE as the number of training instances increase. However, this view is no longer appropriate for relational domains where the training or test data is a single network of interdependent instances. Likewise, the classic statistical optimality of MLE no longer applies to the situation of relational learning in one network either. For this reason, although much of the research work has focused on developing efficient approximations to MLE, it warrants a careful examination of MPLE and MLE-type approaches in relational settings in the first place. We compare these two types of learning algorithms in collective classification tasks where the amount of seeding labels varies.

Many of the empirical results show that relational model performance can vary based on the amount of class label information that is available for use during inference (see e.g., [11]). This is because relational models utilize the class label dependencies among neighboring nodes in the graph and this information is propagated during the collective inference process. When there are few labeled instance in the test graph, there is seeding less information and thus classification performance decreases. Although the relevance of label availability has received attention in the research on collective classification, to our knowledge, the issue of tailoring learning algorithms to different label availability scenarios has not been addressed.

We found that MPLE (which tends to result in strong propagation effect when applied on test networks with a collective inference procedure) and MLE-type algorithms (which tend to result in low propagation effect during collective inference) achieve superior performance in different regimes of the label availability spectrum. We provide an explanation of this phenomenon from a learning theoretic perspective. Our observation and analysis therefore add a new dimension to the comparison between different learning approaches beyond the traditional tradeoff between accuracy and efficiency. We further investigate the error propagation mechanism of collective inference based on MPLE estimates, and develop a quantitative method to characterize this propagation. This characterization is facilitated by the application of the microscopic dependency method, which provides a generic way to decompose complex relational dependencies and approximate long range dependencies when

MLG '11 San Diego, CA, USA

a learned model is applied to a particular network (with specific labeled nodes).

Finally, we use the key insight above to develop a mixture model that can automatically choose between a low propagation model and a high propagation model locally and dynamically during collective inference. The success of this new model lies in the effectiveness of the predictor that we use to determine the activation probability of the high propagation model. This predictor is exactly the quantitative characterization of propagation effect in the high propagation model obtained from our analysis. Empirical evaluation on synthetic and real-world dataset demonstrate that the mixture approach can achieve comparable, or superior, results to both MPLE and low propagation models across the whole spectrum of test set label availability.

2. BACKGROUND

2.1 Collective Classification with Probabilistic Models

Collective classification techniques for network data attempt to explore dependencies between linked instances to improve prediction accuracy [17]. Due to their ability to model complex relationships, probabilistic models are the most popular approach to collective classification. We now review a general probabilistic modeling formulation to relational classification problems. Similar to classification in i.i.d. settings, each data instance i has an attribute vector $x_i \in \mathcal{X}$ and a label $y_i \in \mathcal{Y}$. In relational settings, we further assume a relational structure over the data instances. Therefore, we pre-specify a set $\mathcal{T}$ of clique templates. Within each template type $T \in \mathcal{T}$, there is a set $\mathcal{C}(T)$ of cliques, each clique $C \in \mathcal{C}(T)$ ties together a set of instances $C = \{i_1, i_2, \ldots, i_{|C|}\}$. By making a Markov assumption, the joint probability distribution of labels given the attributes in the network G can be written as the following exponential family form.

$$P(\mathbf{y}_G|\mathbf{x}_G) = \frac{1}{Z(\boldsymbol{\theta}, \mathbf{x}_G)} \prod_{T\in\mathcal{T}} \prod_{C\in\mathcal{C}(T(G))} \Phi_T(\mathbf{x}_C, \mathbf{y}_C; \boldsymbol{\theta}_T) \quad (1)$$

where Z is the normalization factor, and we use $\mathbf{x}_C$ to denote $(x_{i_1}, x_{i_2}, \ldots, x_{i_{|C|}})$ (and similar for $\mathbf{y}_C$). Furthermore, we see in this template formulation that the parameter $\boldsymbol{\theta}$ of cliques within the same template is homogeneous, which makes learning and generalization possible. Therefore, a single potential function Φ_T is used for each template T. Each potential is further formulated as a log-linear function of a set of features $\boldsymbol{\phi}_T$. The feature function $\boldsymbol{\phi}_T$ is predefined and are computed from the vector of attributes and labels within the corresponding clique C.

$$\Phi_T = \exp\{\langle \boldsymbol{\theta}_T, \boldsymbol{\phi}_T(\mathbf{x}_C, \mathbf{y}_C)\rangle\} \quad (2)$$

This representation encompasses a rich class of probabilistic relational models in the literature, including Relational Markov Networks (RMN, [18]), Markov Logic Networks (MLN, [16]) and Relational Dependency Networks (RDN, [15]).

Since the exact inference using these models in general network data is intractable, many collective classification methods have been considered in the literature, e.g., loopy belief propagation, mean field relaxation and Gibbs sampling. To make the discussion concrete, we focus on Gibbs sampling in this paper, since it is theoretically guaranteed to recover the exact probability distribution (defined by the model) in the limit of infinite number of iterations. A Gibbs sampler for collective classification takes a partially labeled (with L denoting the labeled instance set) test network G, attributes $\mathbf{x}_G$, the set of observed labels $\mathbf{y}^*_L$ and the learned parameters $\boldsymbol{\theta}$ as input. It outputs samples from the joint distribution $P(\mathbf{y}_{G\setminus L}|\mathbf{y}^*_L)$. We can then obtain the approximate marginal distributions $P(y_i|\mathbf{y}^*_L)$ [1] from these samples. The error of our collective classification model is simply computed as the per instance error rate:

$$\begin{aligned}\text{Error}(\mathbf{y}^*_G, P_{\boldsymbol{\theta}}) =& \frac{1}{|G\setminus L|}\sum_{i\in G\setminus L} P(y_i \neq y^*_i|\mathbf{y}^*_L)\\ =& 1 - \frac{1}{|G\setminus L|}\sum_{i\in G\setminus L} P(y^*_i|\mathbf{y}^*_L) \quad (3)\end{aligned}$$

2.2 Parameter Estimation in Probabilistic Relational Models

The maximum likelihood estimation (MLE) for model (1) can be written as the following optimization problem.

$$\begin{aligned}\boldsymbol{\theta}^{\text{MLE}} =& \operatorname*{argmax}_{\boldsymbol{\theta}} \log P(\mathbf{y}_G|\mathbf{x}_G)\\ =& \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{T\in\mathcal{T}}\sum_{C\in\mathcal{C}(T(G))} \langle \boldsymbol{\theta}_T, \boldsymbol{\phi}_T(\mathbf{x}_C, \mathbf{y}_C)\rangle - \log Z(\boldsymbol{\theta}, \mathbf{x}_G)\end{aligned} \quad (4)$$

The MLE is generally intractable for large networks due to the normalization factor Z.

Another straightforward method for parameter estimation is the maximum pseudolikelihood estimation (MPLE). Due to its efficiency, MPLE is widely applied to relational data in practice [15]. Let ∂i denote the Markov blanket of i, i.e., the instances that shares some same clique with i. The MPLE optimizes the product of local conditional probability distributions (CPD), $P(y_i|x_i, x_{\partial i}, y_{\partial i})$.

$$\begin{aligned}\boldsymbol{\theta}^{\text{MPLE}} =& \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{i\in G} \log P(y_i|x_i, x_{\partial i}, y_{\partial i})\\ =& \operatorname*{argmax}_{\boldsymbol{\theta}} \sum_{i\in G}\left(\varphi_i - \log\sum_{y_i, \mathbf{y}_{\partial i}} \exp(\varphi_i)\right) \quad (5)\end{aligned}$$

where φ_i denotes the local potential of instance i, i.e., the summation of the potentials of all cliques that involve i. Since the global normalization in (4) is replaced by the local normalization $\log\sum_{y_i, \mathbf{y}_{\partial i}} \exp(\varphi_i)$, exact optimization of (5) is usually tractable.

To facilitate the analysis in this paper, we further decompose the local potential into self potential φ^S_i and interaction potential φ^I_i: $\varphi_i = \varphi^S_i + \varphi^I_i$, where the self potential is only a function of y_i and attributes, while the interaction potential also depends on neighboring labels $y_j : j \in \partial i$.

3. A COMPARISON OF DIFFERENT PARAMETER ESTIMATION METHODS

Although the availability of test set labels has been regarded as an important factor in determining collective classification performance, there has been little work which in-

[1] Alternatively, one may consider using the MAP $\operatorname{argmax}_{\hat{y}_{G\setminus L}} P(\hat{y}_{G\setminus L}|\mathbf{y}^*_L)$ for prediction. However, we adopt the marginal likelihood in this paper as it is widely applied in relational classification on single networks.

vestigates different labeling scenarios in the context of learning algorithm comparison, with the possible exception of the stacking approach [8]. Although not explicitly motivated by the goal, the stacking approach to relational learning is often credited for adjusting for the mismatch in label availability between training data and test data that occurs in pseudo-likelihood type approaches [5]. In this section, we seek to understand the full picture of relative performance of various parameter estimation methods when the amount of observed test set labels varies. In Figure 1, the classification error of MPLE, MLE, independent learning and stacking. All methods have the same form of local potentials, except for the independent learning, which is equivalent to a logistic regression model that applies the same form of self potentials as in the relational models, but does not contain the interaction potentials φ^I.

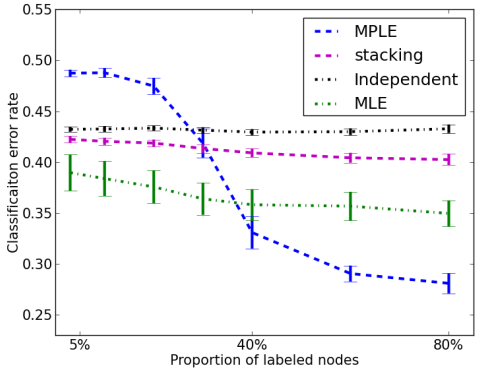

Figure 1: Classification error on test networks. We generate training and test networks of size 200 (for which MLE can be obtained) using a latent group model [13], which emulates the properties of autocorrelation and community structure observed in real networks. In each simulation we train the models on one synthetic network and test them on another. We repeat the simulation 20 times to obtain the error bars on classification error.

3.1 Why MPLE outperforms MLE in the regime of large labeled proportions?

We observe that MPLE outperforms all other methods when there are sufficient amount of observed labels in the test network. In particular, it outperforms MLE, although the latter is often taken for granted to be the preferable way of parameter estimation whenever the computational resource allows.

The superior performance of MPLE is a consequence of learning the parameters from a single network and partially observed test set labels. Let π denote the underlying true generative distribution of the training network G, i.e., the observed training network is a single sample from $\pi(\mathbf{x}_G, \mathbf{y}_G)$. This should be distinguished from the classic learning settings. In the classic settings, during training there are many samples from $\pi(\mathbf{x}_G, \mathbf{y}_G)$, and hence the empirical data distribution $\hat{\pi}(\mathbf{x}_G, \mathbf{y}_G)$ represents the underlying data generative distribution well. In this scenario, since MLE minimizes the Kullback-Leibler divergence between $\hat{\pi}(\mathbf{y}_G|\mathbf{x}_G)$ and $P_{\boldsymbol{\theta}}$: $\mathcal{D}(\hat{\pi}(\mathbf{y}_G|\mathbf{x}_G)\|P_{\boldsymbol{\theta}}(\mathbf{y}_G|\mathbf{x}_G))$, at the same time it approximately minimizes the KL divergence between $\pi(\mathbf{y}_G|\mathbf{x}_G)$ and $P_{\boldsymbol{\theta}}(\mathbf{y}_G|\mathbf{x}_G)$ since $\mathcal{D}(\pi(\mathbf{y}_G|\mathbf{x}_G)\|P_{\boldsymbol{\theta}}(\mathbf{y}_G|\mathbf{x}_G))$ can be bounded by $\mathcal{D}(\hat{\pi}(\mathbf{y}_G|\mathbf{x}_G)\|P_{\boldsymbol{\theta}}(\mathbf{y}_G|\mathbf{x}_G)) + \epsilon(n)$ with high probability, where ϵ typically decreases exponentially or faster in the sample size (see, e.g., [2]). However, in relational learning with one network, it is still an open issue that, under which assumptions, $\mathcal{D}(\pi(\mathbf{y}_G|\mathbf{x}_G)\|P_{\boldsymbol{\theta}}(\mathbf{y}_G|\mathbf{x}_G))$ can be effectively bounded based on $\mathcal{D}(\hat{\pi}(\mathbf{y}_G|\mathbf{x}_G)\|P_{\boldsymbol{\theta}}(\mathbf{y}_G|\mathbf{x}_G))$ when the sample size is 1. Therefore, there is no theoretical guarantee in general situations that minimizing the divergence between the single network sample and the model distribution would lead to good generalization performance on another network sample from the underlying π. Furthermore, even if a bound on $\mathcal{D}(\pi(\mathbf{y}_G|\mathbf{x}_G)\|P_{\boldsymbol{\theta}}(\mathbf{y}_G|\mathbf{x}_G))$ can be obtained under certain assumptions, in the label abundant regime the more suitable objective should be $\mathcal{D}(\hat{\pi}(y_i|x_i, \mathbf{x}_{\partial i}, \mathbf{y}_{\partial i})\|P_{\boldsymbol{\theta}}(y_i|x_i, \mathbf{x}_{\partial i}, \mathbf{y}_{\partial i})$. This mismatch is yet another potential factor that leads to the inferior performance of MLE. Section 3.3 further elaborates on this point.

On the other hand, MPLE directly minimizes the KL divergence between the local CPD of the model distribution and that of the data distribution, i.e., $\mathcal{D}(\hat{\pi}(y_i|x_i, \mathbf{x}_{\partial i}, y_{\partial i})\|$ $P_{\boldsymbol{\theta}}(y_i|x_i, \mathbf{x}_{\partial i}, y_{\partial i})$. It should be noted that although there are n local CPDs in a network of n instances, the *effective sample size* of CPDs is much smaller than n due to the dependence between the CPDs. Nevertheless, under weak dependence assumptions such as exponential correlation decay with network distance, which typically can be satisfied in real networks, the effective sample size $\tilde{n}$ increases with n. Combining this observation with a stationarity assumption that postulates $P(x_i, y_i, \mathbf{x}_{\partial i}, \mathbf{y}_{\partial i})$ to be homogenous for any i, the expected local divergence on unseen data, $\mathcal{D}(\pi(y_i|x_i, \mathbf{x}_{\partial i}, \mathbf{y}_{\partial i})\|P_{\boldsymbol{\theta}}(y_i|x_i, \mathbf{x}_{\partial i}, \mathbf{y}_{\partial i})$ can be effectively bounded based on the local divergence between training data and the model, or explicitly, by $\mathcal{D}(\hat{\pi}(y_i|x_i, \mathbf{x}_{\partial i}, \mathbf{y}_{\partial i})$ $\|P_{\boldsymbol{\theta}}(y_i|x_i, \mathbf{x}_{\partial i}, \mathbf{y}_{\partial i}) + \epsilon(\tilde{n})$ (see, eg. [20]). Therefore, a training algorithm like MPLE, which minimize the local divergence on the training network, also approximately minimizes the local divergence on the test network. In the scenario when there is a large amount of observed test set labels, the predictive probability $P(y_i|\mathbf{x}, \mathbf{y}^*_L)$ is close to $P(y_i|\mathbf{x}, y_{\partial i})$. Since by the above reasoning, MPLE chooses the model parameter $\boldsymbol{\theta}$ so that $P_{\boldsymbol{\theta}}$ approximately minimizes the divergence between the true distribution CPD $\pi(y_i|\mathbf{x}, y_{\partial i})$ and the model CPD $P(y_i|\mathbf{x}, y_{\partial i})$ on the test network, the superior performance of MPLE in this abundant label scenario has thus been explained.

3.2 Why MPLE performs poorly in the regime of small labeled proportions?

At the other end of the spectrum, however, MPLE performs rather poorly, even worse than the independent learning approach. This is due to the fact that MPLE estimates the parameters by separating the local CPDs. It ignores the global coupling among CPDs and attribute all the dependency in the network to local dependencies. Therefore, the local interaction potential φ^I_i accounts for all the dependencies between instance i and the rest of the network. This works fine when we run inference using these CPDs separately on each instance, as in the case where each instance's neighbors are fully labeled. However, problems arise when we apply these CPDs collectively for inference, e.g., through

Gibbs sampling, as the propagation of local dependencies throughout the network results in excessive dependencies. We will analyze this over propagation effect of collective inference with MPLE in more detail in Section 4 by formalizing the notion of propagation error.

When the over propagation effect exceeds the benefit of using the relational information, the relational learning model fails to outperform the independent model which does not perform propagation. The MLE is a global learning approach and thus does not result in over propagation when used in collective classification. The stacking approach compensates for the over-propagation of MPLE by iteratively apply predicted neighboring labels (instead of true training set labels) during training, and yield a performance curve closer to MLE than independent learning without adding overly high computational cost. We call these later models *low propagation* models.

3.3 Why the discrepancy between these two cases?

Given that the different models obtained by different parameter estimation techniques come from the same model family, one may speculate that there should an "optimal" model from the model family which in expectation best predicts the labels in the test network in all scenario, i.e., that there should exist an optimal parameter $\hat{\boldsymbol{\theta}}$ so that the inferential distribution $P_{\hat{\boldsymbol{\theta}}}(\mathbf{y}_{G\setminus L}|\mathbf{y}^*_L, \mathbf{x})$ is the best match for the true distribution for any labeled set L. In fact, this is true in the case of well-specified model families—indeed, it has been shown that MLE and MPLE will converge to the same true parameter when the true data distribution π belongs to the model family (1) with predefined potential functions (see [19]). Unfortunately, in practice, the model family is unlikely to be well specified. In these scenario, the optimal parameter $\hat{\boldsymbol{\theta}}^L$ for certain set L which makes the model $P_{\boldsymbol{\theta}}(\mathbf{y}_{G\setminus L}|\mathbf{y}^*_L, \mathbf{x})$ closest to the true distribution $\pi_{\boldsymbol{\theta}}(\mathbf{y}_{G\setminus L}|\mathbf{y}^*_L, \mathbf{x})$—e.g., in the sense of KL divergence: $\hat{\boldsymbol{\theta}}^L = \operatorname{argmin}_{\boldsymbol{\theta}} \mathcal{D}(P_{\boldsymbol{\theta}}(\mathbf{y}_{G\setminus L}|\mathbf{y}^*_L, \mathbf{x}) \| \pi(\mathbf{y}_{G\setminus L}|\mathbf{y}^*_L, \mathbf{x})$—varies with L. For example, when L only contains 1% of instances in the test network, the MLE tends to be a better approximation of $\hat{\boldsymbol{\theta}}^L$, while when L contains 90% of the instances, the MPLE tends to be a better approximation of $\hat{\boldsymbol{\theta}}^L$. Therefore, we see again that this discrepancy is unique to relational classification in single network domains with partially observed labels.

Although most research work has focused on efficient approximations to MLE, these methods tend to result in low propagation models that improve over MPLE in the small labeled proportion regimes, but are inferior to MPLE in the large labeled proportion regimes. Our analysis provides a balanced view of different learning methods when they are applied to collective classification on network data with partially observed labels. We emphasize that by understanding the pathology of MPLE in the small labeled proportion regime, there is an opportunity of taking full advantage of both MPLE and low propagation models in collective classification tasks. The rest of this paper provides a starting point of exploration in this direction.

4. ERROR ANALYSIS OF COLLECTIVE INFERENCE USING MPLE

To gain further understanding into the collective classification error using MPLE, following the error rate defined by (3), it is useful to consider the following error decomposition [2] for each instance:[3]

$$\begin{aligned}\epsilon_i &= 1 - P(y^*_i|\mathbf{y}^*_L) \\ &= \text{base error} \quad + \quad \text{propagation error} \\ &= (1 - P(y^*_i|\mathbf{y}^*_{\partial i})) + (P(y^*_i|\mathbf{y}^*_{\partial i}) - P(y^*_i|\mathbf{y}^*_L))\end{aligned}$$

The base error is the classification error of a node's label in the scenario that all labels in the rest of the network are observable. Since MPLE optimizes for this scenario, the propagation error term is positive with high probability for any labeling situations. It thus makes sense to use this decomposition. While the base error is decided by the quality of the specification of model family and the feature selection process, the propagation error is the error caused by the collective inference mechanism based on partially observed labels in the rest of the network.

To analyze the propagation error, we first introduce distance measures for probability distributions, which will be used to evaluate *microscopic dependencies*. The following definition of *total variation distance* ν between two probability distributions π_1 and π_2 is standard.

$$\nu(\pi_1, \pi_2) = \frac{1}{2} \sum_{y \in \mathcal{Y}} |\pi_1(y) - \pi_2(y)| \tag{6}$$

The microscopic dependency δ_{ij} and σ_{ij} is defined to be the variation of the conditional probability $P(Y_i|\mathbf{y}_{G\setminus i})$, when only y_j is varied. Formally,

$$\delta_{ij} = \max_{y_j, y'_j, \mathbf{y}_{G\setminus\{i,j\}}} \nu\left(P(y_i|y_j, \mathbf{y}_{G\setminus\{i,j\}}), P(y_i|y'_j, \mathbf{y}_{G\setminus\{i,j\}})\right)$$

While the total variation distance aggregates the difference between two distributions over all states, it will also be convenient for our purpose to consider the maximum difference of two distributions over single states. Hence we define another measure of microscopic dependency, denoted by σ_{ij}.

$$\sigma_{ij} = \max_{y_i \in \mathcal{Y}, y_j, y'_j, \mathbf{y}_{G\setminus\{i,j\}}} \left|P(y_i|y_j, \mathbf{y}_{G\setminus\{i,j\}}) - P(y_i|y'_j, \mathbf{y}_{G\setminus\{i,j\}})\right|$$

For binary classification, $\sigma_{ij} = \delta_{ij}$; otherwise $\sigma_{ij} \leq \delta_{ij}$.

The microscopic dependencies measure the oscillation of node i's label y_i caused only by the change of node j's label y_j, while the rest of the network $\mathbf{y}_{G\setminus\{i,j\}}$ is unchanged.

Analytical methods based on microscopic dependencies are proposed by Dobrushin in his celebrated work [4], where δ_{ij} is used to prove sufficient conditions for the uniqueness of Gibbs measures on lattice data [7]. Our development of error analysis for collective classification that follows, is inspired by Dobrushin's method.

Let $\boldsymbol{\epsilon} = [\epsilon_1, \epsilon_2, \ldots, \epsilon_n]^T$ denote the classification error of instances, $\boldsymbol{\beta}$ denote the base error, and γ denote the propagation error. i.e., $\epsilon_i = \beta_i + \gamma_i$. Furthermore, let $\{j_1, j_2, \ldots, j_{t_i}\}$ index the *unlabeled* nodes within i's Markov

[2]This is not to be confused with the collective classification error decomposition performed in previous work [14, 5]. The previous decomposition is for the squared loss, which facilitates a bias/variance analysis.

[3]Throughout this section, since the model is always conditioned on the attributes, we drop the $\mathbf{x}$ on the right of the conditioning sign | for simplification.

blanket, i.e., $j_k \in \partial i \cap G \setminus L$. Finally, let j_p^q denote the sequence $\{j_p, j_{p+1}, \ldots, j_q\}$. To gain insights into the propagation error of collective classification, we apply the following expansion

$$
\begin{aligned}
\gamma_i &= P(y_i^*|\mathbf{y}^*_{G\setminus i}) - P(y_i^*|\mathbf{y}^*_{\mathcal{L}}) \\
&= \sum_{\mathbf{y}_{j_1^{t_i}}} P(\mathbf{y}_{j_1^{t_i}}) \left[P(y_i^*|\mathbf{y}^*_{j_1^{t_i}}) - P(y_i^*|\mathbf{y}_{j_1^{t_i}})\right] \\
&= \sum_{\mathbf{y}_{j_1^{t_i}}} P(\mathbf{y}_{j_1^{t_i}}) \sum_{k=1}^{t_i} \left[P(y_i^*|\mathbf{y}_{j_1^{k-1}}, \mathbf{y}^*_{j_k^{t_i}}) - P(y_i^*|\mathbf{y}_{j_1^{k}}, \mathbf{y}^*_{j_{k+1}^{t_i}})\right) \\
&= \sum_{k=1}^{t_i} \sum_{\mathbf{y}_{j_1^{t_i}}} P(\mathbf{y}_{j_1^{t_i}}) \left[P(y_i^*|\mathbf{y}_{j_1^{k-1}}, \mathbf{y}^*_{j_k^{t_i}}) - P(y_i^*|\mathbf{y}_{j_1^{k}}, \mathbf{y}^*_{j_{k+1}^{t_i}})\right] \\
&= \sum_{k=1}^{t_i} \sum_{y_{j_k} \neq y^* j_k} P(y_{j_k}) \sum_{\mathbf{y}_{j_1^{k-1}}, \mathbf{y}_{j_{k+1}^{t_i}}} P(\mathbf{y}_{j_1^{k-1}}, \mathbf{y}_{j_{k+1}^{t_i}}|y_{j_k}) \\
&\quad \left[P(y_i^*|\mathbf{y}_{j_1^{k-1}}, \mathbf{y}^*_{j_k^{t_i}}) - P(y_i^*|\mathbf{y}_{j_1^{k}}, \mathbf{y}^*_{j_{k+1}^{t_i}})\right] \\
&\leq \sum_{k=1}^{t_i} \left(1 - P(y^*_{j_k})\right) \sum_{\mathbf{y}_{j_1^{k-1}}, \mathbf{y}_{j_{k+1}^{t_i}}} P(\mathbf{y}_{j_1^{k-1}}, \mathbf{y}_{j_{k+1}^{t_i}}|y_{j_k}) \\
&\quad \max_{y_{j_1^{t_i}}} \left|P(y_i^*|\mathbf{y}_{j_1^{k-1}}, \mathbf{y}^*_{j_k^{t_i}}) - P(y_i^*|\mathbf{y}_{j_1^{k}}, \mathbf{y}^*_{j_{k+1}^{t_i}})\right| \\
&\leq \sum_{k=1}^{t_i} \epsilon_{j_k} \sigma_{ij_k} \leq \left(\sum_{k=1}^{t_i} \sigma_{ij_k}\right) \epsilon_{\partial i, \max}
\end{aligned} \tag{7}
$$

where in the last step, we define $\epsilon_{\partial i,\max} := \max_{k=1,\ldots,t_i} \epsilon_{j_k}$ to gauge the inference error of neighboring labels.

In this way, we have decomposed the propagation error along the edges. By iteratively applying Equation (7), we can see that the inference error of each node is propagated throughout the whole network. Therefore, the prediction error depends on the two factors: the base error and the propagation in the network. Since the base error is typically unknown, we focus on the propagation effect. If the microscopic dependencies σ_{ij_k} are small, the propagation effect decays rapidly with respect to graph distance. If the microscopic dependencies σ_{ij_k} for unlabeled nodes i, j are large, however, long range error propagation is likely to happen, which results in high propagation error.

More specifically, we define the *propagation coefficient* κ_i to evaluate the local propagation effect. κ_i upper bounds how much proportion of neighboring nodes' error are propagated to node i:

$$
\kappa_i := \sum_{k=1}^{t_i} \sigma_{ij_k} \tag{8}
$$

If $\kappa_i < 1$ for every instance i, the influence from far away nodes are guaranteed to decay exponentially with respect to graph distance.

4.1 Propagation upper bound

A direct computation of κ would involve evaluating the microscopic dependencies σ_{ij_k} by enumeration of neighboring labels. This can still be computationally intensive when the network is densely connected. Therefore, we further upper bound σ_{ij_k} based on the oscillation of potential functions of the model so that κ can be efficiently approximated. Lemma 1 will serve for this purpose. Let $\phi^1_{\mathbf{w}}(y), \phi^2_{\mathbf{w}}(y)$ be two set of potential functions each indexed by some vector variable $\mathbf{w} \in \mathcal{W}$. Each instantiation of $\mathbf{w}$ defines the two exponential family distributions $\pi^i(y) (i = 1, 2)$ over the label space $\mathcal{Y}$: $\pi^i(y) = \frac{\exp\left(\phi^i_{\mathbf{w}}(y)\right)}{\sum_y \exp\left(\phi^i_{\mathbf{w}}(y)\right)}$. Define the oscillation of potential δ_ϕ:

$$
\delta_\phi = \max_{\mathbf{w}} \left[\max_y \left(\phi^1_{\mathbf{w}}(y) - \phi^2_{\mathbf{w}}(y)\right) - \min_y \left(\phi^1_{\mathbf{w}}(y) - \phi^2_{\mathbf{w}}(y)\right)\right]
$$

LEMMA 1. *For any $\mathbf{w} \in \mathcal{W}$, the total variation distance between $\pi^1_{\mathbf{w}}$ and $\pi^2_{\mathbf{w}}$ is bounded: $\nu\left(\pi^1_{\mathbf{w}}(Y), \pi^2_{\mathbf{w}}(Y)\right) \leq \frac{1}{4}\delta_\phi$.*

PROOF. Given any $\mathbf{w}$, define

$$
\phi_q(y) = \min\left(\phi^1_{\mathbf{w}}(y), \phi^2_{\mathbf{w}}(y)\right) + q|\phi^1_{\mathbf{w}}(y) - \phi^2_{\mathbf{w}}(y)|
$$

where $q \in [0, 1]$. Then

$$
\phi'_q(y) \equiv \frac{d}{dq}\phi_q(y) = |\phi^1_{\mathbf{w}}(y) - \phi^2_{\mathbf{w}}(y)| \tag{9}
$$

By Equation (6) and the exchangeability of summation and integration,

$$
\begin{aligned}
\nu\left(\pi^1_{\mathbf{w}}(Y), \pi^2_{\mathbf{w}}(Y)\right) &= \frac{1}{2} \sum_y \left| \int_0^1 \frac{d}{dq} \left(\frac{e^{\phi_q(y)}}{\sum_y e^{\phi_q(y)}} \right) dq \right| \\
&\leq \frac{1}{2} \int_0^1 \sum_y \left| \frac{d}{dq} \left(\frac{e^{\phi_q(y)}}{\sum_y e^{\phi_q(y)}} \right) \right| dq
\end{aligned} \tag{10}
$$

where we find

$$
\begin{aligned}
&\left| \frac{d}{dq} \left(\frac{e^{\phi_q(y)}}{\sum_y e^{\phi_q(y)}} \right) \right| \\
=&\frac{e^{\phi_q(y)}}{\sum_y e^{\phi_q(y)}} \left| \phi'_q(y) - \sum_y \frac{e^{\phi_q(y)}\phi'_q(y)}{\sum_y e^{\phi_q(y)}} \right| \\
=&\frac{e^{\phi_q(y)}}{\sum_y e^{\phi_q(y)}} \left| \phi'_q(y) - \mathbb{E}\phi'_q(y) \right|
\end{aligned}
$$

And thus by Cauchy's inequality,

$$
\begin{aligned}
&\sum_y \left| \frac{d}{dq} \left(\frac{e^{\phi_q(y)}}{\sum_y e^{\phi_q(y)}} \right) \right| = \mathbb{E}\left| \phi'_q(y) - \mathbb{E}\phi'_q(y) \right| \\
&\leq \left(\mathbb{E}\left[\phi'_q(y) - \mathbb{E}\phi'_q(y) \right]^2 \right)^{\frac{1}{2}} \\
&\leq \left(\mathbb{E}\left[\phi'_q(y) - \frac{\max_y \phi'_q(y) + \min_y \phi'_q(y)}{2} \right]^2 \right)^{\frac{1}{2}} \\
&\leq \frac{1}{2} \left(\max_y \phi'_q(y) - \min_y \phi'_q(y) \right)
\end{aligned}
$$

Plugging this back into Equation (10), and by (9) we obtain

$$
\begin{aligned}
&\nu\left(\pi^1_{\mathbf{w}}(Y), \pi^2_{\mathbf{w}}(Y)\right) \\
&\leq \frac{1}{2} \int_0^1 \frac{1}{2} \left(\max_y \phi'_q(y) - \min_y \phi'_q(y) \right) dq \\
&= \frac{1}{4} \left(\max_y \phi'_q(y) - \min_y \phi'_q(y) \right) = \frac{1}{4}\delta_\phi
\end{aligned}
$$

□

To apply Lemma 1, we first define the oscillation function of interaction potentials, denoted by $\Delta_{j_k}(\varphi_i^I)$. Let $d_{j_k}(y_i, y_{j_k}^1, y_{j_k}^2, \mathbf{y}_{j_1^{t_i}\setminus j_k})$ denote the difference of unlabeled node i's interaction potential φ^I under two configurations which only differ at one neighboring label y_{j_k}:

$$d_{j_k}(y_i, y_{j_k}^1, y_{j_k}^2 \mathbf{y}_{j_1^{t_i}\setminus j_k}) = \varphi^I(y_i, y_{j_k}^1, \mathbf{y}_{j_1^{t_i}\setminus j_k}) - \varphi^I(y_i, y_{j_k}^2, \mathbf{y}_{j_1^{t_i}\setminus j_k})$$

We define $\Delta_{j_k}(\varphi_i^I)$ to be the maximum oscillation of d_{j_k} as the other unobserved neighboring labels $\mathbf{y}_{j_1^{t_i}\setminus j_k}$ vary:

$$\begin{aligned}\Delta_{j_k}(\varphi_i^I) = &\max_{\mathbf{y}_{j_1^{t_i}\setminus j_k}, y_{j_k}^1, y_{j_k}^2} \big(\max_{y_i} d_{j_k}(y_i, y_{j_k}^1, y_{j_k}^2, \mathbf{y}_{j_1^{t_i}\setminus j_k}) \\ &- \min_{y_i} d_{j_k}(y_i, y_{j_k}^1, y_{j_k}^2, \mathbf{y}_{j_1^{t_i}\setminus j_k}) \big) \\ = &\max_{\mathbf{y}_{j_1^{t_i}\setminus j_k}, y_{j_k}^1, y_{j_k}^2, y_i} 2 d_{j_k}(y_i, y_{j_k}^1, y_{j_k}^2, \mathbf{y}_{j_1^{t_i}\setminus j_k})\end{aligned}$$

PROPOSITION 1. *The propagation coefficient κ_i of each instance i is upper bounded by $\hat{\kappa}_i$:*

$$\hat{\kappa}_i = \frac{1}{8} \sum_{k=1}^{\mathbf{t}_i} \Delta_{j_k}(\varphi_i^I) \tag{11}$$

PROOF. For $k = 1, 2, \ldots, t_i$, we use Lemma 1 to bound σ_{ij_k}. Let $\mathbf{w} = \mathbf{y}_{j_1^{t_i}\setminus j_k}$. For any $y_{j_k}^1, y_{j_k}^2 \in \mathcal{Y}$, let $\phi_{\mathbf{w}}^1(y_i) = \varphi^I_{\mathbf{y}^*_{\mathcal{L}}, \mathbf{x}}(y_i, y_{j_k}^1, \mathbf{y}_{j_1^{t_i}\setminus j_k})$, $\phi_{\mathbf{w}}^2(y_i) = \varphi^I_{\mathbf{y}^*_{\mathcal{L}}, \mathbf{x}}(y_i, y_{j_k}^2, \mathbf{y}_{j_1^{t_i}\setminus j_k})$. Then $\pi_{\mathbf{w}}^1(Y) = \tilde{P}(Y_i|\mathbf{x}, y_{j_k}^1, \mathbf{y}_{j_1^{t_i}\setminus j_k}, \mathbf{y}^*_{\mathcal{L}})$ and $\pi_{\mathbf{w}}^2(Y) = \tilde{P}(Y_i|\mathbf{x}, y_{j_k}^2, \mathbf{y}_{j_2^{t_i}\setminus j_k}, \mathbf{y}^*_{\mathcal{L}})$. Hence

$$\begin{aligned}\sigma_{ij} \le \delta_{ij} &= \max_{y_{j_k}^1, y_{j_k}^2, \mathbf{y}_{j_1^{t_i}\setminus j_k}} \nu\left(\pi_{\mathbf{w}}^1(Y), \pi_{\mathbf{w}}^2(Y)\right) \\ &\le \max_{\mathbf{y}_{j_1^{t_i}\setminus j_k}, y_{j_k}^1, y_{j_k}^2, y_i} d_{j_k}(y_i, y_{j_k}^1, y_{j_k}^2, \mathbf{y}_{j_1^{t_i}\setminus j_k}) = \frac{1}{8}\Delta_{j_k}(\varphi_i^I)\end{aligned}$$

Therefore, we obtain

$$\kappa_i = \sum_{k=1}^{t_i} \sigma_{ij_k} \le \frac{1}{8}\Delta j_k(\varphi_i^I) = \hat{\kappa}_i$$

□

Combine this upper bound with inequality (7), we see that $\hat{\kappa}_i \epsilon_{\partial i, \max}$ provides an upper bound of the propagation error at i. From the definition, it is obvious that the propagation estimate $\hat{\kappa}_i$ decreases monotonically as more and more labels are acquired in the test network.

5. ALGORITHM

Based on the above analysis, we propose a new approach to collective classification. The purpose of this approach is to combine the strength of both MPLE and any low propagation model, so that the resulting algorithm achieves consistently low error across different labeled proportions in the test network.

We directly model the CPDs used in Gibbs sampling by a local mixture model $\mu(y_i)$. Given an MPLE estimate $\boldsymbol{\theta}$ and any low propagation model $\tilde{P}$, the model $\mu(y_i)$ is a mixture of $P_{\boldsymbol{\theta}}(y_i|x_i, \mathbf{x}_{\partial i}, \mathbf{y}_{\partial i})$ and $\tilde{P}(y_i|x_i, \mathbf{x}_{\partial i}, \mathbf{y}_{\partial i})$:

$$\mu(y_i) = \lambda_i P_{\boldsymbol{\theta}}(y_i|x_i, y_i, \mathbf{x}_{\partial i}, \mathbf{y}_{\partial i}) + (1 - \lambda_i)\tilde{P}(y_i|x_i, \mathbf{x}_{\partial i}, \mathbf{y}_{\partial i}) \tag{12}$$

where the mixture coefficient λ_i is a latent variable which represents the confidence of propagation by MPLE in predicting y_i. With probability λ_i, the PL model $P_{\boldsymbol{\theta}}(y_i|x_i)$ is activated. When the test network is fully labeled, there is no propagation error and λ_i should be 1. When the network is partially labeled, however, the propagation error is unknown and thus the confidence variable λ_i is latent. By the analysis in Section 4, it is reasonable to assume that λ_i is negatively correlated with the propagation upper bound κ_i. Thus, we use the following simple model for λ_i:

$$\lambda_i = \exp\left\{-\tau \max(\hat{\kappa}_i - \kappa_0, 0)\right\} \tag{13}$$

When $\hat{\kappa}_i$ is above a threshold κ_0, the MPLE model is activated; otherwise, the MPLE model is activated with probability decaying exponentially with the propagation upper bound.

Alternatively, one may propose to directly use the labeled proportion, instead of the propagation upper bound $\hat{\kappa}$, as the predictor of confidence. However, we argue that the propagation effect is the underlying factor that $\hat{\kappa}$ rightly captures. The labeled proportion, on the other hand may not be in accordance with the propagation error since at the same labeled proportion, different labeling schemes may result in very different propagation strength in the network. For example, consider the comparison of random label acquisition, acquiring labels by node degree and by snowball sampling. As a side note, certain active inference approaches (e.g., the AIGA approach in [1]) can in fact be viewed as reducing the propagation error in the network to the greatest extent within a certain labeling budget.

The mixture model is learned on the training network. Since the mixing coefficients λ_i are latent variables which is coupled with the propagation effect in partially labeled settings, estimating the meta parameters τ and κ_0 is not a trivial task. Fortunately, this is only a two dimensional problem. We develop a simulation method with simple grid search to experimentally validate the model. The experiment results are encouraging. Our method samples multiple times subsets of available labels on the training set to simulate different label availability simulation and hence different propagation strength during collective classification. The details are described in Algorithms 1 and 2. The advantage of this mixture model is that it allows us to dynamically adjust label propagation using MPLE during the collective inference process. Furthermore, since the mixture model is defined on a local level, it allows us to model the heterogeneity of instances due to the difference in network local structures and the difference in label availability at different locations.

6. EXPERIMENTS

We evaluate the strength of the mixture model and verify our understanding of propagation error by comparing the performance of the mixture model with the component models. We tested two low propagation models: the independent model and the stacking method. We first experiment with synthetic data, which allow us to generate multiple identically distributed network samples for training and testing, so we can obtain the error bars of classfication error of various methods. We then test the approaches on three real network datasets. In all experiments, we build a relational Markov network on the networks. Two clique templates are specified. The singleton clique is defined on each instance,

Algorithm 1 Parameter estimation for the local mixture model μ

Input: Training network G, attributes $\mathbf{x}_G$ and labels $\mathbf{y}_G$.
Output: The mixture model μ (consisting of: the MPLE model $P_{\boldsymbol{\theta}}$, the low propagation model $\tilde{P}$ and the optimal meta parameter $(\hat{\tau}, \hat{\kappa}_0)$).
Learn the MPLE parameters $\boldsymbol{\theta}$.
Learn a low propagation model (eg. independent, stacking, etc.) $\tilde{P}$.
Initialize $lowestError = \infty$.
for (τ, κ_0) from a candidate set **do**
  Initialize $error = 0$.
  for $labeledProportion = 0.0, 0.1, \ldots, 0.9$ **do**
    a) Randomly select $labeledProportion$ of training instances as labeled set L.
    b) Run collective inference by Algorithm 2 to obtain samples $\mathbf{y}^{(1)}_{G\setminus L}, \mathbf{y}^{(2)}_{G\setminus L}, \ldots, \mathbf{y}^{(t)}_{G\setminus L}$ from the mixture model.
    c) Evaluate $inferenceError$ from the approximate marginals computed using the samples $\mathbf{y}^{(1)}_{G\setminus L}, \mathbf{y}^{(2)}_{G\setminus L}, \ldots, \mathbf{y}^{(t)}_{G\setminus L}$.
    d) $error = error + inferenceError$.
  end for
  if $error < lowestError$ **then**
    Set $lowestError = error$, $\hat{\tau} = \tau$, and $\hat{\kappa}_0 = \kappa_0$.
  end if
end for

Algorithm 2 Collective inference using the local mixture model μ

Input: Test network G, attributes $\mathbf{x}_G$ and a set of observed labels $\mathbf{y}^*_L$. The local mixture model μ.
Output: Samples $\mathbf{y}^{(1)}_{G\setminus L}, \mathbf{y}^{(2)}_{G\setminus L}, \ldots, \mathbf{y}^{(t)}_{G\setminus L}$ from the inferential distribution $P(\mathbf{y}_{G\setminus L} | \mathbf{x}_G, \mathbf{y}^*_L)$.
for $i \in G \setminus L$ **do**
  a) Compute the propagation upper bound $\hat{\kappa}_i$ by Equation (11).
  b) Compute the latent confidence λ_i by Equation (13).
  c) Compute the CPD $\mu(y_i)$ for every $y_i \in \mathcal{Y}$ by Equation (12).
end for
Use the Gibbs sampler with the CPDs $\mu(y_i)$ for $i \in G \setminus L$ to generate the samples $\mathbf{y}^{(1)}_{G\setminus L}, \mathbf{y}^{(2)}_{G\setminus L}, \ldots, \mathbf{y}^{(t)}_{G\setminus L}$.

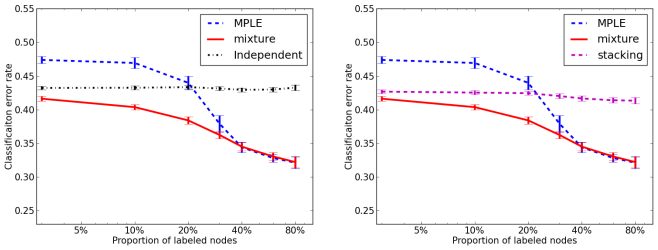

Figure 2: Synthetic data, low linkage. Average over 20 training/test pairs of synthetic networks.

and the potential is a linear combination of the attributes with the weights depending on different values of the label y_i. The edgewise clique is defined on each edge, for which the potential is a weighted indicator function of whether the two related instances have the same label or not.

6.1 Synthetic data

Our synthetic data experiments are based on a latent group model [13], which simulates the autocorrelations and community structures observed in real networks, and by which we can easily vary the linkage in the data. Figures 2(a) and 2(b) show the performance of using independent learning and stacking respectively as the low error model in relatively low link density networks. Figures 2(a) and 2(b) depict the relatively high link density case. We observe that in all cases the error rate of the low error model is lower at the beginning, but it decreases much more slowly than the collective approach, or remains constant (in the independent learning case). Clearly, as the amount of labeled instances grows from small to large, the propagation among unlabeled instances becomes weaker and the relative performance of MPLE and low propagation error models invert. However, the mixture model is able to achieve constantly lowest error rate due to its ability to identify the critical points of propagation error by estimating the latent variable of propagation confidence. In addition, the difference between MPLE and low propagation models is more significant when the link density of the graph is high.

6.2 Real data

We now test these methods on three real datasets. We find that the performance of these methods is quite consistent with our understanding about propagation error and our observation on the synthetic data.

The first dataset was collected by the WebKB Project [3]. The data consist of a set of 3,877 web pages from four computer science departments, labeled with the categories: course, faculty, staff, student, research project, or other. We considered the unipartite co-citation web graph which include all the categories except for "other". We then test the various methods for multiclass classification. The result is shown in Figure 4. Due to the strong dependency caused by the high-linkage of this co-citation graph, the collective inference with MPLE approach tends to assign high probability of the observed class labels to all unlabeled instances in the graph when the labeled proportion is small, which leads to severely poor performance. On the other hand, it also wins over low propagation methods by a large margin on this dataset when the proportion of labeled instances is more than 30%. By identifying and correcting for the excessive propagation of MPLE, the mixture model achieves the lowest error across the whole spectrum.

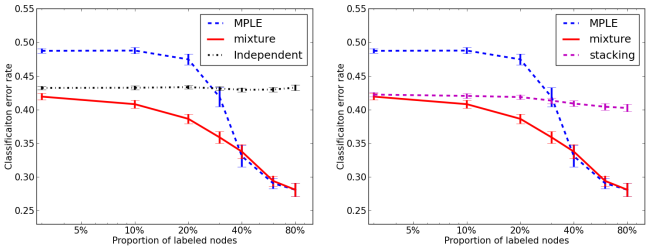

Figure 3: Synthetic data, high linkage. Average over 20 training/test pairs of synthetic networks.

The second dataset is drawn from Facebook. It includes user profile attributes (gender, relationship status, political and religious views), as well as friendship links and transactions (wall posting, picture tagging and common group memberships) among users. Our sample network consists of 7,315 users, which comprises all the students and alumni, with public profiles, from a large "University" network. We further divide the network into 8 articulated subnetworks of comparable sizes for training and testing (e.g., "Class of 2008"). We perform a binary classification task: predicting whether a user's political view is "conservative" or not. We construct relational graphs based on the friendship, wall posting, picture tagging and group sharing links. We train Markov networks on the relational graph, using gender, relationship status and religious view as attributes. Figure 6.2 shows that on this dataset, the MPLE approach performs worse than the low propagation approach when the amount of labeled data is small to moderate. Again the mixture approach improves over both approaches.

Finally, the third dataset is drawn from the Internet Movie Database (www.imdb.com).We used a sample of 1,382 movies released in the U.S. between 1996 and 2001. The binary classification task is to predict movie opening weekend returns ($> \$2$ million). We considered a unipartite graph of movies, where links indicate that the movies share a common actor, producer, director, studio or editor. We build the Markov network based on this graph and two movie genre attributes. Figure 6 shows that the low propagation approaches is worse than MPLE across the whole spectrum, indicating that the average effect of over propagation cannot offset the advantage of the accurate local relational model. However, the mixture approach is still able to improve over MPLE in the regime of small to moderate amount of observed labels. This is due to two facts: First, the curves are averaged over multiple subnetwork splits and random label selections. The mixture model eliminates the occasional pathological performance of MPLE and thus improves it on average. Second, the mixture is defined for the CPDs, so the overall model benefits from targeted adjustments of local propagations.

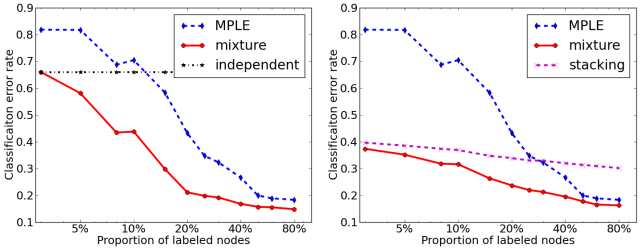

Figure 4: Classification error on the WebKB network. 4 training/test splits are formed based on the 4 school subnetworks. 10 random label selections are run at each labeled proportion.

7. RELATED WORK

A great deal of research focus in the statistical relational learning community has centered on collective classification. Probabilistic models such as Relational Bayesian Networks [6], Relational Markov networks [18], Markov Logic Networks [16], and Relational Dependency Networks [15] have been proposed and widely used for collective classification tasks. Their empirical performance has also been studied (see e.g., [11]).

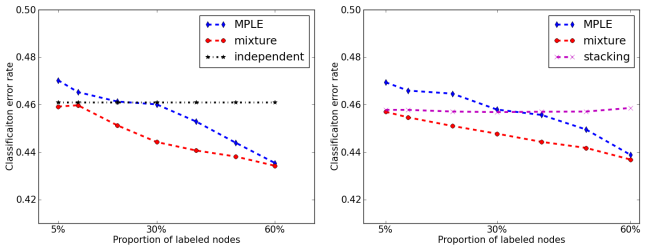

Figure 5: Classification error on the Facebook network. Results are averaged over 4 training/test subnetwork splits and 10 random label selections at each labeled proportion.

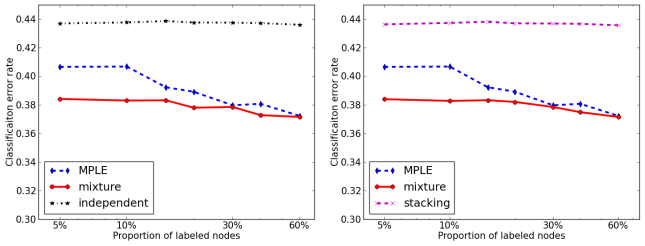

Figure 6: Classification error on the IMDB network. Results are averaged over 4 training/test subnetwork splits and 10 random label selections at each labeled proportion.

There also exist other work which formally analyze the inference error in collective classification [14, 5]. However, these papers attempt to character the performance of relational models from a bias/variance analysis perspective, while the current paper addresses inference error in the context of varying amount of observed test set labels, and suggest ways to improve performance of relational methods.

Another line of research in collective classification that stresses the importance of observed test set labels is the field of active label acquisition. For example, Bilgic and Getoor [1] studied several active label acquisition techniques in probabilistic relational models. Macskassy [12] investigated active labeling in Gaussian field models for network data.

Many parameter estimation techniques that take global propagation into account during learning have been proposed, and can be categorized as low propagation models in our context. Examples of this type that have been applied to a relational learning/collective classification tasks before include the stacking approach [8], the scaled conjugate gradient algorithm [10], and virtual evidence boosting [9]. It would be interesting to examine the performance of using the later models as the low propagation component in our mixture model.

8. CONCLUSIONS AND FUTURE WORK

In this work, we study collective inference error of different models obtained by different parameter estimation techniques in the context of varying availability of test set labels. We observe that the relative performance of different learning approaches is *inconsistent* across the spectrum of label availability. Specifically, we found that the maximum pseudolikelihood estimation performs remarkably well when there are abundant observed test set labels,

but rather poorly when there are very few seeding labels. We analyze the cause of this phenomenon in single network relational classification tasks. This points to a previously unidentified trade-off between different parameter estimation methods: a trade-off based on the test set label availability. We address this trade-off by analyzing the propagation error of MPLE, and propose a mixture modeling approach to achieve a good trade-off between high propagation and low propagation models. The experimental results on both synthetic and real data confirmed our findings, and demonstrated consistently superior performance of the mixture model.

There are a number of future directions to explore with this work. First, while we have developed a mixture model in view of the fact that MPLE better approximates the optimal prediction in the label abundant case while low propagation models better match the label scarce case, is there a direct model that serves for the same purpose and can be efficiently learned from the training network? Second, we would like to gain further understanding into the inference error by incorporating the analysis of base error. This may help us tighten the upper bounds or develop average case bounds on error, and provide novel ways to apply probabilistic relational models for collective classification. Furthermore, the microscopic dependency quantities used in our analysis may find further usage in statistical relational learning, e.g., through the development of active inference methods based on the microscopic dependence matrix.

Acknowledgments

This research is supported by NSF under contract numbers SES-0823313, IIS-1017898, CCF-0939370. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of NSF or the U.S. Government.

9. REFERENCES

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

[2] Olivier Bousquet, StŐphane Boucheron, and GĞbor Lugosi. Introduction to statistical learning theory. In *In , O. Bousquet, U.v. Luxburg, and G. Rsch (Editors*, pages 169–207. Springer, 2004.

[3] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery. Learning to extract symbolic knowledge from the World Wide Web. In *AAAI*, 1998.

[4] P. L. Dobrushin. The description of a random field by means of conditional probabilities and conditions of its regularity. *Theory Probab. Appl.*, 13:197–224, 1968.

[5] Andrew Fast and David Jensen. Why stacked models perform effective collective classification. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2008.

[6] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *IJCAI*, 1999.

[7] Hans-Otto Georgii. *Gibbs measures and phase transitions.* Walter de Gruyter, 1988.

[8] Zhenzhen Kou. Stacked graphical models for efficient inference in markov random fields. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, 2007.

[9] Lin Liao, Tanzeem Choudhury, Dieter Fox, and Henry Kautz. Training conditional random fields using virtual evidence boosting. In *In Proc. of the International Joint Conference on Artificial Intelligence (IJCAI*, 2007.

[10] Daniel Lowd and Pedro Domingos. Efficient weight learning for markov logic networks. In *In Proceedings of the Eleventh European Conference on Principles and Practice of Knowledge Discovery in Databases*, pages 200–211, 2007.

[11] S. Macskassy and F. Provost. Classification in networked data: A toolkit and a univariate case study. *Journal of Machine Learning Research*, 8(May):935–983, 2007.

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

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

[14] J. Neville and D. Jensen. A bias/variance decomposition for models using collective inference. *Machine Learning*, 73:87–106, 2008.

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

[16] M. Richardson and P. Domingos. Markov logic networks. *Mach. Learn.*, 62(1-2):107–136, 2006.

[17] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Galligher, and Tina Eliassi-Rad. Collective classification in network data. *Ai Magazine*, 29(3), 2008.

[18] B. Taskar, P. Abbeel, and Daphne Koller. Discriminative probabilistic models for relational data. In *UAI*, 2002.

[19] R. Xiang and J. Neville. Relational learning with one network: An asymptotic analysis. In *AISTATS*, 2011.

[20] Bin Yu. Rates of convergence for empirical processes of stationary mixing sequences. *Ann. Probab.*, 22(1), 1994.