On the Mismatch Between Learning and Inference for Single Network Domains

Rongjing Xiang and Jennifer Neville RXIANG | NEVILLE@CS.PURDUE.EDU
Department of Computer Science, Purdue University, West Lafayette, IN, USA

Abstract

We observe that the relative performance of statistical relational models learned with different estimation methods changes as the availability of test set labels increases, when learning from single networks. We reason about the cause of this phenomenon and develop an inference error characterization, which leads us to propose a mixture model that can learn the best trade-off between different parameter estimates.

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 (Friedman et al., 1999; Taskar et al., 2002; Richardson & Domingos, 2006; Neville & Jensen, 2007).

A number of learning algorithms have been developed for probabilistic relational models (see e.g. Neville & Jensen, 2007; Lowd & Domingos, 2007; Liao et al., 2007; Kou, 2007), among which the two most representative approaches are MLE and maximum pseudo-likelihood estimation (MPLE). In domains where data instances are independent and identically distributed (i.i.d.), MPLE can be viewed as an efficient approximation of 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 inter-dependent instances. Likewise, the classic statistical optimality of MLE no longer applies for these single networks. For this reason, a more careful examination of MPLE- and MLE-type estimation is warranted for relational settings.

Many recent empirical results show that relational model performance can vary based on the amount and spread of observed class labels on the test network (see e.g. Macskassy & Provost, 2007). We find that MPLE and MLE-type algorithms achieve superior performance in different regimes of the label availability spectrum. This is due to the fact that there is a mismatch between the learning objective and the conditional likelihood used during inference for partially labeled networks in either the MPLE or the MLE approach. Nevertheless, the MPLE objective is closer to the ideal objective function at one end of the label availability spectrum, while the MLE objective is better at the other end. We further characterize different learning methods by the nature of the mismatch. Models estimated by MPLE tends to result in higher global dependencies between distant labels in the test network when the labeled nodes are sparse, and we thus call them *high propagation* models. On the other hand, MLE-type algorithms estimate dependencies more conservatively, and we call them *low propagation* models. Our observation and analysis therefore add a new dimension to the comparison between different learning approaches beyond the traditional trade-off between accuracy and efficiency. We then use this key insight to develop a mixture model that can automatically choose locally and dynamically between low propagation and high propagation models to correct for the mismatch. Empirical evaluation demonstrates that this preliminary solution can achieve comparable, or superior, results to both MPLE and low propagation models across the whole spectrum of test set label availability.[1]

Presented at the International Conference on Machine Learning (ICML) workshop on *Inferning: Interactions between Inference and Learning*, Edinburgh, Scotland, UK, 2012.

[1] Proofs, part of the results, and discussions are omitted in this extended abstract due to space limits. See the full version (Xiang & Neville, 2011b) for more details.

2. Background

To frame our analysis and algorithm development, we outline a general probabilistic modeling formulation for 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 the existence of a relational structure over the data instances. To encode this, we pre-specify a set $\mathcal{T}$ of clique templates, which represent relationships of a particular type (e.g., friendship relations). Within each template type $T \in \mathcal{T}$ there is a set $\mathcal{C}(T)$ of cliques, where each clique $C \in \mathcal{C}(T)$ is an observed relation among 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.

$$P(\mathbf{y}_G|\mathbf{x}_G) = \frac{1}{Z(\boldsymbol{\theta}, \mathbf{x}_G)} \prod_{T \in \mathcal{T}} \prod_{C \in \mathcal{C}(T(G))} \exp \langle \boldsymbol{\theta}_T, \boldsymbol{\phi}_T(\mathbf{x}_C, \mathbf{y}_C) \rangle$$

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, in this template formulation the parameter $\boldsymbol{\theta}$ of cliques within the same template is homogeneous, which makes learning and generalization possible. The feature mapping $\boldsymbol{\phi}_T$ is predefined and are computed from the vector of attributes and labels within the corresponding clique C. This general formulation encompasses a rich class of probabilistic relational models in the literature, including Relational Markov Networks (Taskar et al., 2002), Markov Logic Networks (Richardson & Domingos, 2006), and Relational Dependency Networks (Neville & Jensen, 2007).

To apply the learned model for collective classification, the inference algorithm takes a partially labeled test network G (with L denoting the labeled set and $G \setminus L$ denoting the unlabeled set), the set of observed 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 $P(y_i|\mathbf{y}^*_L)$[2] from these samples. The error of the collective classification model is simply computed as the per instance error rate:

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

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

Parameter Learning in Probabilistic Relational Models. The maximum likelihood estimation (MLE) for the above model can be written as the following optimization problem:

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

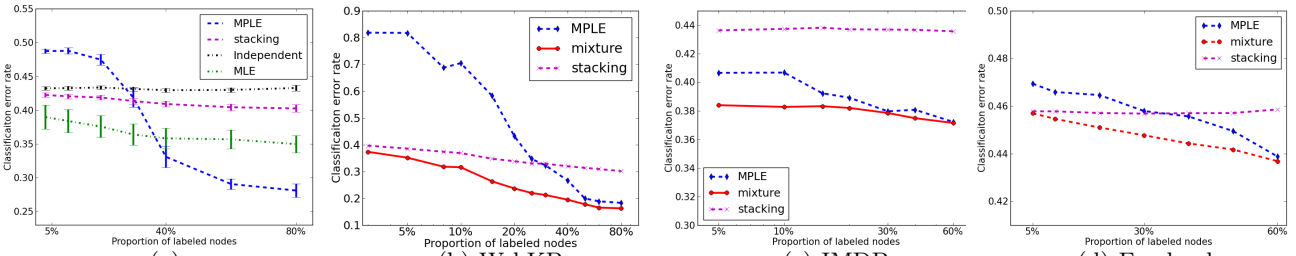
The MLE is generally intractable for large networks due to the normalization factor Z. Another straightforward method for parameter estimation is the maximum pseudolikelihood estimation (MPLE). Due to its efficiency, MPLE is widely applied to relational data in practice (Neville & Jensen, 2007). Let ∂i denote the Markov blanket of i, i.e., the set of instances that share a clique C with i. The MPLE optimizes the product of local conditional probability distributions (CPDs) $P(y_i|x_i, x_{\partial i}, y_{\partial i})$:

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

where φ_i denotes the local potentials of instance i, i.e., the summation of the potentials of all cliques that involve i. Since the global normalization Z is replaced by the local normalization $\log \sum_{y_i, \mathbf{y}_{\partial i}} \exp(\varphi_i)$, exact optimization of MPLE is usually tractable. To facilitate the analysis in this paper, we further decompose the local potentials into self potentials φ^S_i, which are only a function of y_i and attributes $\mathbf{x}_G$, and interaction potentials φ^I_i, which also depend on neighboring labels $y_j : j \in \partial i$. Thus $\varphi_i = \varphi^S_i + \varphi^I_i$.

3. A comparison of different parameter estimation methods

Although the availability of test set labels (i.e., $|L|$) has been regarded as an important factor in determining collective classification performance, there has been little work investigating the impact of different labeling scenarios on *learning*, with the exception of (Kou, 2007; McDowell et al., 2009). The *stacked* modeling approach to relational learning (Kou, 2007) is often credited for adjusting for the mismatch in label availability between training data and test data that occurs in MPLE-type approaches (Fast & Jensen, 2008). In this section, we seek to understand the reasons for performance differences among the various parameter estimation methods, when the amount of observed test set labels is varied. In Figure 1(a), we plot the classification error of MPLE, MLE, *independent* learning, and *stacking* on synthetic network datasets, for varying amounts of labeled test instances. All methods have the same form of local potentials, except for the



(a) (b) WebKB (c) IMDB (d) Facebook

Figure 1. (a): Comparision of various parameter learning methods on synthetic data. (b)-(d): Evaluation of the proposed mixture approach on real network datasets.

independent model, which is equivalent to a logistic regression model that applies the same form of self potentials φ^S as in the relational models, but does not contain the interaction potentials φ^I.

Why does MPLE outperform MLE in the region of large labeled proportions? The superior performance of MPLE is a result of learning the parameters from a single network and then applying them for prediction in a test network with partially observed 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 $\pi(\mathbf{x}_G, \mathbf{y}_G)$. The MLE attempts to minimize $KL(\pi(\mathbf{y}_G|\mathbf{x}_G)\|P_{\boldsymbol{\theta}}(\mathbf{y}_G|\mathbf{x}_G))$ via minimizing $KL(\hat{\pi}(\mathbf{y}_G|\mathbf{x}_G)\|P_{\boldsymbol{\theta}}(\mathbf{y}_G|\mathbf{x}_G))$. While there is no theoretical guarantee in general situations on this minimization using only a single training network in the first place, even if a bound on $KL(\pi(\mathbf{y}_G|\mathbf{x}_G)\|P_{\boldsymbol{\theta}}(\mathbf{y}_G|\mathbf{x}_G))$ can be obtained under certain assumptions, in the label abundant region the more suitable objective should be $KL(\hat{\pi}(y_i|x_i, \mathbf{x}_{\partial i}, \mathbf{y}_{\partial i})\|P_{\boldsymbol{\theta}}(y_i|x_i, \mathbf{x}_{\partial i}, \mathbf{y}_{\partial i}))$. This mismatch is the main factor that leads to the inferior performance of MLE.

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., $KL(\hat{\pi}(y_i|x_i, \mathbf{x}_{\partial i}, y_{\partial i})\|P_{\boldsymbol{\theta}}(y_i|x_i, \mathbf{x}_{\partial i}, y_{\partial i}))$.[3] Under a stationarity assumption that postulates $P(x_i, y_i, \mathbf{x}_{\partial i}, \mathbf{y}_{\partial i})$ to be homogenous for any i, $KL(\pi(y_i|x_i, \mathbf{x}_{\partial i}, \mathbf{y}_{\partial i})\|P_{\boldsymbol{\theta}}(y_i|x_i, \mathbf{x}_{\partial i}, \mathbf{y}_{\partial i}))$ can be effectively bounded using $KL(\hat{\pi}(y_i|x_i, \mathbf{x}_{\partial i}, \mathbf{y}_{\partial i})\|P_{\boldsymbol{\theta}}(y_i|x_i, \mathbf{x}_{\partial i}, \mathbf{y}_{\partial i})) + \epsilon(\tilde{n})$ (see e.g. Yu, 1994). 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 $P(y_i|\mathbf{x}, y_{\partial i})$.

Why does MPLE perform poorly in the region 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 attributes 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 well when we run inference using these CPDs separately on each instance, as in the case when each instance's neighbors are fully labeled. However, problems arise when we apply these CPDs collectively for inference when there are many unobserved nodes since the local dependencies would be propagated globally through unlabeled nodes. We will analyze this *propagation error* of collective inference with MPLE in more detail in Section 4.

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 family, which in expectation best predicts the labels in the test network across all scenarios, 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. 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 with predefined potential functions (Xiang & Neville, 2011a). Unfortunately, in practice though, the model family is unlikely to be well specified. In these scenarios, the optimal parameter $\hat{\boldsymbol{\theta}}^L$ which makes the model $P_{\boldsymbol{\theta}}(\mathbf{y}_{G\setminus L}|\mathbf{y}^*_L, \mathbf{x})$ closest to the true distribution $\pi_{\boldsymbol{\theta}}(\mathbf{y}_{G\setminus L}|\mathbf{y}^*_L, \mathbf{x})$ for a labeled set L

[3] The *effective sample size* $\tilde{n}$ is typically much smaller than the number of CPDs, n, due to the dependence between the CPDs. Nevertheless, under weak dependence assumptions such as exponential correlation decay , $\tilde{n}$ does increase with n.

(i.e., $\hat{\boldsymbol{\theta}}^L = \text{argmin}_{\boldsymbol{\theta}} KL(P_{\boldsymbol{\theta}}(\mathbf{y}_{G\setminus L}|\mathbf{y}^*_L, \mathbf{x}) \| \pi(\mathbf{y}_{G\setminus L}|\mathbf{y}^*_L, \mathbf{x}))$) varies with L. For example, when L contains 1% of the 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 recent research 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 regions, but are inferior to MPLE in the large labeled proportion regions, as indicated by the curve of the stacked learning approach in Figure 1(a). 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 different learning/inference mismatch mechanisms associated with different approaches, there is an opportunity of taking full advantage of both MPLE and low propagation models. The rest of this paper provides an initial exploration in this direction.

4. Inference Error Analysis of MPLE

To gain further understanding into collective classification error using MPLE, following the error rate defined by (1), consider the following error decomposition[4] for each instance:[5]

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

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 observed. Since MPLE optimizes for this scenario, the propagation error term is positive with high probability for any labeling situation. 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 reflects the mismatch between learning and inference due to the partially observed test set labels.

To analyze the propagation error, we borrow the method of microscopic dependencies from the research on Gibbs measures in statistical physics (Dobrushin, 1968). Here we define the microscopic dependency σ_{ij} as the maximum oscillation of the conditional probability $P(Y_i|\mathbf{y}_{G\setminus i})$, when only y_j is varied. Formally,

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

We can show that the following inequality holds:

$$\gamma_i \leq \left(\sum_{j \in \partial i \cap G\setminus L} \sigma_{ij} \right) \max_{j \in \partial i \cap G\setminus L} \epsilon_j \tag{2}$$

In this way, we have decomposed the propagation error of i along the edges from i to *unlabeled* nodes j. By iteratively applying this inequality, we can see that the inference error of each node is propagated throughout the whole network. Therefore, the prediction error depends on 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 the proportion of error on neighboring nodes that is propagated to node i: $\kappa_i := \sum_{j \in \partial i \cap G\setminus L} \sigma_{ij}$. We can use the oscillation in potential functions to upper bound oscillation of probabilities in order to efficiently evaluate the local propagation effect. $\hat{\kappa}_i$ defined as follows is an upper bound on κ_i:

$$\hat{\kappa}_i = \frac{1}{4} \sum_{j \partial i \cap G\setminus L} \max_{y^1_j, y^2_j, y_i, \mathbf{y}_{\partial i \cap G\setminus(L\cup\{j\})}} \left[\varphi^I(y_i, y^1_{j_k}, \mathbf{y}_{j_1^{t_i}\setminus j_k}) - \varphi^I(y_i, y^2_{j_k}, \mathbf{y}_{j_1^{t_i}\setminus j_k}) \right] \tag{3}$$

5. A Local Mixture Approach

Based on the above analysis, we propose a new approach to learning collective classification models. The purpose of this approach is to combine the strength of MPLE with any low propagation model to reduce the mismatch between the estimated and the real dependence strength among unlabeled nodes, so that the resulting algorithm achieves consistently low error across the full range of label availability 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:

$$\begin{aligned}\mu(y_i) =& \lambda_i P_{\boldsymbol{\theta}}(y_i|x_i, y_i, \mathbf{x}_{\partial i}, \mathbf{y}_{\partial i}) \\ &+ (1 - \lambda_i)\tilde{P}(y_i|x_G, \mathbf{y}_{G\setminus i})\end{aligned} \tag{4}$$

where the mixture coefficient λ_i is a latent variable which represents the confidence of propagation by MPLE in predicting y_i. When the test network is fully

[4] This is not to be confused with the collective classification error decomposition in previous work (Neville & Jensen, 2008; Fast & Jensen, 2008) that uses squared loss to facilitate bias/variance analysis.

[5] 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.

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 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:

$$\lambda_i = \exp\{-\tau \max(\hat{\kappa}_i - \kappa_0, 0)\} \quad (5)$$

Instead of the propagation upper bound $\hat{\kappa}$, one may propose to directly use the labeled proportion 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 directly reflect the propagation error since different labeling schemes with the same labeled proportion may result in very different propagation strengths in the network. For example, propagation effects from a randomly distributed set of labels will be quite different from the propagation effects when the labels are in a contiguous subgraph (e.g., from snowball sampling). We also note that some *active inference* methods that query nodes to label (e.g., AIGA method in (Bilgic & Getoor, 2008)) can 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 that are 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 thus develop a simulation method with simple grid search to experimentally validate the model, while leaving the investigation of more sophisticated methods as future work. The details are described in Algorithms 1 and 2. The advantage of this mixture model is that it allows us to dynamically adjust the level of label propagation 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 in the network. We demonstrate the effectiveness of this approach empirically on three real network data sets as shown in Figure 1(b)- 1(d), where we use the stacking approach as the component low propagation model.

Acknowledgements

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.

Algorithm 1 Parameter estimation for mixture model

Input: Training network G, attributes $\mathbf{x}_G$ and labels $\mathbf{y}_G$.
Output: The mixture model μ (the MPLE model $P_{\boldsymbol{\theta}}$, the low propagation model $\tilde{P}$, and meta parameters $(\hat{\tau}, \hat{\kappa}_0)$).
Learn the MPLE model $P_{\boldsymbol{\theta}}$.
Learn a low propagation model $\tilde{P}$ (e.g., independent, stacking).
for $labeledProportion = 0.0, 0.1, \ldots, 0.9$ **do**
  a) Randomly select $labeledProportion$ of training instances as labeled set L.
  b) Compute the propagation upper bound $\hat{\kappa}_i$ for all instances i by Equation (3).
end for
Sort the list of all $\hat{\kappa}_i$'s obtained from the above step.
Initialize $lowestError = \infty$.
for $\tau \in \{0.01, 0.1, 0.5, 1.0\}$, $\kappa_0 \in$ [values ranked at $\{10\%, 30\%, 50\%, 70\%, 90\%\}$ of the sorted $\hat{\kappa}_i$ list] **do**
  for $labeledProportion = 0.0, 0.1, \ldots, 0.9$ **do**
    a) Randomly select $labeledProportion$ of training instances as labeled set L.
    b) Compute the propagation upper bound $\hat{\kappa}_i$ for all instances i by Equation (3).
    c) Initialize $error = 0$.
    d) Run collective inference by Alg. 2 to obtain the marginal predictive probabilities $P(y_i|\mathbf{x}_G, \mathbf{y}^*_L), i \in G \backslash L$ from the mixture model.
    c) Evaluate $inferenceError$ from the marginals, and let $error = error + inferenceError$.
  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 mixture model

Input: Test network G, attributes $\mathbf{x}_G$, set of observed labels $\mathbf{y}^*_L$, and the local mixture model μ.
Output: Predictive marginal probabilities $P(y_i|\mathbf{x}_G, \mathbf{y}^*_L)$.
for $i \in G \setminus L$ **do**
  a) Compute the latent confidence λ_i by Equation (5).
  b) Compute the CPD $\mu(y_i)$ for every $y_i \in \mathcal{Y}$ by Equation (4).
end for
Use the Gibbs sampler with the CPDs $\mu(y_i)$ for $i \in G \setminus L$ to generate samples, and compute marginals $P(y_i|\mathbf{x}_G, \mathbf{y}^*_L), i \in G \setminus L$ from the samples.

References

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

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

Fast, Andrew and Jensen, David. Why stacked models perform effective collective classification. In *ICDM*, 2008.

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

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

Liao, Lin, Choudhury, Tanzeem, Fox, Dieter, and Kautz, Henry. Training conditional random fields using virtual evidence boosting. In *IJCAI*, 2007.

Lowd, Daniel and Domingos, Pedro. Efficient weight learning for markov logic networks. In *PKDD*, pp. 200–211, 2007.

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

McDowell, L., Gupta, K., and Aha, D. Cautious collective classification. *Journal of Machine Learning Research*, 10:2777–2836, 2009.

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

Neville, Jennifer and Jensen, David. Relational dependency networks. *J. Mach. Learn. Res.*, 8:653–692, 2007. ISSN 1532-4435.

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

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

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

Xiang, R. and Neville, J. Understanding propagation error and its effect on collective classification. In *ICDM*, 2011b.

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