Composite Likelihood Data Augmentation for Within-Network Statistical Relational Learning

Joseph J. Pfeiffer III
Department of Computer Science
Purdue University
West Lafayette, IN
jpfeiffer@purdue.edu

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

Paul N. Bennett
Microsoft Research
Redmond, WA
paul.n.bennett@microsoft.com

***Abstract*—The prevalence of datasets that can be represented as networks has recently fueled a great deal of work in the area of Relational Machine Learning (RML). Due to the statistical correlations between linked nodes in the network, many RML methods focus on predicting node features (i.e., labels) using the network relationships. However, many domains are comprised of a *single*, *partially-labeled* network. Thus, relational versions of Expectation Maximization (i.e., R-EM), which jointly learn parameters and infer the missing labels, can outperform methods that learn parameters from the labeled data and apply them for inference on the unlabeled nodes. Although R-EM methods can significantly improve predictive performance in networks that are densely labeled, they do not achieve the same gains in sparsely labeled networks and can perform worse than RML methods.**

In this work, we show the fixed-point methods that R-EM uses for approximate learning and inference result in errors that prevent convergence in sparsely labeled networks. We then propose two methods that do not experience this problem. First, we develop a Relational Stochastic EM (R-SEM) method, which uses stochastic parameters that are not as susceptible to approximation errors. Then we develop a *Relational Data Augmentation* (R-DA) method, which integrates over a range of stochastic parameter values for inference. R-SEM and R-DA can use any collective RML algorithm for learning and inference in partially labeled networks. We analyze their performance with two RML learners over four real world datasets, and show that they outperform independent learning, RML and R-EM—particularly in sparsely labeled networks.

I. Introduction

As online social and information networks have grown in popularity, many domains now consist of a set of items (e.g., people, websites) connected by relationships (e.g., friendships, hyperlinks). These relationships typically encode statistical dependencies, or autocorrelation, between the linked items. *Relational Machine Learning* (RML) aims to leverage these correlations by collectively inferring (i.e., predicting) the attributes of unlabeled items throughout the network [1].

To make predictions, RML methods typically learn a model from observed network data, then the learned model is applied to *collectively* infer the unobserved labels in a network. RML methods typically utilize a *local conditional* model to represent the dependencies of a node's label with the features of neighboring nodes. Local models, such as Relational Naive Bayes (RNB) or Relational Logistic Regression (RLR), are combined with joint inference algorithms (e.g., Gibbs sampling [2] or Variational Mean Field (VMF) [3]) to *collectively* predict labels (see e.g., [4], [5]).

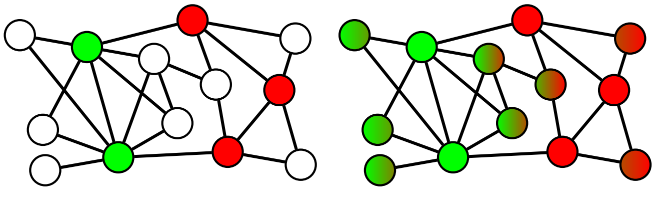


Fig. 1: (a) Partially labeled network. (b) Label Probabilities.

Many RML methods were developed for *across-network* classification, where there is an assumption of separate networks for learning and prediction. However, in practice many datasets are comprised of a *single*, partially-labeled network (Figure 1.a). For this type of domain, *within-network* classification methods, such as *Relational Expectation Maximization* (R-EM), are more appropriate. R-EM methods jointly infer the missing labels (Figure 1.b) in the network while estimating model parameters using a *fixed point* iterative framework.

R-EM generally outperforms traditional RML on within-network classification tasks [6], [7]. However, recent work has reported some problems with collective inference approaches in scenarios where the network is sparsely labeled ([6], [8]). More specifically, [6] showed that fixed point parameters learned through Maximum Composite Likelihood Estimation (MCLE)[1] can create *over propagation* error when performing inference in sparsely labeled networks. Although R-EM methods can significantly improve predictive performance in networks that are densely labeled, they do not achieve the same gains in sparsely labeled networks and can perform worse than RML methods [9]. Since many single-network domains are sparsely labeled, this presents a significant impediment to the adoption of relational methods.

In this paper, we investigate this issue in more detail. First, we introduce the *Relational Stochastic EM* (R-SEM) and *Relational Data Augmentation* (R-DA) approaches for within-network statistical relational learning (Figure 2). Our R-SEM method utilizes samples from the joint distribution to

[1] RML literature generally refers to this as the pseudolikelihood, but composite likelihood is more accurate due to the sole maximization of labeled components given their Markov blankets.

Parameters	Predictions	
	Fixed Point	Stochastic
Fixed Point	R-EM	–
Stochastic	**R-SEM**	**R-DA**

Fig. 2: Comparison of alternatives for incorporating estimates into within-network learning. We introduce R-SEM and R-DA.

iteratively maximize the MCLE, rather than using approximations of the expectations as in R-EM. Our R-DA method moves beyond the fixed point parameters used to make final predictions in both R-EM and R-SEM, by integrating over the posterior distribution of parameters for a stochastic estimate. For R-DA, we provide the corresponding composite likelihood sampling distributions for the RNB and RLR conditional distributions. Further, we provide evidence that substituting the Maximum a Posteriori (MAP) provides a good approximation for distributions where the posterior cannot be easily sampled.

Next, we demonstrate how the *structure* of a network directly impacts the quality of the estimates produced by RML and R-EM. Namely, we demonstrate how applying fixed point MCLE parameters for collective inference leads to distributions of labels that are far from the correct distribution—in many cases the inferred labels are primarily comprised of a single class label. First, we show that the samples drawn from the joint distribution of unlabeled items through Gibbs sampling do not empirically mix (converge) to the correct label distribution. Second, we show how the correct inference solution for VMF can be cast as an equilibrium state of a Nonlinear Dynamical System. By analyzing the first eigenvalue of the solution vector, we show that for sparsely labeled networks the inference method might not converge to a stable solution. Further, even if it does converge to a stable solution, using the predictions to relearn the parameters through MCLE (as is done with R-EM) commonly results in widely varying parameter estimates. Due to these approximation errors, R-EM is no longer covered by EM's guarantees (i.e., [10]) and does not converge.

The contributions of this paper include:

- Introduction of the R-DA and R-SEM within-network learning methods. In particular, R-DA is a Bayesian approach that infers over a distribution of parameters, rather than a fixed point estimate.
- Analysis of the extremums found through Gibbs sampling and VMF inference using parameters learned by MCLE.
- Demonstration that these extremums interfere with the R-EM learning and inference algorithm, showing that R-EM does not converge in sparsely labeled networks.
- Theoretical and empirical motivation for the substitution of samples from the posterior with the MAP for R-DA.

In the next section we give notation and background. In Section III we introduce the proposed R-SEM and R-DA algorithms, and in Section IV we demonstrate that parameters learned through MCLE during R-EM do not converge. Section V demonstrates R-DA and R-SEM's improvement over existing methods on four real world networks. In Section VI we discuss further related work and in Section VII we conclude.

II. Notation and Background

Define a graph $G = \langle \mathbf{V}, \mathbf{E} \rangle$, where $v \in \mathbf{V}$ correspond to vertices and $\mathbf{E} \subseteq \mathbf{V} \times \mathbf{V}$ correspond to edges (or relationships) between the vertices. Let $\mathbf{X}, \mathbf{Y}$ define the sets of *attributes* and *labels*. Every $v_i \in \mathbf{V}$ has a corresponding set of attributes $\mathbf{x}_i \in \mathbf{X}$ and a class label $y_i \in \mathbf{Y}$. We divide the vertices into two disjoint sets: the labeled set L consists of the items where the labels are known, or observed, and the unlabeled set U, where the labels are unknown. Subscripts refer to a subset of associated items from the full graph (e.g., $\mathbf{Y}_L$ and $\mathbf{Y}_U$ refer to item labels in the labeled and unlabeled sets, respectively).

The primary goal of *within-network* relational learning is to jointly infer the unknown labels of $\mathbf{Y}_U$ given the labeled data $\mathbf{Y}_L$, attributes $\mathbf{X}$ and network G: $P(\mathbf{Y}_U|\mathbf{Y}_L, \mathbf{X}, G)$. This contrasts with standard machine learning, where the primary goal is to estimate the parameters $\Theta_{\mathcal{C}}$ of a model $\mathcal{C}$, which are then applied to infer future samples.

A general assumption within RML is that a label $y_i \in \mathbf{Y}$ only depends on the attributes $\mathbf{x}_i \in \mathbf{X}$ and the direct neighbors in the graph. Define the Markov Blanket (e.g., *neighbors*) of a vertex $v_i \in \mathbf{V}$: $\mathcal{MB}(v_i) = \{v_j | (v_i, v_j) \in \mathbf{E}\}$. The corresponding conditional distribution for y_i is then $P(y_i|\mathbf{Y}_{\setminus i}, \mathbf{X}, G) = P(y_i|\mathbf{Y}_{\mathcal{MB}(v_i)}, \mathbf{x}_i, \Theta_{\mathcal{C}})$, which is a chosen local conditional model, such as RNB or RLR.

Collective inference algorithms (e.g., Gibbs Sampling or VMF) iteratively apply the learned local conditional parameters to each unlabeled item to jointly infer the set of unlabeled examples. For future equations, we omit the $\mathbf{X}$ variables as they are fixed and always conditioned on. Let $\Theta_{\mathcal{C}}$ indicate the parameters for a given model $\mathcal{C}$.

From within the frequentist framework, the basic approach to joint inference would correspond to performing:

$$\begin{aligned} &\textbf{Estimate Parameters: } \hat{\Theta}_{\mathcal{C}} = \arg\max_{\Theta_{\mathcal{C}}} P(\mathbf{Y}_L|\Theta_{\mathcal{C}}) \\ &\textbf{Perform Inference: } P(\mathbf{Y}_U|\mathbf{Y}_L, \hat{\Theta}_{\mathcal{C}}) \end{aligned} \tag{1}$$

The first step estimates parameters by *maximizing the likelihood* (i.e., MLE) of the observed data. A common approach is to replace the full likelihood with a corresponding *composite likelihood* for efficiency:[2]

$$\log\ P(\mathbf{Y}_L|\Theta_{\mathcal{C}}) \approx \sum_{y_i \in \mathbf{Y}_L} \log P(y_i|\mathbf{Y}_{\mathcal{MB}_L(v_i)}, \Theta_{\mathcal{C}})$$

Unlike MLE, *Maximum Component Likelihood Estimation* (MCLE) does not compute a partition function over all combinations of labels, which makes it tractable [4].

The second step *jointly infers* the unknown labels in a frequentist manner, using fixed point parameter estimates. Bayesian posterior inference would marginalize over the distribution of parameters $\Theta_{\mathcal{C}}$ given a prior with hyper parameter α:

$$P(\mathbf{Y}_U|\mathbf{Y}_L, \alpha) = \int P(\mathbf{Y}_U|\mathbf{Y}_L, \Theta_{\mathcal{C}}) P(\Theta_{\mathcal{C}}|\mathbf{Y}_L, \alpha)\ \mathrm{d}\Theta_{\mathcal{C}}$$

Since direct computation of this integral is generally hard, approximations are used by sampling from the posterior distribution[3] of $\Theta_{\mathcal{C}}$ (i.e., $P(\Theta_{\mathcal{C}}|\mathbf{Y}_L)$) and averaging the results.

[2] Here we use $\mathcal{MB}_L$ to refer to the Markov blanket in the labeled set.

[3] α is dropped for clarity; it always defines the $\Theta_{\mathcal{C}}$ prior.

Algorithm 1 LearningFromIncompleteData($\mathbf{W}_{obs}, \mathbf{W}_{mis}, \mathcal{C}$)

```
1: $\tilde{\Theta}_{\mathcal{C}}^{0}$ = InitialParameters($\mathbf{W}_{obs}, \mathcal{C}$)
2: while More Iterations or Not Converged do
3:     # The E/I Step, then the M/P Step
4:     $\tilde{P}^{t}(\mathbf{W}_{mis})$ = IterativeAssignment($\mathbf{W}_{obs}$, $\mathcal{C}$, $\tilde{\Theta}_{\mathcal{C}}^{t-1}$)
5:     $\tilde{\Theta}_{\mathcal{C}}^{t}$ = IterativeParameters($\mathbf{W}_{obs}, \tilde{P}^{t}(\mathbf{W}_{mis}), \mathcal{C}$)
6: $\hat{\Theta}_{\mathcal{C}}$ = FinalizeParameters($\mathbf{W}_{obs}, \tilde{P}^{1,\ldots,T}(\mathbf{W}_{mis}), \tilde{\Theta}_{\mathcal{C}}^{0,\ldots,T}, \mathcal{C}$)
7: $\hat{P}(\mathbf{W}_{mis})$ = FinalizeInference($\mathbf{W}_{obs}, \tilde{P}^{1,\ldots,T}(\mathbf{W}_{mis}), \hat{\Theta}_{\mathcal{C}}, \mathcal{C}$)
```

A. *General Learning from Incomplete Data*

For many problem domains, accurate estimation of a set of parameters $\Theta_{\mathcal{C}}$ can be difficult when given a set of partially observed data $\mathbf{W}_{obs}$. In these cases, incorporating latent variables into the model (representing the unobserved data $\mathbf{W}_{mis}$) can improve MLE or posterior estimates of $\Theta_{\mathcal{C}}$.

For domains with unknown latent variables, a general class of methods learn by iteratively evaluating both the latent variables $\mathbf{W}_{obs}$ and parameters $\Theta_{\mathcal{C}}$. Both the deterministic *Expectation Maximization* (EM) method [10] and the Bayesian Data Augmentation (DA) method [11] are examples of methods in this class. Algorithm 1 gives an overview of the general approach for a classifier $\mathcal{C}$.

The algorithm begins by assigning initial values to the parameters (Line 1). This assignment can be random or possibly something more clever if allowed by the domain. Lines 2-5 are the heart of the algorithm, which alternates between inferring the latent variables $\tilde{P}^{t}(\mathbf{W}_{mis})$ (Line 4) and estimating parameters $\tilde{\Theta}_{\mathcal{C}}^{t}$ (Line 5). This continues until convergence or for a fixed number of iterations (denoted T). Lastly, using the set of parameter estimates and latent variable evaluations from the iterations, the algorithm produces final estimates $\hat{\Theta}_{\mathcal{C}}$ and inferences $\hat{P}(\mathbf{W}_{mis})$ (Lines 6-7).

Expectation Maximization: The EM method is an iterative, deterministic method for learning with missing data [10]. Algorithm 1 decribes EM with the following specifications:

E-Step (Line 4): evaluate $\tilde{P}^{t}(\mathbf{W}_{mis}|\mathbf{W}_{obs}, \tilde{\Theta}_{\mathcal{C}}^{t-1})$

M-Step (Line 5): maximize for $\tilde{\Theta}_{\mathcal{C}}^{t}$

$$\arg\max_{\Theta_{\mathcal{C}}} \sum_{\mathbf{W}_{mis}} \tilde{P}(\mathbf{W}_{mis}|\mathbf{W}_{obs}, \tilde{\Theta}_{\mathcal{C}}^{t-1}) \log P(\mathbf{W}_{obs}, \mathbf{W}_{mis}^{t-1}|\Theta_{\mathcal{C}})$$

That is, each iteration first computes the expected values of the missing data, then maximizes the expected log likelihood (over the missing data). Each step maximizes a lower bound of the log likelihood and converges to a local maximum [12]. The estimated $\hat{\Theta}_{\mathcal{C}}$ is the final maximization of $\tilde{\Theta}_{\mathcal{C}}$ (Line 6) and $\hat{\mathbf{W}}_{mis}$ is finally inferred (Line 7) with $\hat{\Theta}_{\mathcal{C}}$ (i.e., $P(\mathbf{W}_{mis}|\mathbf{W}_{obs}, \hat{\Theta}_{\mathcal{C}})$).

For many domains, the 'E'-Step is intractable to compute exactly and various approximate inference techniques exist (e.g., [13], [14]). For example, the Stochastic EM (SEM) algorithm replaces the 'E' step with a sample from the conditional distribution $P(\mathbf{W}_{mis}|\mathbf{W}_{obs}, \tilde{\Theta}_{\mathcal{C}}^{t})$ [13]. Further, averaging over the collection of intermediate parameters can reduce the variance of the final parameter estimates. Note that approximations to the 'E' and 'M' steps do not necessarily carry the same convergence guarantees as the original EM.

Data Augmentation: The DA method is a stochastic Markov Chain Monte Carlo (MCMC) method for computing the joint posterior distributions of $\Theta_{\mathcal{C}}$ and $\mathbf{W}_{mis}$ [11]. Algorithm 1 also describes DA, but instead of the deterministic 'E' and 'M', DA has stochastic *Imputation* (I) and *Posterior* (P) steps in its specification:

I-Step (Line 4): sample $\tilde{\mathbf{W}}_{mis}^{t} \sim P(\mathbf{W}_{mis}|\mathbf{W}_{obs}, \tilde{\Theta}_{\mathcal{C}}^{t-1})$

P-Step (Line 5): sample $\tilde{\Theta}_{\mathcal{C}}^{t} \sim P(\Theta_{\mathcal{C}}|\mathbf{W}_{obs}, \tilde{\mathbf{W}}_{mis}^{t})$

The iterative sampling process forms two correlated Markov Chains from the posterior distributions of $P(\Theta_{\mathcal{C}}|\mathbf{W}_{obs})$ and $P(\mathbf{W}_{mis}|\mathbf{W}_{obs})$. DA can be viewed as a special case of the Gibbs sampler [2] in that both missing data and parameters are jointly sampled. As the samples are drawn from the joint distribution of unlabeled data and parameters, the final Maximum a Posteriori (MAP) estimates/inferences are:

Parameters (Line 6)	**Variables (Line 7)**
$\hat{\Theta}_{\mathcal{C}} \approx \frac{1}{T}\sum_{t} \tilde{\Theta}_{\mathcal{C}}^{t}$	$\hat{P}(\mathbf{W}_{mis}) \approx \frac{1}{T}\sum_{t} \tilde{P}^{t}(\mathbf{W}_{mis})$

B. *Relational Expectation Maximization*

The R-EM method is an application of EM to network domains [9]. In this case, the observed variables $\mathbf{W}_{obs}$ are the label $\mathbf{Y}_{L}$ and attributes $\mathbf{X}$, while the missing data $\mathbf{W}_{mis}$ are the unlabeled $\mathbf{Y}_{U}$. The 'E'-Step in Line 4 involves collective (i.e., joint) inference for $\tilde{P}^{t}(\mathbf{Y}_{U}|\mathbf{Y}_{L}, \tilde{\Theta}_{\mathcal{C}}^{t-1})$, which is intractable to compute exactly. Thus, the expectations are approximated using Gibbs sampling or VMF.

The form of the local conditional distribution (e.g., RNB or RLR) specifies the parameters $\Theta_{\mathcal{C}}$. For tractable estimation, R-EM uses MCLE rather than MLE for the 'M'-step, by assuming conditional independence of the unlabeled components:

R-M-Step (Line 5): maximize $\tilde{\Theta}_{\mathcal{C}}^{t}$ (2)

$$\arg\max_{\Theta_{\mathcal{C}}} \sum_{\mathbf{Y}_{U}} \prod_{y_i \in \mathbf{Y}_{U}} \tilde{P}^{t}(y_i|\tilde{\mathbf{Y}}_{\setminus i}, \tilde{\Theta}_{\mathcal{C}}^{t-1}) \sum_{y_j \in \mathbf{Y}_{L}} \log P(y_j|\tilde{\mathbf{Y}}_{\mathcal{MB}(v_j)}, \Theta_{\mathcal{C}})$$

To produce final parameter estimates $\hat{\Theta}_{\mathcal{C}}$ (e.g., for RNB or RLR) on Line 6, R-EM simply performs one additional learning step (e.g., $\hat{\Theta}_{\mathcal{C}} = \tilde{\Theta}_{\mathcal{C}}^{T}$). Lastly, R-EM performs one final round of collective inference with $\hat{\Theta}_{\mathcal{C}}$ to produce $\hat{P}(\mathbf{Y}_{U}) = P(\mathbf{Y}_{U}|\mathbf{Y}_{L}, \hat{\Theta}_{\mathcal{C}})$ (Line 7).

III. The Relational Stochastic EM and Relational Data Augmentation Methods

The R-EM method described above can be viewed as a series of iterative fixed point updates that incorporate $\mathbf{Y}_{U}$ into the learning process. Due to the complexity of real world networks, algorithms must use approximations for both the 'E' and 'M' steps. As a result, errors with either approximation can interfere with REM's fixed point estimates, which does occur in practice. Section IV explores this issue in more detail.

In this work, we propose two stochastic methods for within network learning and inference instead of using fixed point estimates: (1) Relational Stochastic EM (R-SEM) and (2) Relational Data Augmentation (R-DA). The differences between our proposed methods and conventional R-EM are shown in Table 2. Our proposed R-SEM method utilizes a fixed point $\hat{\Theta}_{\mathcal{C}}$ similar to R-EM to perform a final round of inference. But, R-SEM learns the parameters $\hat{\Theta}_{\mathcal{C}}$ by aggregating over a range of probable values, which reduces parameter estimation error compared to R-EM. Our proposed R-DA method does not use fixed point estimates when inferring $\hat{P}(\mathbf{Y}_U)$. Instead, R-DA performs inference by marginalizing over a distribution of parameters $\Theta_{\mathcal{C}}$, which makes it more robust than utilizing a single, fixed point estimate.

A. Relational Stochastic EM

Our first proposed method, R-SEM, is a stochastic version of the standard R-EM method, where the ‘E’-Step from R-EM is replaced with a stochastic ‘SE’-Step:

$$\begin{aligned}
&\textbf{SE-Step} \text{ (Line 4): sample } \tilde{\mathbf{Y}}_U \sim \tilde{P}^t(\mathbf{Y}_U|\mathbf{Y}_L, \tilde{\Theta}_{\mathcal{C}}^{t-1}) \\
&\qquad \text{(i.e.) sample } \tilde{y}_j \sim P(y_j|\tilde{\mathbf{Y}}_{\mathcal{MB}(v_j)}, \Theta_{\mathcal{C}}^{i-1}) \; \forall \; y_j \in \mathbf{Y}_U \\
&\textbf{M-Step} \text{ (Line 5): maximize } \tilde{\Theta}_{\mathcal{C}}^t \\
&\qquad \operatorname*{arg\,max}_{\Theta_{\mathcal{C}}} \sum_{y_j \in \mathbf{Y}_L} \log P(y_j|\tilde{\mathbf{Y}}_{\mathcal{MB}(v_j)}, \Theta_{\mathcal{C}})
\end{aligned} \tag{3}$$

For the SE-Step, we draw each $\tilde{y}_j$ according to the local conditional distribution (e.g., RNB or RLR) and utilize $\tilde{y}_j$ for subsequent local samples or learning. By sampling across all $y_j \in \mathbf{Y}_U$, the corresponding set of samples represents a *collective* sample from the joint distribution. The M-Step maximizes the parameters $\tilde{\Theta}_{\mathcal{C}}$ for the local conditionals. This produces a key difference between R-SEM and R-EM. R-SEM utilizes a collective sample $\tilde{\mathbf{Y}}_U$ for MCLE estimation, while R-EM assumes conditional independence of the expectations for the unlabeled $\mathbf{Y}_U$ (Equation 2). Thus, R-SEM maximizes the parameters using the joint sample, unlike R-EM.

As suggested by [13], rather than using a single $\tilde{\Theta}_{\mathcal{C}}^T$ as our final estimate we average the set of parameters learned over all iterations and the final parameters are used for inference:

$$\begin{array}{cc}
\textbf{Parameters (Line 6)} & \textbf{Variables (Line 7)} \\
\hat{\Theta}_{\mathcal{C}} \approx \dfrac{1}{T} \displaystyle\sum_t \tilde{\Theta}_{\mathcal{C}}^t & \textbf{evaluate: } \hat{P}(\mathbf{Y}_U|\mathbf{Y}_L, \hat{\Theta}_{\mathcal{C}})
\end{array} \tag{4}$$

Thus, as indicated in Table 2, R-SEM utilizes an aggregated parameter estimate, but inference is a fixed point operation.

B. Relational Data Augmentation

Our proposed R-DA marginalizes over a distribution of parameters for the local conditional (RNB or RLR) rather than using fixed point estimates. In particular, R-DA iteratively samples from the conditional distributions of both labels and parameters:

$$\begin{aligned}
&\textbf{I-Step:} \text{ (Line 4): sample } \tilde{\mathbf{Y}}_U^t \sim \tilde{P}^t(\mathbf{Y}_U|\mathbf{Y}_L, \Theta_{\mathcal{C}}^{t-1}) \\
&\qquad \text{(i.e.) sample } \tilde{y}_j \sim \tilde{P}^t(y_j|\tilde{\mathbf{Y}}_{\mathcal{MB}(v_j)}, \Theta_{\mathcal{C}}^{t-1}) \; \forall \; y_j \in \mathbf{Y}_U \\
&\textbf{P-Step:} \text{ (Line 5): sample } \tilde{\Theta}_{\mathcal{C}}^t \sim P(\Theta_{\mathcal{C}}|\mathbf{Y}_L, \tilde{\mathbf{Y}}_U^t)
\end{aligned} \tag{5}$$

The I-Step repeatedly draws from the local conditionals (RNB or RLR), while the P-Step samples from the posterior distribution of local conditional parameters $\tilde{\Theta}_{\mathcal{C}}$. The resulting draws are from the joint distribution of labels and parameters, forming two intertwined Markov Chains [11]. Importantly, the samples for each are drawn from their corresponding marginal distributions. Thus, the MAP estimate is:

$$\begin{array}{cc}
\textbf{Parameters (Line 6)} & \textbf{Variables (Line 7)} \\
\hat{\Theta}_{\mathcal{C}} \approx \dfrac{1}{T} \displaystyle\sum_{t=1}^{T} \tilde{\Theta}_{\mathcal{C}}^t & \hat{P}(\mathbf{Y}_U) \approx \dfrac{1}{T} \displaystyle\sum_{i=1}^{T} \tilde{P}^t(\mathbf{Y}_U)
\end{array} \tag{6}$$

In contrast with R-EM and R-SEM, R-DA inferences are averages over the prior samples $\tilde{\mathbf{Y}}_U^{1,\ldots,T}$ rather than fixed point inferences based on $\hat{\Theta}_{\mathcal{C}}$. These samples are from the distribution $P(\mathbf{Y}_U|\mathbf{Y}_L)$ that marginalizes over $\Theta_{\mathcal{C}}$, thus inference in no longer dependent on a single fixed point estimate.

Another important distinction exists between the R-SEM and R-DA parameter estimates. R-DA averages over the sampled parameters that are drawn from the marginal probability distributions over the iterations, while R-SEM averages over the maximized parameters $\tilde{\Theta}_{\mathcal{C}}^{1,\ldots,T}$ in order to reduce the variance of a fixed point estimate. This reflects the difference between the frequentist and Bayesian point of view, where frequentists average fixed point estimates to reduce error due to variance in the data and Bayesians view the parameters themselves as random variables that have uncertainty. Thus, despite the apparent similarity in estimation equations, they reflect contrasting viewpoints.

Lastly, the current representation for the full joint posterior of $\Theta_{\mathcal{C}}$ is intractable due to the complexity of computing the full likelihood. Thus, we substitute the composite likelihood:

Composite P-Step:

$$\text{sample } \tilde{\Theta}_{\mathcal{C}}^t \sim P(\Theta_{\mathcal{C}}|\mathbf{Y}_L, \tilde{\mathbf{Y}}_U^t) \overset{\propto}{\sim} \prod_{y_i \in \mathbf{Y}_L} P(y_i|\tilde{\mathbf{Y}}^t_{\mathcal{MB}(v_j)}, \Theta_{\mathcal{C}}) P(\Theta_{\mathcal{C}})$$

This replaces the update on Line 5.

C. Composite Parameter Posteriors and MAP Approximation

In this subsection, we illustrate the simplicity of using the composite posteriors for the local conditionals RNB and RLR within R-DA and R-SEM. We begin by discussing the *sampling* process from the local parameter posteriors for DA (Composite P-Step). For many local conditional forms, such as RNB, selecting the corresponding conjugate prior results in a closed form posterior distribution. For local conditionals such as RLR there is no conjugate prior, but we can use methods such as Metropolis-Hastings (e.g., [15]) to sample. Lastly, we'll discuss theoretical motivations for allowing a replacement of a sample with the MAP estimate for R-DA. This allows virtually all existing relational learning conditional distributions to be incorporated into R-DA. This maximization is similar to the ‘M’-Step for R-SEM; however, R-DA remains distinct from R-SEM as R-DA samples from the posterior of $\mathbf{Y}_U$. As a reminder, each of these methods also condition over the attributes; however, we continue to omit their notation to reduce clutter.

Composite Relational Naive Bayes: We next give an example of the composite posterior corresponding with the RNB [4] local conditional distribution. For this example, we begin by assuming the labels are binary $\{0, 1\}$ and let θ indicate the parameter corresponding with $P(y_j|y_i = 1, \theta)$: that is, the conditional distribution of the neighboring label corresponding with the observed label being $y_i = 1$. The RNB composite likelihood term when $y_i = 1$ is:

$$P(y_i = 1|\tilde{\mathbf{Y}}^t_{\mathcal{MB}(v_i)}, \Theta_{RNB}) \propto P(y_i = 1) \prod_{\tilde{y}^t_j \in \tilde{\mathbf{Y}}^t_{\mathcal{MB}(v_i)}} P(\tilde{y}^t_j|y_i = 1, \theta)$$

We must estimate the posterior distribution of θ (Line 4 and corresponding Equation 5): as a reminder, α is the associated hyper parameter which defines the prior distribution of θ. As the labels are Bernoulli, the corresponding *conjugate prior* distribution for $P(\theta|\alpha)$ is the *Beta*(α_1, α_2) distribution. The posterior of θ is not dependent on either a) the prior $P(y = 1)$ or b) the attribute conditionals $P(\mathbf{x}|y = 1)$. Thus, the corresponding posterior θ for a single datapoint is:

$$\begin{aligned} P(\theta|y_i, \tilde{\mathbf{Y}}^t_{\mathcal{MB}(v_i)}, \alpha) \propto & P(\theta|\alpha) \prod_{\tilde{y}^t_j \in \tilde{\mathbf{Y}}^t_{\mathcal{MB}(v_i)}} P(\tilde{y}^t_j|y_i = 1, \theta) \\ = & \theta^{\alpha_1 - 1}(1-\theta)^{\alpha_2 - 1} \prod_{\tilde{y}^t_j \in \tilde{\mathbf{Y}}^t_{\mathcal{MB}(v_i)}} \theta^{\tilde{y}^t_j}(1-\theta)^{1-\tilde{y}^t_j} \\ = & \theta^{\alpha_1 + \sum \tilde{y}^t_j - 1}(1-\theta)^{\alpha_2 + \sum(1-\tilde{y}^t_j) - 1} \end{aligned}$$

meaning that the posterior again follows a Beta distribution. The corresponding posterior for θ on the full data $\tilde{\mathbf{Y}}^t$ is:

$$P(\theta|\tilde{\mathbf{Y}}^t, \alpha) \propto \ \theta^{\alpha_1 + \sum y_i \sum \tilde{y}^t_j - 1}(1-\theta)^{\alpha_2 + \sum y_i \sum(1-\tilde{y}^t_j) - 1}$$

which also follows a Beta distribution. Thus, after sampling variables for $\mathbf{Y}_U$ in the I-step, we sample from the posterior θ of the relational parameters using the above. A slight generalization would be to use a *multinomial* distribution, rather than Bernoulli, with the corresponding Dirichlet conjugate prior.

Composite Relational Logistic Regression: In this subsection, we give an example of the corresponding composite posterior corresponding to the RLR [5] local conditional distribution. Let $R^i_0 = \sum_{\tilde{y}^t_j \in \tilde{\mathbf{Y}}_{\mathcal{MB}(v_i)}}(1 - \tilde{y}^t_j)$ and $R^i_1 = \sum_{\tilde{y}^t_j \in \tilde{\mathbf{Y}}_{\mathcal{MB}(v_i)}}(\tilde{y}^t_j)$. The composite likelihood is:

$$P(y_i|\tilde{\mathbf{Y}}_{\mathcal{MB}(v_i)}, \Theta_{RLR}) = \left(\frac{1}{1+e^{-(\theta_0 R^i_0 + \theta_1 R^i_1)}}\right)^{y_i}\left(\frac{e^{-(\theta_0 R^i_0 + \theta_1 R^i_1)}}{1+e^{-(\theta_0 R^i_0 + \theta_1 R^i_1)}}\right)^{1-y_i}$$

RLR does not have a conjugate prior, so we instead set the prior distribution to be a Normal with mean $\mu = 0$ and variance σ^2 (the hyper parameters α). Thus, the full composite posterior for Θ over the labeled components becomes:

$$\begin{aligned} & P(\Theta_{RLR}|\tilde{\mathbf{Y}}^t, \sigma^2) \propto \ g(\Theta_{RLR}|\sigma^2) \\ & = \prod_{y_i \in \mathbf{Y}_L} \left(\frac{1}{1+e^{-(\theta_0 R^i_0 + \theta_1 R^i_1)}}\right)^{y_i}\left(\frac{e^{-(\theta_0 R^i_0 + \theta_1 R^i_1)}}{1+e^{-(\theta_0 R^i_0 + \theta_1 R^i_1)}}\right)^{1-y_i} \prod_{\theta \in \Theta} \mathcal{N}(\theta|0, \sigma^2) \end{aligned}$$

This posterior does not have a closed form solution like the RNB methods did. Hence, we must utilize alternative sampling algorithms, such as Metropolis-Hastings [15]. In this example, let $\tilde{\Theta}^t_{RLR}$ be the current assignment of the sampled parameters. Generate a candidate $\tilde{\Theta}'_{RLR} \sim \tilde{\Theta}^t_{RLR} + \mathcal{N}(0, \sigma^2)$. Let $U \sim Uniform(0, 1)$. The next iteration of $\tilde{\Theta}^{t+1}_{RLR}$ is:

$$\tilde{\Theta}^{t+1}_{RLR} = \begin{cases} \tilde{\Theta}'_{RLR} & \text{if } U < \min\left(\frac{g(\tilde{\Theta}'_{RLR}|\sigma^2)}{g(\tilde{\Theta}^t_{RLR}|\sigma^2)}, 1\right) \\ \tilde{\Theta}^t_{RLR} & \text{otherwise} \end{cases}$$

In this example we have used Normal priors over the parameters, which is equivalent to a L2-regularization.

Algorithm 2 RelationalDataAugmentation($\mathbf{Y}_{obs}, \mathbf{Y}_{mis}, \mathcal{C}$)

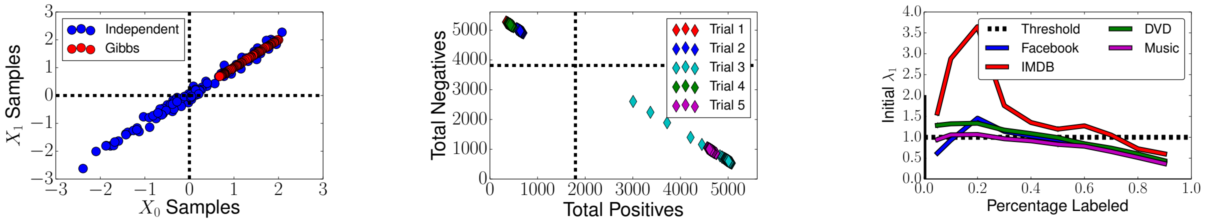
```
1: $\tilde{\Theta}^0_{\mathcal{C}}$ = MaximizeMCLE_MAP($\mathbf{Y}_{obs}, \mathcal{C}$)
2: while More Iterations do
3:     # I Step, then P Step
4:     $\tilde{\mathbf{Y}}_{mis}$ = SampleLabels($\mathbf{Y}_{obs}$, $\mathcal{C}$, $\tilde{\Theta}^{t-1}_{\mathcal{C}}$)
5:     $\tilde{\Theta}^t_{\mathcal{C}}$ = MaximizeMCLE_MAP($\mathbf{Y}_{obs}, \tilde{\mathbf{Y}}_{mis}, \mathcal{C}$)
6: $\hat{\Theta}_{\mathcal{C}}$ = AverageParameters($\tilde{\Theta}^{0,\ldots,T}_{\mathcal{C}}$)
7: $\hat{P}(\mathbf{Y}_{mis})$ = AverageSamples($\tilde{\mathbf{Y}}^{1,\ldots,T}_{mis}$)
```

MAP Substitution: In [16], the authors note that $P(\mathbf{Y}_U|\mathbf{Y}_L) = P(\mathbf{Y}_U|\hat{\theta}, \mathbf{Y}_L)(1 + O(n^{-1}))$, meaning that the distribution of the unlabeled data given the MAP is a close approximation to the posterior distribution. This motivated them to introduce the 'Poor Man's Data Augmentation', in order to estimate the probability of the posterior parameters by maximizing the MAP and sampling multiple times. In this work, we wish to take advantage of that approximation in a different way: that is, we replace the composite P-Step with the maximization of the local parameters (for, e.g., RNB or RLR) instead of a sample:

$$\textbf{MaxComposite P-Step:} \ \text{maximize } \tilde{\Theta}^t_{\mathcal{C}} \\ \arg\max_{\Theta_{\mathcal{C}}} \prod_{v_j \in \mathbf{Y}_L} P(y_j|\tilde{\mathbf{Y}}_{\mathcal{MB}(v_j)}, \Theta_{\mathcal{C}})P(\Theta_{\mathcal{C}}) \tag{7}$$

Our motivation for this is the abundance of previous work on relational algorithms which may require significant work to be transferrable to the Bayesian framework (e.g., choice of proposal distribution). Prior work which focuses on the maximization includes the Relational Generative Models (e.g., RNB) [1], Relational Logistic Regression [5] and others (e.g., [4], [17]). By utilizing this MAP approximation step, we can directly apply each of these respective local learners without the overhead of determining the acceptance steps. Further, we effectively learn a *distribution* of maximizations to apply for inference, rather than a fixed point estimate. Thus, we again avoid any instabilities that could result from a single fixed point parameter estimate. The I-Step will not change, and our inference is still performed by aggregating the samples from the marginal distribution (i.e., Equation 6).

In Algorithm 2, we give just our R-DA algorithm. The algorithm begins by determining initial MAP parameters (Line 1). The algorithm then alternates between sampling from the posterior distribution of labels (Line 4) and maximizing the MCLE MAP (Line 5) until the desired number of iterations are performed. Lastly, on Line 6 we average the previously sampled parameters, and on Line 7 we average the previously sampled labels to recover our predictions. Note that in most domains, the actual parameters are unnecessary to know as we simply desire the final predictions.



(a) Bivariate Normal

(b) RNB

(c) RNB - Initial Eigenvalues

Fig. 3: In each subfigure, dashed lines indicate the expected values of the corresponding axis. (a) A simple Bivariate normal (Independent vs. Gibbs). (b) RLR - number of positive/negative samples. (c) RNB - First Eigenvalue of the Jacobian of the converged solution.

IV. Fixed Point Learning Error and Its Effect on R-EM

In this section, we discuss the learning error of MCLE using the corresponding Gibbs sampling and Variational Mean Field inference methods in relational networks. In particular, we find that the parameter estimates create equilibriums that are far from the true label distribution. These effects compound themselves during R-EM, with the parameter estimates failing to converge for sparsely labeled networks. Out analysis is with respect to a single, fixed point iteration method with respect to a single set of parameters being learned (or iteratively updated). For space, we primarily give results here utilizing the RLR conditional; however, we will also show the parameters of RNB do not converge.

A. Empirical Convergence of Gibbs Sampling

The Gibbs sampler is a theoretically guaranteed MCMC method to sample from the joint distribution of a set of (possibly correlated) variables [2]. For relational inference, this corresponds to repeatedly sampling from the conditional distributions of the unlabeled items: i.e., $\tilde{y}_i^m \sim P^m(y_i|\tilde{\mathbf{Y}}_{\mathcal{MB}(v_i)}, \Theta_{\mathcal{C}})\ \forall\ y_i \in \mathbf{Y}_U$. The samples correspond to draws from the joint distribution and the MAP inference is performed by averaging: i.e., $\tilde{y}_i^t = \frac{1}{M}\sum_m \tilde{y}_i^m$.

In the R-EM framework, this corresponds to Line 4 of Algorithm 1. As our state space is finite, when the probabilities for all vertices and labels is nonzero the chain is ergodic, meaning it will sample from the states in a finite number of steps [18]. However, the mixing rate, or time it takes to converge, can be greatly affected by the correlation of variables [19]. As an example, we present a simple bivariate normal in Figure 3.a, where the variables X_1 and X_2 are highly correlated. In this example, we draw 100 samples independently from the bivariate normal, which we compare with 100 samples drawn utilizing a Gibbs sampler. When trapped in an extremum, the high correlation limits the Gibbs sampler (red) to only a small portion of the space.

When performing Gibbs sampling for relational networks, we find a similar problem exists when the parameters are learned from varying amounts of labeled data. Figure 3.b shows the number of positives and negatives recorded per iteration of Gibbs sampling using the RLR conditional distribution. In particular, we show five different trials for each local conditional distribution, with different randomly assigned labeled values for learning. For each trial, we label 10% of the Facebook network to learn from (dataset discussed in Section V-A) and report results over 1000 iterations over the unlabeled data. Our analysis shows that for each trial, the Gibbs sampling iterations give different results for the number of positives and negatives existing. Of the 10 trials for RLR, no trial gives a reasonable coverage of the space, with sampling from each set of parameters converging to a single (incorrect) point. Although 1000 iterations seems moderate, recall that each iteration involves sampling from over 5000 conditional distributions, resulting in over 5,000,000 total samples drawn. Thus, the parameters learned from the sparsely labeled set create highly correlated estimates of the unlabeled vertices. This results in the Gibbs sampler converging to an incorrect fixed point estimate without fully exploring the label space.

B. Empirical Stability of Variational Mean Field

As the theoretically correct Gibbs sampler fails to efficiently search the space, we next analyze the Variational Mean Field (VMF) inference approximation [3]. VMF approximates the full joint distribution of $\tilde{\mathbf{Y}}_U$ through the approximating distribution $Q(\tilde{\mathbf{Y}}_U) = \prod_{y_i \in \tilde{\mathbf{Y}}_U} Q(y_i)$. Each component $Q(y_i)$ is iteratively updated in a coordinate ascent algorithm:

$$Q(y_i) = \frac{1}{Z_{Q(i)}} \exp\left\{\mathbb{E}_{\mathbf{Y}_{U\setminus i}\sim Q}[\log f(y_i|\tilde{\mathbf{Y}}_{\mathcal{MB}(v_i)}\tilde{\Theta}_{\mathcal{C}})]\right\} \tag{8}$$

where $f(\cdot)$ is the unnormalized energy function and $Z_{Q(i)}$ is the partition function for the local $Q(y_i)$ conditional. VMF is guaranteed to converge to a fixed point equilibrium [20]; thus, VMF inference can be cast as a *Nonlinear Dynamical System* (NLDS). A useful theorem exists about the *stability* of a NLDS system at an equilibrium:

Theorem 1: [Asymptotic Stability ([21])] The system given by $\mathbf{P}^* = Q(\mathbf{P}^*)$ is asymptotically stable at an equilibrium point $\mathbf{P}^* = \tilde{\mathbf{y}}$ if the eigenvalues of the Jacobian $\mathcal{J} = \nabla Q(\tilde{\mathbf{y}})$ are less than 1 in absolute value, where: $\mathcal{J}_{i,j} = \frac{\partial\ Q(y_i)}{\partial\ Q(y_j)}$.

Hence, given a set of labeled data $\mathbf{Y}_L$ and unlabeled vertices $\mathbf{Y}_U$ to infer, we can determine whether or not the system will stay in an equilibrium $\mathbf{P}^*$ using the partial derivatives of the VMF update in Equation 8. In particular, the labeled data is a fixed value (1 or 0, depending on the state and label), meaning partial derivatives with respect to all other variables is 0. The corresponding Jacobian matrix $\mathcal{J}$ is:

$$\mathcal{J} = \begin{array}{c|c|c|} & \mathbf{Y}_U & \mathbf{Y}_L \\ \hline \mathbf{Y}_U & \mathcal{J}_{U\times U} & \mathcal{J}_{U\times L} \\ \hline \mathbf{Y}_L & \mathcal{J}_{L\times U} & \mathcal{J}_{L\times L} \end{array} = \begin{array}{c|c|c} & \mathbf{Y}_U & \mathbf{Y}_L \\ \hline \mathbf{Y}_U & \mathcal{J}_{U\times U} & 0 \\ \hline \mathbf{Y}_L & 0 & 0 \end{array}$$

where $\mathcal{J}_{U\times L} = 0$ as the corresponding rows $\mathcal{J}_L$ are 0 (they do not affect the maximal eigenvalue [21]). Thus, we need only evaluate the partial derivatives of the unlabeled $Q(y_i)$ conditionals (with learned parameterization $\hat{\Theta}_{\mathcal{C}}$) at the stationary convergence equilibrium $\mathbf{P}^*$. Solutions for the RNB and RLR partials can be found in Appendix A.

In Figure 3.c, we evaluate the eigenvalues at the converged $\mathbf{P}^*$ for four datasets. For this starting example, we use a single fixed parameter estimate (i.e., $\tilde{\Theta}^0_{\mathcal{C}}$) and perform inference with respect to those parameters. This corresponds to traditional RML, without performing R-EM (i.e., the inferences are not used to relearn). In general, the eigenvalues reach a fairly stable state, with the average eigenvalues largely being at or below the 1 threshold. However, for some cases of RLR it is clear the achieved equilibriums are not necessarily stable, meaning small perturbations during inference could have a large effect on $\mathbf{P}^*$.

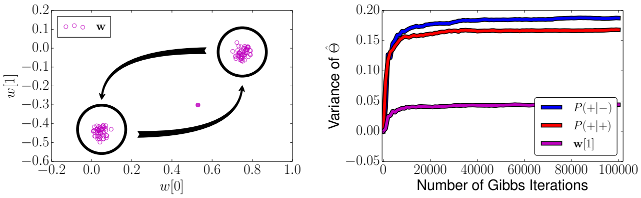
C. MCLE Error on R-EM

In this section we study the empirical error produced by the R-EM algorithm. In particular, we analyze whether the algorithm ever converges (in practice) to a stationary point, whether using Gibbs sampling or VMF.

We first analyze the convergence of R-EM utilizing Gibbs sampling for inference. For the RLR relational classifier, we analyze the Facebook network for convergence (Section V). The networks are initially assigned 10% of the data labeled: the rest is unlabeled and must be inferred. The model is then utilized to compute the expectations of the unlabeled instances utilizing the Gibbs sampler and the process is repeated. We allow each method 100,000 passes over the unlabeled data for performing the Gibbs sampling, with maximizations performed every 1000 passes.

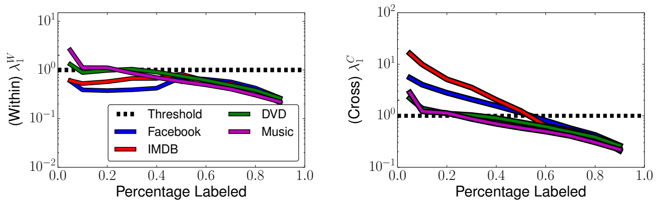
We show results in Figure 4, with Figure 4.a containing the learned relational conditional distributions for RLR. The scatterplots illustrate the learning parameters after each 'M'-Step; we observe that they follow a *periodic* behavior. That is, for example, in Figure 4.a when a learned state corresponds to the bottom left state, the next maximization will result in parameters from the upper right state. This occurs despite the initial parameter estimates beginning in a less extreme portion of the parameter space; even though each method has (potentially) started near a good solution, the estimates quickly degrade. Figure 4.b demonstrates that even over 100,000 passes of the data, the estimates of $\hat{\Theta}$ never converge (we plot both the weight variance for RLR and the neighboring conditionals' variance for RNB).

The Variational R-EM approach allows us to draw a more general conclusion regarding the *convergence* R-EM. Let $\mathcal{J}^W$ be the *within*-iteration Jacobian, where first the parameters $\tilde{\Theta}^t_{\mathcal{C}}$ are learned; then we estimate $\mathbf{P}^*$ using the parameters. As an alternative, let $\mathcal{J}^C$ be the *cross*-iteration Jacobian matrix. For $\mathcal{J}^C$, we use the equilibrium $\mathbf{P}^*$ to learn a new set of parameters ($\tilde{\Theta}^{t+1}_{\mathcal{C}}$). We then define $\mathcal{J}^C$ using $\mathbf{P}^*$ and $\tilde{\Theta}^{t+1}_{\mathcal{C}}$.



(a) (b)

Fig. 4: Empirical (lack of) convergence of R-EM. (a) RLR relational parameters. (b) Variance of the parameter estimates does not decrease with number of iterations.



(a) Within Eigenvalues (b) Cross Eigenvalues

Fig. 5: Parameter convergence. (a) The within-iteration Jacobian max eigenvalue, (b) the cross-iteration Jacobian max eigenvalue.

Corollary 1: [Parameter Convergence] If the first eigenvalue λ_1^W of $\mathcal{J}^W$ is less than 1 in absolute value, and the parameters $\tilde{\Theta}^t_{\mathcal{C}} = \tilde{\Theta_{\mathcal{C}}}^{t+1}$, then the first eigenvalue λ_1^C of $\mathcal{J}^C$ is less than 1 in absolute value.

This is a consequence of Equation 8 having equivalence for $\tilde{\Theta}^t_{\mathcal{C}}$ and $\tilde{\Theta}^{t+1}_{\mathcal{C}}$, and $\mathcal{J}$ comprising the partial derivatives with respect to Equation 8. This is easily seen as a consequence of $\mathcal{J}^W = \mathcal{J}^C$ when $\tilde{\Theta}^t_{\mathcal{C}} = \tilde{\Theta}^{t+1}_{\mathcal{C}}$. In Figure 5a-b, we plot the λ_1^W and λ_1^C within R-EM. Note that the within-iteration eigenvalue is small, and usually indicates a stable convergence to $\mathbf{P}^*$. However, λ_1^C in Figure 5.b is exceptionally large for small amounts of labeled data. Thus, we conclude that the parameters have not reached a stable equilibrium (even after 100 iterations). For each dataset, when using both RLR and RNB (not shown for space), R-EM does not converge prior to the 20% labeled data mark. This is an extreme limitation to the method as most partially labeled datasets have few labels.

V. Experiments

In this section, we compare our R-SEM and R-DA frameworks against the existing R-EM within-network relational learning approach. We test each method on four large, real world datasets, and compare against independent and collective inference methods based on two local conditional implementations (RNB and RLR) combined with Gibbs sampling.[4]

A. Datasets

We compare each of the above methods on four datasets. The full statistics for the datasets are compiled in Table I.

[4] Implementation can be found at: *http://nld.cs.purdue.edu/*.

Dataset	N_v	N_e	W	ρ	$P(+)$
Facebook	5,906	73,374	2	0.174	0.320
IMDB	11,280	426,167	37	0.207	0.494
DVD	16,118	75,596	28	0.177	0.510
Music	56,891	272,544	26	0.114	0.491

TABLE I: From left: dataset, number vertices, number edges, number attributes, label correlation across edges, positive prior.

When possible, we set thresholding for the labels such that the label set is closely balanced, to keep skew from impacting our error measurements.

Facebook: This is a snapshot of the Purdue University Facebook network. We use the users' Political views as the label, with Religious Views and Gender as attributes.

IMDB: This is the IMDB dataset (www.imdb.com), where we predict whether a movie is *successful*. We discretize the label by assigning the value 1 if the gross receipts were greater than \$300 million. For features, we use the genres and average user ratings, which gives a total of 37 features. Edges in the network represent when two moves share a producer.

DVD: This is the Amazon copurchase network compiled by [22], but we only select the DVD items. This allows us to incorporate 24 *genres* of movies as features in addition to the 1 through 5 star ratings for a total of 28 features. The label we predict is whether the item is a top seller. We use the provided sales rank and set the top seller threshold at 20000.

Music: This is the Amazon copurchase network compiled by [22], but we only select the Music items. This allows us to incorporate 22 *styles* of music as features. We keep our user rating features which gives us 26 features total, and also set our sales rank threshold at 65000.

B. Methods Compared

We test the RNB and RLR conditionals with six different learning and representations, ranging from independent learning and inference to the proposed R-DA. The collective approaches are allowed a *total* of 1000 iterations of Gibbs sampling over the unlabeled dataset, regardless of the method, allowing us to directly compare their relative performance on the same number of iterations over the data. The parameters in the RNB formulation have a $Beta(\alpha_1 = \alpha_2 = .5)$ prior; the parameters in the RLR formulation have a $\mathcal{N}(0, 1)$ prior. Each uses the MAP approximation and we use LibLinear [23] for optimization. Each method can be viewed as different implementations of various lines in Algorithm 1—we mention each specifically.

Ind (NB and LR): (Lines 6 & 7, Equation 1). This method uses just the attribute components of the data, and ignores the relational components.

Rel (IND) (RNB and RLR): (Lines 6 & 7, Equation 1). This method estimates from the *observed* attributes and relational components. These estimates are applied on the remaining data. It does not utilize the unlabeled data when learning, and does not perform collective inference.

Rel (CI) (RNB and RLR): (Lines 6 & 7, Equation 1). This method estimates from the *observed* attributes and relational components. These estimates are applied on the remaining data. It does not utilize the unlabeled data when learning, but does perform collective inference.

R-EM (RNB and RLR): (Lines 1–7, Equation 2). This is the fixed point estimation method of [9], and is the first iterative method. The method begins by computing the expectations of the unlabeled data, then utilizes these to maximize the full data likelihood. We allow 10 iterations of the full EM loop, with 100 iterations of Gibbs sampling each EM iteration. As EM can have extreme variance, we average in 10 and 11 iterations to give the expected error.

R-SEM (RNB and RLR): (Lines 1–7). This is the first of our proposed methods. We allow 900 iterations of R-SEM (Equation 3) and averages over the intermediate parameters are used for the final parameters. This final parameter set is used for a final round of collective inference using an additional 100 iterations of Gibbs sampling (Equation 4).

R-DA (RNB and RLR): (Lines 1–7, Equation 3). This is the second of our proposed methods. We allow R-DA 1000 iterations of Gibbs sampling (Equation 5), and utilize the MAP approximation to the parameters between each iteration (Equation 7). We perform the final inference by aggregating over the intermediate Gibbs samples (Equation 6).

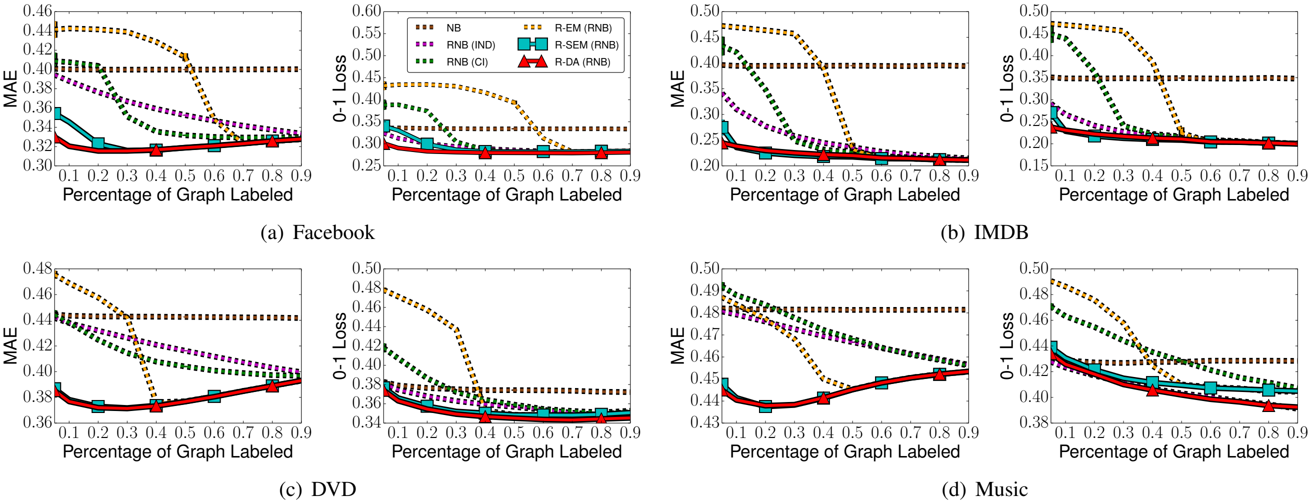
C. Methodology

We compare each method on each dataset. For each percentage of labeled information a random subset was selected from the respective networks and used for learning/inference. All methods are given the same starting set for each of the 25 recorded trials (10 for the larger Music dataset). For error, we measure the *Mean Absolute Error* (MAE) and the 0-1 Loss. Standard error bars are plotted but small (i.e., $< .01$).

D. Results

In Figure 6 we report error results when applying the methods, each using the RNB conditional distribution. Note that R-DA is equal or better than all competitors across all label percentages, regardless of the error measure used. Importantly, R-DA exceeds previous methods with small percentages of labeled data. It is important to notice the previous R-EM suffers for small amounts of labeled data in comparison to relational RNB which does not perform collective inference. This is due to the unstable collective inference impacting the learned parameters of R-EM. Not surprisingly, NB performs well with small amounts of information but never improves.

In Figure 7 we report the RLR conditional distribution error results. In each dataset R-DA outperforms or equals the corresponding R-EM collective inference algorithm, particularly when fewer labels are available. In these examples, the RLR without collective inference performs competitively with R-DA on the denser datasets, even outperforming R-DA on IMDB. However, this is not generally the case—for most datasets R-DA largely outperforms the independent relational method. Our experiments demonstrate R-DA's ability to compete and outperform competing methods, across a variety of datasets and label percentages.



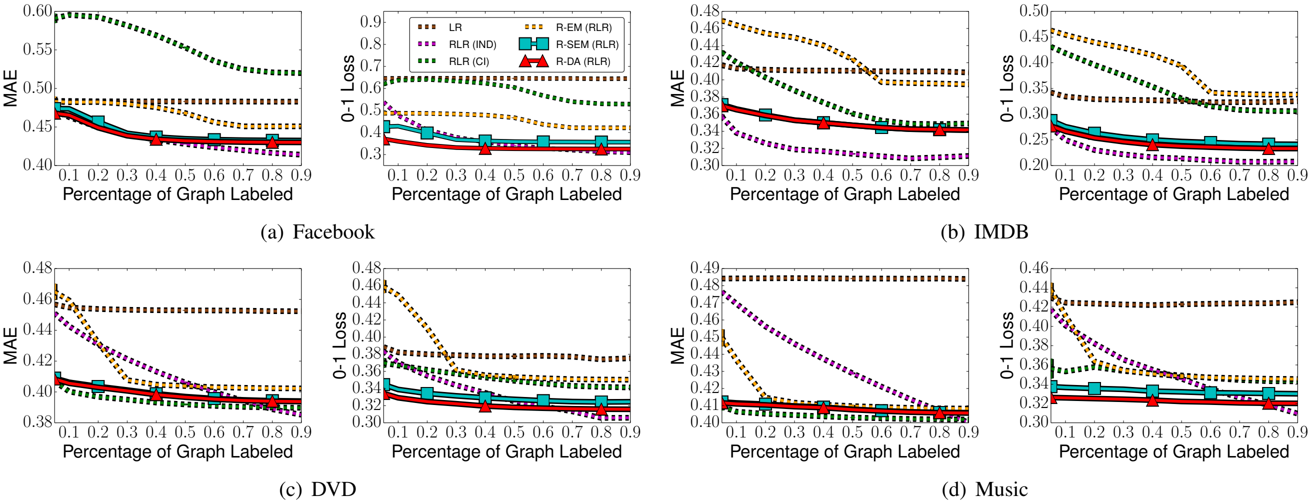
(a) Facebook

(b) IMDB

(c) DVD

(d) Music

Fig. 6: **RNB Conditionals.** We show the MAE and 0/1 Loss on a) Facebook, b) IMDB, c) DVD and d) Music.



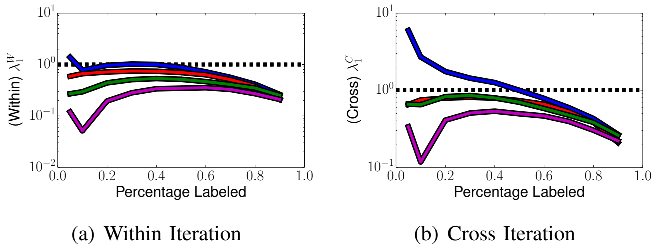
(a) Facebook

(b) IMDB

(c) DVD

(d) Music

Fig. 7: **RLR Conditionals.** We show the MAE and 0/1 Loss on a) Facebook, b) IMDB, c) DVD and d) Music.



(a) Within Iteration

(b) Cross Iteration

Fig. 8: For the RLR conditional, (a) the within iteration eigenvalues for SEM and (b) the SEM cross iteration eigenvalues.

As a final note, we see that R-SEM outperforms R-EM across all datasets and performs nearly as well as R-DA in many cases, despite being a fixed point estimate. However, we cannot always use the inferences that result from R-SEM for additional fixed point estimations. This is shown in Figure 8 for the RLR conditional, where the Facebook network has low within iteration Jacobian eigenvalues but still has high cross iteration eigenvalues. Thus, even with largely correct inferences MCLE can still learn unstable parameters.

VI. Discussion and Further Related Work

Our proposed R-SEM and R-DA tie several research areas together. First, we demonstrated that the approximations necessary for tractable learning and inference substantially interfere with the guarantees provided by EM [10]. However, by utilizing a distribution of maximizations, R-SEM is able to find a reasonable fixed point in the parameter space which results in empirically stable inference. We further improve on the R-SEM implementation and remove the fixed point inference process, introducing the Bayesian R-DA method. These methods facilitate the application of RML techniques (e.g., [1], [4]) to make predictions over entire networks from minimal amounts of label data using collective inference—improving on independent inference, despite using approximations for scalable learning (e.g., the component likelihood).

Collective Inference (CI) error for sparsely labeled datasets has been noted before, although we carry out the first empirical analysis of the Gibbs mixing rate and Variational Inference stability when parameters are learned through MCLE. Our methods fit easily within current *Cautious Collective Inference*

(CCI) methods [8]; these methods only utilize inferred labels with high confidence during CI to overcome the possible error in the parameter estimates. R-SEM and R-DA provide better confidence estimates for these methods to use during CCI. Specialized conditionals, as proposed by [24], place more weight on the attributes of neighboring instances to improve CI. By weighting the attributes more heavily, these conditionals implicitly stabilize the inference process. R-SEM and R-DA can again improve these methods by incorporating the unlabeled data into the conditionals' learning, while maintaining their implicit stability.

The Gibbs mixing and VMF inference stability creates connections to other areas of Statistical Network Analysis, notably virus propagation. The stability analysis of VMF was partially motivated by the work of [21], which showed common virus propagation models can be tied to the stability of the network and the maximal eigenvalue. Our R-DA model can be tied to Ensemble Methods [25]. In particular, as each fixed point MCLE step has error, R-DA takes an *ensemble* of estimates over the missing data for inference. Although each individual value may only be weakly correlated with the correct solution, the aggregation over these methods can produce a good solution.

VII. Conclusions

In this work we introduced the R-DA and R-SEM methods for within-network relational learning and inference. We began with an analysis of the fixed point relational inference methods in conjunction with MCLE learning methods. In particular, we demonstrated that Gibbs sampling and VMF inference are inaccurate when the parameters are learned through MCLE, and that these errors interfere with R-EM's convergence. By introducing the R-SEM method, we were able to learn fixed point parameter estimates with a reasonable inference solution. R-DA further extends this idea and removes fixed point inference, replacing it with a distribution of inferences. We demonstrated that R-DA significantly outperforms competing methods when utilized in conjunction with multiple learning algorithms. Most importantly, R-DA improves prediction in sparsely labeled networks, an important practical application where RML techniques have traditional struggled.

This work implicates multiple avenues for future work. A central finding is the tie between relational inference and the maximal eigenvalue of the inference solution. By selectively labeling items in the network that reduce the maximal eigenvalue, *active* learning methods could improve estimates through the creation of a more stable inference task. The improved estimates can be used to further improve algorithms such as cautious collective inference. Lastly, analyzing the extremums that result from MCLE estimation can hopefully motivate new, stable learning methods.

Acknowledgements

This research is supported by NSF under contract number(s) IIS-1149789, CCF-0939370 and IIS-1219015. 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.

Appendix A

Let $h(y_i|y_j) = \exp\left\{\mathbb{E}_{\tilde{\mathbf{Y}}_U \sim Q}[\log f(y_i|y_j, \tilde{\mathbf{Y}}_{\mathcal{MB}(v_i)\setminus j})]\right\}$. The partials for $\mathcal{J}_{ij}$ are:

$$\frac{\partial\, Q(y_i)}{\partial\, Q(y_j)} = \frac{\phi(y_i, y_j)h(y_i|y_j)Z_{Q(i)} - h(y_i|y_j)\sum_{y' \in \mathcal{Y}} \phi(y, y_j)h(y|y_j)}{Z^2_{Q(i)}}$$

where $\phi(y, y_j) = \log P(y_j|y)$ (RNB) or $\phi(y, y_j) = \theta_y$ (RLR).

References

[1] L. Getoor and B. Taskar, *Introduction to Statistical Relational Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2007.

[2] S. Geman and D. Geman, "Stochastic relaxation, gibbs distributions, and the bayesian restoration of images," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 6, no. 6, pp. 721–741, Nov. 1984.

[3] Jordan, Ghahramani, Jaakkola, and Saul, "An introduction to variational methods for graphical models," *Mach. Learn.*, vol. 37, no. 2.

[4] Neville and Jensen, "Relational dependency networks," *JMLR*, 2007.

[5] P. Sen, G. M. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi-Rad, "Collective classification in network data," *AI Magazine*, vol. 29, no. 3, pp. 93–106, 2008.

[6] R. Xiang and J. Neville, "Understanding propagation error and its effect on collective classification," in *In ICDM*, 2011.

[7] T. Khot, S. Natarajan, K. Kersting, and J. Shavlik, "Learning relational probabilistic models from partially observed data – opening the closed–world assumption," ser. LNAI, August, 28–30 2013.

[8] L. K. McDowell, K. M. Gupta, and D. W. Aha, "Cautious collective classification," *J. Mach. Learn. Res.*, vol. 10, pp. 2777–2836, Dec. 2009.

[9] R. Xiang and J. Neville, "Pseudolikelihood em for within-network relational learning." in *ICDM*. IEEE Computer Society, 2008.

[10] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, no. 1, pp. 1–38, 1977.

[11] M. A. Tanner and W. Wong, "The calculation of posterior distributions by data augmentation," *JASA*, vol. 82, pp. 528–540, 1987.

[12] C. F. J. Wu, "On the Convergence Properties of the EM Algorithm," *The Annals of Statistics*, pp. 95–103, 1983.

[13] G. Celeux, D. Chauveau, and J. Diebolt, "On Stochastic Versions of the EM Algorithm," Tech. Rep.

[14] Bernardo, Bayarri, Berger, Dawid, Heckerman, Smith, and West, "The variational bayesian em algorithm for incomplete data," 2003.

[15] S. Chib and E. Greenberg, "Understanding the Metropolis-Hastings Algorithm," *The American Statistician*, no. 4.

[16] G. C. G. Wei and M. A. Tanner, "A mc implementation of the em algorithm," *JASA*, no. 411, pp. 699–704.

[17] Natarajan, Khot, Kersting, Gutmann, and Shavlik, "Gradient-based boosting for statistical relational learning: The relational dependency network case," *Mach. Learn.*, vol. 86, no. 1, pp. 25–56.

[18] S. M. Ross, *Stochastic Processes (Wiley Series in Probability and Statistics)*, 2nd ed. Wiley, Feb. 1995.

[19] R. Neal, "Slice sampling," *Annals of Statistics*, vol. 31, 2000.

[20] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[21] B. A. Prakash, D. Chakrabarti, M. Faloutsos, N. Valler, and C. Faloutsos, "Got the flu (or mumps)? check the eigenvalue!" *arXiv:1004.0060v1 [physics.soc-ph]*, 2010.

[22] J. Leskovec, L. A. Adamic, and B. A. Huberman, "The dynamics of viral marketing," *ACM Trans. Web*, vol. 1, 2007.

[23] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "LIBLINEAR: A library for large linear classification," *Journal of Machine Learning Research*, vol. 9, pp. 1871–1874, 2008.

[24] McDowell and Aha, "Labels or attributes?: Rethinking the neighbors for collective classification in sparsely-labeled networks," ser. CIKM, 2013.

[25] H. Eldardiry and J. Neville, "Across-model collective ensemble classification," in *AAAI*, 2011.