Journal of Machine Learning Research 21 (2020) 1-37 Submitted 9/14; Revised 2/20; Published 3/20

Ensemble Learning for Relational Data

Hoda Eldardiry HDARDIRY@VT.EDU
Department of Computer Science
Virginia Tech
114 McBryde Hall
225 Stanger Street
Blacksburg, VA 24061-0106 USA

Jennifer Neville NEVILLE@CS.PURDUE.EDU
Department of Computer Science and Department of Statistics
Purdue University
307 N. University Street
West Lafayette, IN 47907 USA

Ryan A. Rossi RROSSI@ADOBE.COM
Adobe Research
345 Park Avenue
San Jose, CA 95110 USA

Editor: Luc De Raedt

Abstract

We present a *theoretical analysis framework* for relational ensemble models. We show that ensembles of collective classifiers can improve predictions for graph data by reducing errors due to variance in both learning and inference. In addition, we propose a *relational ensemble framework* that combines a relational ensemble learning approach with a relational ensemble inference approach for collective classification. The proposed ensemble techniques are applicable for both single and multiple graph settings. Experiments on both synthetic and real-world data demonstrate the effectiveness of the proposed framework. Finally, our experimental results support the theoretical analysis and confirm that ensemble algorithms that explicitly focus on both learning and inference processes and aim at reducing errors associated with both, are the best performers.

Keywords: Ensemble learning, relational ensemble, collective classification, collective inference, bias-variance decomposition, relational machine learning, theoretical framework

1. Introduction

Ensemble methods have been widely studied as a means of reducing classification error by combining multiple *base* models for prediction. These methods have been used successfully for many practical applications and real-world systems. Despite the fundamental importance of these techniques, most existing work has focused on i.i.d. data where objects are independent and models use exact inference techniques. The few works that have investigated

ensembles for relational data (Heß and Kushmerick, 2004; Preisach and Schmidt-Thieme, 2008) have a number of limitations: (a) the techniques reduce only one type of error; due to learning, (b) techniques are designed for networks with multiple relation types, and (c) there is no theoretical analysis to show the mechanism by which the ensembles reduce model error in relational data.

In this paper, we formulate a *theoretical analysis framework* to compare the errors made by different relational ensembles and demonstrate theoretically the reason why some methods perform better than others. Moreover, we also present a *relational ensemble framework* that combines a relational ensemble learning approach with a relational ensemble inference approach. The first approach, for learning the ensemble, focuses on reducing error due to variance in learning whereas the second approach, that applies the ensemble for inference, focuses on reducing the error due to variance in inference. The combination of these methods is shown to offer the largest improvement in classification accuracy compared to the baseline approaches on both synthetic and real-world data. Furthermore, the proposed approach is applicable in both single- and multi-graph settings (i.e., where there are multiple graphs with the same nodes but different link types).

The different ensemble design choices are shown in Figure 1. Ensembles for i.i.d. data have mainly considered design choices including methods for input data treatment shown in Figure 1a and methods for aggregating the output of the models shown in Figure 1d. The goal of input data treatment is to ensure a variety among the learned models, e.g., bagging approaches use resampling to generate multiple bootstraps of the input data to learn the models and then aggregate predictions from them Breiman (1996a), while boosting approaches construct the models in a coupled fashion such that their weighted vote provides a good fit to the data Schapire et al. (1997); Quinlan (1996); Freund and Schapire (1996). For more complex relational data, different design choices must be considered. First, the treatment of input data must consider the relational data characteristics. Second, relational or collective inference models can be used as the base component models of the ensemble since they have been shown to improve predictions for relational data. For example, some recent work Natarajan et al. (2012) proposed using boosting for learning relational dependency networks (RDNs) to reduce the bias component of error. Third, instead of running the base models independently for inference, and aggregating the predictions to obtain the final predictions, this work takes advantage of the collective inference process and allows a prediction made by one model to influence the prediction made for the same node by another model (Figure 1c). This notion of *across-model* inference aggregation facilitates an additional reduction of variance due to *inference* (called *inference variance*) on top of the traditional learning variance reduction achieved by the final step of output aggregation (Figure 1d).

The goal of this work is to theoretically analyze the error components of relational ensembles to better understand the mechanisms by which different approaches reduce classification error. Ensemble methods for non-relational data have been shown to reduce classification error by reducing *learning variance* (e.g., bagging Breiman (1996a)) or reducing bias (e.g., boosting Freund and Schapire (1996)). Previous ensembles that reduce variance have focused on reducing one type of variance called *learning variance*. This is the variance due to learning the models from different training data sets. On the other hand, collective inference models applied to relational data have been shown to have additional sources of

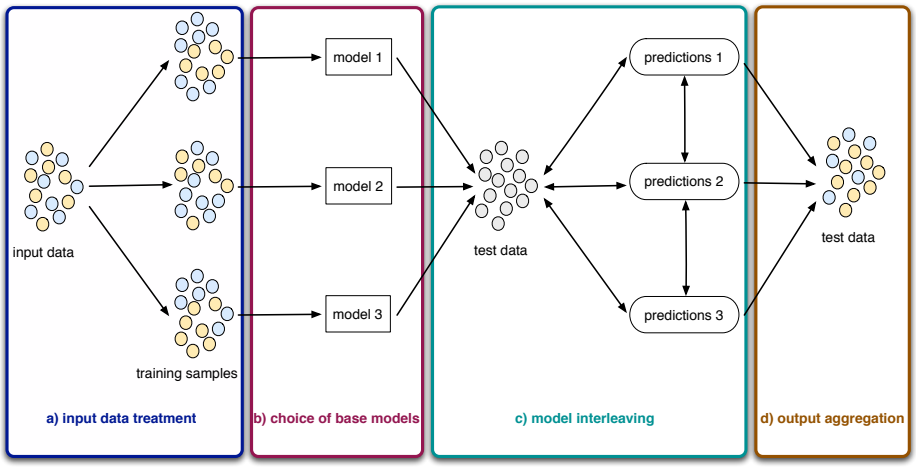

Figure 1: Design dimensions for ensembles

error due to variance in the inference process Neville and Jensen (2008). We refer to the variance in predictions made by the same model given different subsets of true labels for nodes in the test set, as the *inference variance*. This paper focuses on reducing variance in both learning *and* inference.

We propose a theoretical analysis framework for analyzing the error reduction of relational ensembles models. We extend relational bias/variance decomposition Neville and Jensen (2008) for the ensemble setting to consider not only a single collective inference model, but an ensemble of collective inference models. We theoretically analyze two classes of ensemble models: (i) a relational ensemble model that uses the component classifiers independently for inference and aggregates the final predictions, and (ii) an across-model approach that uses the component models simultaneously for collective inference and aggregates the intermediate predictions across the models during inference. The aim of the theoretical analysis is to decompose the errors associated with each ensemble and show how the different ensemble approaches are able to reduce the error of a single model. Specifically, we show that the interleaved *across-model* ensemble produces the greatest reduction in error due to its ability to reduce learning and inference error without an increase in bias. To our knowledge this is the first theoretical analysis of error for relational ensembles.

1.1. Summary of Main Contributions

The main contributions of this work are as follows:

- Theoretical analysis framework for relational ensembles that extends bias/variance decomposition to not only consider a single model but an ensemble of collective inference models (Section 3.1).
- Theoretical analysis of the reduction in error offered by different relational ensemble models (Section 3.2). This is the first theoretical analysis of relational ensembles.
- A relational ensemble framework that combines a relational ensemble learning approach with a relational ensemble inference approach to reduce both learning and

inference error due to variance (Section 4). For the relational ensemble *learning*, we propose a relational subgraph resampling approach called RSR (Section 4.1) whereas for the relational ensemble *inference* approach we propose CEC (Section 4.2).

- Empirical evaluation on real and synthetic data that demonstrates the effectiveness of the proposed framework as it achieves significant performance gains compared to alternative ensembles (Section 5).

1.2. Organization of this Article

This article is organized as follows. Section 2 begins with preliminaries. Section 3 proposes an error analysis framework for theoretical analysis of relational ensembles. Section 4 proposes a relational ensemble framework that reduces error due to variance in both learning (Section 4.1) *and* inference (Section 4.2). Section 5 demonstrates the effectiveness of the proposed techniques empirically using both real-world and synthetic graph data. Section 6 discusses the related work. Finally, Section 7 concludes.

2. Preliminaries

2.1. Collective classification

The general relational learning and collective classification problem are defined as follows. Note collective classification consists of relational learning and collective inference.

Relational learning: Given a fully-labeled training set composed of a graph $G_{tr} = (V_{tr}, E_{tr})$ with nodes V_{tr} and edges E_{tr} along with observed features X_{tr} and observed class labels Y_{tr}, the goal is to learn a model f defining a joint probability distribution over the labels of V_{tr} conditioned on the observed attributes and graph structure in G_{tr}.

Collective inference: Given a partially-labelled test set composed of a graph $G_{te} = (V_{te}, E_{te})$ with nodes V_{te} and edges E_{te} along with observed features X_{te} and partially-observed class labels $\tilde{Y}_{te} \subset Y_{te}$, the learned model f is applied for collective inference to output a set of marginal probability distributions P (i.e., predictions) for each unlabeled node in V_{te}. In this work, the G_{tr} used for learning is different from the G_{te} used for collective inference. Further, we also focus primarily on node classification (i.e., predicting labels for the nodes), though the techniques are more generally applicable for other relational learning tasks such as link classification.

2.2. Relational ensembles

The general relational ensemble classification problem is defined as follows. Given a fully-labeled training set graph $G_{tr} = (V_{tr}, E_{tr})$ with nodes V_{tr} and edges E_{tr} along with observed features X_{tr} and observed class labels Y_{tr}, the first step is to generate a set of m sample graphs $\{G_{tr_1}, \ldots, G_{tr_m}\}$ from G_{tr}, which we refer to as *pseudosamples*. Next, a set of models $F = \{f_1, \ldots, f_m\}$ are learned from $\{G_{tr_1}, \ldots, G_{tr_m}\}$ using a relational learning algorithm (i.e., one model is learned for each pseudosample graph). During prediction, each learned model $f_i \in F$ is applied for collective inference on $G_{te} = (V_{te}, E_{te})$ to output a set of marginal probability distributions P_i (i.e., predictions) for each unlabeled node in V_{te}. Finally, the

predictions $P_1, \ldots, P_m$ from the m models are aggregated and the final predictions P are generated for nodes in G_{te}.

3. Theoretical Analysis

This section describes an error analysis framework for relational ensembles. In particular, we extend bias/variance decomposition for relational ensembles and use it to theoretically analyze different relational ensembles. We use squared loss as a measure of classification performance and show the error reduction offered by the different types of ensembles. To our knowledge, this is the first theoretical exploration of classification error for relational ensembles.

3.1. Theoretical Framework

We formalize the collective classification task in order to describe the setting we use for this analysis. Let $\mathcal{D}$ be a population of attributed graphs G.[1] Each sample $D = [G=(V, E), X_V, Y_V]$ is drawn from $\mathcal{D}$, where V is the set of nodes (instances) and E is the set of links. Let $f = P(\mathbf{Y}|\mathbf{X}, G)$ represent a model of the joint distribution over class labels $\mathbf{Y}$ of nodes in a graph G, given attributes of the nodes $\mathbf{X}$. Let $D_L \in \mathcal{D}$ be a training graph. Let $D_I \in \mathcal{D}$ be a partially labeled test graph where $\mathcal{T} \subseteq V_I$ is the set of labeled nodes in G_I. Let $\mathbf{Y}_\mathcal{T}$ be the set of known labels available to the inference process. We also use t^i to denote the true label of an unlabeled node v_i. For this analysis, we assume that D_L and D_I are drawn independently from $\mathcal{D}$. Further, we assume the train and test graph are generated by the same underlying mechanism.

The goal is to learn f from the training set D_L and apply it to the test set D_I to collectively predict class labels for each unlabeled node $v_i \in V_{I \setminus \mathcal{T}}$:

$$y_f^i = f(v_i, D_I, \mathcal{T}) = P(Y^i{=}t^i|\mathbf{Y}_\mathcal{T}, \mathbf{X}, G_I) \tag{1}$$

Since relational models that use collective inference have an additional source of error due to the inference process, we need to isolate the errors due to learning from the errors due to inference. To achieve this, we also consider the performance of an *exact inference* model, which does not use collective inference and simply makes a prediction for v_i conditioned on the set of Bayes-optimal values for all instances except v_i. Below, we use $\tilde{\mathbf{Y}}_{V_{I \setminus v_i}}$ to refer to the Bayes-optimal prediction for all instances in the dataset D_I except v_i. A model that uses $\tilde{\mathbf{Y}}_{V_{I \setminus v_i}}$ to make a prediction for an instance v_i is referred to as an *exact inference* model and we will use it to isolate errors due to learning from the errors due to collective inference.

3.1.1. Model definitions

We consider four models in our analysis: the "true" model (f_*), a single collective inference model (f_s), a simple relational ensemble model (f_e), and a interleaved collective inference model (f_c). We define each of these models below.

True model: We define f_* as the "true" model for the population $\mathcal{D}$, where P_* is the "true" joint distribution, which can be estimated as the expected model f_s that will be

1. In attributed graphs, every node is assigned attributes and possibly a label.

learned over samples drawn from $\mathcal{D}$:

$$f_* = P_*(\mathbf{Y}|\mathbf{X}, G) = E[f_s] = \sum_{D_L \in \mathcal{D}} f_s * p(D_L) \tag{2}$$

Single collective inference model: Let f_s be a single collective inference model learned from a sample $D_L \in \mathcal{D}$, which estimates P_s. The model f_s is then used to make predictions for each unlabeled node v_i in a partially labeled dataset $< D_I, \mathcal{T} >$:

$$\begin{aligned} y^i_{f_s} &= f_s(v_i, D_I, \mathcal{T}) \\ &= P_s(Y^i{=}t^i|\mathbf{Y}_\mathcal{T}, \mathbf{X}, G_I) \end{aligned} \tag{3}$$

Simple relational ensemble model: Let $F = \{f_{s_1}, \ldots, f_{s_m}\}$ be a set of m collective inference models learned from $\{D_{L_1}, \ldots, D_{L_m}\}$ generated from D_L. Each f_s gives a different estimate of the true joint distribution P_*. See Section 3.2.6 for a description of how $\{D_{L_1}, \ldots, D_{L_m}\}$ are generated from a given D_L.

Let f_e be a simple relational ensemble model that aggregates predictions from $F = \{f_{s_1}, \ldots, f_{s_m}\}$, where each $f_s \in F$ uses n Gibbs iterations independently for inference. A prediction $y^i_{f_e}$ for a node v_i is then calculated by averaging the final predictions for node v_i from all m models $F = \{f_{s_1}, \ldots, f_{s_m}\}$. Each base model makes its predictions as described for the single collective inference model above.[2]

$$\begin{aligned} y^i_{f_e} &= \frac{1}{m} \sum_{k=1}^{m} f_k(v_i, D_I, \mathcal{T}) \\ &= \frac{1}{m} \sum_{k=1}^{m} P_k(Y^i{=}t^i|\mathbf{Y}_\mathcal{T}, \mathbf{X}, G_I) \end{aligned} \tag{4}$$

Interleaved ensemble model: As described above, let $F = \{f_{s_1}, \ldots, f_{s_m}\}$ be a set of m collective inference models learned from $\{D_{L_1}, \ldots, D_{L_m}\}$ generated from D_L. Then let f_c be an interleaved model that aggregates predictions by taking the average of the probabilities from the m collective inference base models in F, at each Gibbs iteration $j \in \{1, \ldots, n\}$. At each iteration j, predictions made by all the base models are aggregated and used to make a prediction for each model $k \in \{1, \ldots, m\}$.[3] These predictions are for $V_{I \setminus \mathcal{T}}$. For the instances in $\mathcal{T}$, we use the true labels. The final prediction for a node v_i is estimated from the average of the component models' predictions at the last inference iteration n. This defines the interleaved model $f_c = \check{f}_{k,n}$.

$$\begin{aligned} \check{y}^i_{k,j} &= \frac{1}{m} \sum_{k'=1}^{m} f_{k',j}(v_i, D_I, \mathcal{T}) \\ &= \frac{1}{m} \sum_{k'=1}^{m} P_{k'}(Y^i{=}t^i|\mathbf{Y}_\mathcal{T}, \hat{Y}_{V_{I \setminus \{\mathcal{T} \cup v_i\}}, j}, \mathbf{X}, G_I) \\ y^i_{f_c} &= \check{y}^i_{k,n} \end{aligned} \tag{5}$$

2. At the final Gibbs iteration, a prediction is made for a node using the inferred probability distribution.
3. Note the interleaved ensemble model is a Collective Ensemble Classification (CEC) model. For more details, see Section 4.2.

3.1.2. Error decomposition

We decompose error of collective classification models into bias, variance and noise components based on the work of Neville and Jensen (2008). Here we consider squared loss as a measure of classification performance. The loss for model f on node v_i is defined as the expected squared loss for prediction y_f^i given v_i's true label of t^i:

$$L_f^i = E\left[(t^i - y_f^i)^2\right] \tag{6}$$

where E refers to the total expectation taken over the training sets ($D \in \mathcal{D}$) used to learn the model f and subsets of true labels $\mathcal{T}$ available for inference. For readability, the superscript i and subscript f are dropped whenever it is clear from context. Note that in conventional settings, the expectation E would refer to aspects of *learning* and represent the effect of training sets on models/predictions. However, in collective inference settings the relational inference process introduces another source of error (Neville and Jensen, 2008). Thus, to reason about the performance of different relational ensembles, we need to make a distinction between the expectation over *learning* and the expectation over *inference* and the expectation over both. We define these expectations below.

To analyze performance differences, loss can be decomposed into bias, variance, and noise components, and compared across models. For squared loss, the decomposition is additive:

$$L = V + B + N \tag{7}$$

We show the decomposition and define each component below.

$$\begin{aligned}
E[L] &= E[(t-y)^2] \\
&= E[t^2 - 2ty + y^2] \\
&= E[y^2] - 2E[t]E[y] + E[t^2] \\
&= E[y^2] - 2E[t]E[y] + E[t^2] + E[y]^2 - E[y]^2 \\
&= V + E[y]^2 - 2E[t]E[y] + E[t^2] \\
&= V + E[y]^2 - 2E[t]E[y] + E[t^2] + E[t]^2 - E[t]^2 \\
&= V + (E[t] - E[y])^2 - E[t]^2 + E[t^2] \\
&= V + B + E[t^2] - E[t]^2 \\
&= V + B + N
\end{aligned}$$

Variance defined as $V = E\left[(E[y] - y)^2\right]$ is the average loss incurred by all predictions y, relative to the mean prediction $E[y]$.

Bias defined as $B = (E[t] - E[y])^2$ is the loss incurred by the mean prediction, relative to the Bayes-optimal value for node v_i: $E[t]$ (the expected value of the true label).

Noise defined as $N = E\left[(t - E[t])^2\right]$ is the loss incurred due to noise in the labels of the data, which is independent of the learning algorithm.

3.1.3. Expectations

We define the three types of expectations that will be used in the proofs—expectations over *learning*, *inference*, and *total*. Note these expectations are defined for the predictions that will be made by the single model f_s for a test data set D_I.

Expected learning prediction: This is the expectation over *learning*, where the prediction for a node v_i is estimated using *exact inference* based on the set of Bayes-optimal predictions for the rest of the graph, $\tilde{\mathbf{Y}}_{V_{I\setminus v_i}}$:

$$\begin{aligned} E_L[y^i_{f_s}|D_I] &= \sum_{D_L\in\mathcal{D}} P_s(Y^i{=}t^i|\tilde{\mathbf{Y}}_{V_{I\setminus v_i}},\mathbf{X},G_I) * p(D_L) \\ &= P_*(Y^i{=}t^i|\tilde{\mathbf{Y}}_{V_{I\setminus v_i}},\mathbf{X},G_I) \end{aligned} \tag{8}$$

Expected inference prediction: This is the expectation over *inference*, where the prediction for a node v_i is estimated using the model $f_s^{D_L}$ learned from a single training set D_L:

$$\begin{aligned} E_I[y^i_{f_s}|D_I,f_s^{D_L}] &= \sum_{\mathcal{T}} P_s(Y^i{=}t^i|\mathbf{Y}_{\mathcal{T}},\mathbf{X},G_I) * p(\mathbf{Y}_{\mathcal{T}}) \\ &= P_s(Y^i{=}t^i|\mathbf{X},G_I) \end{aligned} \tag{9}$$

Expected total prediction: This is the *total* expectation over learning and inference, where the prediction for a node v_i reflects the prediction that would be made from the true distribution:

$$\begin{aligned} E_T[y^i_{f_s}|D_I] &= E_{LI}[y^i_{f_s}|D_I] \\ &= \sum_{\mathcal{T}} p(\mathbf{Y}_{\mathcal{T}}) \sum_{D_L\in\mathcal{D}} P_s(Y^i{=}t^i|\mathbf{Y}_{\mathcal{T}},\mathbf{X},G_I) * p(D_L) \\ &= P_*(Y^i{=}t^i|\mathbf{X},G_I) \end{aligned} \tag{10}$$

Note that all possible learning scenarios is covered by all possible training graphs, therefore the expectation is over all possible training graphs in (8). On the other hand, all possible inference scenarios is covered by all possible/alternative (partial) labelings provided on the test set, therefore the expectation is over all possible labelings in (9).

3.2. Analysis

Given the framework described above, we compare the performance of the ensemble models to the single model and show that the ensembles reduce total loss. Specifically, we decompose the error of the single collective inference model f_s, the relational ensemble model f_e, and the interleaved ensemble model f_c. Our analysis shows that the interleaved ensemble results in the greatest reduction in error, through its reduction of *both* learning and inference variance.

We refer to y_s as an arbitrary prediction from a single collective inference model f_s, y_e as an arbitrary prediction from a relational ensemble f_e, and y_c as an arbitrary prediction from

an interleaved ensemble model f_e. The proofs below make use of the following assumptions.

Noise equivalence: The noise component of error is dependent upon the data set, and is independent of the classification algorithm. Therefore:

$$N_s = N_e = N_c \tag{11}$$

Dataset independence: The data graph samples $\{D_{L_s}\}_{s=1,\ldots,m}$ used for learning the m models and D_I used for inference are drawn independently from $\mathcal{D}$. When the datasets are independent, the total expectation can be computed from the learning and inference expectations as follows:

$$E_T[.] = E_I[E_L[.]] \tag{12}$$

Predictions from relational ensemble: In the simple relational ensemble f_e, when the number of base models m approaches ∞, the ensemble prediction $y^i_{f_e}$ approaches the expected prediction of the single model f_s, when the expectation is over *learning* (i.e., $E_L[y^i_s]$). Since the predictions from f_e are conditioned on a single labeling $\mathcal{T}$, the ensemble prediction does not approach the *total* expected prediction of the single model (i.e., it does not reflect the variation over inference).

$$\lim_{m\to\infty} y_e = E_L[y_s] = P_*(Y^i{=}t^i|\tilde{\mathbf{Y}}_{V_{I\setminus v_i}}, \mathbf{X}, G_I) \tag{13}$$

Predictions from interleaved relational ensemble: In the interleaved relational ensemble f_c, when both the number of base models m and the number of inference iterations n approach ∞, the interleaved prediction $y^i_{f_c}$ approaches the expected prediction of the single model f_s, where the expectation is over *both* learning and inference (i.e., $E_T[y^i_s]$). This is because the interleaving process, which conditions on $\hat{Y}_{D_{I\setminus\{\mathcal{T}\cup v_i\}},j}$ at each inference iteration j, simulates draws from alternative labelings $\mathcal{T}$ over the course of inference.

$$\lim_{m,n\to\infty} y_c = E_T[y_s] = P_*(Y^i{=}t^i|\mathbf{X}, G_I) \tag{14}$$

This is discussed further in Section 3.2.6.

3.2.1. Variance reduction

When squared loss is decomposed, the variance component is $V_T = E_T\left[(E_T[y] - y)^2\right]$. Here we consider the expected *total* error, over both learning and inference. We now show that a simple relational ensemble reduces the variance of a single model, and an interleaved ensemble reduces the variance of a relational ensemble.

Theorem 1 *Let f_s be a single collective inference model. Let V_s be the expected variance over f_s obtained from randomly drawing train and test sets. Let f_e be a simple relational ensemble. Let V_e be the expected variance over f_e obtained from randomly drawing train and test set. Let f_c be an interleaved ensemble model. Let V_c be the expected variance over f_c. Then $V_s \geq V_e \geq V_c$.*

$$(1.1)\quad V_s - V_e \geq 0$$
$$(1.2)\quad V_e - V_c \geq 0$$

Proof of Theorem 1.1

$$
\begin{aligned}
V_s - V_e &= E_T\left[(E_T[y_s]-y_s)^2\right] - E_T\left[(E_T[y_e]-y_e)^2\right] \\
&= E_T\big[E_T[y_s]^2 - 2y_s E_T[y_s] + y_s^2\big] - E_T\big[E_T[y_e]^2 - 2y_e E_T[y_e] + y_e^2\big] \\
&= E_T[y_s]^2 - 2E_T[y_s]^2 + E_T[y_s^2] - E_T[y_e]^2 + 2E_T[y_e]^2 - E_T[y_e^2] \\
&= -E_T[y_s]^2 + E_T[y_s^2] + E_T[y_e]^2 - E_T[y_e^2] \\
&= -E_T[y_s]^2 + E_T[y_s^2] + E_T\left[E_L[y_s]\right]^2 - E_T\left[E_L[y_s]^2\right] && \textbf{(by 13)} \\
&= -E_I[E_L[y_s]]^2 + E_T[y_s^2] + E_I[E_L[y_s]]^2 - E_T\big[E_L[y_s]^2\big] && \textbf{(by 12)} \\
&= E_T[y_s^2] - E_T\big[E_L[y_s]^2\big] \\
&= E_I\big[E_L[y_s^2]\big] - E_I\left[E_L[y_s]^2\right] && \textbf{(by 12)} \\
&= E_I\big[E_L[y_s^2] - E_L[y_s]^2\big] \\
&\geq 0 && (E_L[y_s^2] - E_L[y_s]^2 \geq 0 \textbf{ by Jensen's Inequality})
\end{aligned}
$$

■

Proof of Theorem 1.2

$$
\begin{aligned}
V_e - V_c &= E_T\left[(E_T[y_e]-y_e)^2\right] - E_T\left[(E_T[y_c]-y_c)^2\right] \\
&= E_T\big[E_T[y_e]^2 - 2y_e E_T[y_e] + y_e^2\big] - E_T\big[E_T[y_c]^2 - 2y_c E_T[y_c] + y_c^2\big] \\
&= E_T[y_e]^2 - 2E_T[y_e]^2 + E_T[y_e^2] - E_T[y_c]^2 + 2E_T[y_c]^2 - E_T[y_c^2] \\
&= -E_T[y_e]^2 + E_T[y_e^2] + E_T[y_c]^2 - E_T[y_c^2] \\
&= -E_T\left[E_L[y_s]\right]^2 + E_T\left[E_L[y_s]^2\right] + E_T[y_c]^2 - E_T[y_c^2] && \textbf{(by 13)} \\
&= -E_T\left[E_L[y_s]\right]^2 + E_T\left[E_L[y_s]^2\right] + E_T\left[E_T[y_s]\right]^2 - E_T\left[E_T[y_s]^2\right] && \textbf{(by 14)} \\
&= -E_T\left[E_L[y_s]\right]^2 + E_T\left[E_L[y_s]^2\right] + E_T[y_s]^2 - E_T[y_s]^2 \\
&= -E_I\left[E_L[y_s]\right]^2 + E_I\left[E_L[y_s]^2\right] && \textbf{(by 12)} \\
&= E_I\left[E_L[y_s]^2\right] - E_I\left[E_L[y_s]\right]^2 \\
&\geq 0 && \textbf{(by Jensen's Inequality)}
\end{aligned}
$$

■

Single collective models f_s have two sources of variance in their predictions—variance due to learning the models from different training graphs, and variance due to applying the model for inference given different labeled subsets of the test graph. Relational ensembles f_e average model predictions from different learned models and reduce the variance due to learning. Thus, $V_s \geq V_e$.

Similar to relational ensembles, interleaved ensembles f_c reduce the variance due to learning. Moreover, interleaving predictions across the base models during each collective inference iteration simulates draws from alternative labeled subsets of the inference graph, and prevents any of the base models from converging to extreme state. This allows an additional reduction of the inference variance. Thus, $V_c \geq V_e$.

3.2.2. Bias reduction

When squared loss is decomposed, the bias component is $B_T = (E_T[t] - E_T[y])^2$. We consider the expected *total* error, over both learning and inference. We now show that the two relational ensembles have the same bias as the single model. Since bias depends on how well the models can approximate the true model, it is not corrected by the relational or interleaved ensemble.

Theorem 2 *Let f_s be a single collective inference model with bias B_s, f_e be a relational ensemble with bias B_e, and f_c be an interleaved ensemble model with bias B_c. Then $B_s = B_e = B_c$.*

$$(2.1) \quad B_s - B_e = 0$$
$$(2.2) \quad B_e - B_c = 0$$

Proof of Theorem 2.1

$$\begin{aligned} B_s - B_e &= (E_T[t] - E_T[y_s])^2 - (E_T[t] - E_T[y_e])^2 \\ &= (E_T[t] - E_T[y_s])^2 - (E_T[t] - E_T[E_L[y_s]])^2 && \textbf{(by 13)} \\ &= (E_T[t] - E_T[y_s])^2 - (E_T[t] - E_T[y_s])^2 \\ &= 0 \end{aligned}$$ ■

Proof of Theorem 2.2

$$\begin{aligned} B_e - B_c &= (E_T[t] - E_T[y_s])^2 - (E_T[t] - E_T[y_c])^2 \\ &= (E_T[t] - E_T[E_L[y_s]])^2 - (E_T[t] - E_T[E_T[y_s]])^2 && \textbf{(by 13, 14)} \\ &= (E_T[t] - E_T[y_s])^2 - (E_T[t] - E_T[y_s])^2 \\ &= 0 \end{aligned}$$ ■

3.2.3. Loss reduction

Given the reduction in variance and equivalent bias, we can analyze the reduction in error that the ensembles offer. Recall that we define total loss as the expected error over learning and inference $L = E_T[(t^i - y_f^i)^2]$ and this decomposes into variance, bias and noise components: $L = V + B + N$. We now show that a relational ensemble reduces the loss of a single model, and an interleaved ensemble reduces the loss of a relational ensemble.

Corollary 1 *Let f_s be a single collective inference model with loss L_s, f_e be a relational ensemble with loss L_e, and f_c be an interleaved ensemble model with loss L_c. Then $L_s \geq L_e \geq L_c$.*

$$(1.1) \quad L_s - L_e \geq 0$$
$$(1.2) \quad L_e - L_c \geq 0$$

Proof of Corollary 1.1

$$\begin{aligned} L_s - L_e &= (V_s + B_s + N_s) - (V_e + B_e + N_e) \\ &= (V_s + B_s + N_s) - (V_e + B_s + N_s) && \textbf{(by 11, Thm. 2)} \\ &= V_s - V_e \\ &\geq 0 && \textbf{(by Thm. 1.1)} \end{aligned}$$

■

Proof of Corollary 1.2

$$\begin{aligned} L_e - L_c &= (V_e + B_e + N_e) - (V_c + B_c + N_c) \\ &= (V_e + B_s + N_s) - (V_c + B_s + N_s) && \textbf{(by 11, Thm. 2)} \\ &= V_e - V_c \\ &\geq 0 && \textbf{(by Thm. 1.2)} \end{aligned}$$

■

Following the results of Theorems 1 and 2, and according to the definition of noise, it is straightforward to make the above conclusion about reduction in error. A relational ensemble model will reduce the error of a single collective inference model by reducing the learning variance, and an interleaved ensemble will reduce the error even further by reducing *both* learning variance *and* inference variance.

3.2.4. Learning variance reduction

In Section 3.2.1, we presented the reduction of total variance component of error of the two ensemble models. Total variance can be decomposed in learning and inference variance components. Next, we analyze the learning and inference variance components of the ensemble models, to show how they reduce total variance.

Learning variance: Here learning variance, $V_L = E_L[(E_L[y] - y)^2]$ is the average loss incurred by all predictions y, relative to the mean learning prediction $E_L[y]$. This measures the variance in predictions made for the same instances by models learned from different training datasets.

Theorem 3 *Let f_e be a relational ensemble with learning variance V_{L_e}, and f_c be an interleaved ensemble model with learning variance V_{L_c}. Then in the limit, as the number of base models m approaches ∞, both f_e and f_c are able to eliminate learning variance components V_{L_e} and V_{L_c}.*

$$V_{L_e} = 0 \quad (3.1)$$
$$V_{L_c} = 0 \quad (3.2)$$

Proof of Theorem 3.1

$$V_{L_e} = E_L\left[(E_L[y_e] - y_e)^2\right]$$

$$
\begin{aligned}
&= E_L\big[E_L[y_e]^2 - 2y_e E_L[y_e] + y_e^2\big] \\
&= E_L[y_e]^2 - 2E_L[y_e]^2 + E_L[y_e^2] \\
&= -E_L[y_e]^2 + E_L[y_e^2] \\
&= -E_L\left[E_L[y_s]\right]^2 + E_L\left[E_L[y_s]^2\right] && \textbf{(by 13)} \\
&= -E_L[y_s]^2 + E_L[y_s]^2 \\
&= 0 && \blacksquare
\end{aligned}
$$

Proof of Theorem 3.2

$$
\begin{aligned}
V_{L_c} &= E_L\left[(E_L[y_c] - y_c)^2\right] \\
&= E_L\big[E_L[y_c]^2 - 2y_c E_L[y_c] + y_c^2\big] \\
&= E_L[y_c]^2 - 2E_L[y_c]^2 + E_L[y_c^2] \\
&= -E_L[y_c]^2 + E_L[y_c^2] \\
&= -E_L\left[E_{LI}[y_s]\right]^2 + E_L\left[E_{LI}[y_s]^2\right] && \textbf{(by 14)} \\
&= -E_{LI}[y_s]^2 + E_{LI}[y_s]^2 \\
&= 0 && \blacksquare
\end{aligned}
$$

Learning variance measures the variation in predictions due to learning the models from different training graphs. Both relational ensembles f_e and interleaved ensembles f_c average model predictions from different learned models to eliminate learning variance. Thus in the limit, $V_{L_s} \geq V_{L_e} = V_{L_c}$.

3.2.5. Inference variance reduction

Inference variance: Here inference variance is defined as $V_I = \alpha - \beta$, where $\alpha = E_{LI}[(E_L[y] - y)^2]$ is the average loss incurred by all predictions y relative to the mean learning prediction $E_L[y]$, while $\beta = E_L[(E_{LI}[y] - y)^2]$ is the average loss incurred by the predictions for y that use exact inference (using Bayes-optimal predictions for all other instances in the data), relative to the overall mean prediction $E_{LI}[y]$. Inference variance measures the variation in predictions made for the same node by the same model given different labeled subsets of the test graph. Inference variance can also be defined as the difference between total variance and learning variance.

Corollary 2 *Let V_T, V_L, and V_I be the total, learning, and inference variance, respectively. Then $V_I = V_T - V_L$.*

Proof of Corollary 2

$$
\begin{aligned}
V_I &= \alpha - \beta \\
&= E_{LI}[(E_L[y] - y)^2] - E_L[(E_{LI}[y] - y)^2]
\end{aligned}
$$

$$
\begin{aligned}
&= E_{LI}[(E_L[y])^2 - 2yE_L[y] + y^2] - E_L[(E_{LI}[y])^2 - 2yE_{LI}[y] + y^2] \\
&= (E_L[y])^2 - 2E_{LI}[y]E_L[y] + E_{LI}[y^2] - (E_{LI}[y])^2 + 2E_L[y]E_{LI}[y] - E_L[(y)^2] \\
&= (E_{LI}[y^2] - (E_{LI}[y])^2) - (E_L[(y)^2] - (E_L[y])^2) \\
&= V_T - V_L
\end{aligned}
$$

■

Theorem 4 *Let f_e be a simple relational ensemble with inference variance V_{I_e}, and f_c be an interleaved ensemble model with inference variance V_{I_c}. Then in the limit, as the number of base models m and the number of inference iterations n both approach ∞, f_e can not eliminate inference variance V_{I_e}, while f_c can eliminate inference variance V_{I_c}.*

$$
\begin{aligned}
&(4.1) \quad V_{I_e} \geq 0 \\
&(4.2) \quad V_{I_c} = 0
\end{aligned}
$$

Proof of Theorem 4.1

$$
\begin{aligned}
V_{I_e} &= V_{T_e} - V_{L_e} \\
&= (E_{LI}[(E_{LI}[y_e] - y_e)^2]) - (E_L[(E_L[y_e] - y_e)^2]) \\
&= (E_{LI}[y_e^2] - (E_{LI}[y_e])^2) - (E_L[(y_e)^2] - (E_L[y_e])^2) \\
&= E_{LI}[y_e^2] - (E_{LI}[y_e])^2 - E_L[(y_e)^2] + (E_L[y_e])^2 \\
&= E_{LI}[y_e^2] - (E_{LI}[y_e])^2 - E_L[(E_L[y_s])^2] + (E_L[E_L[y_s]])^2 && \textbf{(by 13)} \\
&= E_{LI}[y_e^2] - (E_{LI}[y_e])^2 - (E_L[y_s])^2 + (E_L[y_s])^2 \\
&= E_{LI}[y_e^2] - (E_{LI}[y_e])^2 \\
&= E_{LI}[(E_L[y_s])^2] - (E_{LI}[E_L[y_s]])^2 && \textbf{(by 13)} \\
&\geq 0 && \textbf{(by Jensen's Inequality)}
\end{aligned}
$$

■

Proof of Theorem 4.2

$$
\begin{aligned}
V_{I_c} &= V_{T_c} - V_{L_c} \\
&= (E_{LI}[y_c^2] - (E_{LI}[y_c])^2) - (E_L[(y_c)^2] - (E_L[y_c])^2) \\
&= E_{LI}[y_c^2] - (E_{LI}[y_c])^2 - E_L[(y_c)^2] + (E_L[y_c])^2 \\
&= E_{LI}[y_c^2] - (E_{LI}[y_c])^2 - E_L[(E_{LI}[y_s])^2] + (E_L[E_{LI}[y_s]])^2 && \textbf{(by 14)} \\
&= E_{LI}[y_c^2] - (E_{LI}[y_c])^2 - (E_{LI}[y_s])^2 + (E_{LI}[y_s])^2 \\
&= E_{LI}[y_c^2] - (E_{LI}[y_c])^2 \\
&= E_{LI}[(E_{LI}[y_s])^2] - (E_{LI}[E_{LI}[y_s]])^2 && \textbf{(by 14)} \\
&= (E_{LI}[y_s])^2 - (E_{LI}[y_s])^2 \\
&= 0
\end{aligned}
$$

■

Inference variance measures the variation in predictions due to applying the model given different labeled subsets of the test graph. Interleaved ensembles f_c eliminate inference

variance by interleaving predictions across the base models during each collective inference iteration, which simulates draws from alternative labeled subsets of the inference graph, and prevents any of the base models from converging to extreme state. However, simple relational ensembles f_e can not achieve this inference variance elimination because they only average the predictions of the models after the inference process is complete.

3.2.6. Ensemble learning

The error analysis presented above holds for ensembles constructed from either i.i.d. resampling where instances are sampled independently with replacement or relational subgraph resampling (RSR shown in Algorithm 2). As the number of pseudosamples m approaches ∞, the bootstrap samples approximate the true population distribution $\mathcal{D}$ and the models in F approximate P_* Breiman (1996a). This indicates that for the ensemble model f_e, Eq. 13 holds regardless of the learning approach. In other words, in expectation the ensemble prediction $y^i_{f_e}$ approaches the expected prediction of the single model f_s over *learning* (i.e., $E_L[y^i_s]$) for both resampling methods:

$$\lim_{m\to\infty} y_e^{RSR} = \lim_{m\to\infty} y_e^{IID} = E_L[y_s] \tag{15}$$

Furthermore, y_e^{RSR} converges faster than y_e^{IID} since pseudosamples from RSR (Algorithm 2) more accurately reflect the correlations in relational data. Thus, given a finite ensemble size m, predictions made by models learned from RSR pseudosamples will capture and reduce more learning variance (due to RSR more accurately capturing the increased variance in network data). The same argument applies to f_c. Thus, Eq. 14 holds regardless of the resampling approach, but in finite ensemble sizes, RSR pseudosamples will capture and reduce more variance.

Our analysis illustrates the errors due to different phases of an ensemble algorithm. This understanding points to an additional way of reducing error due to variance in learning. In particular, the better the set of training samples can approximate the true population variance, the more reduction in learning variance the final model aggregation can achieve. Using RSR can more accurately capture the increased variance in relational data, specifically using a small number of bootstrap samples. Following this observation, we propose to use RSR to enable the final predictions aggregation to reduce more learning variance. We combine the use of RSR with the interleaved inference aggregation (i.e., CEC) to additionally reduce inference variance. We outline the algorithmic details in the next section.

4. Relational Ensemble Framework

In this section, we describe the relational ensemble framework. Given a training dataset, the proposed relational ensemble framework uses a resampling approach (such as RSR in Algorithm 2) to generate m bootstrap pseudosamples to learn an ensemble of m models (Section 4.1). The models are applied for collective inference on a single test set using CEC, which iteratively interleaves the inferences across the m models (Section 4.2). After inference is finished, the predictions given by each base model are aggregated for each node independently as in traditional ensembles.

4.1. Ensemble Learning

We propose two alternative approaches for learning the base models of the ensemble, depending on the network setting. The first method is suitable for networks composed of a single graph (Section 4.1.1), while the second method is for multi-graph networks (Section 4.1.2). Using either of these methods reduces the error due to variance in learning.

4.1.1. Ensemble learning from a single graph

Given a single graph $G_{tr} = (V_{tr}, E_{tr})$, the goal is to generate multiple training graphs to learn an ensemble of base models. We use the bagging approach in which the psuedosamples used for learning are networks sampled with replacement from the training graph. However, instead of sampling node instances independently with replacement as typically done in bagging, we propose a *relational subgraph resampling* approach to generate m bootstrap pseudosamples for learning an ensemble of m models.

The ensemble learning approach using bootstrap sampling is summarized in Algorithm 1 and demonstrates how an ensemble of m models is constructed. A pseudosample $G_j = (V_j, E_j)$ is generated by resampling from G_{tr} (line 3) and a model F_j is learned from G_j (line 4). F_j is a joint probability distribution over the labels of V_j, conditioned on the observed attributes and graph structure in G_j. The ensemble set $\mathcal{M}$ of m learned models is returned (line 6). Note that the two main components needed for an implementation of Algorithm 1 are a resampling algorithm (line 3) and a learning algorithm (line 4).

Algorithm 1 Ensemble Learning: EL($G_{tr} = (V_{tr}, E_{tr}), m$)

```
1 Set M ← ∅
2 for j = 1, ..., m do
3     G_j = Resample(G_tr)              ▷ Construct pseudosample of G_tr
4     F_j = LearnModel(G_j)             ▷ Learn model F_j from pseudosample
5     M = M ∪ {F_j}                     ▷ Add model F_j to ensemble M
6 return ensemble of relational models M
```

Resampling. In this work, we propose Relational Subgraph Resampling (RSR) for resampling relational data to accurately capture the increased variance due to linkage and autocorrelation.[4] However, any resampling approach can be used instead. RSR samples *subgraphs* with replacement instead of the typical independent sampling technique that samples *instances* (i.e., nodes) with replacement.

The proposed RSR approach is shown in Algorithm 2. Given a relational data graph $G = (V, E)$, it returns a pseudosample data graph $G_{PS} = (V_{PS}, E_{PS})$. A set of $N_S = \lceil \frac{|V|}{b} \rceil$ subgraphs of size b are sampled with replacement from G. Each of the N_S subgraphs are sampled using a breadth-first search (BFS) from a randomly selected seed node. As a node v is added to the sampled subgraph node set V_S, the neighbors of v are added to a queue Q, from which the next node v is popped. This continues until the subgraph size b is reached. Note that the sampling is with replacement from the graph, so a node may appear

4. Autocorrelation is the statistical dependency of the same attribute on related (neighboring) node instances Leenders (2002); Xiang et al. (2010).

Algorithm 2 Relational Subgraph Resampling: RSR($G = (V, E), b$)

$V_{PS} \leftarrow \emptyset$, $E_{PS} \leftarrow \emptyset$ ▷ Initialize pseudosample node and edge set
for $s = 1, \ldots, \lceil \frac{|V|}{b} \rceil$ **do** ▷ For each sampled subgraph s
  Set $V_S \leftarrow \emptyset$, $E_S \leftarrow \emptyset$, $Q \leftarrow \emptyset$
  Select v uniformly at random from V
  $V_S \leftarrow V_S \cup \{v\}$ ▷ Add node v to the set of V_S sampled nodes
  $Q \leftarrow Q \cup$ neighbors of v ▷ Add neighbors of v to the queue Q (for BFS step)
  while $(|V_S| < b) \wedge (|Q| > 0)$ **do** ▷ Add at most $b - 1$ nodes to V_S via BFS
    $v = \text{pop}(Q)$ ▷ Set next node in Q to v and remove v from Q
    $V_S \leftarrow V_S \cup \{v\}$ ▷ Add v to the sampled node set V_S of subgraph s
    $Q \leftarrow Q \cup$ neighbors of v ▷ Add neighbors of v to the end of Q
  $E_S = \{e_{ij} \in E \ \ s.t. \ \ v_i, v_j \in V_S\}$ ▷ All edges in E with both endpoints in V_S
  Set $V_{PS} \leftarrow V_{PS} \cup V_S$ and $E_{PS} \leftarrow E_{PS} \cup E_S$
return a pseudosample $G_{PS} = (V_{PS}, E_{PS})$

in multiple subgraphs, one subgraph, or none. The pseudosample node set (V_{PS}) consists of all the nodes selected in the subgraphs (suitably relabeled so multiple copies of the same original node are distinguishable for the learning algorithm). The pseudosample edge set (E_{PS}) consists of all the edges within the selected subgraphs. Note sets are assumed to be multisets and therefore can contain multiple instances of the same element.

The intuition behind sampling subgraphs with replacement is that when autocorrelation is high (i.e., neighbor labels are correlated), the effective sample size is going to be closer to the number of "groups" of correlated instances than the number of nodes in the network. To account for this, RSR attempts to sample these "groups" instead of single instances, thus it more accurately approximates the effective sample size of the data. Moreover, sampling subgraphs preserves the local relational dependencies among instances in the subgraph so the relational model is better able to utilize the interrelated attribute dependencies to improve classification. In the traditional independent sampling technique, a node in the pseudosample will not necessarily have its neighbors from the original sample, and therefore the model will be less capable of exploiting the link structure.

4.1.2. Ensemble learning from multiple link graphs

Consider the problem of collective node classification in domains where a single set of objects (i.e., V) is connected through multiple link graphs (i.e., $G_1 = (V, E_1), G_2 = (V, E_2), ...$). For instance, a friendship graph in an online social network consists of links connecting users listed as friends, a message graph connects users that communicate via messages, and a photo graph can also be constructed where a photo-tag link connects users that tag one another in photos. For these types of networks with different types of *relations* (link types), each graph provides complementary information about the same set of objects and can thus be viewed as a different "source" of link information. For predicting a single class label Y (e.g., political views) over the set of nodes V given multiple types of links among V, the goal is to combine the link sources to improve the quality of inferences produced from collective classification. There are two primary ways to combine the various link sources to improve prediction—either by combining the sources before learning and then learning a joint model across all graphs, or by combining the sources after learning, which can be done

by learning an ensemble of models, one from each source. As discussed previously, in order to reduce the prediction error due to variance (particularly due to the collective inference process), this work focuses on the latter.

For learning an ensemble in the multi-graph setting, each base model is learned independently from one link graph using an arbitrary relational learning (RL) method. The resulting models comprise a set of joint probability distributions over the labels of the nodes of the training network. This is analogous to learning a set of ensemble models by using different feature subsets Cunningham and Carney (2000), but in this case link types are treated as features. For the Facebook example, this will correspond to learning one model from each of the friendship, message exchange, and photo-tagging graphs. This method of ensemble learning uses the complete set of nodes in the training network for learning each model, as opposed to the proposed bootstrap sampling approach described in Section 4.1 that learns models from subsets of a single graph.

4.1.3. Relational learning

For learning a set of models (line 4 in Algorithm 1) given psuedosamples from a single training graph (Section 4.1.1) or a set of graphs with different link types (from Section 4.1.2), we can use any arbitrary relational learning (RL) method. For the experiments in Section 5, we use relational dependency network (RDN) Neville and Jensen (2007) models as the component collective classification models. RDNs are selective models based on decision trees and therefore exhibit the instability that typically works well in bagged ensembles. They use pseudolikelihood estimation to efficiently learn a full joint probability distribution over the labels of the data graph and use Gibbs sampling for collective inference. As an aside, the full joint distribution over the test data does not need to be estimated for accurate inference and it is sufficient to accurately estimate the per instance conditional likelihoods, which is easy to do with Gibbs sampling and has been shown to converge within 500-2000 Gibbs iterations Neville and Jensen (2007).

4.2. Ensemble Inference

In this section, we propose an *across-model* collective classification method where inferences are propagated across the models of the ensemble during collective inference (See Figure 2). The proposed method is called Collective Ensemble Classification (CEC) and is outlined in Algorithm 3. Given a test network G with partially labeled nodes V and m base models $F_1, F_2, \ldots, F_m$ learned using either approaches described previously in Section 4.1, the models are applied simultaneously to collectively predict the values of unknown labels (lines 5-11). First, the labels are randomly initialized (lines 1-4). In particular, line 3 selects a node v_j with an unknown label and line 4 initializes the label of v_j randomly. Next, at each collective inference iteration, the model F_i is used to infer a label for each node v conditioned on the current labels of the neighbors of v (line 8). This corresponds to a typical collective inference iteration. Then instead of using the prediction from F_i directly for the next round, it is averaged with the inferences for v made by each other model F_j s.t. $j \neq i$ (line 9). This interleaves inferences across the component models and pushes the variance reduction gains into the collective inference process itself. At the end, the predictions are calculated for each model based on the stored prediction values from each

Algorithm 3 Collective Ensemble Classification (CEC)

CEC($F_1, F_2, \ldots, F_m$, G=(V, E), X,$\tilde{Y}$, F_m=$P(Y_i|G, X, Y)$)

/* initialize labels for each model */
1 **for** $i = 1, \ldots, m$ **do**
2 Set $\hat{Y}^i = \tilde{Y}$ and $\mathbf{Y_T^i} = \emptyset$
3 **for each** node v_j with unobserved label **do**
4 Randomly initialize $\hat{y}_j^i$ and set $\hat{Y}^i = \hat{Y}^i \cup \{\hat{y}_j^i\}$
5 **repeat**
  /* inference within each model */
6 **for** $i = 1, \ldots, m$ **do**
7 **for each** node v_j with unobserved label **do**
  /* use F^i to infer label of v_j conditioned on current neighbor labels */
8 $\hat{y}_j^{i_{new}}$ = label inferred from $F^i := P^i(Y_j|\mathbf{X_j^i}, \mathbf{X_R^i}, \hat{\mathbf{Y}}_\mathbf{R}^\mathbf{i})$ where $\mathbf{R} = \{v_k : e_{jk} \in E_i\}$
  /* merge predictions across models */
9 $\hat{y}_j^{i_{agg}} = \frac{1}{m} \sum_{j=1}^{m} \hat{y}_j^{i_{new}}$
  /* store merged prediction */
10 Set $\hat{Y}^i = \hat{Y}^i \setminus \{\hat{y}_j^i\} \cup \{\hat{y}_j^{i_{agg}}\}$ and $\mathbf{Y_T^i} = \mathbf{Y_T^i} \cup \{\hat{y}_j^{i_{agg}}\}$
11 **until** terminating condition is satisfied
/* make predictions for each model */
12 **for** $i = 1, \ldots, m$ **do**
13 Compute predictions $\mathbf{P^i}$ (for all nodes with unobserved labels) using $\mathbf{Y_T^i}$
/* combine predictions across models (voting models outputs) */
14 $P = \emptyset$
15 **for each** node v_j with unobserved label **do**
16 $p_j = \frac{1}{m} \sum_{i=1}^{m} p_j^i$ and set $P = P \cup \{p_j\}$
17 **return** P

collective inference iteration (lines 12-13). Finally, model outputs are averaged to produce the final predictions (lines 15-16).

The manner that CEC uses inferences from other models (for the same node) provides more information to the inference process that is not available if the collective inference processes are run independently on each base model. Since each collective inference process can experience error due to variance from approximate inference, the ensemble averaging during inference can reduce these errors before they propagate throughout the network. This results in significant reduction of inference variance, which is achieved solely by CEC. CEC assumes a collective classification model as the base component of the ensemble. In this work, we use RDNs, though any collective classification model can be used instead. However, our analysis shows that the approach will work particularly well for models that exhibit learning and/or inference variance.

For the multi-graph network setting, the CEC approach shown in Algorithm 3 can be used directly since in this case the m base models $F_1, F_2, \ldots, F_m$ from the ensemble learning (Section 4.1) correspond to m different graphs with the same nodes but different link types (multi-graph setting).

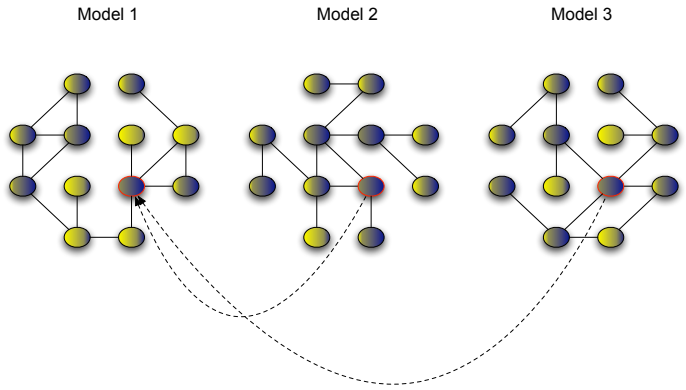

Figure 2: Model interleaving aggregates predictions for the same nodes across the models.

4.3. Complexity Analysis

Let m denote the number of models in the ensemble and let the runtime complexity of the general *relational learning* (RL) algorithm be C_l, then the worst-case time complexity of CEC is $\mathcal{O}(mC_l)$. Further, let the time complexity of *inference* using the general collective classification (CC) algorithm be C_i. Algorithm 3 uses CC m times to learn m models and aggregates over m predictions. Therefore, the time complexity of CEC is $\mathcal{O}(mC_i)$. Since m is a small constant, CEC is computationally efficient with a time complexity that is comparable to a single relational model. As an aside, since the proposed relational ensemble framework can leverage any RL and CC algorithm, we use C_l and C_i above to denote their runtime complexity, respectively. Suppose, the RL and CL algorithms have a worst-case time complexity that is linear in the number of edges, then the proposed relational ensemble approach is also linear in the number of edges since k is typically a small constant.

5. Experimental Evaluation

We evaluate the ensemble method on both synthetic and real world data. Furthermore, we demonstrate the effectiveness of the proposed methods for two different network settings: (i) the single graph setting (Section 5.1) and (ii) the multi-graph setting where there are multiple graphs available with different link types (Section 5.2).

5.1. Results for single graph setting

This section compares the proposed ensemble methods for the single graph setting that assumes there is only a single training graph available for learning a relational ensemble. This is in contrast to Section 5.2 that leverages multiple graphs that all have the same nodes but with different link types. The single graph setting is the most common and general setting for learning graph-based ensembles. We refer the reader to Section 4 for more details on the differences and technical details.

5.1.1. Baseline approaches

We use a number of baseline methods to compare our proposed model to alternative approaches while controlling for model representation.

IID-RE This model uses IID resampling for generating the training pseudosamples and learns a relational model for each base classifier. IID resampling works by sampling instances independently at random from the network with replacement. A link in the original sample will only appear in the pseudosample if both nodes it connects were selected. A simple *relational ensemble* (RE) approach is then used for inference, where each base model is applied independently for collective inference to produce a set of probability estimates for nodes predictions. Then for each node, the base models' predictions are averaged to get the node's final prediction. We compare to this approach to evaluate the combined improvement achieved by using RSR for resampling and CEC for inference over a method that does not use either approach. The goal is to show the total variance reduction offered by RSR and CEC.

RSR-RE This baseline uses RSR for constructing the ensemble and RE for inference. Comparing the performance of our proposed model to this approach allows us to evaluate the improvement achieved by CEC for inference, while controlling for the resampling method (RSR) used by our proposed approach.

IID-CEC This baseline uses IID resampling for ensemble construction and CEC for inference. Comparing the performance of our proposed model to this approach allows us to evaluate the improvement achieved by RSR for sampling, while controlling for the inference method (CEC) used by our proposed approach.

SM A *single model* baseline is used to evaluate the improvement achieved by each ensemble approach. Here, a collective classification model is learned from the original training sample and applied once on the given test set. Note that all sampling based ensembles generate the bootstrap pseudosamples from this original training sample, and use the same collective classification algorithm as the base component model.

5.1.2. Data sets

We evaluate the methods on synthetic and real world network data. Synthetic data sets are generated with a latent group model Neville and Jensen (2005). They are homogeneous (i.e., with a single object type) data graphs with autocorrelation due to an underlying (hidden) group structure. Each object has a boolean class label C (that is determined by the type of group to which it belongs), and three attributes. The class label C has an autocorrelation level of 0.75. We independently constructed five training and test pairs of such data sets, each consisting of 500 objects. Further details including an algorithm summarizes the synthetic generator is provided in Appendix A.

The Facebook dataset used in this work is a sample of Purdue University Facebook network. We construct a friendship graph from the links between friends. Each user has a boolean class label which indicates whether their political view is 'Conservative'. In addition, we considered nine node features which record user profile information. We use 4 sampled networks of users (based on membership in various Purdue subnetworks): [Pur-

due Alum'07, Purdue'08, Purdue'09, Purdue'10] with node sizes of: [921, 827, 1268, 1384] respectively. Then we construct 4 different training and test pairs by testing on one subnetwork and training on two subnetworks from the previous and preceding class networks. For example we learn the model from Purdue Alum'07 and Purdue'09, and apply the model on Purdue'08.

5.1.3. Methodology

The RSR algorithm uses a subgraph size $b = 50$ and $b = 10$ for the synthetic and Facebook experiment, respectively. The methods described are learned and evaluated using RDNs as the base collective classification model, using $450 - 500$ Gibbs iterations for collective inference. We use the following setting to compare the various approaches.

For each experiment, the proportion of the test set that is labeled before inference is specified, and for each trial a random set of nodes is chosen to label. The random labeling process is repeated 10 times. AUC ROC is measured to assess the prediction accuracy of each model. The 10 trials are repeated for 4 training and test pairs, and the averages of the $10 \times 4 = 40$ AUC measurements from each approach are reported. Note that, all methods are run on the same random labeling of the test set. From each training test set and for each sampling approach, we construct 5 bootstrap pseudosamples and learn the ensemble models (i.e., $m = 5$). This is repeated for 4 different labeling proportions (l) in each experiment. Note $l = \{10\%, 30\%, 50\%, 70\%\}$ denotes the x-axis in the figures, while the y-axis represents AUC.

5.1.4. Results

The results for the synthetic and Facebook experiments are shown in Figure 3 and Figure 4, respectively. Overall, the proposed RSR-CEC approach is shown to achieve significantly

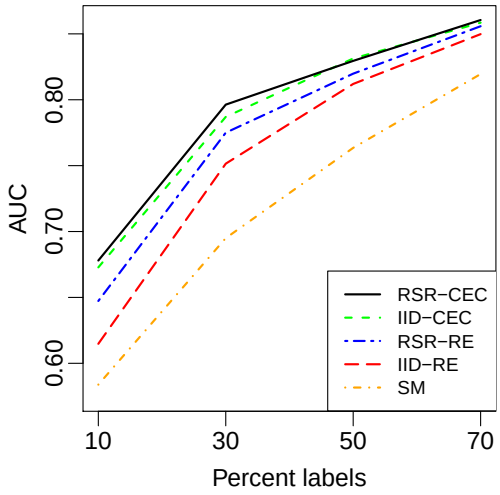

Figure 3: Synthetic experiments comparing the methods using different proportions of known labels in the test graph. The proposed RSR-CEC model is shown to significantly outperform all other baselines across all percent labelings.

higher classification accuracy than all other baseline methods for all percent labelings and across both synthetic and Facebook experiments. The significance is measured using paired t-tests and all significance reported here correspond to $p < 0.0001$ unless stated otherwise. The superior performance of RSR-CEC can be explained by the combined benefit of learning and inference variance reduction.

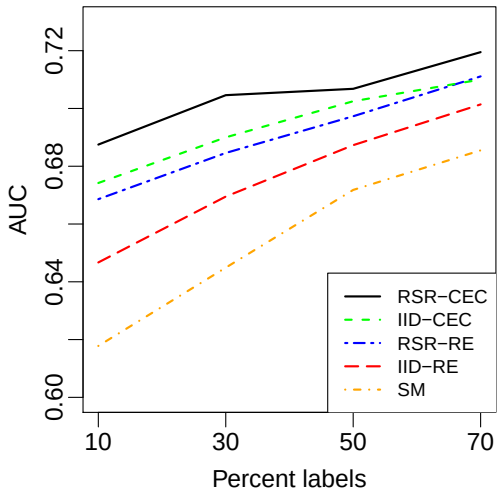

Figure 4: Facebook experiments comparing the methods using different proportions of known labels in the test graph. RSR-CEC is shown to significantly outperform all other baselines across all percent labelings.

In addition, the accuracy of the single model baseline is significantly less than *all* the ensemble models, at all percent labelings for both experiments. Moreover, IID-CEC significantly outperforms IID-RE at all percent labelings for both experiments. This is because CEC reduces inference variance while RE only reduces learning variance. RE applies the models independently for inference which does not reduce inference variance–since prediction aggregation happens after inference, possibly after inference variance has propagated through the graph. Furthermore, RSR-RE significantly outperforms IID-RE at all percent labelings for both experiments, with $p < 0.01$ and $p < 0.03$ for the 50% and 70% synthetic experiments. This is because RSR captures more variance in the data than IID resampling. Therefore, RE can reduce more learning variance when used with RSR. Finally, IID-CEC significantly outperforms RSR-RE at $\{10\%, 30\%, 50\%\}$ for the synthetic experiment. This shows that CEC can reduce both learning and inference variance, even when combined with IID resampling. To summarize the empirical findings:

- Ensembles using RSR outperform ensembles using IID resampling, since RSR reduces more learning variance than IID resampling.
- Ensembles using CEC outperform ensembles using RE, since CEC reduces inference variance which is not reduced by RE.
- Combining RSR with CEC results in significant gains in accuracy, since the combination reduces the largest amount of variance (due to learning and inference).

5.2. Results on multi-source networks

We also evaluate the proposed collective ensemble classification (CEC) approach for the multi-source network setting where each component model is learned from a different link graph with the same nodes. The models are applied interdependently for inference to reduce inference variance. As shown below, this significantly improves classification accuracy for relational graph data where with multiple link types.

5.2.1. Datasets

The first dataset is from a public University Facebook dataset. Three link sources describing different relationships between the same set of users are used. The friendship graph has undirected friendship links. The wall graph has directed links extracted from users' interactions through a public message board on their profile *wall* page. The photo graph has directed links extracted from users tagging others in their profile photo page. Each user has a boolean class label which indicates whether their political view is 'Conservative'. In addition, nine node features and two link features are considered. The object features record user profile information. Wall links have one link feature that counts the number of wall posts exchanged between any two users, while photo links have one link feature that counts the number of photos shared between any two users. Further details about this dataset are provided in Appendix B.

The second data set is from IMDb (Internet Movie Database), which contains movie release information. Five link sources are used. The actors graph links movies that share an actor. Similarly, the studios, producers, directors and editors graphs link movies that share the corresponding aspect. Each movie has a boolean class label which indicates whether the movie is a 'Block buster'. See Appendix C for further details.

The third data set consists of synthetically generated relational data graphs, where relational data characteristics (i.e., linkage and autocorrelation) can be varied. We generate 10 different link sources (for the same set of objects) with different link density structures and link types. Each node has one binary class label. Further details including an algorithm summarizes the synthetic generator is provided in Appendix A.

5.2.2. Baseline approaches

The following baseline methods are considered in order to compare the proposed approach to related work, while controlling for model representation.

Relational Ensemble (RE): The RE baseline uses the same ensemble learning procedure of CEC, but applies each model independently for inference to produce a set of probability estimates for nodes predictions. Then it averages the resulting set of predictions for each node independently to get the final predictions P. This is used to evaluate the improvement achieved by our proposed across-model inference approach (since RE uses the same learning and final prediction averaging as CEC), and is intended to show that the increase in accuracy of CEC cannot be achieved by a straightforward ensemble classification that combines different relations, e.g., as described by Preisach and Schmidt-Thieme (2006).

The limitation of RE is that inference is applied independently on each base model, so the availability of multiple predictions from the ensemble models is only utilized to average the final ensemble predictions—after inference is done and after inference variance has propagated through the graph. Our key insight is that collective classification offers a unique opportunity to jointly utilize information from all the models during collective inference.

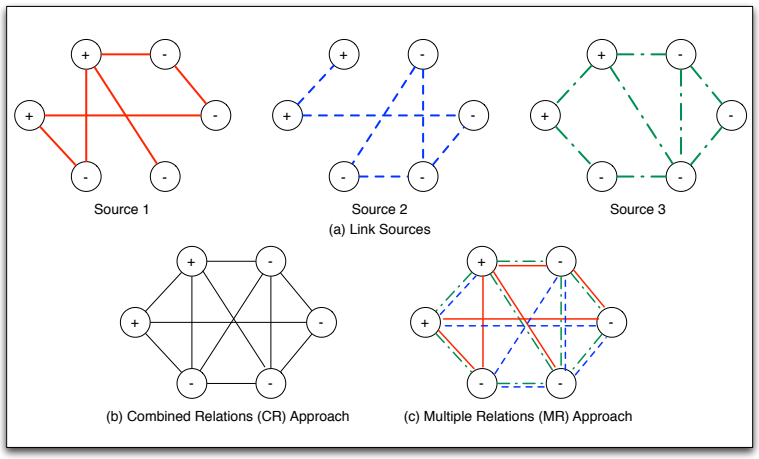

Figure 5: Merging multiple link sources on the same network of nodes.

Multiple Relations (MR): The MR baseline is a single model approach that learns one model from the set of training graphs, using the multiple relation types as features in the model. The learned model is applied collectively to the test graph, producing a single set of predictions. This is used to evaluate the improvement achieved by the relational ensemble approach, by comparing to just using a single model approach that uses the link types as features for learning. MR is similar to methods mentioned in the related work (Section 6) that combine multiple data sources into a single network for learning. Figure 5(c) shows an example merged graph using the MR approach on three example link sources shown in Figure 5(a).

Combined Relations (CR): The CR baseline is another single model approach that learns one model from the set of training graphs. However this method ignores the relation types and just uses the single-source (i.e., attribute) features. The model is also applied collectively on a single, merged test graph that contains all link source information but no link type features, resulting in a single set of predictions. The goal of comparing to this simple method which does not consider the various link types is to assess any gains achieved by considering link types as features in MR. Figure 5(b) shows an example merged graph using the CR approach on three example link sources shown in Figure 5(a).

Single Relation (SR): The SR baseline learns one model from a *single* link source and applies the model collectively to the test network from the same source. One SR model is learned and evaluated for each link source separately. The goal of comparing to this method is to assess the intrinsic value of each relationship in the network when used for

classification by itself. In the experimental results, the average performance of the set of single models is reported.

5.2.3. Methodology

Each of the above methods are evaluated using relational dependency networks (RDNs) Neville and Jensen (2007) as the collective classification model. RDNs use pseudolikelihood estimation to efficiently learn a full joint probability distribution over the labels of the data graph, and are typically applied with Gibbs sampling for collective inference. Note that the full joint distribution over the test data need not be estimated for accurate inference and it is sufficient to accurately estimate the per instance conditional likelihoods.

For each experiment, the proportion of the test set that is labeled before inference is varied, and for each trial a random set of nodes is chosen to label. The labeling process is repeated 5 times, then 5 rounds of inference are run for each random labeling. AUC is measured to assess the prediction accuracy of each model. The $5 \times 5 = 25$ trials are repeated for 5 training and testing pairs, and the averages of the 125 AUC measurements from each approach are reported. The robustness of the methods to missing labels (in the test set) is evaluated by varying the proportion of labeled test data at 10% through 90%. For the synthetic data experiment, results using 3 link sources, high autocorrelation, and low link density setting are reported. For the Facebook dataset, 3 link sources are used: friendship, wall, and photo graphs. For IMDB, 5 link sources are used: actor, studio, producer, director and editor graphs.

The effect of increasing the number of link sources is tested by generating synthetic data with 1, 3, 6 and 9 sources. When there is one source, this corresponds to the SR baseline. In this evaluation, the reported results use 10% labeled nodes in the test set, high autocorrelation, and low link density setting. Note that the same nodes are labeled across all the link graphs, and therefore increasing the number of link graphs does not mean there is more labeled data available, just that more link information is being considered.

Since collective inference in general, and the RDN specifically, have been shown to exploit relational autocorrelation and linkage in relational data Neville and Jensen (2007), the effects of increasing both levels are investigated. The autocorrelation level is varied at low and high using 3 link graphs, each with low link density and 10% labeled test data. Then the linkage level in the data is varied from low to high, using 3 sources, each with high autocorrelation and 10% labeled test data.

5.2.4. Results

Overall, we find that CEC consistently and significantly outperforms the baselines across all experiments. We summarize the key findings below:

- CEC has significantly higher classification accuracy than all the baselines.
- CEC is the most robust to missing labels (due to its ability to best exploit the available label information).
- CEC best exploits the information from additional sources, as well as information due to higher linkage and autocorrelation.

In Figures 6, 7(a) and 7(b), we observe that accuracy increases as a function of the proportion of labeled nodes. Notably, CEC is the most robust technique to missing labels across all data sets. Moreover, CEC significantly ($p < 0.01$) outperforms RE at all label proportions on the synthetic and Facebook data sets, and on the IMDb at labeled proportions through 50%. It is clear that CEC results in huge performance gains over other methods with very few labeled instances. This is because when there is a limited number of labeled neighbors available, CEC is able to best exploit the link information available from the multiple sources to reduce inference error. Although the mean SR performance is plotted, the CEC also outperforms the *best* SR model. Furthermore, CEC is able to improve performance even when the SR models do not have similar performance (e.g., when some perform poorly).

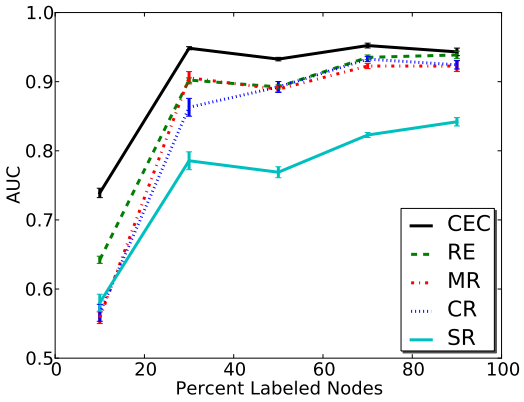

Figure 6: Synthetic experiments show significant accuracy improvement of proposed CEC ensemble model at various proportions of available true labels in the test graph.

In Figure 8, the ensemble methods are shown to improve overall model performance as more sources are considered. Furthermore, CEC significantly outperforms RE ($p < 0.01$). On the other hand, the performance of the single model baselines (MR, CR) degrade. This can be explained by the fact that an ensemble approach (RE) reduces the learning variance, and that interleaving the collective inference processes (CEC) reduces the inference variance on top of that. In contrast, the degradation in performance for the single model baselines can be attributed to the increased variance in the learned model due to the increased number of links and features in the merged graph.

Table 1 shows that the ensemble methods better exploit autocorrelation and link density than the single model baselines. CEC again significantly outperforms RE at both low and high levels of autocorrelation and link density ($p < 0.01$). The performance of SR models improve as autocorrelation and link density increase, because RDNs use collective inference, which exploits autocorrelation and link density to use predictions of related instances to improve one another. As discussed briefly, RE aggregates those improved predictions and hence improves the overall predictions accuracy. CEC improves node predictions even further, using predictions made by other models simultaneously during collective inference.

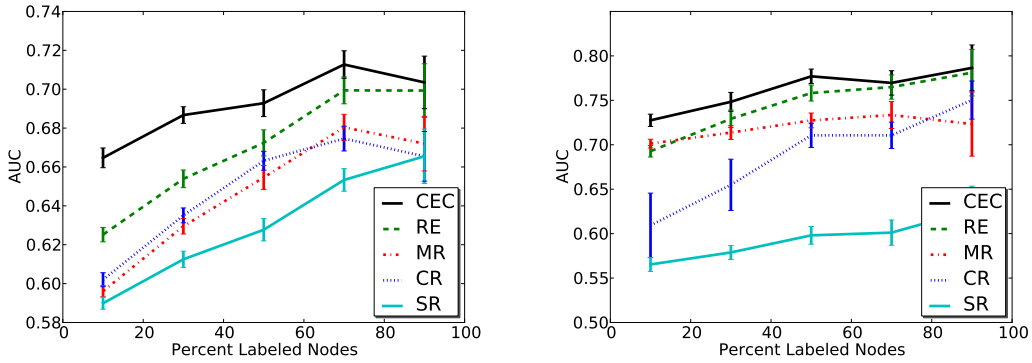

Figure 7: Facebook and IMDB experiments show significant accuracy improvement of proposed CEC ensemble model at various proportions of available true labels in the test graph.

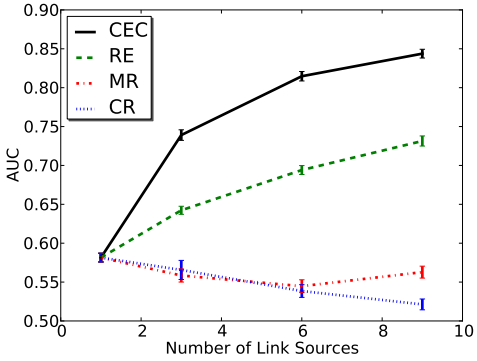

Figure 8: The accuracy of the CEC ensemble model improves as more link graphs are considered by the ensemble.

While MR and CR also improve as autocorrelation and link density increase, they are not able to achieve the same gains as the ensemble methods.

The difference between CEC and RE is due to the intermediate averaging of predictions across the models that is used by CEC. We conjecture that this process reduces the error due to inference variance and that the magnitude of the effect is related to the number of models/sources that are averaged during the inference process. To investigate this, We evaluate a *hybrid* version of RE and CEC—where an ensemble of 10 models is learned on 10 link sources, but vary the number of models that are interleaved during the collective inference process. When 10 models are interleaved, it corresponds to CEC, and when 0 models are interleaved, it corresponds to RE. In between these two extremes, the hybrid model performance shows the effect of propagating prediction information during inference. The blue, dashed line in Figure 9 shows a smooth increment in the overall predictive per-

Table 1: Experimental results for varying autocorrelation and linkage on synthetic data, reporting AUC values.

	Autocorrelation		**Linkage**	
Method	Low	High	Low	High
SR	0.51	0.58	0.58	0.63
CR	0.53	0.57	0.57	0.63
MR	0.52	0.56	0.56	0.68
RE	0.53	0.64	0.64	0.73
CEC	**0.55**	**0.74**	**0.74**	**0.82**

formance as the proportion of propagated predictions during inferences increases, which illustrates the relationship between CEC and RE. The dotted red line shows the average inference variance measured from the same set of experiments, indicating that the accuracy improvement coincides with a similar reduction in inference variance.

6. Related work

Many studies have shown that ensembles of classifiers, including both bagging and boosting methods, usually achieve higher accuracy than individual classifiers Dietterich (2000); Bauer and Kohavi (1999); Breiman (1996b); Kohavi and Kunz (1997); Bauer and Kohavi (1999); Maclin and Opitz (1997). These methods typically assume i.i.d. data and a single information source, but some work has been done to extend ensemble techniques to structured and/or multi-source settings. For example, Blum and Mitchell (1998) propose multi-view learning for i.i.d. data, while Ganchev et al. (2008) propose multi-view learning for structured data. However, none of these methods are suitable for collective classification in a multi-source, relational domain—since they either assume i.i.d. data, multiple structured examples, or a single source. There are many machine learning methods that use multiple information sources to improve classification—by either combining data sources *at the input to learning* (Eldardiry et al., 2014), or by combining predictions *at the output of inference*. Our method is the first to combine information *during* inference instead of *after* inference.

Related to the approach we propose here, are methods that combine source information before learning, including work on integrating multiple networks for label propagation methods (Kato et al., 2008; Tsuda et al., 2005). Since these methods combine multiple information sources and exploit relational structure to propagate inferences via label propagation, they may seem similar to our work. However, in contrast to our method, these approaches combine the source information before inference and focus on label propagation to improve transductive inference within a single network—the methods do not learn complex relational models to generalize to unseen networks, nor do they combine information across networks during inference. There are several other works in this category (Allen and Salzberg, 2005; Lanckriet et al., 2004; Xu et al., 2007).

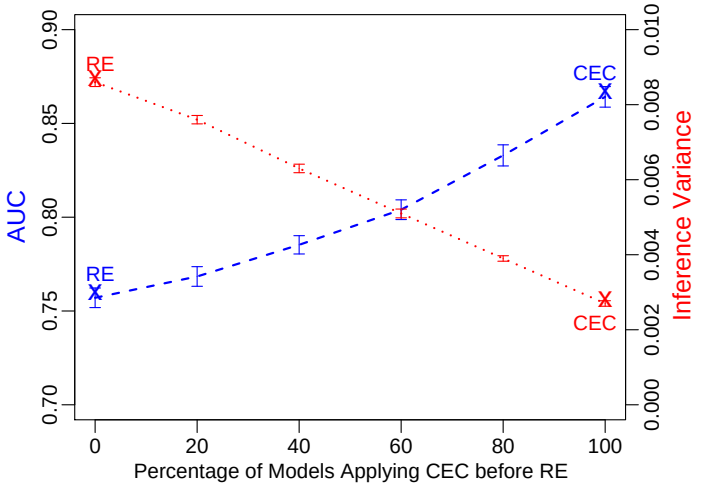

Figure 9: AUC and inference variance for a hybrid model that uses CEC on a limited number of models. As more models are applied using CEC, accuracy increases and inference variance decreases.

In statistical relational learning Getoor and Taskar (2007); De Raedt (2008), there are general learning methods that treat heterogeneous information of multiple object and link types as a single information source and use a single model approach for classification (Getoor and Taskar, 2007; McGovern et al., 2003; Neville et al., 2005; Singh and Gordon, 2008). There has also been some work that augments the observed relational data with additional 'sources' of information to improve performance (Macskassy, 2007; Gallagher et al., 2008). However, once again, the methods combine this information before learning. The MR results presented here are intended to serve as a baseline to compare to this broad class of methods, while controlling for model representation, since the MR models combine all the source information before learning a single model.

Another related line of research contains work that combines prediction information at the output level. Preisach and Schmidt-Thieme (2006) learn a separate classifier from each relational source then combine the classifiers using voting and stacking. This is similar to the proposed CEC method since it uses an ensemble approach to combine multiple link sources. However, their method is intended to reduce learning error, not inference error. The RE results presented here are intended to serve as a baseline comparison to this straightforward relational ensemble method. The work of Gao et al. (2009a,b) presents methods to maximize consensus among the decisions of multiple models for heterogeneous data. These methods are similar to our approach since they combine predictions from multiple models and use label propagation for prediction. However, the label propagation is designed to maximize consensus among the model outputs after inference, rather than during a collective inference process over a relational network. In addition, the method is designed primarily for i.i.d. learners where again, there will be no inference error. There are many other works in this category (Chen et al., 2007; Sehgal et al., 2004; Zhilkin and Somorjai, 1996).

Section 4.1 has focused on developing methods to improve resampling from network data so bagging can reduce more learning variance. Although bagging has been shown to reduce total classification error by reducing the error due to variance (Breiman, 1996a), existing work has focused only on i.i.d. data. Therefore, i.i.d. resampling has been used to generate bootstrap samples from which ensembles are learned. Graph data has an increased variance due to linked object interdependencies. Unfortunately, i.i.d. resampling does not capture all the variance present in the data. Moreover, the i.i.d. models that use exact inference assume the only type of variance is due to learning error. Consequently, conventional bagging approaches only reduce variance due to learning, and if the graph structure is ignored, their ability to reduce learning variance may be limited. Our graph-based resampling method, on the other hand, accounts for the increased variance of network data during ensemble learning. We have evaluated the approach in the context of collective classification and have shown that it significantly improves classification accuracy for network data.

Other related work on ensembles for collective classification contexts has considered alternatives for creating multiple training samples and/or choice of base models. McDowell and Aha (2012) uses a feature subset type approach to learn a set of models to combine in an ensemble, but the models are developed for semi-supervised learning in a partially labeled network (i.e., transductive classification). Llerena et al. (2012) also develop an ensemble approach for transductive settings, where i.i.d. cross-validation is used to learn an ensemble of models. There has also been related work that uses boosting to significantly reduce error due to bias while learning relational dependency networks (Natarajan et al., 2012) and in consensus-based relational models for use with multiple, non-overlapping sources of information (Shi et al., 2012). These methods follow a conventional boosting approach in that they reweight the examples/sources based on prediction errors made during estimation. However, we note that the weights in (Natarajan et al., 2012) are computed individually for each example, rather than jointly across the network of examples. Our experience with relational resampling indicates that this may limit the amount of error reduction achieved by the ensemble due to dependencies in relational data. In contrast, while the weights in (Shi et al., 2012) are computed from a joint estimate of error, the method reweights the information sources (i.e., based on utility) rather than individual examples.

In addition to learning error due to the estimation process, Neville and Jensen (2008) showed that collective classification adds an additional source of error due to the inference process. Straightforward implementations of bagging and boosting for relational models only reduce errors due to learning, because they focus on parameter estimation and ignore errors due to variation in the collective inference process. Other related work by Fast and Jensen (2008) showed that relational stacking (Kou and Cohen, 2007) improves collective classification by reducing inference bias. Their analysis only evaluated model performance in single source relational datasets, but it is interesting to note that stacking reduces inference bias, while our method reduces inference variance. A question for future work is whether the combination of boosted RDNs (Natarajan et al., 2012) and stacking (Kou and Cohen, 2007) would reduce *both* learning and inference bias.

Finally, there is research related to the analytical characterizations we present in this paper. Error analysis for ensemble classifiers and collective classification models, and work on relational methods that reduce bias or variance. For error analysis, earlier work has used conventional bias/variance analysis to evaluate model performance (Domingos, 2000;

Friedman, 1997; Geman et al., 1992; James, 2003). However, the focus has been on single models and on errors in learning. For error analysis of ensembles, Breiman (1996a) has shown theoretically that bagging reduces total classification error by reducing the error due to variance. However, the work is based on the assumption that the data is i.i.d. and therefore the models run exact inference. Consequently, Breiman's work has focused on theoretical analysis for this type of models where the error is only associated with the learning process. Other work has presented an analytical framework to quantify the improvements in classification results due to combining or integrating the outputs of several classifiers (Tumer and Ghosh, 1996). However the analysis by Tumer and Ghosh (1996) is based on analysis of decision boundaries and is applied on linearly combined neural classifiers. For error analysis of collective classification models, Neville and Jensen (2008) have shown that collective classification introduces an additional source of error due to variation in the inference process. More recent work by Xiang and Neville (2011), presented another type of error decomposition for collective classification in single network domains, by studying the propagation error in collective inference with maximum pseudolikelihood estimation.

7. Conclusion

This work introduced a theoretical analysis framework for relational ensemble models. Using the framework, we demonstrated theoretically how ensembles of collective classifiers can improve predictions for graph data. Furthermore, we showed that collective ensemble classification reduces errors due to variance in learning and more interestingly inference. Based on the theoretical analysis, we proposed a relational ensemble framework that combines a relational ensemble learning approach with a relational ensemble inference approach to reduce error due to variance in both learning and inference. Experiments on both synthetic and real-world data demonstrated the effectiveness of the proposed framework.

Acknowledgments

We thank Luc De Raedt for many insightful suggestions and feedback that greatly improved the manuscript. We also thank all the reviewers for many helpful suggestions and feedback. We would like to acknowledge support for this project from the National Science Foundation (NSF grant IIS-9988642) and the Multidisciplinary Research Program of the Department of Defense (MURI N00014-00-1-0637).

Appendix A. Synthetic data

Synthetic datasets are generated with a latent group model (Neville and Jensen, 2005) using the procedure described in Table A. The data graphs are homogeneous (i.e., single object type) data graphs with autocorrelation due to an underlying (hidden) group structure. Each object has a boolean class label C (that is determined by the type of group to which it belongs), and two boolean attributes X_0 and X_1. The class label C has an autocorrelation level of 0.5 and the probabilities of intra- and inter-group linkage are 0.4

and 0.004 respectively. The attribute X_0 is correlated with C, and X_1 has no dependencies (i.e., it is random).

Table 2: Algorithm for generating synthetic dataset with a relational group structure.

For each group g, $1 \leq g \leq (N_G = N_O/G_S)$:
Choose a value for group type t_g from $p(T)$.
For each object i, $1 \leq i \leq N_O$:
Choose a group g_i uniformly in $[1, N_G]$.
Choose a class value C_i from $p(C\|T_{G_i})$.
Choose a value for X_{0i} from $p(X_0\|C)$.
Choose a values for X_{1i} from $p(X_1)$.
For each object j, $1 \leq j \leq N_O$:
For each object k, $j < k \leq N_O$:
Choose whether the two objects are linked from $p(E\|G_j = G_k)$.

Appendix B. Facebook data

The facebook dataset used in this work is a sample of the Purdue University Facebook network (www.facebook.com). Facebook is an online social network site where users maintain a personal profile page and interact with 'friends'. Four sampled networks of users (based on users membership in various University subnetworks) were used in the experiments: [University Alum '07, University '08, University '09, University '10] of sizes: [921, 827, 1268, 1384] users respectively.

We constructed three link graphs. The friendship graph has undirected friendship links. The wall graph has directed links extracted from users' interactions through a public message board on their profile page called wall. The photo graph has directed links extracted from users tagging others in their profile photo page. Each user has a boolean class label which indicates whether their political view is 'Conservative'. In addition, we considered nine node features and two link features. The object features record user profile information: "interested in", "looking for", "relation","gender", "home state", "home", and boolean features "profile public", "friends public" and "christian". Wall links have one link feature that counts the number of wall posts exchanged between any two users, while photo links have one link feature that counts the number of photos shared between any two users.

Appendix C. IMDB data

The IMDb data set is drawn from the Internet Movie Database (www.imdb.com), which contains movie release information. A sample of 1,382 movies released in the United States between 1996 and 2007 was collected. In addition to movies, the data set contains objects representing actors, directors, and studios. In total, this sample contains approximately 42,000 objects and 61,000 links. Five link graphs among movies were constructed. The actors graph links movies that share an actor. Similarly, the studios, producers, directors

and editors graphs were constructed. Seven networks of movies (based on movie release years) were sampled: [2002, 2003, 2004, 2005, 2006, 2007] of sizes: [269, 253, 264, 314, 305, 249] movies respectively. Each movie has a boolean class label which indicates whether the movie is a 'Block buster' (earnings > \$60mil; inflation adjusted). The binary prediction task for movies is to predict blockbuster movies.

References

Jonathan E. Allen and Steven L. Salzberg. Jigsaw: integration of multiple sources of evidence for gene prediction. *Bioinformatics*, 21(18):3596–3603, 2005.

Eric Bauer and Ron Kohavi. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning*, 36(1-2):105–139, 1999.

Avrim Blum and Tom Mitchell. Combining labeled and unlabeled data with co-training. In *Proceedings of the 11th Annual Conference on Computational Learning Theory*, 1998.

Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996a.

Leo Breiman. Bias variance and arcing classifiers. Technical Report 460, Department of Statistics, University of California, Berkley, 1996b.

Xiujuan Chen, Yong Li, Robert Harrison, and Yan-Qing Zhang. Genetic fuzzy classification fusion of multiple svms for biomedical data. *Journal of Intelligent and Fuzzy Systems*, 18 (6):527–541, 2007.

Padraig Cunningham and John Carney. Diversity versus quality in classification ensembles based on feature selection. In *European Conference on Machine Learning*, pages 109–116. Springer, 2000.

Luc De Raedt. *Logical and relational learning*. Springer Science & Business Media, 2008.

Thomas G. Dietterich. Ensemble methods in machine learning. *Multiple Classifier Systems*, 2000.

Pedro Domingos. A unified bias-variance decomposition for zero-one and squared loss. In *Proceedings of the 17th AAAI*, pages 564–569, 2000.

Hoda Eldardiry, Kumar Sricharan, Juan Liu, John Hanley, Bob Price, Oliver Brdiczka, and Eugene Bart. Multi-source fusion for anomaly detection: using across-domain and across-time peer-group consistency checks. *Journal of Wireless Mobile Networks, Ubiquitous Computing, and Dependable Applications*, 5(2):39–58, 2014.

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

Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. In *Proceedings of the 13th International Conference on Machine Learning*, pages 546–551. Morgan Kaufman, 1996.

Jerome H. Friedman. On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery*, 1(1):55–77, 1997.

Brian Gallagher, Hanghang Tong, Tina Eliassi-Rad, and Christos Faloutsos. Using ghost edges for classification in sparsely labeled networks. In *In Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008.

Kuzman Ganchev, Joao Graca, John Blitzer, and Ben Taskar. Multi-view learning over structured and non-identical outputs. In *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, 2008.

Jing Gao, Wei Fan, Yizhou Sun, and Jiawei Han. Heterogeneous source consensus learning via decision propagation and negotiation. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 339–348, 2009a.

Jing Gao, Wei Fan, Yizhou Sun, and Jiawei Han. Graph-based consensus maximization among multiple supervised and unsupervised models. In *Proceedings of 23rd Annual Conference on Neural Information Processing Systems*, 2009b.

Stuart Geman, Elie Bienenstock, and René Doursat. Neural networks and the bias/variance dilemma. *Neural Computation*, 4:1–58, 1992.

Lise Getoor and Ben Taskar. *Introduction to statistical relational learning*. MIT press, 2007.

Andreas Heß and Nicholas Kushmerick. Iterative ensemble classification for relational data: A case study of semantic web services. In *Proceedings of the 15th European Conference on Machine Learning*, 2004.

Gareth M. James. Variance and bias for general loss functions. *Machine Learning*, 51: 115–135, 2003.

Tsuyoshi Kato, Hisashi Kashima, and Masashi Sugiyama. Integration of multiple networks for robust label propagation. In *Proceedings of the SIAM Conference on Data Mining*, 2008.

Ron Kohavi and Clayton Kunz. Option decision trees with majority votes. In *Proceedings of the Conference on Computational Learning Theory*, pages 161–169, 1997.

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

Gert Lanckriet, Tijl De Bie, Nello Cristianini, Michael I. Jordan, and William Stafford Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004.

Roger Leenders. Modeling social influence through network autocorrelation: Constructing the weight matrix. *Social Networks*, 24:21–47, 2002.

Nils Ever Murrugarra Llerena, Lilian Berton, and Alneu de Andrade Lopes. Graph-based cross-validated committees ensembles. In *Fourth International Conference on Computational Aspects of Social Networks (CASoN)*, 2012.

Richard Maclin and David Opitz. An empirical evaluation of bagging and boosting. In *Proceedings of the AAAI*, pages 546–551, 1997.

Sofus Macskassy. Improving learning in networked data by combining explicit and mined links. *Association for the Advancement of Artificial Intelligence*, 2007.

Luke McDowell and David W. Aha. Semi-supervised collective classification via hybrid label regularization. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2012.

Amy McGovern, Lisa Friedland, Michael Hay, Brian Gallagher, Andrew Fast, Jennifer Neville, and David Jensen. Exploiting relational structure to understand publication patterns in high-energy physics. *SIGKDD Explorations*, 5(2):165–172, 2003.

Sriraam Natarajan, Tushar Khot, Kristian Kersting, Bernd Gutmann, and Jude Shavlik. Gradient-based boosting for statistical relational learning: The relational dependency network case. *Machine Learning*, 86(1):25–56, 2012.

Jennifer Neville and David Jensen. Leveraging relational autocorrelation with latent group models. In *Proceedings of the 5th IEEE International Conference on Data Mining*, pages 322–329, 2005.

Jennifer Neville and David Jensen. Relational dependency networks. *Journal of Machine Learning Research*, 8:653–692, 2007.

Jennifer Neville and David Jensen. A bias/variance decomposition for models using collective inference. *Machine Learning Journal*, 2008.

Jennifer Neville, Özgür Şimşek, David Jensen, John Komoroske, Kelly Palmer, and Henry Goldberg. Using relational knowledge discovery to prevent securities fraud. In *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 449–458, 2005.

Christine Preisach and Lars Schmidt-Thieme. Relational ensemble classification. In *The 6th IEEE International Conference on Data Mining*, 2006.

Christine Preisach and Lars Schmidt-Thieme. Ensembles of relational classifiers. *Knowledge and Information Systems*, 2008.

Ross Quinlan. Bagging, boosting and c4.5. In *Proceedings of the 13th AAAI*, pages 725–730. Cambridge, MA: AAAI Press/MIT Press, 1996.

Robert E. Schapire, Yoav Freund, Peter Bartlett, and Wee Sun Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. In *Proceedings of 14th International Conference on Machine Learning*, 1997.

Muhammad Shoaib Sehgal, Iqbal Gondal, and Laurence Dooley. Support vector machine and generalized regression neural network based classification fusion models. In *Proceedings of 4th International Conference on Hybrid Intelligent Systems*, 2004.

Xiaoxiao Shi, Jean-Francois Paiement, David Grangier, and Philip S Yu. Learning from heterogeneous sources via gradient boosting consensus. In *Proceedings of the 12th SIAM International Conference on Data Mining*, 2012.

Ajit P. Singh and Geoffrey J. Gordon. Relational learning via collective matrix factorization. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008.

Koji Tsuda, Hyunjung Shin, and Bernhard Schlkopf. Fast protein classification with multiple networks. *Bioinformatics*, 21, 2005.

Kagan Tumer and Joydeep Ghosh. Analysis of decision boundaries in linearly combined neural classifiers. *Pattern Recognition*, 29:341–348, 1996.

Rongjing Xiang and Jennifer Neville. Understanding propagation error and its effect on collective classification. In *Proceedings of the 11th IEEE International Conference on Data Mining*, 2011.

Rongjing Xiang, Jennifer Neville, and Monica Rogati. Modeling relationship strength in online social networks. In *Proceedings of the 19th International Conference on World Wide Web*, pages 981–990. ACM, 2010.

Zenglin Xu, Irwin King, and Michael R. Lyu. Web page classification with heterogeneous data fusion. In *Proceedings of the 16th International World Wide Web Conference*, 2007.

Peter A. Zhilkin and Ray L. Somorjai. Application of several methods of classification fusion to magnetic resonance spectra. *Connection Science*, 8(3, 4):427–442, 1996.