Computer Science Technical Reports

Department of Computer Science

2012

An ensemble model for collective classification that reduces learning and inference variance

Hoda Eldardiry
Purdue University, hdardiry@cs.purdue.edu

Jennifer Neville
Purdue University, neville@cs.purdue.edu

Report Number:
12-003

An ensemble model for collective classification that reduces learning and inference variance

Hoda Eldardiry
Purdue University
West Lafayette, IN, USA
hdardiry@cs.purdue.edu

Jennifer Neville
Purdue University
West Lafayette, IN, USA
neville@cs.purdue.edu

ABSTRACT

Ensemble learning can improve classification of relational data. Previous attempts to do so include methods that have focused primarily on reducing learning or inference variance, but not both at the same time. We present an ensemble model that reduces error due to variance in both learning and collective inference. Our model uniquely combines two strategies tailored specifically for relational data and relational models to achieve a larger reduction in variance than using either method alone, which results in significant accuracy gains. In addition, we present the first theoretical analysis for ensembles of collective classifiers in relational domains, to show the reasons for the superior performance of our proposed method. We also use synthetic and real world data to demonstrate the improvement empirically.

1. INTRODUCTION

Ensemble methods have been widely studied as a means of reducing classification error by combining multiple models for prediction. However, much of this work has focused on i.i.d. domains (where objects are independent and models use exact inference techniques). While there has been some recent investigation of ensembles for relational domains [4, 11, 19], these previous works have a number of limitations in that: (1) they focus on the reduction of only one type of error (due to either learning or inference), (2) they restrict their attention to datasets with multiple relations, and (3) there is no theoretical analysis to show the mechanism by which ensembles reduce model error in relational domains. In this work, we go beyond previous work and develop an ensemble that can reduce both types of errors (learning and inference), and is also applicable for networks with only a single relation (e.g., email networks, citation networks). Moreover, we formulate a theoretical framework to compare the errors made by different relational ensembles and show the reason for the superior performance of our proposed method.

Traditional design choices for i.i.d. ensembles included methods to ensure variety among the learned models and methods to aggregate the output of the models. For example, bagging approaches (e.g., [2]) aggregate predictions from multiple models and boosting approaches (e.g., [1, 20, 24]), construct the models in a coupled fashion so that their weighted vote gives a good fit to the data. Previous work on relational ensembles [4, 19] focused on an opportunity offered by relational networks with multiple edge types in order to learn the component models of a relational ensemble in a new way. Specifically, the multiple link types in the network are used to subset the data (instead of a conventional feature subset approach which would sample from the node features). Also, [4] proposed a new approach to aggregating predictions, which aggregates across the models during collective inference in addition to the conventional aggregation of the final model output. This method utilized another unique opportunity offered by relational domains, which stems from the use of collective classification [10, 14, 18, 21].

In this work, we formulate a novel ensemble approach for relational domains through a unique combination of design choices. First, we choose to use *collective classifiers* as base models for the ensemble, since they can significantly improve classification accuracy. In addition, link structures and dependencies in relational data require new approaches for data sampling to learn the base models. Instead of the traditional way of sampling objects independently to create multiple *pseudosamples* for learning, our approach uses a *relational resampling* method that considers *subgraphs* of connected objects to capture the dependencies in the data. This type of resampling more accurately captures the increased variance in network data [6]. Learning the base models from pseudosamples constructed in this way will allow the ensemble to capture and reduce more learning variance in predictions. In addition, the collective classifiers will be able to utilize the link dependencies in the data, which are only preserved by this type of resampling approach. For prediction, our approach uses an *interleaved inference* process that aggregates predictions across the models during inference. This generalizes the previous interleaved method [4] to single-network domains. We empirically compare our proposed ensemble approach to several baselines using synthetic and real-world classification tasks and demonstrate its superior performance. In addition to empirical validation, we also analyze the relational ensembles theoretically to show the mechanism by which they reduce classification error.

In i.i.d. domains, ensemble learning methods have been shown to reduce classification error by reducing variance (e.g., bagging [2]) or reducing bias (e.g., boosting [24]). However, since the analysis focuses on i.i.d. data, the models are assumed to use exact inference techniques that have no associated error—thus the only error is attributed to the learning process. On the other hand, collective inference models applied to relational data have been shown to have additional sources of error due to the inference process [16]. Furthermore, the correlation in relational data has been shown to increase variance [13]. We show that our ensemble design choices combine to reduce errors due to *both* the learning and inference process in relational data. Specifically, our relational resampling approach aims to capture the increased variance in relational data, allowing the ensembles to reduce more of the variance due to learning. This is combined with the interleaved inference, which allows the ensembles to reduce more of the variance due to inference. In contrast, the previous work has developed methods focused on reducing errors due to variance in learning [19], or due to variance in inference [4], but not both simultaneously.

We use a bias/variance decomposition similar to that of [16] for our analysis, but extend it for the ensemble setting—to consider not just a single collective inference model, but an ensemble of collective inference models. Specifically, we reason about two ensemble models: (1) a simple relational ensemble model that runs the component classifiers independently for inference and aggregates the final predictions, and (2) an across-model approach, which runs the component models simultaneously for collective inference and aggregates intermediate predictions across the models during inference. In the remainder of this paper, the first model is referred to as the relational ensemble model, while the second model is referred to as the interleaved model. The goal of our 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 an interleaved 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 analytical investigation of error for relational ensembles.

The main contributions of this work are:

- A novel ensemble method for relational domains that can reduce both learning and inference errors, which is applicable for networks with only a single relation.
- Empirical evaluation on real and synthetic data, which shows significant performance gains for our proposed method compared to alternative ensembles.
- Formalization of an error analysis framework for relational ensemble models.
- Theoretical analysis to show the error reduction offered by alternative relational ensembles, which demonstrates the mechanism by which our proposed model improves model accuracy.

2. PROPOSED ENSEMBLE MODEL

We propose an ensemble model that uses relational subgraph resampling (RSR) for generating the bootstrap pseudosamples to learn the ensembles from, and collective ensemble classification (CEC) for inference. RSR was originally proposed for accurate estimation of variance for network data [6]. We utilize RSR to accurately capture the learning variance during ensemble construction. This enables our ensemble method to reduce more variance in learning than traditional independent resampling approaches. In addition, using CEC in our method facilitates the reduction of inference error. In contrast, the majority of existing ensembles focus on reducing learning error alone. Using these two approaches allows a combined reduction of learning and inference variance, and extends the utility of CEC to single network settings. Note that, CEC was developed for domains with multiple types of relations (e.g., a network with email, phone, and SMS links)—and the method requires multiple link types to learn a model from each typed subnetwork. However, using RSR for learning enables a generalization of the method to domains that do not necessarily have multiple link graphs. The psuedosamples we use for learning are networks sampled with replacement from a *single* training graph (regardless of the link types in the graph).

Given a training dataset, our algorithm uses RSR to generate m bootstrap pseudosamples to learn an ensemble of m models. The models are applied for collective inference on a single test set using CEC, which iteratively interleaves the inferences across the m models. After inference is done, the predictions output by each base model are aggregated for each node independently as in traditional ensembles.

2.1 Problem Formulation

The general relational learning and collective classification problem can be described as follows. Given a fully-labeled training set composed of a graph $G_{tr} = (V_{tr}, E_{tr})$ with nodes V_{tr} and edges E_{tr}, observed features X_{tr}, and observed class labels Y_{tr}, 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} is learned. Given a partially-labelled test set composed of a graph $G_{te} = (V_{te}, E_{te})$ with nodes V_{te} and edges E_{te}, 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}. Note that, in this work, we assume the G_{tr} used for learning is different from the G_{te} used for collective inference.

2.2 Ensemble learning

Given the setting described above, the ensemble learning approach using bootstrap sampling is outlined in Algorithm 1, showing how an ensemble of size m models is constructed. A pseudosample $G_{ps} = (V_{ps}, E_{ps})$ is generated by resampling from G_{tr} (line 3) and a model F is learned from G_{ps} (line 4). F is a joint probability distribution over the labels of V_{ps}, conditioned on the observed attributes and graph structure in G_{ps}. The ensemble set of m learned models is returned (line 6). Note that the two main components needed for an implementation of the algorithm are: a resampling algorithm (step 3) and a learning algorithm (step 4). We describe each below.

2.2.1 *Resampling*

RSR is an approach for resampling relational data to accurately capture the increased variance due to linkage and autocorrelation. RSR samples *subgraphs* with replacement

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

```
$Ensemble \leftarrow \emptyset$
for $j := 1$ to $m$ do
    $G_{ps_j} = Resample(G_{tr})$
    $F_j = LearnModel(G_{ps_j})$
    $Ensemble = Ensemble \cup \{F_j\}$
return $Ensemble$
```

Algorithm 2 Relational Subgraph Resampling (RSR)

```
RSR($G = (V, E), b$)
$V_{PS} \leftarrow \emptyset$;  $E_{PS} \leftarrow \emptyset$
for $s := 1$ to $\lceil \frac{|V|}{b} \rceil$ do
    $V_S \leftarrow \emptyset$;  $E_S \leftarrow \emptyset$;  $Q \leftarrow \emptyset$
    $v_s$ = randomly select node from $V$
    $V_S \leftarrow V_S \cup v_s$
    push neighbors of $v_s$ onto $Q$
    while $(|V_S| < b) \wedge (|Q| > 0)$ do
        $v_s$ = pop $Q$
        $V_S \leftarrow V_S \cup v_s$
        push neighbors of $v_s$ onto $Q$
    $E_S = \{e_{ij} \in E \;\; s.t. \;\; v_i, v_j \in V_S\}$;  $V_{PS} \leftarrow V_{PS} + V_S$;  $E_{PS} \leftarrow E_{PS} + E_S$
return $G_{PS} = (V_{PS}, E_{PS})$
```

Algorithm 3 Collective Ensemble Classification (CEC)

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

 1: for all i in 1 to $m$ do
 2:     $\hat{Y}^i = \tilde{Y}; \mathbf{Y^i_T} = \emptyset$
 3:     for all $v_j \in V \; s.t. \; y_j \notin \tilde{Y}$ do
 4:         Randomly initialize $\hat{y}^i_j$ ; $\hat{Y}^i = \hat{Y}^i \cup \hat{y}^i_j$
 5: repeat
 6:     for all $i = 1$ to $m$ do
 7:         for all $v_j \in V \; s.t. \; y_j \notin \tilde{Y}$ do
 8:             $\hat{y}^{i_{new}}_j = F^i : P^i(Y_j|\mathbf{X}_{i.j}, \mathbf{X_{i.R}}, \mathbf{\hat{Y}^i_R})$ $where$ $\mathbf{R} = \{v_k : e_{jk} \in E_i\}$
 9:             $\hat{y}^{i_{agg}}_j = \frac{1}{m} \sum_{j=1}^{m} \hat{y}^{i_{new}}_j$
10:             $\hat{Y}^i = \hat{Y}^i - \{\hat{y}^i_j\} + \{\hat{y}^{i_{agg}}_j\}$ ; $\mathbf{Y^i_T} = \mathbf{Y^i_T} \cup \hat{y}^{i_{agg}}_j$
11: until $terminating\_condition$
12: for all $i = 1$ to $m$ do
13:     Compute $\mathbf{P^i} = \{P^i_j : y_j \notin \tilde{Y}\}$ using $\mathbf{Y^i_T}$
14: $P = \emptyset$
15: for all $v_j \in V$ do
16:     $p_j = \frac{1}{m} \sum_{i=1}^{m} p^i_j$ ; $P = P \cup \{p_j\}$
17: return $P$
```

instead of the typical independent sampling technique that samples *instances* (i.e., nodes) with replacement. When instances are resampled independently at random the resulting pseudosamples *underestimate* the amount of variance if the data exhibits network autocorrelation.

The RSR procedure is outlined in Algorithm 2. Given a sample 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 from G. Each of N_S subgraphs is sampled using a breadth-first search from a randomly selected seed nodes. As a node v_s is added to the sampled subgraph node set V_s, $v_s's$ neighbors are added to a list Q, from which the next node v_s is taken. This continues until the subgraph size b is reached.

Note that the sampling is with replacement from the graph, so a node may appear 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.

The key idea behind sampling subgraphs is that when autocorrelation is high (i.e., neighbors 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. We compare to a method that uses independent sampling as a baseline. It is described in more detail the experimental section.

2.2.2 *Learning*

We learn relational dependency network (RDN) [18] models as the component collective classification models. Since RDNs are selective models based on decision trees, they exhibit the instability that typically works well in bagged ensembles. 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, which is easy to do with Gibbs sampling (i.e., has been shown to converge within 500-2000 Gibbs iterations [18]).

2.3 Ensemble inference

For inference, we use collective ensemble classification (CEC). However, instead of learning the ensemble from multiple link graphs as previously proposed [4], we learn the ensemble from bootstrap pseudosamples constructed using RSR as described above. This has an additional advantage of being applicable in single-graph network settings. The CEC procedure is included in Algorithm 3 for completeness. CEC uses *across-models* collective classification for inference, which propagates predictions across the component models during collective inference.

Given a test network G with partially labeled nodes V, and m base models $F_1, F_2, \ldots, F_k$ learned as described in section 2.2, 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). 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 collective inference iteration (lines 12-13). Finally, model outputs are averaged to produce the final predictions (lines 15-16).

Note that the manner in which 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, we use RDNs, but 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.

2.4 Experimental Evaluation

We refer to our proposed ensemble as RSR-CEC. We evaluate the ensemble method on both synthetic and real world datasets, and the results show that combining RSR with CEC significantly outperforms using either approach alone.

2.4.1 Baseline approaches

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

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 the ensembles we discuss below, including the proposed model, generate the bootstrap pseudosamples from this original training sample, and use the same collective classification algorithm as the base component model.

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.

2.4.2 Datasets

We evaluate the methods on synthetic and real world network data. Synthetic datasets are generated with a latent group model [17]. 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 datasets, each consisting of 500 objects.

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): [Purdue 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.

2.4.3 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. The area under the ROC (AUC) 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.

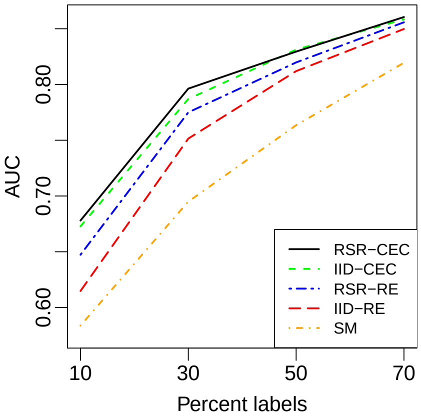

Figure 1: Synthetic experiments show significant accuracy improvement of proposed RSR-CEC ensemble model at various proportions of available true labels in the test graph.

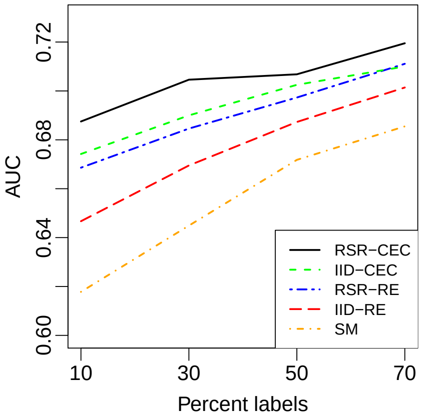

Figure 2: Facebook experiments show significant accuracy improvement of proposed RSR-CEC ensemble model at various proportions of available true labels in the test graph.

$l = \{10\%, 30\%, 50\%, 70\%\}$ denotes the x-axis in the figures, while the y-axis plots the AUC values.

2.4.4 *Results*

Figures 1 and 2 show the results of the synthetic and Facebook experiments, respectively. The main finding is that our proposed RSR-CEC approach has significantly higher classification accuracy than all the baseline comparison methods—at all percent labelings, across both the synthetic and Facebook experiments. We measured significance 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.

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).

3. THEORETICAL ANALYSIS

In this section we use bias/variance analysis to explore the differences between single collective models and the various relational ensembles. Specifically, we focus on squared loss as a measure of classification performance and show the error reduction offered by the different types of ensembles. The analytical results confirm our empirical findings, and shows how the simple relational ensemble improves performance over the single collective classifier, as well as how the CEC improves performance over the simple relational ensemble. To this best of our knowledge, this is the first analytical exploration of classification error for relational ensembles.

3.1 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. Each sample $D := [G=(V,E), X_V, Y_V]$ is drawn from $\mathcal{D}$, where V is the set of instances in D, E is the set of links, and $|V| = g$.

Let $f := P(\mathbf{Y}_g|\mathbf{X}_g, G)$ represent a model of the joint distribution over class labels Y of instances in a graph G, given attributes of the instances $\mathbf{X}$. Let $D_L \in \mathcal{D}$ be a training graph. Let $D_I \in \mathcal{D}$ be a partially labeled test graph where $\mathcal{T} \in V_I$ is the set of labeled instances in G_I. Let $\mathbf{Y}_\mathcal{T}$ be the set of known labels available to the inference process. For this analysis, we assume that D_L and D_I are drawn independently from $\mathcal{D}$ and that $D_I \neq D_L$.

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 instance $i \in V_{I/\mathcal{T}}$:

$$y^i_f := f(i, D_I, \mathcal{T}) = P(Y^i{=}t^i | \mathbf{Y}_\mathcal{T}, \mathbf{X}, G_I) \quad (1)$$

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 an *exact inference* model, which does not use collective inference and simply makes a prediction for i conditioned on the set of Bayes-optimal values for all instances except i. Below, we use $\check{\mathbf{Y}}_{V_{I/i}}$ to refer to the Bayes-optimal prediction for all instances in the dataset D_I except i.

3.1.1 Model definitions

We consider four models in our analysis: a single collective inference model (f_s), a simple relational ensemble model (f_e), our interleaved collective inference model (f_c), and the "true" model (f_*). 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 learned over samples drawn from the population $\mathcal{D}$:

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

Single model: Let f_s be a single collective inference model learned from a sample D_L, which estimates P_s. Note that each f_s learned from a different sample D_L gives a different estimate of the true joint distribution P_*. The model f_s is then used to make predictions for each unlabeled instance i in a partially labeled dataset $< D_I, \mathcal{T} >$:

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

Simple relational ensemble model (RE): Let f_e be a simple relational ensemble model that aggregates predictions from m collective inference base models that each run n Gibbs iterations independently. A prediction $y^i_{f_e}$ for an instance i is calculated by averaging the final predictions for i from all m models. Each base model makes its predictions as described for the single model above.

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

Interleaved ensemble model (CEC): Let f_c be an interleaved model that aggregates predictions from m collective inference base models at each Gibbs iteration $j \in \{1..n\}$. At each iteration j, predictions made by all the base models are aggregated and used to make a prediction for each model $k \in \{1..m\}$. These predictions are for $V_{I/\mathcal{T}}$. For the instances in $\mathcal{T}$, we use the true labels. The final prediction for an instance i is estimated from the average of the component models' predictions at the last inference iteration n. This defines the interleaved model $f_c = \check{f}_{k,n}$.

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

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 [16]. Here we consider squared loss as a measure of classification performance. The loss L for model f on instance i is defined as the expected squared loss for prediction y^i_f given i's true label of t^i:

$$\text{Loss: } L^i_f = E\Big[(t^i - y^i_f)^2\Big] \quad (6)$$

Here E refers to the total expectation, which is taken over training sets ($D \in \mathcal{D}$) used to learn the model f and subsets of true labels $\mathcal{T}$ available for inference. For ease of reading, when it is clear from context, we drop the superscript i and the subscript f.

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 [16]. 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: $L = V + B + N$. 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: Here variance, $V = E\left[(E[y] - y)^2\right]$, is the average loss incurred by all predictions y, relative to the mean prediction $E[y]$.

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

Noise: Here noise, $N = E\big[(t - E[t])^2\big]$, 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 an instance i is estimated using *exact inference* based on the set of Bayes-optimal predictions for the rest of the graph, $\tilde{\mathbf{Y}}_{V_{I/i}}$:

$$\begin{aligned} E_L[y^i_{f_s}|D_I] &= \sum_{D_L \in \mathcal{D}} P_s(Y^i = t_i|\tilde{\mathbf{Y}}_{V_{I/i}}, \mathbf{X}, G_I) * p(D_L) \\ &= P_*(Y^i = t_i|\tilde{\mathbf{Y}}_{V_{I/i}}, \mathbf{X}, G_I) \end{aligned} \tag{7}$$

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

$$\begin{aligned} E_I[y^i_{f_s}|D_I, f_s^{D_L}] &= \sum_{\mathcal{T}} P_s(Y^i = t_i|\mathbf{Y}_{\mathcal{T}}, \mathbf{X}, G_I) * p(\mathbf{Y}_{\mathcal{T}}) \\ &= P_s(Y^i = t_i|\mathbf{X}, G_I) \end{aligned} \tag{8}$$

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

$$\begin{aligned} E_T[y^i_{f_s}|D_I] &= E_{LI}[y^i_{f_s}|D_I] \\ &= \sum_{\mathcal{T}} p(\mathbf{Y}_{\mathcal{T}}) \sum_{D_L \in \mathcal{D}} P_s(Y^i = t_i|\mathbf{Y}_{\mathcal{T}}, \mathbf{X}, G_I) * p(D_L) \\ &= P_*(Y^i = t_i|\mathbf{X}, G_I) \end{aligned} \tag{9}$$

3.2 Analysis

Given the framework described above, we compare the performance of the ensemble models to the single model and show how the ensembles reduce total loss. Specifically, we decompose the error of the single collective inference model f_s, the simple relational ensemble model f_e, and our proposed 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 simple 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: We note that the noise component of error is dependent upon the data set, and is independent of the classification algorithm. Therefore:

$$N_s = N_e = N_c \tag{10}$$

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

$$E_T[.] = E_I[E_L[.]] \tag{11}$$

Predictions from simple 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., $E_L[y^i_s]$). But 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).

$$\lim_{m \to \infty} y_e = E_L[y_s] = P_*(Y^i = t^i|\tilde{\mathbf{Y}}_{V_{I/i}}, \mathbf{X}, G_I) \tag{12}$$

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 $\hat{Y}_{D_{I/\{\mathcal{T}+i\}},j}$ at each inference iteration j, simulates draws from alternative labelings $\mathcal{T}$ over the course of inference.

$$\lim_{m,n \to \infty} y_c = E_T[y_s] = P_*(Y^i = t^i|\mathbf{X}, G_I) \tag{13}$$

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 simple relational ensemble.

Theorem 1: Let f_s be a single collective inference model with variance V_s, f_e be a simple relational ensemble with variance V_e, and f_c be an interleaved ensemble model with variance V_c. Then $V_s \geq V_e \geq V_c$.

1.1 $V_s - V_e \geq 0$
1.2 $V_e - V_c \geq 0$

Proof of Theorem 1.1

$$\begin{aligned}
&V_s - V_e \\
&= E_T\left[(E_T[y_s] - y_s)^2\right] - E_T\left[(E_T[y_e] - y_e)^2\right] \\
&= E_T\big[E_T[y_s]^2 - 2y_s E_T[y_s] + y_s^2\big] - E_T\big[E_T[y_e]^2 - 2y_e E_T[y_e] + y_e^2\big] \\
&= E_T[y_s]^2 - 2E_T[y_s]^2 + E_T[y_s^2] - E_T[y_e]^2 + 2E_T[y_e]^2 - E_T[y_e^2] \\
&= -E_T[y_s]^2 + E_T[y_s^2] + E_T[y_e]^2 - E_T[y_e^2] \\
&= -E_T[y_s]^2 + E_T[y_s^2] + E_T\left[E_L[y_s]\right]^2 - E_T\left[E_L[y_s]^2\right] \quad \textbf{(by 12)} \\
&= -E_I[E_L[y_s]]^2 + E_T[y_s^2] + E_I[E_L[y_s]]^2 - E_T\left[E_L[y_s]^2\right] \quad \textbf{(by 11)} \\
&= E_T[y_s^2] - E_T\left[E_L[y_s]^2\right] \\
&= E_I\left[E_L[y_s^2]\right] - E_I\left[E_L[y_s]^2\right] \quad \textbf{(by 11)}
\end{aligned}$$

$$
\begin{aligned}
&= E_I\left[E_L[y_s^2] - E_L[y_s]^2\right] \\
&\geq 0 \qquad (E_L[y_s^2] - E_L[y_s]^2 \geq 0 \textbf{ by Jensen's Inequality})
\end{aligned}
$$

□

Proof of Theorem 1.2

$$
\begin{aligned}
&V_e - V_c \\
&= E_T\left[(E_T[y_e] - y_e)^2\right] - E_T\left[(E_T[y_c] - y_c)^2\right] \\
&= E_T\left[E_T[y_e]^2 - 2y_e E_T[y_e] + y_e^2\right] - E_T\left[E_T[y_c]^2 - 2y_c E_T[y_c] + y_c^2\right] \\
&= E_T[y_e]^2 - 2E_T[y_e]^2 + E_T[y_e^2] - E_T[y_c]^2 + 2E_T[y_c]^2 - E_T[y_c^2] \\
&= -E_T[y_e]^2 + E_T[y_e^2] + E_T[y_c]^2 - E_T[y_c^2] \\
&= -E_T\left[E_L[y_s]\right]^2 + E_T\left[E_L[y_s]^2\right] + E_T[y_c]^2 - E_T[y_c^2] && \textbf{(by 12)} \\
&= -E_T\left[E_L[y_s]\right]^2 + E_T\left[E_L[y_s]^2\right] + E_T\left[E_T[y_s]\right]^2 \\
&\qquad - E_T\left[E_T[y_s]^2\right] && \textbf{(by 13)} \\
&= -E_T\left[E_L[y_s]\right]^2 + E_T\left[E_L[y_s]^2\right] + E_T[y_s]^2 - E_T[y_s]^2 \\
&= -E_I\left[E_L[y_s]\right]^2 + E_I\left[E_L[y_s]^2\right] && \textbf{(by 11)} \\
&= E_I\left[E_L[y_s]^2\right] - E_I\left[E_L[y_s]\right]^2 \\
&\geq 0 && \textbf{(by Jensen's Inequality)}
\end{aligned}
$$

□

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. Simple relational ensembles f_e average models predictions from different learned models and reduce the variance due to learning. Thus, $V_s \geq V_e$.

Similar to simple 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 $B_T = (E_T[t] - E_T[y])^2$. Again 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 variance B_s, f_e be a simple relational ensemble with variance B_e, and f_c be an interleaved ensemble model with variance B_c. Then $B_s = B_e = B_c$

$$
\begin{aligned}
&2.1 \quad B_s - B_e = 0 \\
&2.2 \quad B_e - B_c = 0
\end{aligned}
$$

Proof of Theorem 2.1

$$
\begin{aligned}
&B_s - B_e \\
&= (E_T[t] - E_T[y_s])^2 - (E_T[t] - E_T[y_e])^2 \\
&= (E_T[t] - E_T[y_s])^2 - (E_T[t] - E_T\left[E_L[y_s]\right])^2 && \textbf{(by 12)} \\
&= (E_T[t] - E_T[y_s])^2 - (E_T[t] - E_T[y_s])^2 \\
&= 0
\end{aligned}
$$

□

Proof of Theorem 2.2

$$
\begin{aligned}
&B_e - B_c \\
&= (E_T[t] - E_T[y_s])^2 - (E_T[t] - E_T[y_c])^2 \\
&= (E_T[t] - E_T\left[E_L[y_s]\right])^2 - (E_T[t] - E_T\left[E_T[y_s]\right])^2 && \textbf{(by 12, 13)} \\
&= (E_T[t] - E_T[y_s])^2 - (E_T[t] - E_T[y_s])^2 \\
&= 0
\end{aligned}
$$

□

3.2.3 Loss reduction

Now, 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 additively into variance, bias and noise components: $L = V + B + N$. We now show that a simple relational ensemble reduces the loss of a single model, and an interleaved ensemble reduces the loss of a simple relational ensemble.

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

$$
\begin{aligned}
&1.1 \quad L_s - L_e \geq 0 \\
&1.2 \quad L_e - L_c \geq 0
\end{aligned}
$$

Proof of Corollary 1.1

$$
\begin{aligned}
&L_s - L_e \\
&= (V_s + B_s + N_s) - (V_e + B_e + N_e) \\
&= (V_s + B_s + N_s) - (V_e + B_s + N_s) && \textbf{(by 10, Thm 2)} \\
&= V_s - V_e \\
&\geq 0 && \textbf{(by Thm 1.1)}
\end{aligned}
$$

□

Proof of Corollary 1.2

$$
\begin{aligned}
&L_e - L_c \\
&= (V_e + B_e + N_e) - (V_c + B_c + N_c) \\
&= (V_e + B_s + N_s) - (V_c + B_s + N_s) && \textbf{(by 10, Thm 2)} \\
&= V_e - V_c \\
&\geq 0 && \textbf{(by Thm 1.2)}
\end{aligned}
$$

□

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 simple relational ensemble model will reduce the error 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 Resampling

The error analysis presented above applies to ensembles learned from bootstrap pseudosamples generated using *either* IID resampling or RSR. In both sampling methods, when the number of pseudosamples m approaches ∞, the bootstrap samples approximate the true population distribution $\mathcal{D}$. This indicates that for the ensemble model f_e, assumption 12 holds regardless of the resampling approach. In other words, 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 IID and RSR sampling:

$$\lim_{m\to\infty} y_e^{RSR} = \lim_{m\to\infty} y_e^{IID} = E_L[y_s] \tag{14}$$

However, y_e^{RSR} converges faster than y_e^{IID}. Thus, given a finite ensemble size m, because RSR can more accurately capture the increased variance in network data, predictions made by models learned from RSR pseudosamples will capture and reduce more learning variance. The same argument applies to f_c. Thus assumption 13 holds regardless of the resampling approach, but in finite ensemble sizes, RSR pseudosamples will capture and reduce more variance.

4. RELATED WORK

There are two main lines of research related to the analysis we present here. 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 [3, 8, 9, 12]. However, the focus has been on single models and on errors in learning.

For error analysis of ensembles, Breiman [2] 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 [22]. Their work 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 [16] have shown that collective classification introduces an additional source of error due to variation in the inference process. While other work has presented another type of error decomposition for collective classification [23], by studying the propagation error in collective inference with maximum pseudolikelihood estimation.

Related works [4, 5, 6] have extended ensembles to improve classification accuracy for relational domains. This includes a method for construcring ensembles while accounting for the increased variance of network data [6], a method for ensemble classification on multi-source networks [5], and an ensemble method for reducing variance in the inference process for collective classification [4]. Moreover, recent work [7] recently showed that stacking [15] improves collective classification by reducing inference bias. This work compares to our model as it evaluated model performance in single source relational datasets. However, it is interesting to note that stacking reduces inference bias, while our method reduces inference variance.

5. CONCLUSION

We proposed an ensemble model that significantly improves classification accuracy of network data by reducing errors due to variance in both learning and inference. We evaluated it using both synthetic and real-world classification tasks. We presented theoretical analysis that confirms our empirical findings. We showed that an interleaved ensemble model reduces total loss over a simple relational ensemble model which reduces total loss over a single model (corollary 1). We showed that this is achieved by the reduction of variance (theorem 1), not bias (theorem 2).

6. REFERENCES

[1] R. S. Y. F. P. Bartlett and W. Lee. Boosting the margin: A new explanation for the effectiveness of voting methods. In *ICML'97*.

[2] L. Breiman. Bagging predictors. *MLJ'96*.

[3] P. Domingos. A unified bias-variance decomposition for zero-one and squared loss. In *AAAI'00*.

[4] H. Eldardiry and J. Neville. Across-model collective ensemble classification. In *AAAI'11*.

[5] H. Eldardiry and J. Neville. Multi-network fusion for collective inference. In *MLG'10*.

[6] H. Eldardiry and J. Neville. A resampling technique for relational data graphs. In *SNA-SIGKDD'08*.

[7] A. Fast and D. Jensen. Why stacked models perform effective collective classification. In *ICDM'08*.

[8] J. Friedman. On bias, variance, 0/1-loss, and the curse-of-dimensionality. *DMKD'97*.

[9] S. Geman, E. Bienenstock, and R. Doursat. Neural networks and the bias/variance dilemma. *NC'92*.

[10] L. Getoor, N. Friedman, D. Koller, and A. Pfeffer. Learning probabilistic relational models. In *RDM'01*.

[11] A. HeB and N.Kushmerick. Iterative ensemble classification for relational data: a case study of semantic web services. In *ECML'04*.

[12] G. James. Variance and bias for general loss functions. *MLJ'03*.

[13] D. Jensen and J. Neville. Linkage and autocorrelation cause feature selection bias in relational learning. In *ICML'02*.

[14] D. Jensen, J. Neville, and B. Gallagher. Why collective inference improves relational classification. In *SIGKDD'04*.

[15] Z. Kou and W. W. Cohen. Stacked graphical models for effecient inference for markov random fields. In *SDM'07*.

[16] J. Neville and D. Jensen. A bias/variance decomposition for models using collective inference. *MLJ'08*.

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

[18] J. Neville and D. Jensen. Relational dependency networks. *JMLR'07*.

[19] C. Preisach and L. Schmidt-Thieme. Ensembles of relational classifiers. *KIS'08*.

[20] J. Quinlan. Bagging, boosting and c4.5. In *AAAI'96*.

[21] B. Taskar, P. Abbeel, and D. Koller. Discriminative probabilistic models for relational data. In *UAI'02*.

[22] K. Tumer and J. Ghosh. Analysis of decision boundaries in linearly combined neural classifiers. *Pattern Recognition'96*.

[23] R. Xiang and J. Neville. Understanding propagation error and its effect on collective classification. In *ICDM'11*.

[24] Y.Freund and R.E.Schapire. Experiments with a new boosting algorithm. In *ICML'96*.