Graduate School ETD Form 9
(Revised 12/07)

PURDUE UNIVERSITY
GRADUATE SCHOOL
Thesis/Dissertation Acceptance

This is to certify that the thesis/dissertation prepared

By Hoda Eldardiry

Entitled Ensemble Classification Techniques for Relational Domains

For the degree of Doctor of Philosophy ▾

Is approved by the final examining committee:

Jennifer Neville
Chair

Christopher Clifton

Luo Si

Ahmed Elmagarmid

To the best of my knowledge and as understood by the student in the *Research Integrity and Copyright Disclaimer (Graduate School Form 20)*, this thesis/dissertation adheres to the provisions of Purdue University's "Policy on Integrity in Research" and the use of copyrighted material.

Approved by Major Professor(s): Jennifer Neville

Approved by: Sunil Prabhakar / William Gorman 02/16/2012
Head of the Graduate Program Date

PURDUE UNIVERSITY
GRADUATE SCHOOL

Research Integrity and Copyright Disclaimer

Title of Thesis/Dissertation:
Ensemble Classification Techniques for Relational Domains

For the degree of Doctor of Philosophy ▾

I certify that in the preparation of this thesis, I have observed the provisions of *Purdue University Executive Memorandum No. C-22,* September 6, 1991, *Policy on Integrity in Research.**

Further, I certify that this work is free of plagiarism and all materials appearing in this thesis/dissertation have been properly quoted and attributed.

I certify that all copyrighted material incorporated into this thesis/dissertation is in compliance with the United States' copyright law and that I have received written permission from the copyright owners for my use of their work, which is beyond the scope of the law. I agree to indemnify and save harmless Purdue University from any and all claims that may be asserted or that may arise from any copyright violation.

Hoda Eldardiry
Printed Name and Signature of Candidate

01/04/2012
Date (month/day/year)

*Located at http://www.purdue.edu/policies/pages/teach_res_outreach/c_22.html

ENSEMBLE CLASSIFICATION TECHNIQUES FOR RELATIONAL DOMAINS

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Hoda M. Eldardiry

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2012

Purdue University

West Lafayette, Indiana

To my beloved family.

TABLE OF CONTENTS

Page

LIST OF TABLES vii

LIST OF FIGURES viii

ABSTRACT x

1 INTRODUCTION 1

1.1 Ensemble design 2

1.1.1 Input data treatment 2

1.1.2 Choice of base model 3

1.1.3 Output aggregation 3

1.1.4 Model interleaving 4

1.2 Ensemble framework and analysis 4

2 BACKGROUND 7

2.1 Ensemble learning 7

2.1.1 Independent ensemble construction 7

2.1.2 Coordinated ensemble construction 9

3 RELATIONAL ENSEMBLE CONSTRUCTION 10

3.1 Motivation 10

3.2 Problem formulation 11

3.3 Relational subgraph resampling 12

3.4 Experimental evaluation 16

3.4.1 Baseline approach 17

3.4.2 Methodology 17

3.4.3 Results 19

3.5 Related work 22

3.6 Conclusion 24

Page

4 RELATIONAL ENSEMBLE INFERENCE 26

4.1 Motivation 26

4.2 Problem formulation 27

4.3 Collective ensemble classification 30

4.4 Experimental evaluation 32

4.4.1 Baseline approaches 34

4.4.2 Methodology 36

4.4.3 Results 37

4.5 Related work 42

4.6 Conclusion 44

5 RELATIONAL ENSEMBLE CLASSIFICATION FRAMEWORK 46

5.1 Motivation 46

5.2 Problem formulation 47

5.3 Relational ensemble framework 49

5.3.1 Ensemble learning 49

5.3.2 Ensemble inference 50

5.4 Experimental evaluation 52

5.4.1 Baseline approaches 52

5.4.2 Methodology 53

5.4.3 Results 54

5.5 Related work 56

5.6 Conclusion 58

6 THEORETICAL ANALYSIS 59

6.1 Motivation 59

6.2 Framework 60

6.2.1 Model definitions 61

6.2.2 Error decomposition 62

6.2.3 Expectations 64

Page

6.3 Analysis 65

6.3.1 Assumptions 65

6.3.2 Variance reduction 66

6.3.3 Bias reduction 68

6.3.4 Loss reduction 69

6.3.5 Learning variance reduction 71

6.3.6 Inference variance reduction 72

6.3.7 Effect of resampling method on error 75

6.4 Related work 75

6.5 Conclusion 76

7 CONTRIBUTIONS 78

8 CONCLUSION 79

8.1 Summary 79

8.2 Future work 81

LIST OF REFERENCES 82

A APPENDIX 87

A.1 Synthetic data 87

A.2 Real world datasets 88

A.2.1 IMDB 88

A.2.2 Facebook 88

A.2.3 WebKB 89

VITA 90

LIST OF TABLES

Table Page

4.1 Experimental results for varying autocorrelation and linkage on synthetic data, reporting AUC values. 41

A.1 Algorithm for generating synthetic dataset with a relational group structure. 87

LIST OF FIGURES

Figure Page

1.1 Graphic illustration of ensemble classification. 1

1.2 Graphic illustration of the various design dimensions for ensemble classification. 5

3.1 Graphic illustration of Relational Subgraph Resampling (RSR). 13

3.2 Bagging experimental results on WebKB data for various class labels. . 20

3.3 Bagging and variance estimation experimental results on synthetic data. 21

4.1 Graphic illustration of model interleaving, showing how predictions for the same instance are aggregated across the models. 31

4.2 Graphic illustration of merging multiple link sources on the same network of objects. 35

4.3 Synthetic experiments show significant accuracy improvement of proposed CEC ensemble model at various proportions of available true labels in the test graph. 38

4.4 Facebook experiments show significant accuracy improvement of proposed CEC ensemble model at various proportions of available true labels in the test graph. 39

4.5 IMDB experiments show significant accuracy improvement of proposed CEC ensemble model at various proportions of available true labels in the test graph. 40

4.6 Synthetic experiments show significant accuracy improvement of proposed CEC ensemble model as more link graphs are considered by the ensemble. 41

4.7 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. 42

5.1 Synthetic experiments show significant accuracy improvement of proposed RSR-CEC ensemble model at various proportions of available true labels in the test graph. 55

Figure Page

5.2 Facebook experiments show significant accuracy improvement of proposed RSR-CEC ensemble model at various proportions of available true labels in the test graph. 56

8.1 Graphic illustration of our contributions with respect to the various ensemble design dimensions. 81

ABSTRACT

Eldardiry, Hoda M. Ph.D., Purdue University, May 2012. Ensemble Classification Techniques for Relational Domains. Major Professor: Jennifer Neville.

Ensemble learning techniques combine predictions of multiple models to improve classification, while relational learning methods focus on utilizing link information to improve classification for network data. Our goal is to combine these two machine learning directions by applying ensemble classification to improve relational learning.

There are many domains in which data exhibits complex and heterogeneous relational structures. However, applying traditional ensemble methods in relational domains has a number of limitations that have neither been studied nor addressed before. This dissertation (1) explores these limitations, (2) gives explanations for why they exist, (3) provides solutions for them by proposing a relational ensemble framework, (4) applies the proposed relational ensemble framework to combine link information for collective classification in multi-network domains, (5) develops a more general framework that works for single-network settings, and (6) presents a theoretical analysis framework to support the empirical findings.

Traditional ensemble methods assume independent and identically distributed (i.i.d.) data and exact inference models. Both assumptions are violated in relational domains, where data has heterogeneous link structures, and models use collective inference techniques. Ensemble methods that assume i.i.d. data use independent sampling approaches during ensemble learning. This underestimates the increased variance exhibited by network data, so the ensemble is unable to reduce the full amount of variance in learning. We propose a novel method for learning ensembles from relational data, which can capture and reduce more learning variance. The exact inference assumption overlooks inference variance, introduced by collective classification techniques. We propose a novel ensemble classification method for relational data, specifically when the ensemble uses collective inference base models. This is the first ensemble method that accounts for and reduces inference variance.

1. INTRODUCTION

Ensemble classification methods have been widely studied as a means of reducing classification error by combining multiple *base* models for prediction. However, much of the previous 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 [1,2], these previous works have a number of limitations in that: (1) they focus on the reduction of only one type of error (due to learning), (2) they focus only on networks composed of multiple relations, and (3) there is no theoretical analysis to show the mechanism by which ensembles reduce model error in relational domains. Figure 1.1 shows an ensemble classifier.

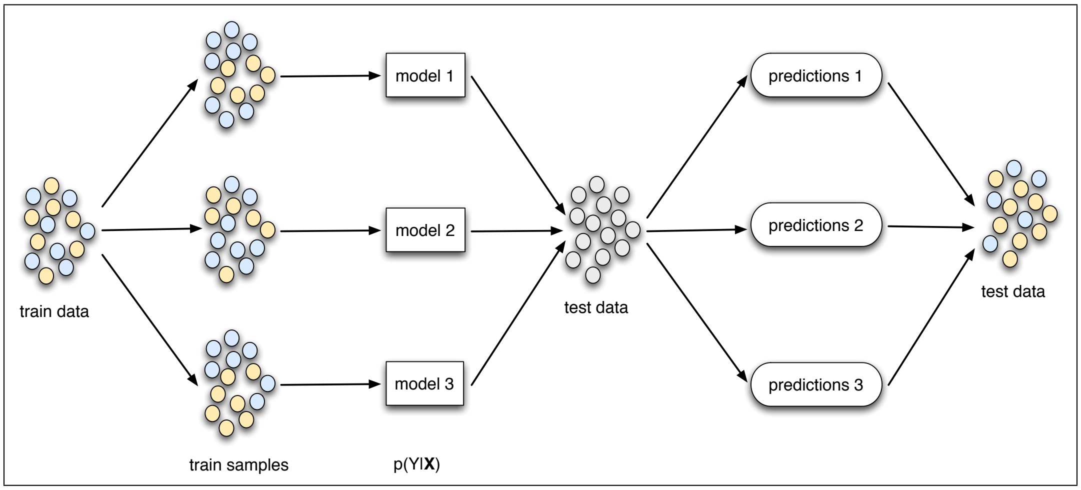


Fig. 1.1. Graphic illustration of ensemble classification.

In this work, we go beyond previous work and develop ensemble methods that consider various aspects of relational data and relational models. Our proposed approaches focus on the reduction of errors due to both learning and inference, and are applicable in both single- and multiple- relation settings. Moreover, we propose the first theoretical framework that analyzes various error components of relational ensembles. Our empirical results show significant classification accuracy gains by our proposed methods, and our theoretical analysis explains and confirms the empirical results.

1.1 Ensemble design

Traditional ensembles have considered three design dimensions, with a limited number of alternative choices for each dimension. In this work, we formulate novel ensemble techniques for relational domains through unique combinations of design choices. Figure 1.2 shows the different design choices we discuss below.

1.1.1 Input data treatment

In a typical classification scenario, only a single training dataset is given. However, for ensemble learning, multiple training datasets are needed to learn multiple models. The treatment should produce datasets of the same size as the input data to learn different models from them (Figure 1.2(a)).

One way to treat input data is using resampling (i.e., sampling with replacement), to generate multiple pseudosamples to learn the base models from. For example, bagging approaches (e.g., [3]) use IID resampling (i.e., sampling instances independently with replacement), then aggregate the multiple models predictions. Other methods learn each base models from the full set of instances, but use a different subset of features in each model [4–7]. Moreover, boosting approaches (e.g., [8–10]) construct the models in a coupled fashion reweighting instances so that their weighted vote gives a good fit to the data.

In this work, we propose two alternative methods for input data treatment, which consider characteristics of relational data, to learn more useful ensembles. In Chapter 3, we propose a method for resampling from relational data that allows the ensemble to reduce more error due to learning than the IID resampling method considered by bagging. In Chapter 4 we propose learning the base models in a new way, in which each model is learned from a different link graph. This allows our ensemble to utilize the different relational structures present in the network, while separating them during learning to reduce noise that can result from too many links in the same graph. Specifically, the multiple link types in the network are used to subset the data (instead of the conventional feature subset approach which would sample from the node features).

1.1.2 Choice of base model

An ensemble is typically composed of multiple base models (shown in Figure 1.2(b)). Traditional ensembles assume i.i.d. base classifiers, which model information about each instance in isolation, and use exact inference techniques. We note that once we move beyond the i.i.d. assumption (for models), to the richer relational settings, much higher classification performance can be achieved.

In Chapter 3 we propose to learn a relational classifier for the base models of the ensemble. Since relational classifiers use the dependencies between attributes of interrelated objects to improve classification, this necessitates a relational approach to resampling, which can preserve the link structure in the data, so the models can utilize the link structure.

In Chapter 4 we propose to learn collective classification base models, which models the dependencies between class labels of interrelated objects. In collective inference, predictions made at one inference iteration can be used to improve predictions to be made at the next iteration [11–14]. This offers an opportunity for a novel ensemble design dimension which we present in section 1.1.4.

1.1.3 Output aggregation

The output aggregation step is shown in Figure 1.2(d). Traditional ensembles aggregate the set of predictions made by the base models for each instance independently. This happens after the base models are run independently for inference. Aggregation strategies include simple averaging, weighted averaging and majority voting. In this dissertation, we focus on simple averaging for output aggregation, since we explore another level of aggregation that precedes the final output aggregation. We discuss this next.

1.1.4 Model interleaving

In Chapter 4, we propose a new ensemble design dimension that we call model interleaving. This method utilizes a unique opportunity offered by relational domains, which stems from the use of collective classification. Instead of running the base models independently for inference, and aggregating the final models' output, we take advantage of the collective inference process and allow a prediction made by one model to influence the prediction made for the same instance by another model. Our proposed approach of *across-model* inference aggregation facilitates an additional reduction of inference error on top of the traditional learning error reduction achieved by the final step of output aggregation. Figure 1.2(c) shows that this model interleaving step takes place before the final output aggregation step.

1.2 Ensemble framework and analysis

In Chapter 5 we combine our ensemble design choices to reduce errors due to *both* the learning and inference processes in relational data. Specifically, we use our relational resampling approach presented in Chapter 3, which aims to capture the increased variance in relational data, enabling the ensembles to reduce more of the variance due to learning. This is combined with the interleaved inference presented in Chapter 4, which enables the ensembles to reduce more of the variance due to inference. This combination also facilitates the application of model interleaving to networks with single relations, not just networks with multiple relations.

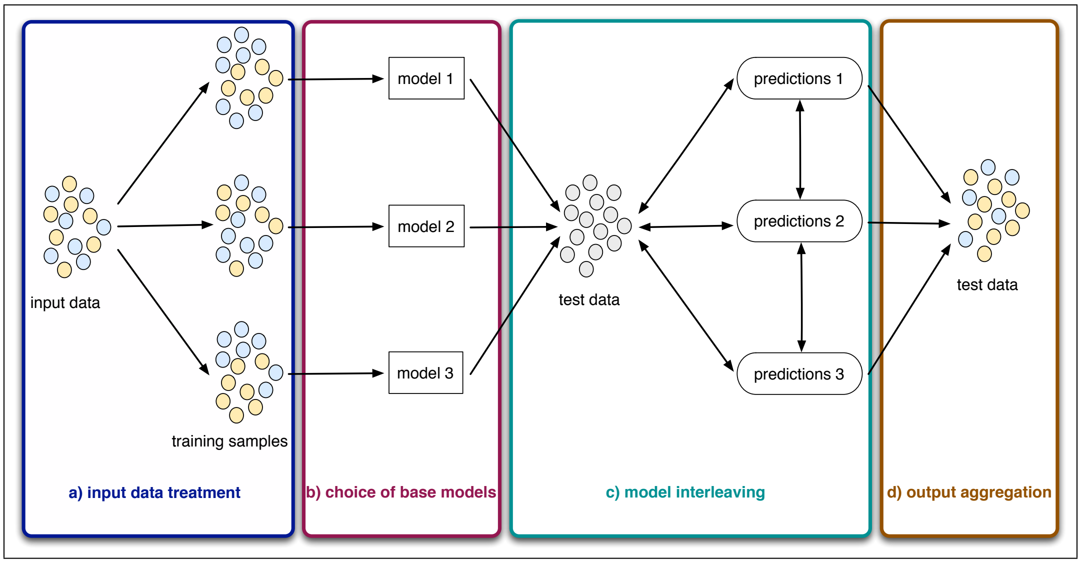


Fig. 1.2. Graphic illustration of the various design dimensions for ensemble classification.

Classification error is typically decomposed into variance and bias components [15–18]. In i.i.d. domains, ensemble learning methods have been shown to reduce classification error by reducing variance (e.g., bagging [3]) or reducing bias (e.g., boosting [10]). In this work, we focus on the reduction of variance.

Previous ensembles that reduce variance have focused on reducing one type of variance, which we refer to as the *learning variance*. This is the variance due to learning the models from different training datasets. On the other hand, collective inference models applied to relational data have been shown to have additional sources of error due to variance in the inference process [19]. We refer to the variance in predictions made by the same model given different subsets of true labels for objects in the test set, as the *inference variance*.

In Chapter 6, we propose a theoretical framework to analyze the error reduction of relational ensembles. We use a relational bias/variance decomposition similar to that of [19] 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) our *across-model* approach, which runs the component models simultaneously for collective inference and aggregates intermediate predictions across the models during inference. 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 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 analytical investigation of error for relational ensembles.

2. BACKGROUND

2.1 Ensemble learning

Ensemble classification methods have been shown to produce more accurate predictions than the *base* component models [20, 20–23]. Due to their effectiveness, ensemble approaches have been applied in a wide range of domains to improve classification. Ensemble learning algorithms work by running a "base learning algorithm" multiple times, and combining the predictions from the resulting models. There are two main approaches to designing ensemble learning algorithms.

2.1.1 Independent ensemble construction

The first approach is to construct each model independently in such a way that the resulting set of models are accurate and diverse–that is, each individual model has a reasonably low error rate for making new predictions and yet the models disagree with each other in many of their predictions. If such an ensemble is constructed, it is easy to see that it will be more accurate than any of its component classifiers, because the disagreements will "cancel out" [24]. This approach improves classification accuracy by reducing error due to variance.

Bagging

One way to force a learning algorithm to construct multiple models is to run the algorithm several times and provide it with somewhat different training data in each run [3,20,25,26]. For example, Breiman [3] introduced the Bagging ("Bootstrap Aggregating") method which works as follows. Given a set of m training data points, Bagging chooses in each iteration a set of data points of size m by sampling uniformly with replacement from the original data points. This creates a resampled data set in which some data points appear multiple times and other data points do not appear at all. If the learning algorithm is unstable–that is, if small changes in the training data lead to large changes in the resulting model–then Bagging will produce a diverse ensemble of models.

The traditional methods assume i.i.d. data, so they use i.i.d. resampling techniques. However, these methods do not work well for relational data. Chapter 3 presents a method for bagging from relational data that uses a novel relational resampling approach to ensure accuracy and diversity of the ensembles.

Feature subsets

Another way to force diversity is to provide a different subset of the input features in each call to the learning algorithm [4–7]. For example, in a project to identify volcanoes on Venus, Cherkauer [4] trained an ensemble of 32 neural networks. The 32 networks were based on 8 different subsets of the 119 available input features and 4 different network sizes. The input feature subsets were selected to group together features that were based on different image processing operations. The resulting ensemble classifier was significantly more accurate than any of the individual neural networks.

Chapter 4 presents a method for ensemble inference that uses a similar learning approach in which each link types are used as features, and each model is learned using a different link type from the network.

Other methods

There are other methods that independently construct ensembles, but are not relevant to this work. Dietterich and Bakiri [27] describe a technique called error-correcting output coding in which they manipulate the output labels of the training data to force diversity. Another way of generating accurate and diverse ensembles is to inject randomness into the learning algorithm. (E.g., see [28–31].)

2.1.2 Coordinated ensemble construction

The second approach to designing ensembles is to construct the models in a coupled fashion so that their weighted vote gives a good fit to the data [8, 9]. This approach improves classification accuracy by reducing error due to bias. These methods view an ensemble as an additive model that predicts the class of a new data point by taking a weighted sum of a set of component models. In statistics, such ensembles are known as generalized additive models. Freund and Schapire [8, 10] introduced the Adaboost algorithm for constructing an additive model. However, since variance reducing ensembles are more general and have been more widely studied [32–34], this dissertation focuses on these types of ensembles.

We discuss related work at the end of each chapter.

3. RELATIONAL ENSEMBLE CONSTRUCTION

3.1 Motivation

Bagging (i.e., bootstrap aggregating) is an ensemble method that improves classification accuracy by reducing the error due to variance in learning [3]. Bagging works by generating multiple versions of a model and using them to get aggregated predictions. For a typical classification task, one training dataset and one test dataset are given. In bagging, bootstrapping (sampling with replacement, a.k.a. resampling) is used to generate multiple training datasets from the given one. Multiple models are learned from the generated datasets. Then, each model is applied to predict label values for the test set instances. Each instance's final prediction is produced by aggregating the various models' predictions for that instance.

Although bagging was initially developed for classification of independent and identically distributed (i.i.d.) data, it can be directly applied for relational data by using a relational classifier as the base model. This straightforward approach of applying bagging for a relational classification task can improve accuracy, but suffers from a number of limitations. First, relational data characteristics (that improve prediction when considered) will only be exploited by the base *relational* classifier, and not by the bagging mechanism itself. However, explicitly accounting for the structured nature of relational data, can significantly improve bagging. Second, typical bagging methods that assume i.i.d. data fail to preserve the relational structure of non-i.i.d. data. This can (1) prevent the relational base classifiers from exploiting these structures, and (2) fail to accurately capture properties of the dataset which can lead to inaccurate models and classifications.

Bagging typically uses sampling independently (i.e., at random) with replacement to generate multiple bootstrap samples to learn the models from. The key idea behind this work is that constructing the ensembles using i.i.d. sampling can underestimate the variance in learning, when applied to relational data since it will not account for the reduced effective sample size due to dependencies among interrelated objects [35]. Reduction in effective sample size leads to an increased population distribution variance [36], but i.i.d. bootstrap samples (and consequently the models learned on them) will underestimate this variance. When the ensemble construction (i.e. resampling) procedure fails to accurately capture the population variance, bagging fails to reduce the full amount of learning variance. Additionally, independent sampling from a relational dataset does not preserve the relational structure, since a given node will not necessarily have all of its neighbors in the sample. This limits fully exploiting *autocorrelation* and *linkage*, and the increased accuracy that can otherwise be achieved.

This work proposes a relational ensemble construction method that explicitly accounts for the structured nature of relational data and significantly improves classification accuracy of bagging in relational domains. Our proposed method can enable learning more accurate ensembles from relational data, by expanding the reduction of error due to variance in learning, while preserving the relational characteristics in the data.

The proposed bagging approach uses a relational subgraph resampling algorithm in lieu of the traditional i.i.d. resampling mechanism. We will present empirical results which show that bagging using the proposed relational resampling method significantly outperforms bagging using i.i.d. resampling, on both synthetic and real-world datasets. In addition, bagging using relational resampling can better exploit increased autocorrelation in the data, due to its ability to preserve the relational structure during sampling.

3.2 Problem formulation

The general bagging approach is outlined in Algorithm 3.1. Given a fully-labeled training set composed of a graph $G_{tr} = (V_{tr}, E_{tr})$ with nodes V_{tr} and edges E_{tr}, and an unlabeled test set composed of a graph $G_{te} = (V_{te}, E_{te})$ with nodes V_{te} and edges E_{te}; an ensemble of size k models is constructed as follows.

A pseudosample $G_{ps} = (V_{ps}, E_{ps})$ is generated by resampling from G_{tr} (line 2) and a model F is learned from G_{ps} (line 3). F is composed of a joint probability distribution over the labels of V_{ps}, conditioned on the observed attributes and graph structure in G_{ps}. Each learned model F is applied to G_{te}, and a set of marginal probability distributions P (i.e., predictions) over the labels of nodes V_{te} are produced (line 4). The final step of bagging involves averaging the k models' predictions P^v for each node v to produce the final aggregate node prediction (line 6).

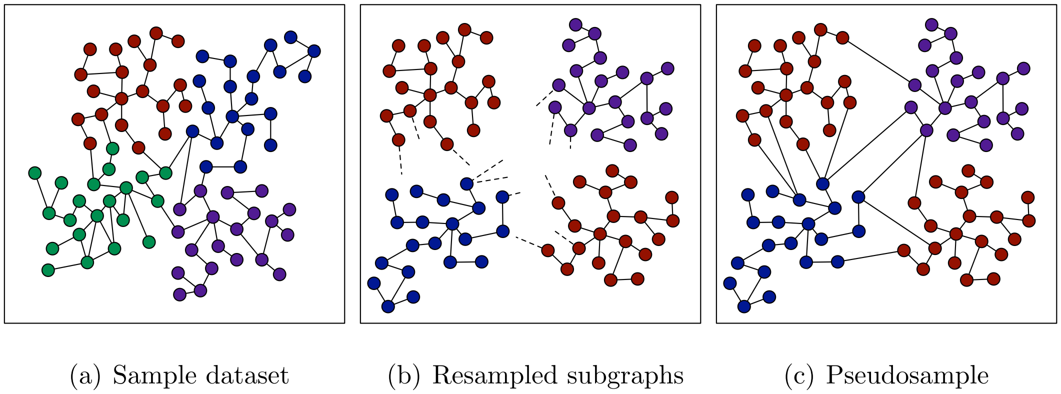
Note that any relational learner can be used in line 3, and that the conventional resampling method in which nodes are sampled independently with replacement is used in line 2. Next we present our proposed resampling method that can be used instead of the typical i.i.d. resampling when the data exhibits a relational structure.

Algorithm 3.1 Bagging($G_{tr} = (V_{tr}, E_{tr}), G_{te} = (V_{te}, E_{te}), k$)

```
1: for $j := 1$ to $k$ do
2:     $G_{ps_j} = Resample(G_{tr})$ {construct pseudosample}
3:     $F_j = LearnModel(G_{ps_j})$ {learn model}
4:     $P_j = F_j(G_{te})$ {apply model}
5: for all node $v_i \in V_{te}$ do
6:     $P^v = (\sum P_i^v)/n$ {aggregate models' predictions}
7: return $P$
```

3.3 Relational subgraph resampling

Relational subgraph resampling (RSR) is a novel approach for resampling relational data. This approach can be used for more accurate ensemble construction when bagging is used for relational classification tasks. By using RSR instead of the typical independent sampling for generating bootstrap datasets for learning the ensemble.



(a) Sample dataset

(b) Resampled subgraphs

(c) Pseudosample

Fig. 3.1. Graphic illustration of Relational Subgraph Resampling (RSR).

Given a relational sample dataset (e.g., see figure 3.1(a)), the first phase of the algorithm selects subgraphs based on snowball sampling [37]. It repeatedly selects a subgraph of size b via breadth-first search from a randomly selected seed node (figure 3.1(b)). The second phase then links up the selected subgraphs (figure 3.1(c)). The aim is to preserve the local relational dependencies among instances in the subgraph, while generating a pseudosample with sufficient global variance by linking up the set of selected subgraphs. The key idea behind this approach is that when autocorrelation is high, the effective sample size is determined by the number of underlying groups in the data. As such, this approach attempts to sample these groups instead of single instances, thus preserving the effective sample size of the data.

One challenge is how to link up the subgraphs into a single relational data graph. Due to the varied link structure of relational data, there will be a large number of nodes on the periphery of the selected subgraphs. If the peripheral nodes are missing a significant portion of their neighbors, this could bias the properties of the sample. The potential for bias due to peripheral nodes is much greater in relational data with varied link structure than temporal or spatial data with regular link structure. Consider a lattice subgraph where each interior node has four neighbors. The peripheral nodes each have three neighbors, except for the four corners which have two. Each peripheral node is missing at most 50% of its neighbors. However, in relational data with concentrated linkage, if the peripheral nodes in the sample are hub nodes with high degree from the original data, they may be missing almost all their neighbors (i.e., ≈100%). To deal with this issue, we will outline a procedure to link up the peripheral nodes in the selected subgraphs, which attempts to maintain the global graph properties and attribute dependencies of the original data. More specifically, the relational autocorrelation is maintained by maximizing attribute similarity between nodes as they are linked, while the link structure is maintained by considering the neighborhood similarity when selecting nodes to link.

The procedure for RSR is outlined in Algorithm 3.3. Given a sample relational data graph $G = (V, E)$, it returns a pseudosample data graph $G_{PS} = (V_{PS}, E_{PS})$. The first phase samples a set of $N_S = \lceil \frac{|V|}{b} \rceil$ subgraphs of size b from G, using breadth-first search from N_S randomly selected seed nodes. Note that the sampling is with replacement from the graph, so a node may appear in multiple subgraphs, one subgraph, or none (lines 2-5). 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). The pseudosample edge set (E_{PS}) initially consists of all the edges within the selected subgraphs.

This is augmented by a two-pass process that links up the peripheral nodes across subgraphs, choosing the links that are most *similar* to the links that were broken by the subgraph selection process. For example, if a peripheral node v_p is linked to node v_m in the original dataset but v_m was not selected as a member of v_p's subgraph, and a node *similar* to v_m is available (and is missing a neighbor) in another subgraph, it is linked to v_p.

In lines 6-8, missing neighbors in each subgraph are identified. An initial pass attempts to link up peripheral nodes from the various subgraphs that were originally linked in the data sample. Then an additional pass links up peripheral nodes from different subgraphs that have not been linked during the first pass, and the selections are based on *similarity*.

Algorithm 3.2 Relational Subgraph Resampling: RSR$((G = (V, E), b))$

```
$V_{PS} \leftarrow \emptyset$;  $E_{PS} \leftarrow \emptyset$
for $s := 1$ to $\lceil \frac{|V|}{b} \rceil$ do
    choose a seed node $v_s$ randomly from $V$
    construct $V^S$ by selecting $b-1$ nodes around $v_s$ using breadth-first search
    $E^S = \{e_{ij} \in E \ \ s.t. \ \ v_i, v_j \in V^S\}$;  $V_{PS} \leftarrow V_{PS} + V^S$;  $E_{PS} \leftarrow E_{PS} + E^S$
for all $V^S$ do
    for all $v_i \in V^S$ do
        $N_i^S = \{v_j \ \ s.t. \ \ e_{ij} \in E \wedge v_j \notin V^S\}$
while true do
    update = false
    for all node $v_i \in V_{PS}$ do
        if $|N_i^S| > 0$ then
            randomly select $v_j$ from $N_i^S$;  let $C_j = \{v_k : v_k \equiv v_j \wedge v_k \in V^{S' \neq S} \wedge v_i \in N_k^S\}$
            Select $v_m \in C_j$ $s.t.$ $v_m = argmin \mathrm{Path}(v_m, v_i)$, maximize $|N_m|$ to break ties
            if $v_m \neq null$ then
                $N_i^S = N_i^S - \{v_j\}$;  $N_m^{S'} = N_m^{S'} - \{v_i\}$;  $E_{PS} = E_{PS} + \{e_{im}\}$
                update = true
    break if update = false
while true do
    update = false
    for all node $v_i \in V_{PS}$ do
        if $|N_i^S| > 0$ then
            randomly select $v_j$ from $N_i^S$;  let $C_j = \{v_k : |N_k| > 0 \wedge v_k \in V^{S' \neq S}\}$
            Select $v_m \in C_j$ $s.t$ $v_m = argmax \mathrm{Sim}(v_m, v_j)$, break ties by $argmin \mathrm{Path}(v_m, v_i), argmax |N_m|$
            if $v_m \neq null$ then
                $N_i^S = N_i^S - \{v_j\}$;  $N_m^{S'} = N_m^{S'} - \{v_i\}$
                $E_{PS} = E_{PS} + \{e_{im}\}$
                update = true
    break if update = false
return $G_{PS} = (V_{PS}, E_{PS})$
```

In the first pass (lines 9-18), v_p links to a copy of v_m if available (and is missing a neighbor) in another subgraph in the pseudosample. If there are multiple copies, the copy with the shortest path length to v_p and with the greatest number of missing neighbors is chosen. Then links are created for any nodes with neighbors still missing after the first pass. For example, if there were no copies of v_m selected for the pseudosample, then a corresponding link for v_p is not created in the first pass.

The second pass (lines 19-29) looks for the node in the pseudosample that is most similar to v_m. Node similarity is calculated based on both the attributes of the nodes and on their link structure (i.e., the number of neighbors they have in common in the original data). Again, if there are multiple nodes with the same (maximum) similarity to v_m, the node with the the shortest path length to v_p and with the greatest number of missing neighbors is chosen.

The following similarity function is used to compare nodes based on both attributes and links:

$$\text{Sim}(v_i, v_j) = \alpha * \text{aSim}(v_i, v_j) + (1 - \alpha) * \text{lSim}(v_i, v_j)$$

where the attribute similarity is defined as $\text{aSim}(v_i, v_j) = \#$ shared attribute values between v_i and v_j, and the link similarity is defined as $\text{lSim}(v_i, v_j) = \#$ common neighbors between v_i and v_j. In the experiments reported in this section, α is set to 0.15 to upweight the importance of matching on link structure.

3.4 Experimental evaluation

The proposed resampling methodology is evaluated in two different relational settings. First, to improve the accuracy of bagging. And second, to estimate a sampling distribution of feature scores and calculate an accurate estimate of the variance of the feature score distribution.

3.4.1 Baseline approach

In both scenarios RSR is compared to a baseline i.i.d. resampling method outlined in Algorithm 3.4.1, where n nodes are sampled randomly with replacement from a given sample V of size n to construct one pseudosample.

Algorithm 3.3 Independent Resampling: IR($V = v_1, .., v_n$)

```
$V_{PS} \leftarrow \emptyset$
for $j := 1$ to $n$ do
    Randomly select $v_s$ from $V$
    $V_{PS} \leftarrow \{V_{PS} + v_s\}$
return $V_{PS}$
```

3.4.2 Methodology

Synthetic relational datasets that exhibit relational autocorrelation and concentrated linkage are generated for evaluation in both experiments (described in Section A.1). The area under the ROC curve (AUC) of each type of model is measured. Relational probability trees (RPTs) [38] is the classification model used to predict the values of the class label. Any other relational model can be used.

Bagging Experiments Bagging using RSR for bootstrapping is compared to bagging using IR and to using just a single model. Each resampling algorithm is used by Algorithm 3.1 (line 2), to construct $k = 5$ pseudosamples and learn an ensemble of 5 models.

The first set of bagging experiments uses synthetic datasets with increasing levels of autocorrelation {0.25,0.50,0.75} to test the hypothesis that as autocorrelation increases the improvement of RSR over IR should increase as well (due to a lower effective sample size). The RPTs are learned using MODE, COUNT, and PROPORTION as the aggregation functions in feature construction. Four training and testing sets of sizes 120 and 255 are generated respectively, for a total of 16 training-test pairs, and the error reduction of each bagging approach compared to the single model is measured.

The second set of bagging experiments evaluates the models using the Webkb dataset (described in Section A.2.3), where the classification task is to predict page category. As in previous work on this dataset, the category "other" is not predicted. The performance of bagging using RSR (using subgraph size of $b = 50$ and $\alpha = 0.15$), bagging using IR and a single model are compared. RPTs are learned using MODE features. For each of the three models, 12 training-testing pairs based on the four disjoint websites in WebKB are used. The AUC for each class label value is measured separately. Model robustness is evaluated by adding random attributes to the data. The results for 0, 3 and 6 random attributes are presented. This is to test the hypothesis that RSR is more accurate at determining which features are irrelevant in relational data.

Variance Estimation Experiments The goal of this experiment is test our hypothesis that for a relational dataset RSR bootstrap samples can capture the population variance of a statistic more accurately than IR bootstrap samples. As mentioned earlier this is because IR accounts for the reduced effective sample size due to the structured nature of the dataset, and therefore does not underestimate the population variance unlike the IR approach.

In this experiment, resampling is used to calculate an approximation of the unknown sampling distribution of relational features scores and estimate the variance of their distribution. To estimate the sampling distribution of a statistic, each of RSR and IR are applied 20 times to create 20 pseudosamples of the data. The statistic is then calculated on each pseudosample and the empirical distribution of values is returned as an approximation of the statistic's sampling distribution. Synthetic datasets described in A.1 are used with 270 objects and groups of size 15.

To calculate variance, 20 pseudosamples are created, a feature score for each sample is calculated, then the variance (Var_{est}) of the distribution of the 20 feature scores is calculated. Two relational features are considered: one that is correlated with the class (i.e., MODE(linked.X_0)) and one that is random (i.e., MODE(linked.X_1)). The feature score calculation assesses the correlation of the feature values with the class labels C using Pearson's corrected contingency coefficient [39].

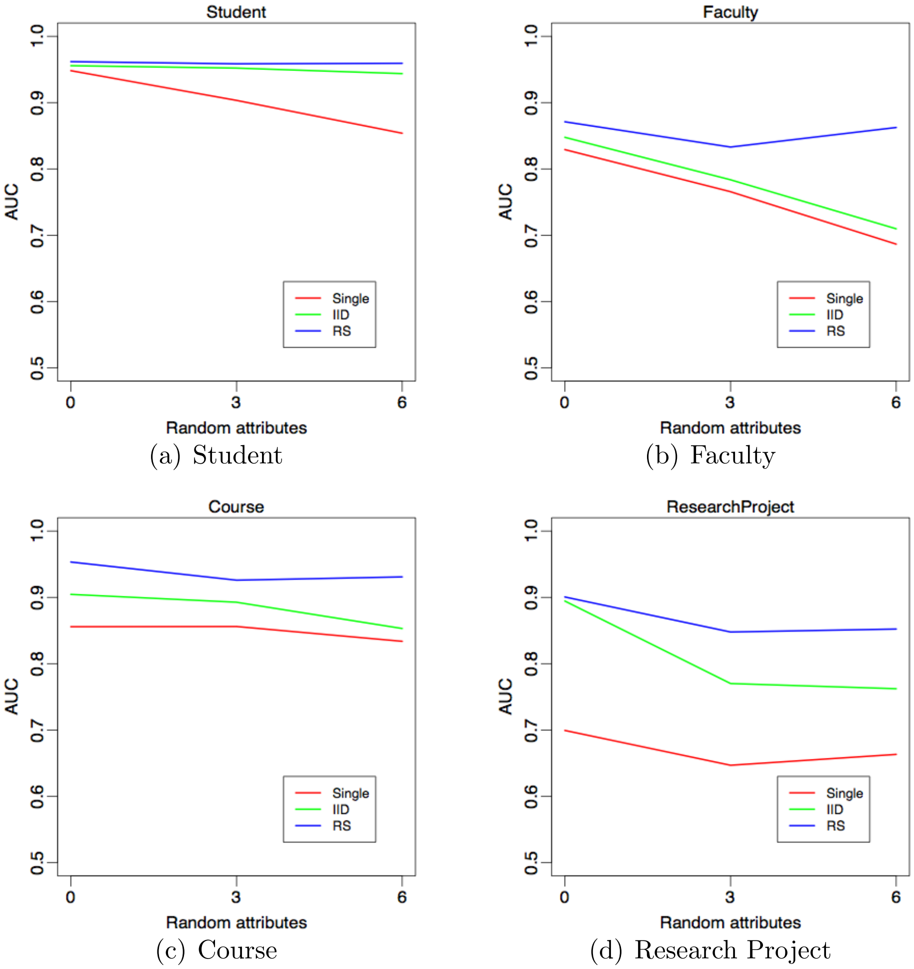
To evaluate the accuracy of the feature score variance estimates, they are compared to the empirical variance of the feature scores in the synthetic datasets. The population variance Var_{pop} of the features is estimated by generating 100 different datasets and calculating the variance from the empirical distribution of features scores in these datasets. The relative error is used as a measure of accuracy: $\frac{(Var_{pop}-Var_{est})}{Var_{pop}}$.

The feature score variances are calculated using RSR and IR and the relative error for both approaches is measured. The average relative error over 10 trials is reported. For RSR, performance is evaluated on subgraphs of varying sizes of b: $\{1,5,15,25,35,45\}$. Since RSR aims to exploit the underlying groups structure, it is expected to outperform IR most significantly when the subgraph size is the same as the average group size (15) of the generated data.

3.4.3 Results

The results show that using the proposed relational ensemble construction method for bagging results in significant performance improvements over both the single model and bagging with i.i.d. resampling. Furthermore, compared to the IR, RSR results in more accurate variance estimates on both correlated and random attributes.

Figure 3.2 shows the results for the WebKB data, plotting the AUC values for each class label value: Student, Faculty, Course and Research Project. Bagging with IR produces higher accuracy than the single model. However, bagging with RSR is not only significantly better than the single model, it also achieves equivalent or better performance compared to IR for all datasets. As more random attributes are



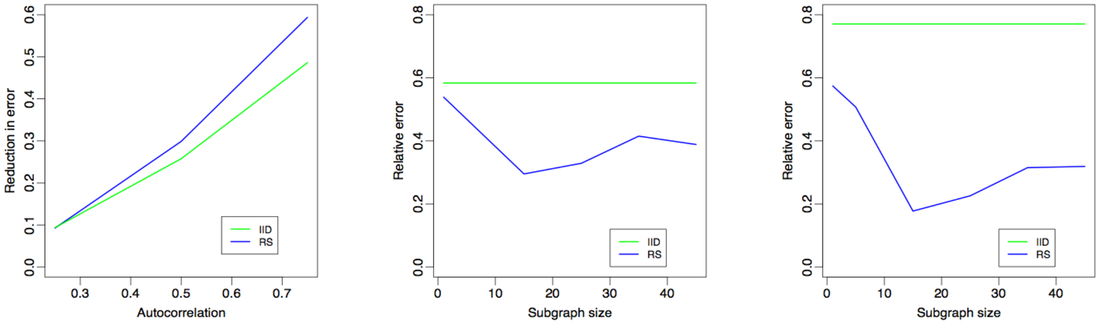
(a) Student (b) Faculty

(c) Course (d) Research Project

Fig. 3.2. Bagging experimental results on WebKB data for various class labels.

included in the learning process, the single model and the i.i.d bagging model both experience a degradation in performance while bagging using RSR is more robust.

Figure 3.3(a) presents the results for synthetic data bagging experiments for different levels of autocorrelation. Reduction of AUC error achieved by each of the bagging models over the single model is graphed. Notice that as autocorrelation increases, the difference between RSR and IR increases. These synthetic data experiments are conducted with relatively simple relational datasets. Performance difference between the two approaches is expected to increase on complex, real-world relational datasets.



(a) Reduction in error for bagging.

(b) Feature variance estimation error for correlated attribute.

(c) Feature variance estimation error for random attribute.

Fig. 3.3. Bagging and variance estimation experimental results on synthetic data.

Figure 3.3(b) and 3.3(c) graphs the average relative error in variance for both IR and RSR using different subgraph sizes. Figure 3.3(b) graphs the results for the correlated feature and Figure 3.3(c) graphs the results for the random feature. Both plots show that RSR results in lower error than IR. Furthermore, RSR estimates of variance increase in accuracy as the subgraph size approaches the underlying group size (15). Notice also that RSR shows a more significant reduction in estimation error for the feature formed from the random attribute (Figure 3.3(c)).

Accurately estimating the variance of random (or irrelevant) features is likely to impact model learning more significantly than accurate estimation for real features, since reduction in effective sample size increases the risk of Type I errors [36]. Improved resampling techniques can be used to develop more accurate feature selection models, reducing the risk that random features are selected for inclusion in relational models when the data exhibits linkage and autocorrelation.

3.5 Related work

Resampling is a statistical technique that approximates sampling from the true underlying population by sampling with replacement from a single dataset D to create a set of *pseudosamples* $\mathbf{D}'$. Each pseudosample contains as many instances as the original data set. Some instances in the original data set will occur multiple times in a given pseudosample, and others will not occur at all. The basic idea of resampling is that, in the absence of any other information about the population, the observed sample contains the best available information about the underlying population. Thus, resampling from the sample is the best way to approximate draws from the population.

Initially, resampling was introduced for i.i.d. data [40]. However, when the data instances are interdependent, pseudosamples generated by i.i.d. resampling are likely to exhibit less variance than the underlying population distribution. Dependencies among instances reduce the *effective* sample size of the data and thus increase the variance of statistics estimated from those data [36]. Resampling techniques that ignore the dependencies and sample independently from the instances will be replicating the actual sample size, not the effective sample size, and thus they are likely to underestimate the variance of statistics calculated from the data.

Previous work in spatial statistics has investigated graph-based *reuse sampling* techniques for lattice graphs, which use small, overlapping subgraphs as pseudosamples [41, 42]. A statistic is repeatedly calculated on smaller subgraphs to estimate the variance of its sampling distribution. This estimate is then rescaled to reflect the number of instances in the original data sample. For example, consider a regular lattice graph with degree four, contiguous subgraphs of length four (i.e., 4×4 squares) can be used as the pseudosamples and then scale the estimate of variance to approximate the original sample size.

In spatial and temporal datasets, where the link structure is generally homogeneous (either a line graph or a lattice of fixed degree), the choice of scaling factor is relatively straightforward. In relational data it is difficult to determine the effective sample size of a relational data set analytically due to heterogeneous link structure.

For example, consider a bipartite graph with 1000 objects X connected to 100 objects Y. There is a binary class label on the objects X and a binary attribute on the objects Y. When calculating feature scores concerning X, the actual sample size is $N_X = 1000$. However, if the class labels are perfectly autocorrelated through the objects Y (i.e., all X connected to the same object Y_i share the same class label value), then the *effective* sample size is $N_Y = 100$. Again one can think of this as having an urn filled with bunches of grapes—when you reach in to grab a single X you end up pulling out a single Y and all of its neighbors $\mathbf{X}$.

In practice, when the level of autocorrelation is somewhere between 0 and 1, the effective sample size N_{ESS} will be between the number of coordinating objects and the number of instances (i.e., $N_Y \leq N_{ESS} \leq N_X$). The goal of this work is thus to develop a relational resampling technique that accurately preserves the effective sample size of the data, thus producing more accurate estimates of the sampling distributions of statistics for heterogeneous, dependent data.

In order to maintain the dependencies among related data instances, the relational subgraph resampling technique proposed is used. Subgraph sampling is used to identify and sample sets of interconnected instances with each selection. Then, the selected subgraphs are linked back together in an attempt to preserve various relational properties throughout the sample.

Other than bagging, resampling is also used for estimating the sampling distribution of a statistic θ empirically. In practice, it is used to assess a wide variety of statistics including: the generalization accuracy of models, feature scores, predicted class labels, and model parameter estimates.

Moreover, machine learning techniques that use resampling are generally concerned with estimating the mean and/or variance of sampling distributions. For example, model selection techniques may use resampling to estimate the mean generalization error of different models in order to identify the model with lowest average error [43]. Alternatively, feature selection techniques may use resampling to estimate the standard errors of feature scores or model coefficients in an effort to identify which features are most relevant to the task [44].

Another application of resampling is active learning. The goal of active learning is to learn an accurate model with as few labeled instances as possible. Many criteria have been proposed to determine the most valuable instance for labeling. In particular, some methods have proposed selecting the instances whose prediction have highest variance, which is determined by resampling (see e.g., [45]). Our proposed RSR technique has been used for uncertainty estimation for active learning in relational domains [46].

3.6 Conclusion

Accurate resampling methods are important for many machine learning algorithms, including ensemble methods, active learning, and feature selection. Although it is straightforward to sample with replacement from i.i.d. data, it is more difficult to sample with replacement from an interconnected relational data graph in a manner that preserves the link structure and relational attribute dependencies.

This chapter presents a novel method for resampling from relational data, which accounts for the link structure and attribute dependencies of the data. Resampling in this manner maintains the local autocorrelation dependencies while allowing the global structure to vary as if the sampling is done from the population.

Since RSR explicitly accounts for the local structure in the data, it avoids overestimating the effective sample size and thus can be used for accurate variance estimation. To our knowledge, this is the first estimation algorithm that can effectively estimate sampling distributions in data with autocorrelation and heterogeneous link structure.

The presented methodology is evaluated on a real-world relational classification task, showing that it improves the accuracy of bagging when compared to IID resampling. In addition, the approach is evaluated on synthetic data, showing that compared to an IID approach, RSR results in significantly lower error when used to estimate the variance of feature scores.

The significance of the presented method has been evaluated for bagging. Moreover a variance estimation experiment confirms the conjectures made about why the proposed method improves bagging. Which is the ability of RSR to accurately capture the population variance of graph data.

4. RELATIONAL ENSEMBLE INFERENCE

4.1 Motivation

Ensemble classification methods learn an *ensemble* of models, apply them each for classification, then combine the models' predictions to produce more accurate classification decisions than the individual *base* models constituting the ensemble [20]. These methods were initially developed for classification of independent and identically distributed (i.i.d.) data, but they can be directly applied to relational data just by using a relational classifier as the base model. This straightforward approach can increase prediction accuracy in relational domains, but only to a limited extent. This is because relational data characteristics (which are often exploited to improve classification) will be considered only by the base classifier and not the ensemble method itself, thus opportunities to further exploit these characteristics in the ensemble would be ignored. Furthermore, since the typical ensemble methods were initially developed for i.i.d. datasets, their aim is to reduce errors associated with i.i.d. classification models, thus errors specific to relational classifiers would not be reduced by a straightforward application of previous methods.

Some recent work has addressed the first limitation by incorporating relational data characteristics directly into the ensemble method. For example, Preisach and Schmidt-Thieme [47] use voting and stacking methods to combine relational data with multiple relations. Moreover, Chapter 3 outlines a relational ensemble construction method to improve bagging in relational domains. However, these methods were developed with the conventional goal of ensembles in mind, which is to reduce errors associated with i.i.d. models; i.e., errors due to learning. There has been no work that has focused on the second limitation—to extend ensemble techniques to focus on reducing additional types of errors that can result from relational classification techniques; i.e., errors due to inference.

The key observation which motivated this work is that *collective classification* models in statistical relational learning suffer from two sources of variance error [19]. Collective classification methods [11, 14, 48, 49] learn a model of the dependencies in relational graph (e.g., social network) and then apply the learned model to *collectively* (i.e., jointly) infer the unknown class labels in the graph. The first source of error for these models is the typical variance due to *learning*—as variation in the data used for estimation causes variation in the learned models. The second source of error is due to variance in *inference*—since predictions are propagated throughout the network during inference, variation due to approximate inference and variation in the test data can both increase prediction variance.

The focus here is on reducing error due to variance by proposing a relational ensemble framework that uses a novel form of *across-model* collective inference for collective classification. The method proposed in this chapter propagates inference information across simultaneous collective inference processes running on the base models of the ensemble to reduce *inference variance.* Then the final model predictions are combined to reduce *learning variance.* This is the first ensemble technique that aims to reduce error due to inference variance.

The proposed method is evaluated using real-world and synthetic datasets, and is shown to outperform the baseline alternative solutions, including a straightforward relational ensemble approach. The results show that while prediction accuracy is improved using a straightforward ensemble approach, the method proposed here achieves significant additional gains by reducing error due to inference variance.

4.2 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 $G_{tr} =$

Algorithm 4.1 Relational Learning: RL$(G{=}(V,E), X, Y)$

```
Use $G$, $X$, $Y$ to learn a node classifier $F$ for $v_i \in V$
$F := P(Y_i|\mathbf{X}_i, \mathbf{X_R}\mathbf{Y_R})\ where\ \mathbf{R} = \{v_j : e_{ij} \in E\}$
return F
```

Algorithm 4.2 Collective Classification: CC$(G{=}(V,E), X,\tilde{Y}\ , F{=}P(Y_i|G,X,Y))$

```
$\hat{Y} = \tilde{Y}; \mathbf{Y_T} = \emptyset$
for all $v_i \in V\ s.t.\ y_i \notin \tilde{Y}$ do
    Randomly initialize $\hat{y}_i$ ; $\hat{Y} = \hat{Y} \cup \hat{y}_i$
repeat
    for all $v_i \in V\ s.t.\ y_i \notin \tilde{Y}$ do
        $\hat{y}_i^{new} = P(Y_i|\mathbf{X}_i, \mathbf{X_R}\hat{\mathbf{Y}}_\mathbf{R})\ where\ \mathbf{R} = \{v_j : e_{ij} \in E\}$
        $\hat{Y} = \hat{Y} - \{\hat{y}_i\} + \{\hat{y}_i^{new}\}$ ; $\mathbf{Y_T} = \mathbf{Y_T} \cup \hat{y}_i^{new}$
until $terminating\_condition$
Compute $\mathbf{P} = \{P_i : y_i \notin \tilde{Y}\}$ using $\mathbf{Y_T}$
return $P$
```

(V_{tr}, E_{tr}) with nodes V_{tr} and edges E_{tr}; observed features X_{tr}; and observed class labels Y_{tr}, the relational learning procedure (RL) outlined in Algorithm 4.1, outputs a model F composed of a joint probability distribution over the labels of V_{tr}, conditioned on the observed attributes and graph structure in G_{tr}. 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 $\tilde{Y}_{te} \subset Y_{te}$, and the model F learned using RL, the collective classification procedure (CC) outlined in Algorithm 4.2, outputs a set of marginal probability distributions P (i.e., predictions) over the labels of nodes V_{te}. Note that G_{tr} used for RL is different from G_{te} used for CC. The collective classification pseudocode primarily describes inference based on Gibbs sampling. However, many other approximate inference methods (see e.g., [48]) are quite similar.

Collective Classification with Multiple Networks Consider the problem of relational learning and collective 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 example, in an online social network, a friendship graph 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 and many other relational domains with different types of *relations*, each graph provides complementary information about the same set of objects and can thus be viewed as a different "source" of link information.

Consider the task of predicting a single class label Y (e.g., political views) over the set of nodes V, given multiple types of relationships 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. The proposed ensemble method is described next.

Algorithm 4.3 Collective Ensemble Classification (CEC)

```
CEC($F_1, F_2, \ldots, F_k$, $G{=}(V, E), X, \tilde{Y}$, $F_k{=}P(Y_i|G, X, Y)$)
for all i in 1 to $k$ do
    $\hat{Y}^i = \tilde{Y}; \mathbf{Y^i_T} = \emptyset$
    for all $v_j \in V$ $s.t.$ $y_j \notin \tilde{Y}$ do
        Randomly initialize $\hat{y}^i_j$ ; $\hat{Y}^i = \hat{Y}^i \cup \hat{y}^i_j$
repeat
    for all $i = 1$ to $k$ do
        for all $v_j \in V$ $s.t.$ $y_j \notin \tilde{Y}$ do
            $\hat{y}^{i_{new}}_j = F^i : P^i(Y_j|\mathbf{X}_{i.j}, \mathbf{X_{i.R}}, \mathbf{\hat{Y}^i_R})$ $where$ $\mathbf{R} = \{v_k : e_{jk} \in E_i\}$
            $\hat{y}^{i_{agg}}_j = \frac{1}{k}\sum_{j=1}^{k} \hat{y}^{i_{new}}_j$
            $\hat{Y}^i = \hat{Y}^i - \{\hat{y}^i_j\} + \{\hat{y}^{i_{agg}}_j\}$ ; $\mathbf{Y^i_T} = \mathbf{Y^i_T} \cup \hat{y}^{i_{agg}}_j$
until $terminating\_condition$
for all $i = 1$ to $k$ do
    Compute $\mathbf{P^i} = \{P^i_j : y_j \notin \tilde{Y}\}$ using $\mathbf{Y^i_T}$
$P = \emptyset$
for all $v_j \in V$ do
    $p_j = \frac{1}{k}\sum_{i=1}^{k} p^i_j$ ; $P = P \cup \{p_j\}$
return $P$
```

4.3 Collective ensemble classification

Ensemble Learning: Each base model is learned independently from one link graph using the RL method outlined in Algorithm 4.1. 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 [7], 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

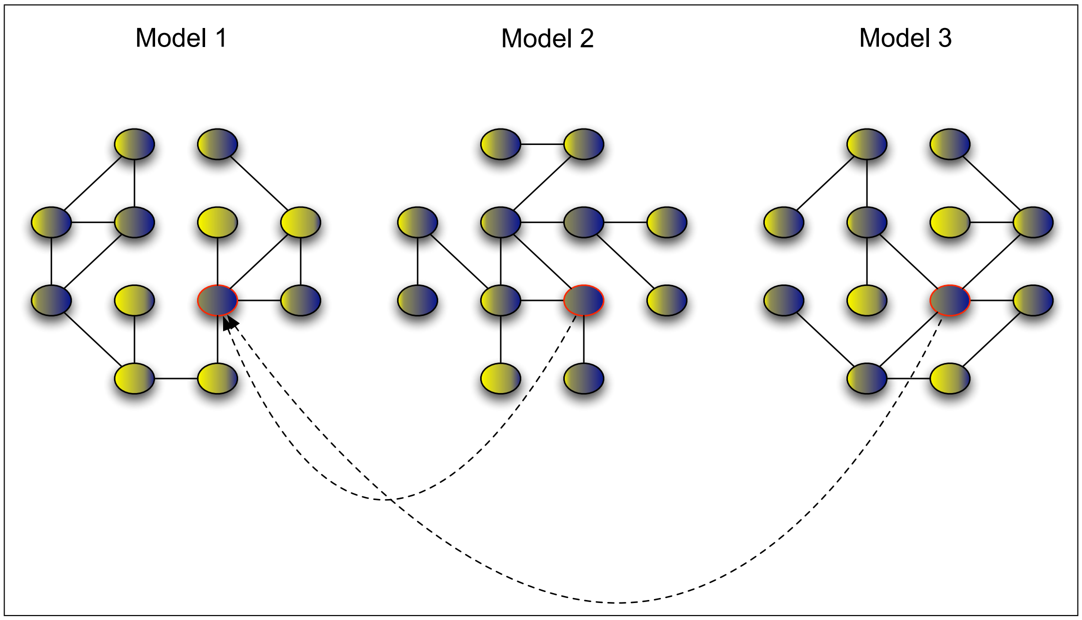


Fig. 4.1. Graphic illustration of model interleaving, showing how predictions for the same instance are aggregated across the models.

ensemble learning uses the complete set of nodes in the training network for learning each model, as opposed to bootstrap sampling [50] that learns models from subsets of a single graph.

Ensemble Inference: For inference, a novel *across-models* collective classification method is proposed. Where inferences are propagated across the models of the ensembles during collective inference (see figure 4.2). The method is called Collective Ensemble Classification (CEC) and is outlined in Algorithm 4.3.

Given a test network G with partially labeled nodes V, and k base models $F_1, F_2, \ldots, F_k$ learned, as described above, from different link sources, 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 the inferences made across the set of ensemble 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 or from the underlying network structure, 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 the proposed method.

Complexity: Let the number of component models in the ensemble be k, and let the complexity of learning using the general RL algorithm be C_l. Then CEC learning complexity is $k * C_l$. Also, let the complexity of inference using the general CC algorithm be C_i. Algorithm 4.3 loops over CC k times (for k models), and aggregates over k predictions within that loop. Therefore CEC complexity is $k^2 * C_i$. Since k is usually a small constant, the efficiency of CEC is comparable to a single relational model.

4.4 Experimental evaluation

This section presents the results of running experiments using synthetic and real-world datasets to evaluate the proposed approach. The results show that CEC significantly outperforms a set of alternative methods under a variety of conditions.

Furthermore, the results demonstrate that the accuracy gains coincide with a reduction in inference variance.

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 (see A.2.2 for more information about this dataset.)

The second dataset is from the IMDb (Internet Movie Database) dataset, 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 A.2.1 for more information about this dataset.)

The third dataset consists of synthetically generated relational data graphs, where relational data characteristics (i.e., linkage and autocorrelation) can be varied. 10 different link sources (for the same set of objects) are generated with different link density structures and link types. Each node has one binary class label (see A.1 for more information about this dataset.)

4.4.1 Baseline approaches

Three baselines methods are considered in order to compare the proposed approach to related work, while controlling for model representation. Each method uses the RL and CC algorithms for learning and inference, respectively.

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 [47]).

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.

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 6.4) that combine multiple data

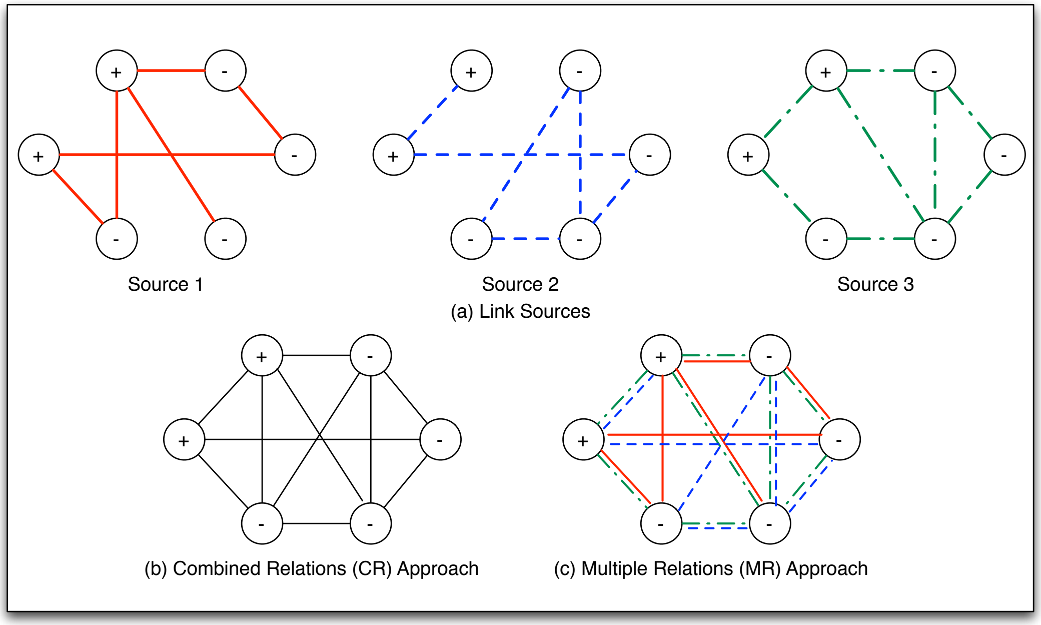

Source 1
Source 2
Source 3
(a) Link Sources
(b) Combined Relations (CR) Approach
(c) Multiple Relations (MR) Approach

Fig. 4.2. Graphic illustration of merging multiple link sources on the same network of objects.

sources into a single network for learning. Figure 4.2(c) shows an example merged graph using the MR approach on three example link sources shown in figure 4.2(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 4.2(b) shows an example merged graph using the CR approach on three example link sources shown in figure 4.2(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.

4.4.2 Methodology

Each of the above methods is evaluated using a relational dependency network (RDN) collective inference model [13]. 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 [13]).

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. The area under the ROC (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 the IMDB dataset, 5 link sources are used: actors, studios, producers, directors and editors 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 [13], 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.

4.4.3 Results

The main finding across all experiments is that CEC consistently and significantly outperforms the baselines. To summarize the findings of this work:

- 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.

Figures 4.3, 4.4 and 4.5 show that as the proportion of labeled nodes increases, accuracy increases. CEC is the most robust technique to missing labels across all datasets. Moreover, CEC significantly ($p < 0.01$) outperforms RE at all label proportions on the synthetic and Facebook datasets, and on the IMDb at labeled proportions through 50%. (significance is analyzed using paired t-tests). 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).

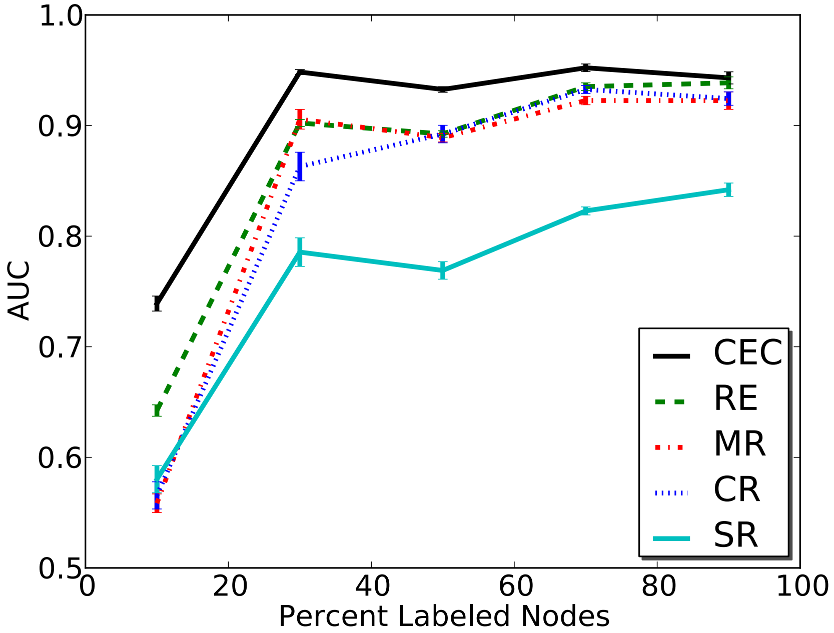


Fig. 4.3. Synthetic experiments show significant accuracy improvement of proposed CEC ensemble model at various proportions of available true labels in the test graph.

Figure 4.6 shows that the ensemble methods improve overall model performance as more sources are considered, although again CEC achieves significantly higher accuracies compared to 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

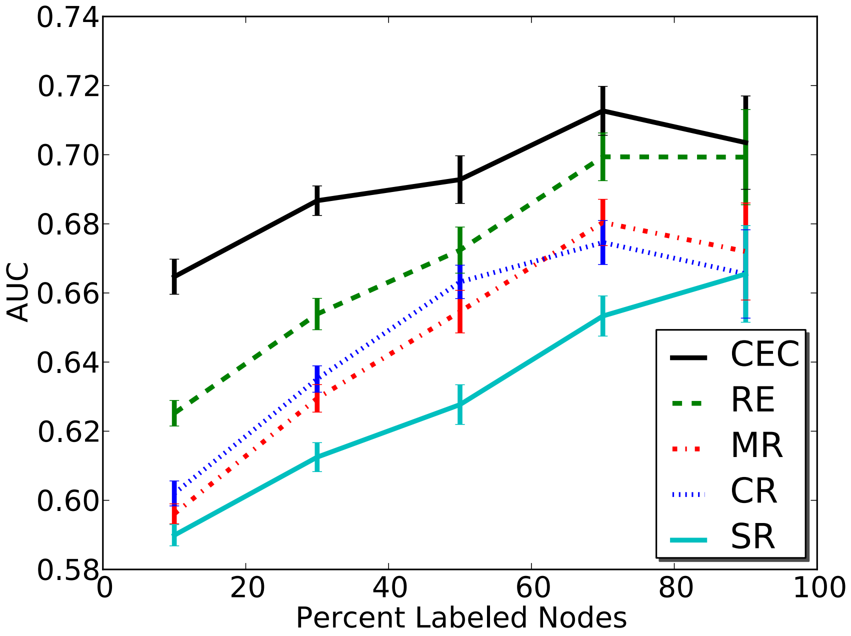


Fig. 4.4. Facebook experiments show significant accuracy improvement of proposed CEC ensemble model at various proportions of available true labels in the test graph.

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 4.1 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

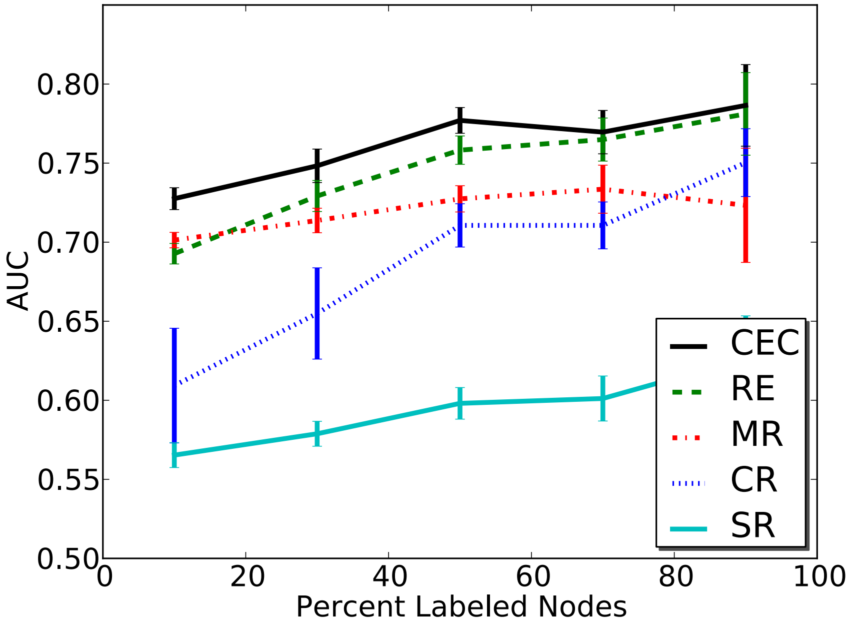


Fig. 4.5. IMDB experiments show significant accuracy improvement of proposed CEC ensemble model at various proportions of available true labels in the test graph.

accuracy. CEC improves node predictions even further, using predictions made by other models simultaneously during collective inference. 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

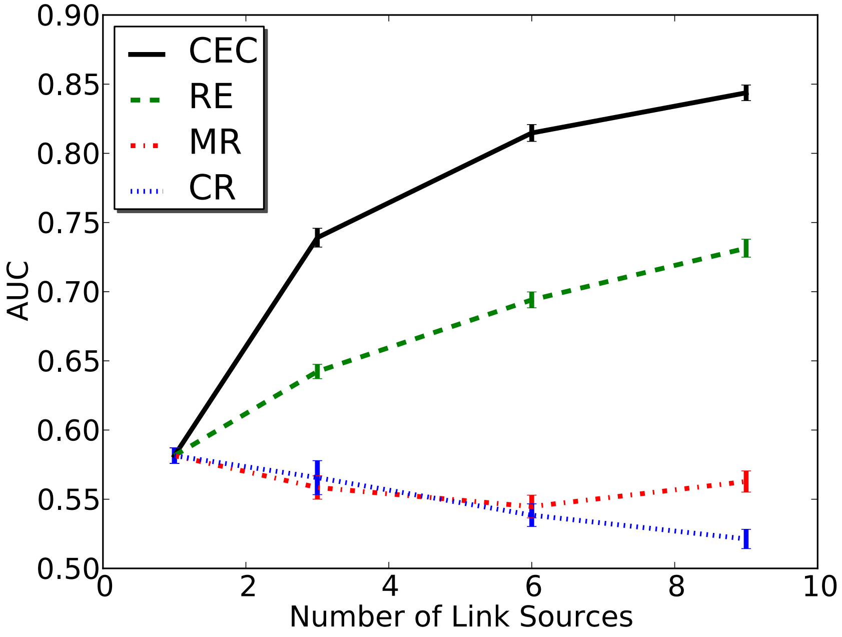


Fig. 4.6. Synthetic experiments show significant accuracy improvement of proposed CEC ensemble model as more link graphs are considered by the ensemble.

Table 4.1
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	0.630
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**

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 4.7 shows a smooth increment in the overall predictive performance 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.

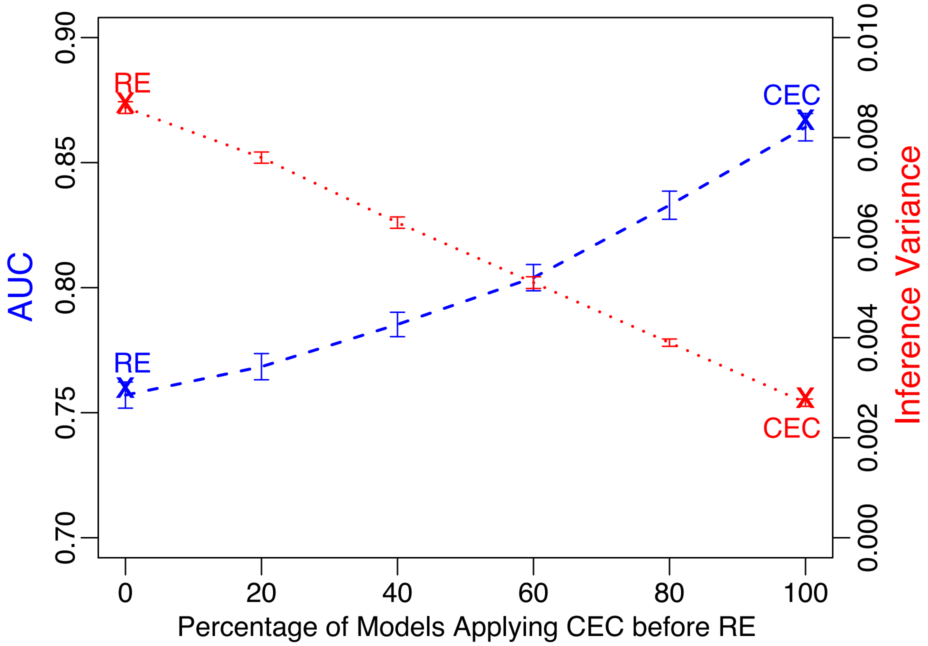


Fig. 4.7. AUC and inference variance for a hybrid model that uses CEC on a limited number of models. As more models are applied using CEC, accuracy increases and inference variance decreases.

4.5 Related work

Many studies have shown that ensembles of multiple classifiers can usually achieve higher accuracy than individual classifiers [28]. 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, [51] propose multi-view learning for i.i.d. data, while Gancchev et al, propose multi-view learning for structured data [52]. 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), 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 [53, 54]. 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 [55–57].

In statistical relational learning, 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 [58–61]. There has also been some work that augments the observed relational data with additional 'sources' of information to improve performance [62, 63]. 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 [47] 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 [64] presents a method to maximize consensus among the decisions of multiple supervised and unsupervised models. The method is similar to our approach since it combines 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 [65–67].

Recent work [68] recently showed that stacking [69] improves collective classification by reducing inference bias. Although this work evaluated model performance in single source relational datasets, it is interesting to note that stacking reduces inference bias, while our method reduces inference variance.

4.6 Conclusion

Ensemble techniques were initially developed for i.i.d. data, so they focus on reducing error due to learning. However, collective inference methods, which are widely used for classification of relational data, introduce a significant amount of inference variance due to the use of approximate inference techniques. This chapter presents a novel ensemble method for collective classification domains with multiple link types, which can reduce the error due to inference variance (in addition to the reduction in learning variance typically achieved by ensembles).

The CEC method takes advantage of an opportunity unique to multi-source relational domains, which is that inferences can be propagated *across* a set of collective inference processes running simultaneously on the various link sources. This approach spans collective inference process across the component models to maximize agreement between the predictions made by the models and stop errors due to inference variance from propagating throughout the network. The experiments show that CEC results in significant performance gains compared to more straightforward ensemble and relational classification methods that do not attempt to reduce variance in the collective inference process.

5. RELATIONAL ENSEMBLE CLASSIFICATION FRAMEWORK

5.1 Motivation

This chapter proposes a relational ensemble classification framework consisting of a unified model that combines the first (ensemble learning) and second (ensemble inference) components of this dissertation. The goal is to present a complete relational ensemble framework that improves the accuracy of both learning and inference for relational domains. Chapter 6 complements the proposed framework with theoretical analysis.

The relational subgraph resampling (RSR) method presented in Chapter 3 assumes a single-source network setting, where each model of the ensemble is learned from a bootstrap sample from the original training graph. Then the models are applied independently for inference. Using RSR for constructing ensembles has been shown to significantly improve classification performance by reducing more learning variance than traditional bagging approaches which use independent sampling.

The collective ensemble classification (CEC) approach presented in Chapter 4 assumes a multi-source network setting, where each component model is learned from a different link graph. The models are applied interdependently for inference to reduce inference variance. This method has shown a significant impact on improving classification accuracy for relational data, and therefore it is important to extend its applicability to domains which do not necessarily have multiple link types.

The key observation that motivated the work presented here is that combining RSR with CEC will provide two-fold benefits. RSR will extend the applicability of CEC to single-source network settings, while CEC will complement the learning variance reduction achieved by RSR with an additional inference variance reduction. This can be achieved by a unified model that uses relational ensemble construction (with RSR) for learning, and CEC for inference. The result is a novel ensemble framework that can reduce the prediction error variance components associated with both learning and inference for relational data. The hypothesis here is that reduction in both learning and inference variance will improve model performance the most. Furthermore, using RSR will broaden the applicability of the CEC approach to domains that do not necessary have multi-source networks or multiple link types.

RSR is applied to a given training dataset to generate m bootstrap samples to learn the ensemble set of m models from. CEC is then applied for inference on the same test set, by interleaving the m model. Recall that the learning method proposed for CEC assumes multiple link types (one model is learned per type). However, using RSR for learning allows generalization to domains that do not necessarily have multiple link types. The bootstraps used for learning are sampled with replacement from the nodes of the graph regardless of the link types present. The proposed framework is evaluated using synthetic data experiments.

5.2 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}, the relational learning procedure (RL) outlined in Algorithm 4.1, outputs a model F composed of a joint probability distribution over the labels of V_{tr}, conditioned on the observed attributes and graph structure in G_{tr}. 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 $\tilde{Y}_{te} \subset Y_{te}$, and the model F learned using RL, the collective classification procedure (CC) outlined in Algorithm 4.2, outputs a set of marginal probability distributions P (i.e., predictions) over the labels of nodes V_{te}. Note that G_{tr} used for RL is different from G_{te} used for CC. The collective classification pseudocode primarily describes inference based on Gibbs sampling. However, many other approximate inference methods (see e.g., [48]) are quite similar.

Algorithm 5.1 Relational Subgraph Resampling: 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 $V$
    $V_S \leftarrow V_S \cup v_s$
    $Q \leftarrow Q \cup$ neighbors of $v_s$
    while $(|V_S| < b) \wedge (|Q| > 0)$ do
        $v =$ pop $(Q)$
        $V_S \leftarrow V_S \cup v$
        $Q \leftarrow Q \cup$ neighbors of $v$
    $E_S = \{e_{ij} \in E \ \ s.t. \ \ v_i, v_j \in V_S\}$;  $V_{PS} \leftarrow V_{PS} + V_S$;  $E_{PS} \leftarrow E_{PS} + E_S$
return $G_{PS} = (V_{PS}, E_{PS})$
```

Algorithm 5.2 EnsembleLearning($G_{tr} = (V_{tr}, E_{tr})), m$)

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

5.3 Relational ensemble framework

5.3.1 Ensemble learning

Given the setting described above, the general ensemble learning approach using bootstrap sampling is outlined in Algorithm 5.2, 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). Where F is composed of 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 k learned models is produced (line 6).

Note that any relational learner can be used in line 4. In this work, the relational learning procedure (RL) outlined in Algorithm 4.1 is used. Moreover, any resampling method can be used in line 3. Here the RSR method, which is a modified version of the method described in Chapter 3, is used.

RSR is an approach for resampling relational data to learn more accurate ensembles for relational classification task. Instead of the typical independent sampling, RSR resamples subgraphs.

The procedure is outlined in Algorithm 5.2. 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, using breadth first search from N_S randomly selected seed nodes. As a node v is added to the sampled subgraph node set V_s, $v's$ neighbors are added to a queue Q, from which the next node v is popped. This continues until the subgraph size b is reached.

The key idea behind sampling in subgraphs is that when autocorrelation is high, the effective sample size is determined by the number of underlying groups in the data. As such, this approach attempts to sample these groups instead of single instances, thus preserving the effective sample size of the data.

The main goal is to preserve the local relational dependencies among instances in the subgraph by sampling subgraphs.

Note that 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). The pseudosample edge set (E_{PS}) consists of all the edges within the selected subgraphs.

5.3.2 Ensemble inference

For inference, the Collective Ensemble Classification (CEC) presented in Chapter 4 is used. However, instead of learning the ensemble from multiple link graphs, here the ensemble is learned using RSR as described above. For a quicker reference, the procedure is outlined again in Algorithm 5.3. Recall that CEC uses an *across-models* collective classification approach, which propagates inferences across the component models during collective inference.

Given a test network G with partially labeled nodes V, and k base models $F_1, F_2, \ldots, F_k$ learned as described in section 5.3.1, 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). 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 avail-

Algorithm 5.3 Collective Ensemble Classification (CEC)

```
CEC($F_1, F_2, \ldots, F_k$, $G$=$(V, E), X$,$\tilde{Y}$, $F_k$=$P(Y_i|G, X, Y)$)
for all i in 1 to $k$ do
    $\hat{Y}^i = \tilde{Y}; \mathbf{Y^i_T} = \emptyset$
    for all $v_j \in V$ $s.t.$ $y_j \notin \tilde{Y}$ do
        Randomly initialize $\hat{y}^i_j$ ; $\hat{Y}^i = \hat{Y}^i \cup \hat{y}^i_j$
repeat
    for all $i = 1$ to $k$ do
        for all $v_j \in V$ $s.t.$ $y_j \notin \tilde{Y}$ do
            $\hat{y}^{i_{new}}_j = F^i : P^i(Y_j|\mathbf{X}_{i.j}, \mathbf{X_{i.R}}, \mathbf{\hat{Y}^i_R})$ $where$ $\mathbf{R} = \{v_k : e_{jk} \in E_i\}$
            $\hat{y}^{i_{agg}}_j = \frac{1}{k}\sum_{j=1}^{k} \hat{y}^{i_{new}}_j$
            $\hat{Y}^i = \hat{Y}^i - \{\hat{y}^i_j\} + \{\hat{y}^{i_{agg}}_j\}$ ; $\mathbf{Y^i_T} = \mathbf{Y^i_T} \cup \hat{y}^{i_{agg}}_j$
until $terminating\_condition$
for all $i = 1$ to $k$ do
    Compute $\mathbf{P^i} = \{P^i_j : y_j \notin \tilde{Y}\}$ using $\mathbf{Y^i_T}$
$P = \emptyset$
for all $v_j \in V$ do
    $p_j = \frac{1}{k}\sum_{i=1}^{k} p^i_j$ ; $P = P \cup \{p_j\}$
return $P$
```

able 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.

5.4 Experimental evaluation

The proposed ensemble model is evaluated on both synthetic and real world datasets, and the results show that combining RSR with CEC significantly outperforms using either approach alone.

Datasets For the synthetic experiment, we used a relational dataset with a high level of autocorrelation, generated with a group structure as described in A.1. We independently constructed four 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, described in A.2.2. For this experiment, we constructed four 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.

5.4.1 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 samples 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 bootstrap samples and learns a relational model for each base classifier. IID resampling works by sampling instances at random independently with replacement. A link in the original sample will only appear in the bootstrap sample 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 (learning and inference) variance reduction.

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.

5.4.2 Methodology

The RSR used a subgraph size of $b = 50$ for the synthetic experiment and $b = 10$ for the Facebook experiment. Each of the baseline methods described is learned and evaluated using RDNs with 450 Gibbs iterations as the base collective classification model. We expect our ensemble approach to be applicable using other collective classification models as well. The following setting is used 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 5 bootstraps are constructed. This is repeated for 4 different random labelings in each experiment. $l = \{10\%, 30\%, 50\%, 70\%\}$ denotes the x-axis in the figures, while the y-axis plots the AUC values.

5.4.3 Results

Figures 5.1 and 5.2 show the results of the synthetic and Facebook experiments, respectively.

The main finding is that the proposed RSR-CEC has significantly higher classification accuracy than all the baselines at all percent labelings for both 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 our proposed model can be explained by the combined benefit of learning and inference variance reduction.

In addition, all ensemble models significantly outperform the single model baseline 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.

Additionally, 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.

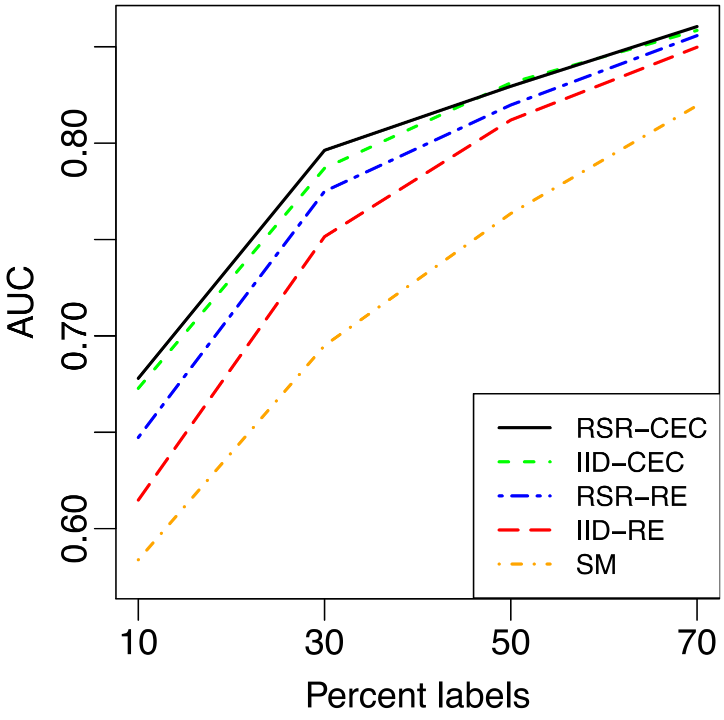


Fig. 5.1. Synthetic experiments show significant accuracy improvement of proposed RSR-CEC ensemble model at various proportions of available true labels in the test graph.

Furthermore, 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 our findings. RSR allows ensembles to reduce more learning variance that IID resampling, CEC reduces learning variance which is not reduced by RE, and combining RSR with CEC reduces the largest amount of learning and inference variance.

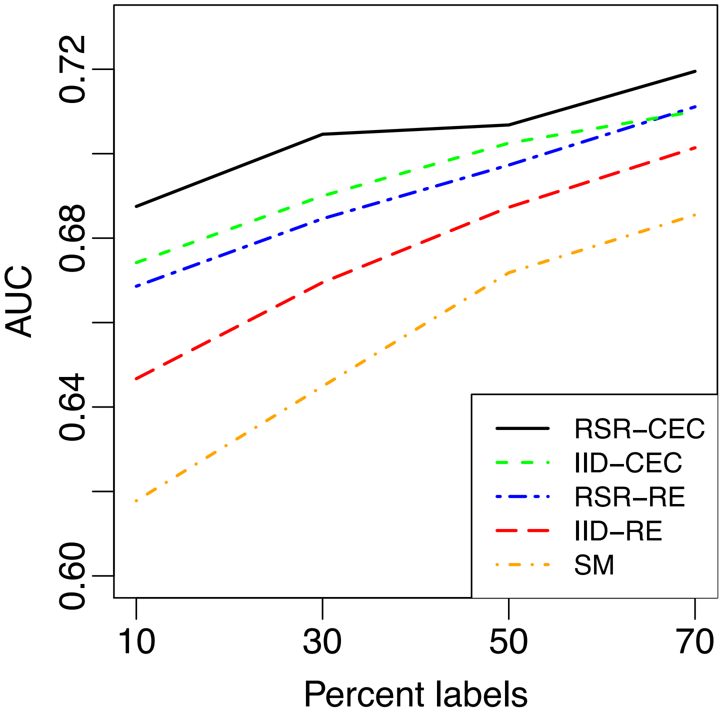


Fig. 5.2. Facebook experiments show significant accuracy improvement of proposed RSR-CEC ensemble model at various proportions of available true labels in the test graph.

5.5 Related work

Breiman [3] has shown that bagging reduces total classification error by reducing the error due to variance. However, his work assumes i.i.d. data. Therefore, i.i.d. resampling is used to generate the bootstrap samples from which the ensemble is learned. Moreover, the models are assumed to use exact inference, so the only type of variance is assumed to be due to learning.

Consequently, bagging only aims at reducing variance due to learning. Furthermore, graph data has an increased variance due to linked objects interdependencies, so i.i.d. resampling capture less amount of variance than that present in the data. As a result, bagging does not reduce as much learning variance.

Chapter 3 has focused on developing methods to improve resampling from network data so bagging can reduce more learning variance. Using these resampling methods accounts for the increased variance of network data during ensemble learning. This work has been evaluated for collective classification [13], which significantly improves classification accuracy for network data.

However, Neville and Jensen [19] have shown that collective classification introduces an additional source of error due to variation in the inference process, but bagging only reduces learning variance because of the exact inference model assumption.

Chapter 4 proposed collective classification ensembles that besides the common learning variance reduction, can additionally reduce inference variance.

The work in this chapter uses the learning variance reduction method from Chapter 3 for learning and the inference variance reduction method from Chapter 4 for inference. Where both approaches are combined in a unified framework that can improve classification accuracy for network data by reducing variance due to both learning and inference.

Other related work [68] recently showed that stacking [69] improves collective classification by reducing inference bias. This work compares to our framework 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.

5.6 Conclusion

This chapter presents a unified relational ensemble classification framework that combines the benefits of the ensemble methods presented in this dissertation. RSR is used for ensemble learning to reduce the error due to variance in learning, and CEC is used for ensemble inference, to reduce error due to variance in inference. Also, using RSR for ensemble learning extends the utility of CEC to single-graph network settings. The framework significantly improves accuracy of collective classification models over several baselines. Chapter 6 presents a theoretical analysis for the proposed framework to provide a theoretical foundation for this work.

6. THEORETICAL ANALYSIS

6.1 Motivation

This dissertation presents a relational ensemble classification framework for collective inference. This chapter presents a theoretical analysis for the proposed framework. The work presented so far has shown the significant impact of the proposed methods. Furthermore, the theoretical conjectures of why the methods improve accuracy have been confirmed empirically. The goal of the work in this chapter is to justify the conjectures theoretically. Specifically, about why the proposed framework improves performance. This is done using a bias/variance analysis of the error associated with the proposed relational ensemble classification framework. The goal is to use the results of the analysis to draw more conclusions that can direct useful improvements or modifications to the relevant state of the art methods, which can ultimately lead to further classification accuracy improvement in relational domains.

Ensemble classifiers have been shown to improve classification performance by reducing bias or variance components of expected loss. However, ensemble methods were developed for exact inference models, where the only types of errors are those due to learning. Therefore, ensemble techniques have a limited focus on the reduction of errors associated with learning. Earlier work on ensemble classification [3] has limited the analysis to reason about errors associated with i.i.d. models, and therefore only focused on errors due to learning. This is because i.i.d. models use exact inference techniques that have no associated inference error.

Earlier work has used conventional bias/variance analysis to evaluate model performance [15–18]. However, the focus as been on errors in learning. More recently, work on collective classification [19] introduced reasoning about inference errors, which result from approximate inference techniques. However, the focus has been on single models. Some work [70] has recently shown that ensembles can take advantage of collective inference to reduce additional errors due to inference. The goal of the analysis presented here is to decompose the errors associated with an ensemble of collective inference models, and explore how different ensemble mechanisms are able to reduce more errors than can a single model.

In this chapter 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 presented in Chapter 5, and show 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 the best of our knowledge, this is the first analytical exploration of classification error for relational ensembles.

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

$$y^i_f := f(i, D_I, \mathcal{T}) = P(Y^i{=}t^i|\mathbf{Y}_{\mathcal{T}}, \mathbf{X}, G_I) \tag{6.1}$$

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 $\tilde{\mathbf{Y}}_{V_{I/i}}$ to refer to the Bayes-optimal prediction for all instances in the dataset D_I except i.

6.2.1 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 $\mathcal{D}$:

$$f_* = P_*(\mathbf{Y}_g|\mathbf{X}_g, G) = E[f_s] \ = \sum_{D_L \in \mathcal{D}} f_s * p(D_L) \tag{6.2}$$

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 $< D_I, \mathcal{T} >$:

$$\begin{aligned} y^i_{f_s} &:= f_s(i, D_I, \mathcal{T}) \\ &= P_s(Y^i{=}t^i|\mathbf{Y}_{\mathcal{T}}, \mathbf{X}, G_I) \end{aligned} \tag{6.3}$$

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.

$$\begin{aligned} y^i_{f_e} &:= \frac{1}{m}\sum_{k=1}^{m} f_k(i, D_I, \mathcal{T}) \\ &= \frac{1}{m}\sum_{k=1}^{m} P_k(Y^i{=}t^i|\mathbf{Y}_{\mathcal{T}}, \mathbf{X}, G_I) \end{aligned} \tag{6.4}$$

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 $f_c = \check{f}_{k,n}$.

$$\begin{aligned} \check{y}^i_{k,j} &= \frac{1}{m}\sum_{k'=1}^{m} f_{k',j}(i, D_I, \mathcal{T}) \\ &= \frac{1}{m}\sum_{k'=1}^{m} P_{k'}(Y^i{=}t^i|\mathbf{Y}_{\mathcal{T}}, \hat{Y}_{V_{I/\{\mathcal{T}+i\}},j}, \mathbf{X}, G_I) \\ y^i_{f_c} &= \check{y}^i_{k,n} \end{aligned} \tag{6.5}$$

6.2.2 Error decomposition

We decompose error of collective classification models into bias, variance and noise components based on the work of Neville and Jensen [19]. 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:

$$\text{Loss: } L^i_f = E\big[(t^i - y^i_f)^2\big] \tag{6.6}$$

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 [19]. 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:

$$L = V + B + N \tag{6.7}$$

We show the decomposition and define each component below.

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

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

6.2.3 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, $\tilde{\mathbf{Y}}_{V_{I/i}}$:

$$\begin{aligned} E_L[y^i_{f_s}|D_I] &= \sum_{D_L \in \mathcal{D}} P_s(Y^i = t_i | \tilde{\mathbf{Y}}_{V_{I/i}}, \mathbf{X}, G_I) * p(D_L) \\ &= P_*(Y^i = t_i | \tilde{\mathbf{Y}}_{V_{I/i}}, \mathbf{X}, G_I) \end{aligned} \tag{6.8}$$

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:

$$\begin{aligned} E_I[y^i_{f_s}|D_I, f_s^{D_L}] &= \sum_{\mathcal{T}} P_s(Y^i = t_i | \mathbf{Y}_{\mathcal{T}}, \mathbf{X}, G_I) * p(\mathbf{Y}_{\mathcal{T}}) \\ &= P_s(Y^i = t_i | \mathbf{X}, G_I) \end{aligned} \tag{6.9}$$

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:

$$\begin{aligned} E_T[y^i_{f_s}|D_I] &= E_{LI}[y^i_{f_s}|D_I] \\ &= \sum_{\mathcal{T}} p(\mathbf{Y}_{\mathcal{T}}) \sum_{D_L \in \mathcal{D}} P_s(Y^i{=}t_i|\mathbf{Y}_{\mathcal{T}}, \mathbf{X}, G_I) * p(D_L) \\ &= P_*(Y^i{=}t_i|\mathbf{X}, G_I) \end{aligned} \tag{6.10}$$

6.3 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.

6.3.1 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:

$$N_s = N_e = N_c \tag{6.11}$$

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:

$$E_T[.] = E_I[E_L[.]] \tag{6.12}$$

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).

$$\lim_{m\to\infty} y_e = E_L[y_s] = P_*(Y^i{=}t^i|\tilde{\mathbf{Y}}_{V_{I/i}}, \mathbf{X}, G_I) \tag{6.13}$$

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

$$\lim_{m,n\to\infty} y_c = E_T[y_s] = P_*(Y^i{=}t^i|\mathbf{X}, G_I) \tag{6.14}$$

6.3.2 Variance reduction

When squared loss is decomposed into its variance, bias and noise components, the is defined as $V_T = E_T\left[(E_T[y] - y)^2\right]$. 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 $V_s \geq V_e \geq V_c$.

$$1.1 \quad V_s - V_e \geq 0$$

$$1.2 \quad V_e - V_c \geq 0$$

Proof of Theorem 1.1

$$V_s - V_e$$

$$= E_T\left[(E_T[y_s] - y_s)^2\right] - E_T\left[(E_T[y_e] - y_e)^2\right]$$

$$= E_T\left[E_T[y_s]^2 - 2y_s E_T[y_s] + y_s^2\right] - E_T\left[E_T[y_e]^2 - 2y_e E_T[y_e] + y_e^2\right]$$

$$= E_T[y_s]^2 - 2E_T[y_s]^2 + E_T[y_s^2] - E_T[y_e]^2 + 2E_T[y_e]^2 - E_T[y_e^2]$$

$$= -E_T[y_s]^2 + E_T[y_s^2] + E_T[y_e]^2 - E_T[y_e^2]$$

$$= -E_T[y_s]^2 + E_T[y_s^2] + E_T\left[E_L[y_s]\right]^2 - E_T\left[E_L[y_s]^2\right] \qquad \text{(by 6.13)}$$

$$= -E_I[E_L[y_s]]^2 + E_T[y_s^2] + E_I[E_L[y_s]]^2 - E_T\left[E_L[y_s]^2\right] \qquad \text{(by 6.12)}$$

$$= E_T[y_s^2] - E_T\left[E_L[y_s]^2\right]$$

$$= E_I\left[E_L[y_s^2]\right] - E_I\left[E_L[y_s]^2\right] \qquad \text{(by 6.12)}$$

$$= E_I\left[E_L[y_s^2] - E_L[y_s]^2\right]$$

$$\geq 0 \qquad (E_L[y_s^2] - E_L[y_s]^2 \geq 0)$$

(by Jensen's Inequality)

□

Proof of Theorem 1.2

$$V_e - V_c$$

$$= E_T\left[(E_T[y_e] - y_e)^2\right] - E_T\left[(E_T[y_c] - y_c)^2\right]$$

$$= E_T\left[E_T[y_e]^2 - 2y_e E_T[y_e] + y_e^2\right] - E_T\left[E_T[y_c]^2 - 2y_c E_T[y_c] + y_c^2\right]$$

$$= E_T[y_e]^2 - 2E_T[y_e]^2 + E_T[y_e^2] - E_T[y_c]^2 + 2E_T[y_c]^2 - E_T[y_c^2]$$

$$= -E_T[y_e]^2 + E_T[y_e^2] + E_T[y_c]^2 - E_T[y_c^2]$$

$$= -E_T\left[E_L[y_s]\right]^2 + E_T\left[E_L[y_s]^2\right] + E_T[y_c]^2 - E_T[y_c^2] \qquad \text{(by 6.13)}$$

$$= -E_T\left[E_L[y_s]\right]^2 + E_T\left[E_L[y_s]^2\right] + E_T\left[E_T[y_s]\right]^2 - E_T\left[E_T[y_s]^2\right] \qquad \text{(by 6.14)}$$

$$= -E_T\left[E_L[y_s]\right]^2 + E_T\left[E_L[y_s]^2\right] + E_T[y_s]^2 - E_T[y_s]^2$$

$$= -E_I\left[E_L[y_s]\right]^2 + E_I\left[E_L[y_s]^2\right] \qquad \text{(by 6.12)}$$
$$= E_I\left[E_L[y_s]^2\right] - E_I\left[E_L[y_s]\right]^2$$
$$\geq 0 \qquad \text{(by Jensen's Inequality)}$$

□

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, $V_c \geq V_e$.

6.3.3 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 $B_s = B_e = B_c$

2.1 $B_s - B_e = 0$

2.2 $B_e - B_c = 0$

Proof of Theorem 2.1

$$\begin{aligned}
&B_s - B_e \\
&=(E_T[t] - E_T[y_s])^2 - (E_T[t] - E_T[y_e])^2 \\
&=(E_T[t] - E_T[y_s])^2 - (E_T[t] - E_T\left[E_L[y_s]\right])^2 && \text{(by 6.13)} \\
&=(E_T[t] - E_T[y_s])^2 - (E_T[t] - E_T[y_s])^2 \\
&=0
\end{aligned}$$

□

Proof of Theorem 2.2

$$\begin{aligned}
&B_e - B_c \\
&=(E_T[t] - E_T[y_s])^2 - (E_T[t] - E_T[y_c])^2 \\
&=(E_T[t] - E_T\left[E_L[y_s]\right])^2 - (E_T[t] - E_T\left[E_T[y_s]\right])^2 && \text{(by 6.13, 6.14)} \\
&=(E_T[t] - E_T[y_s])^2 - (E_T[t] - E_T[y_s])^2 \\
&=0
\end{aligned}$$

□

6.3.4 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 $L = E_T[(t^i - y^i_f)^2]$ 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

$$\begin{aligned} & L_s - L_e \\ =& (V_s + B_s + N_s) - (V_e + B_e + N_e) \\ =& (V_s + B_s + N_s) - (V_e + B_s + N_s) && \text{(by 6.11, Thm 2)} \\ =& V_s - V_e \\ \geq& 0 && \text{(by Thm 1.1)} \end{aligned}$$

□

Proof of Corollary 1.2

$$\begin{aligned} & L_e - L_c \\ =& (V_e + B_e + N_e) - (V_c + B_c + N_c) \\ =& (V_e + B_s + N_s) - (V_c + B_s + N_s) && \text{(by 6.11, Thm 2)} \\ =& V_e - V_c \\ \geq& 0 && \text{(by Thm 1.2)} \end{aligned}$$

□

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.

6.3.5 Learning variance reduction

In section 6.3.2 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 simple 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}.

$$3.1 \quad V_{L_e} = 0$$

$$3.2 \quad V_{L_c} = 0$$

Proof of Theorem 3.1

$$\begin{aligned}
&V_{L_e}\\
&\quad = E_L\left[(E_L[y_e] - y_e)^2\right]\\
&\quad = E_L\big[E_L[y_e]^2 - 2y_e E_L[y_e] + y_e^2\big]\\
&\quad = E_L[y_e]^2 - 2E_L[y_e]^2 + E_L[y_e^2]\\
&\quad = -E_L[y_e]^2 + E_L[y_e^2]\\
&\quad = -E_L\left[E_L[y_s]\right]^2 + E_L\left[E_L[y_s]^2\right] && \text{(by 6.13)}
\end{aligned}$$

$$
\begin{aligned}
&= -E_L[y_s]^2 + E_L[y_s]^2 \\
&= 0
\end{aligned}
$$

□

Proof of Theorem 3.2

$$
\begin{aligned}
&V_{L_c} \\
&= E_L\left[(E_L[y_c] - y_c)^2\right] \\
&= E_L\big[E_L[y_c]^2 - 2y_c E_L[y_c] + y_c^2\big] \\
&= E_L[y_c]^2 - 2E_L[y_c]^2 + E_L[y_c^2] \\
&= -E_L[y_c]^2 + E_L[y_c^2] \\
&= -E_L\left[E_{LI}[y_s]\right]^2 + E_L\left[E_{LI}[y_s]^2\right] && \text{(by 6.14)} \\
&= -E_{LI}[y_s]^2 + E_{LI}[y_s]^2 \\
&= 0
\end{aligned}
$$

□

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

6.3.6 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 instance 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: $V_I = V_T - V_L$.

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

□

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

$$
\begin{aligned}
&V_{I_e} \\
&=V_{T_e} - V_{L_e} \\
&=(E_{LI}[(E_{LI}[y_e]-y_e)^2]) - (E_L[(E_L[y_e]-y_e)^2]) \\
&=(E_{LI}[y_e^2] - (E_{LI}[y_e])^2) - (E_L[(y_e)^2] - (E_L[y_e])^2)
\end{aligned}
$$

$$
\begin{aligned}
&= E_{LI}[y_e^2] - (E_{LI}[y_e])^2 - E_L[(y_e)^2] + (E_L[y_e])^2 \\
&= E_{LI}[y_e^2] - (E_{LI}[y_e])^2 - E_L[(E_L[y_s])^2] + (E_L[E_L[y_s]])^2 && \text{(by 6.13)} \\
&= E_{LI}[y_e^2] - (E_{LI}[y_e])^2 - (E_L[y_s])^2 + (E_L[y_s])^2 \\
&= E_{LI}[y_e^2] - (E_{LI}[y_e])^2 \\
&= E_{LI}[(E_L[y_s])^2] - (E_{LI}[E_L[y_s]])^2 && \text{(by 6.13)} \\
&\geq 0 && \text{(by Jensen's Inequality)}
\end{aligned}
$$

□

Proof of Theorem 4.2

$$
\begin{aligned}
&V_{I_c} \\
&= V_{T_c} - V_{L_c} \\
&= (E_{LI}[y_c^2] - (E_{LI}[y_c])^2) - (E_L[(y_c)^2] - (E_L[y_c])^2) \\
&= E_{LI}[y_c^2] - (E_{LI}[y_c])^2 - E_L[(y_c)^2] + (E_L[y_c])^2 \\
&= E_{LI}[y_c^2] - (E_{LI}[y_c])^2 - E_L[(E_{LI}[y_s])^2] + (E_L[E_{LI}[y_s]])^2 && \text{(by 6.14)} \\
&= E_{LI}[y_c^2] - (E_{LI}[y_c])^2 - (E_{LI}[y_s])^2 + (E_{LI}[y_s])^2 \\
&= E_{LI}[y_c^2] - (E_{LI}[y_c])^2 \\
&= E_{LI}[(E_{LI}[y_s])^2] - (E_{LI}[E_{LI}[y_s]])^2 && \text{(by 6.14)} \\
&= (E_{LI}[y_s])^2 - (E_{LI}[y_s])^2 \\
&= 0
\end{aligned}
$$

□

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.

6.3.7 Effect of resampling method on error

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 6.13 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:

$$\lim_{m\to\infty} y_e^{RSR} = \lim_{m\to\infty} y_e^{IID} = E_L[y_s] \tag{6.15}$$

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 6.14 holds regardless of the resampling approach, but in finite ensemble sizes, RSR pseudosamples will capture and reduce more variance.

6.4 Related work

There are two main lines of research related to the analysis we present here. Error analysis for ensemble classifiers and error analysis of collective classification models.

For error analysis of ensembles, Breiman [3] 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 [34]. 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 [19] 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 [71], by studying the propagation error in collective inference with maximum pseudolikelihood estimation.

Related works [50, 70, 72] have extended ensembles to improve classification accuracy for relational domains. This includes a method for constructing ensembles while accounting for the increased variance of network data [50], a method for ensemble classification on multi-source networks [72], and an ensemble method for reducing variance in the inference process for collective classification [70].

This work presents a theoretical analysis for the relational ensemble classification framework proposed in this dissertation. The work follows the bias/variance analysis direction in [19], but extends it for the ensemble setting. It is also based on the theoretical analysis of why bagging works, presented by Breiman [3], but extends it to decompose error into learning and inference components to account for inference errors due to collective classification.

6.5 Conclusion

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), without an increase in bias (theorem 2).

We have also shown that the reason why an interleaved ensemble has less variance than a simple relational ensemble is the following. While both ensembles can eliminate learning variance (theorem 3), only the interleaved ensemble is able to eliminate inference variance, but the simple relational ensemble is not (theorem 4).

7. CONTRIBUTIONS

This work studies the problem of ensemble classification for relational domains, by focusing on the reduction of error due to variance. We have proposed a relational ensemble framework which explicitly accounts for the structured nature of relational data during both learning and inference. This research work consists of four components. (1) A method for learning accurate ensembles from relational data, focusing on the reduction of error due to variance in learning, while preserving the relational characteristics in the data which can be exploited to improve both learning and inference. (2) A method for applying ensembles for collective classification, focusing on the additional reduction of error due to variance in inference, which is an error specific to collective inference techniques and have been ignored by state of the art ensemble methods. (3) A unified framework that puts the first and second components together, to exploit both contributions. (4) A theoretical analysis for the presented framework to validate the conjectures and support the empirical findings. This work resulted in a number of publications [50, 70, 72, 73].

8. CONCLUSION

8.1 Summary

This dissertation focuses on improving the quality of classification in relational domains through the use of ensemble techniques. Ensemble methods can improve classification accuracy by reducing bias or variance components of error. The methods considered in this work focus on the reduction of variance.

While it is evident that ensemble approaches improve classification accuracy, certain characteristics of relational data and relational models that are crucial to classification accuracy have been overlooked by state of the art ensemble methods. In particular, state of the art ensemble mechanisms have overlooked inference variance which results from collective classification, and have underestimated learning variance which results from learning models from relational data.

Focusing on unique characteristics of relational data and relational classification models, we have developed ensemble methods that are able to significantly improve classification accuracy in relational domains, over traditional ensemble approaches. Figure 8.1 summarizes the contributions of our work with respect to the design choices we discussed in the introduction.

Chapter 3 presents a novel method for constructing ensembles from relational data. We have proposed a relational subgraph resampling (RSR) method that accounts for the link structure and attribute dependencies of relational data. RSR is necessary because relational data violates the assumptions of traditional resampling approaches about the data being i.i.d. Therefore, applying traditional resampling methods to relational data prevents ensemble mechanisms from achieving their goal of reducing the variance component of prediction error that results from the learning process. We use RSR to generates bootstrap samples that accurately capture the true variance in relational data. We learn the ensemble base models from the generated bootstrap samples. The ensemble algorithm can then reduce the variance in the models' predictions to improve classification performance. RSR is the first data treatment method (Figure 8.1(a)) developed in this work.

Chapter 4 presents a novel technique that increases prediction accuracy for collective classification given multi-source network datasets, which can be represented by multiple link graphs. The method learns an ensemble of models, one on each source, then applies collective inference on each model of the ensemble in parallel. This enables using inferences for one instance made by one model, to improve inferences for the same instance made by other models. It is shown that while a basic ensemble approach improves overall prediction accuracy by averaging final predictions of the ensembles, the proposed collective ensemble classification (CEC) approach improves predictions accuracies of each model of the ensemble *during collective inference*, which further improves the overall prediction accuracy. The algorithm is: (1) novel, utilizing neighborhood information from multiple link sources simultaneously during collective inference; (2) effective, achieving significant accuracy gains compared to three alternative approaches; (3) general, developed for collective inference algorithms using Gibbs sampling but can be applied to various other iterative inference algorithms. Learning the base models from multiple link graphs is the second data treatment method (Figure 8.1(a)) used in this dissertation, while CEC is the first proposed model interleaving approach (Figure 8.1(c)).

Chapter 5 combines the first and second components of this dissertation into a larger framework that achieves both of their benefits to improve classification for relational domains even further. Our framework uses RSR for learning CEC for inference. This combination enables our ensembles to reduce the greatest amount of learning and inference variance. In addition, using RSR enables the applicability of CEC for networks that consist of single relations.

Finally, Chapter 6 completes the framework by thoroughly investigating how the ensemble framework improves classification, which on top of the empirical justifications, theoretically confirms the underlying conjectures for this work. Moreover, this is the first theoretical analysis for errors associated with relational ensemble models.

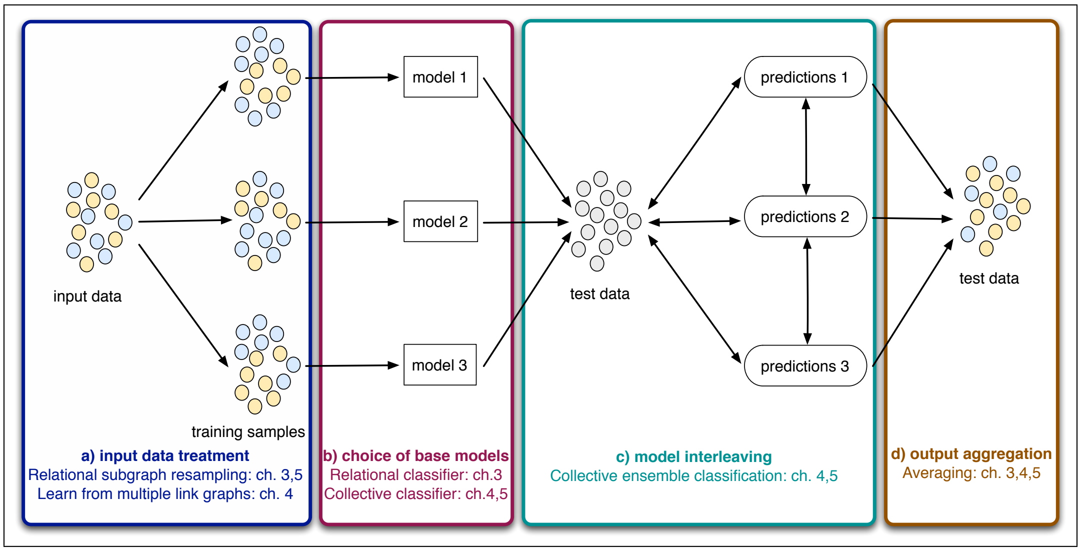


Fig. 8.1. Graphic illustration of our contributions with respect to the various ensemble design dimensions.

8.2 Future work

This work has revolved around ensemble methods that independently construct ensembles, and therefore the main focus has been variance reduction. It would be interesting to use the same mindset to improve ensemble classification for relational domains, using the other major family of ensemble models, which construct ensembles in a coordinated manner to reduce bias.

Recent work [68] has shown that stacking [69] improves collective classification by reducing inference bias. Although this work evaluated model performance in single source relational datasets, it is interesting to note that stacking reduces inference bias, while our proposed CEC method reduces inference variance. We plan to explore whether the two can be combined in a larger ensemble framework that can reduce both bias and variance error components.

LIST OF REFERENCES

LIST OF REFERENCES

[1] A. Heß and N. 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.

[2] C. Preisach and L. Schmidt-Thieme, “Ensembles of relational classifiers,” *Knowledge and Information Systems*, 2008.

[3] L. Breiman, “Bagging predictors,” *Machine Learning*, vol. 24, no. 2, pp. 123–140, 1996.

[4] K. Cherkauer, “Human expert-level performance on a scientific image analysis task by a system using combined artificial neural networks,” in *Proceedings of the AAAI-96 Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms*, 1996.

[5] T. Ho, J. Hull, and S. Srihari, “Decision combination in multiple classifier systems,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, no. 1, pp. 66–75, 1994.

[6] L. Xu, A. Krzyzak, and C. Suen, “Methods of combining multiple classifiers and their applications to handwriting recognition,” *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 22, no. 3, pp. 418–435, 1992.

[7] P. Cunningham and J. Carney, “Diversity versus quality in classification ensembles based on feature selection,” in *Machine Learning: ECML 2000*, vol. 1810, pp. 109–116, 2000.

[8] R. S. Y. F. P. Bartlett and W. Lee, “Boosting the margin: A new explanation for the effectiveness of voting methods,” in *Proceedings of 14th International Conference on Machine Learning*, 1997.

[9] J. Quinlan, “Bagging, boosting and c4.5,” in *Proceedings of the 13th National Conference on Artificial Intelligence*, pp. 725–730, Cambridge, MA: AAAI Press/MIT Press, 1996.

[10] Y. Freund and R. Shapire, “Experiments with a new boosting algorithm,” in *Proceedings of 13th International Conference on Machine Learning*, 1996.

[11] L. Getoor, N. Friedman, D. Koller, and A. Pfeffer, “Learning probabilistic relational models,” in *Relational Data Mining*, pp. 307–335, Springer-Verlag, 2001.

[12] D. Jensen, J. Neville, and B. Gallagher, “Why collective inference improves relational classification,” in *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 593–598, 2004.

[13] J. Neville and D. Jensen, "Relational dependency networks," *Journal of Machine Learning Research*, vol. 8, pp. 653–692, 2007.

[14] B. Taskar, P. Abbeel, and D. Koller, "Discriminative probabilistic models for relational data," in *Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence*, pp. 485–492, 2002.

[15] S. Geman, E. Bienenstock, and R. Doursat, "Neural networks and the bias/variance dilemma," *Neural Computation*, vol. 4, pp. 1–58, 1992.

[16] J. Friedman, "On bias, variance, 0/1-loss, and the curse-of-dimensionality," *Data Mining and Knowledge Discovery*, vol. 1, no. 1, pp. 55–77, 1997.

[17] P. Domingos, "A unified bias-variance decomposition for zero-one and squared loss," in *Proceedings of the 17th National Conference on Artificial Intelligence*, pp. 564–569, 2000.

[18] G. James, "Variance and bias for general loss functions," *Machine Learning*, vol. 51, pp. 115–135, 2003.

[19] J. Neville and D. Jensen, "A bias/variance decomposition for models using collective inference," *Machine Learning Journal*, 2008.

[20] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting and variants," *Machine Learning*, vol. 36, no. 1/2, pp. 105–139, 1999.

[21] L. Breiman, "Bias variance and arcing classifiers," Tech. Rep. 460, Department of Statistics, University of California, Berkley, 1996.

[22] R. Kohavi and C. Kunz, "Option decision trees with majority votes," in *Machine Learning: Proceedings of the 1998 Conference on Computational Learning Theory*, pp. 161–169, Morgan Kaufmann Publishers, 1997.

[23] R. Maclin and D. Optiz, "An empirical evaluation of bagging and boosting," in *Proceedings of the 14th National Conference on Artificial Intelligence*, pp. 546–551, Cambridge, MA: AAAI Press/MIT Press, 1997.

[24] L. Breiman, "Heuristics of instability and stabilization in model selection," Tech. Rep. 416, Dept of Statistics, University of California, Berkley, 1994.

[25] M. Skurichina and R. Duin, "Stabilizing classifiers for very small sample sizes," in *Proceedings of International Conference on Pattern Recognition*, 1996.

[26] J. Kittler, M. Hatef, R. Duin, and J. Matas, "On combining classifiers," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1998.

[27] T. Dietterich and G. Bakiri, "Solving multiclass learning problems via error-correcting output codes," *Journal of Atificial Intelligence Research*, vol. 2, pp. 263–286, 1995.

[28] T. Dietterich, "Ensemble methods in machine learning," *Multiple Classifier Systems*, 2000.

[29] T. Ho, "Random decision forests," in *Proceedings of the 3rd International Conference on Document Analysis and Recognition*, pp. 278–282, 1995.

[30] D. Wolpert, "Satcked generalization," *Neural Networks*, vol. 5, no. 2, pp. 241–260, 1992.

[31] L. Breiman, "Random forests," *Machine Learning*, 2001.

[32] J. Kittler, "Improving recognition rates by classifier combination: A theoretical framework," *Frontiers of Handwriting Recognition 5, A.G. Downton and S. Impedovo, eds. World Scientific*, vol. 2, pp. 231–247, 1997.

[33] K. Tumer and J. Ghosh, "Classifier combining: Analytical results and implications," in *Proceedings of the AAAI-96 Workshop on Integrating Multiple Learned Models for Improving and Scaling Machine Learning Algorithms*, 1996.

[34] K. Tumer and J. Ghosh, "Analysis of decision boundaries in linearly combined neural classifiers," *Pattern Recognition*, vol. 29, pp. 341–348, 1996.

[35] J. Neville and D. Jensen, "Leveraging relational autocorrelation with latent group models," in *Proceedings of the 5th IEEE International Conference on Data Mining*, pp. 322–329, 2005.

[36] D. Jensen and J. Neville, "Linkage and autocorrelation cause feature selection bias in relational learning," in *Proceedings of the 19th International Conference on Machine Learning*, pp. 259–266, 2002.

[37] L. Goodman, "Snowball sampling," *Annals of Mathematical Statistics*, vol. 32, pp. 148–170, 1961.

[38] J. Neville, D. Jensen, L. Friedland, and M. Hay, "Learning relational probability trees," in *Proceedings of the 9th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 625–630, 2003.

[39] L. Sachs, *Applied Statistics*. Springer-Verlag, 1992.

[40] E. Noreen, *Computer Intensive Methods for Testing Hypotheses*. Wiley, 1989.

[41] E. Carlstein, "The use of subseries values for estimating the variance of a general statistic from a stationary sequence," *The Annals of Statistics*, vol. 14, pp. 1171–1179, 1986.

[42] P. Hall and B. Jing, "On sample reuse methods for dependent data," *Journal of the Royal Statistical Society, Series B*, vol. 58, pp. 727–737, 1996.

[43] J. Shao, "Bootstrap model selection," *Journal of the American Statistical Association*, vol. 91, 1996.

[44] D. Freedman and S. Peters, "Bootstrapping a regression equation: Some empirical results," *Journal of the American Statistical Association*, vol. 79, no. 385, 1984.

[45] M.-T. F.Provost, "Active sampling for class probability estimation and ranking," *Machine Learning*, vol. 54, no. 2, pp. 153–178, 2004.

[46] A. Kuwadekar and J. Neville, "Relational active learning for joint collective classification models," in *Proceedings of the 28th International Conference on Machine Learning*, 2011.

[47] C. Preisach and L. Schmidt-Thieme, "Relational ensemble classification," in *The 6th IEEE International Conference on Data Mining*, 2006.

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

[49] J. Neville and D. Jensen, "Collective classification with relational dependency networks," in *Proceedings of the 2nd Multi-Relational Data Mining Workshop, KDD2003*, pp. 77–91, 2003.

[50] H. Eldardiry and J. Neville, "A resampling technique for relational data graphs," in *Proceedings of KDD Workshop on Social Network Mining and Analysis, in conjunction with the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008.

[51] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proceedings of the 11th Annual Conference on Computational Learning Theory*, 1998.

[52] K. Ganchev, J. Graca, J. Blitzer, and B. Taskar, "Multi-view learning over structured and non-identical outputs," in *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence*, 2008.

[53] T. Kato, H. Kashima, and M. Sugiyama, "Integration of multiple networks for robust label propagation," in *Proceedings of the 2008 SIAM Conference on Data Mining*, 2008.

[54] K. Tsuda, H. Shin, and B. Scholkoopf, "Fast protein classification with multiple networks," *Bioinformatics*, vol. 21, 2005.

[55] J. Allen and S. Salzberg, "Jigsaw: integration of multiple sources of evidence for gene prediction," *Bioinformatics*, vol. 21, no. 18, pp. 3596–3603, 2005.

[56] G. Lanckriet, T. D. Bie, N. Cristianini, M. Jordan, and W. Noble, "A statistical framework for genomic data fusion," *Bioinformatics*, vol. 20, no. 16, pp. 2626–2635, 2004.

[57] Z. Xu, I. King, and M. Lyu, "Web page classification with heterogeneous data fusion," in *Proceedings of the 16th International World Wide Web Conference*, 2007.

[58] L. Getoor and B. Taskar, eds., *Introduction to Statistical Relational Learning.* MIT Press, 2007.

[59] A. McGovern, L. Friedland, M. Hay, B. Gallagher, A. Fast, J. Neville, and D. Jensen, "Exploiting relational structure to understand publication patterns in high-energy physics," *SIGKDD Explorations*, vol. 5, no. 2, pp. 165–172, 2003.

[60] J. Neville, O. Şimşek, D. Jensen, J. Komoroske, K. Palmer, and H. Goldberg, "Using relational knowledge discovery to prevent securities fraud," in *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 449–458, 2005.

[61] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2008.

[62] S. Macskassy, "Improving learning in networked data by combining explicit and mined links," *Association for the Advancement of Artificial Intelligence*, 2007.

[63] T. Eliassi-Rad, B. Gallagher, H. Tong, and C. 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.

[64] J. Gao, F. Liang, W. Fan, Y. Sun, and J. Han, "Graph-based consensus maximization among multiple supervised and unsupervised models," in *Proceedings of 23rd Annual Conference on Neural Information Processing Systems*, 2009.

[65] X. Chen, Y. Li, R. Harrison, and Y. Zhang, "Genetic fuzzy classification fusion of multiple svms for biomedical data," *Journal of Intelligent and Fuzzy Systems*, vol. 18, no. 6, pp. 527–541, 2007.

[66] M. S. B. Sehgal, I. Gondal, and L. Dooley, "Support vector machine and generalized regression neural network based classification fusion models for cancer diagnosis," in *Proceedings of 4th International Conference on Hybrid Intelligent Systems*, 2004.

[67] P. A. Zhilkin and R. L. Somorjai, "Application of several methods of classification fusion to magnetic resonance spectra," *Connection Science*, vol. 8, no. 3, 4, pp. 427–442, 1996.

[68] A. Fast and D. Jensen, "Why stacked models perform effective collective classification," in *Proceedings of the 2008 IEEE International Conference on Data Mining*, 2008.

[69] Z. Kou and W. W. Cohen, "Stacked graphical models for effecient inference for markov random fields," in *Proceedings of the 2007 SIAM International Conference on Data Mining*, 2007.

[70] H. Eldardiry and J. Neville, "Across-model collective ensemble classification," in *Proceedings of the 25th Conference on Artificial Intelligence*, 2011.

[71] R. Xiang and J. Neville, "Understanding propagation error and its effect on collective classification," in *Proceedings of the 11th IEEE International Conference on Data Mining*, 2011.

[72] H. Eldardiry and J. Neville, "Multi-network fusion for collective inference," in *Proceedings of the 8th Workshop on Mining and Learning with Graphs*, 2010.

[73] H. Eldardiry and J. Neville, "An ensemble model for collective classification that reduces learning and inference variance," Tech. Rep. 12-003, Department of Computer Science, Purdue University, 2012.

[74] M. Craven, D. DiPasquo, D. Freitag, A. McCallum, T. Mitchell, K. Nigam, and S. Slattery, "Learning to extract symbolic knowledge from the World Wide Web," in *Proceedings of the 15th National Conference on Artificial Intelligence*, pp. 509–516, 1998.

APPENDIX

A. APPENDIX

A.1 Synthetic data

Synthetic datasets are generated with a latent group model [35] using the procedure described in Table A.1. 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 A.1
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
        $p(E|G_j = G_k)$.
```

A.2 Real world datasets

A.2.1 IMDB

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.

A.2.2 Facebook

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”, ”sex”, ”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.

A.2.3 WebKB

The WebKB data set was collected by the WebKB Project [74]. The data consists of a set of 4,135 web pages from four computer science departments. The web pages have been manually labeled with the categories: course, faculty, staff, student, research project, or other. The collection contains approximately 4,000 web pages and 8,000 hyperlinks among those pages. The classification task is to predict page category. As in previous work on this dataset, the category ‘other’ is not predicted; these instances are removed from the data after creating the co-citation graph. The page features considered by our models are ”department name”, ”server information”, ”url hierarchy” and ”url protocol”. We constructed 12 training-test pairs based on the four disjoint websites of the four departments.

VITA

VITA

Hoda Eldardiry received her B.Sc. degree in Computer and Systems Engineering from Alexandria University, Egypt in 2003. She received her M.S. and Ph.D. degrees in Computer Science from Purdue University in December 2006 and May 2012, respectively. Her research interests include machine learning, knowledge discovery, data mining and social network analysis. In particular, her work focuses on statistical relational learning and ensemble classification.