Across-Model Collective Ensemble Classification

Hoda Eldardiry and Jennifer Neville
Computer Science Department
Purdue University
West Lafayette, IN 47907
(hdardiry | neville)@cs.purdue.edu

Abstract

Ensemble classification methods that independently construct component models (e.g., bagging) improve accuracy over single models by reducing the error due to *variance*. Some work has been done to extend ensemble techniques for classification in relational domains by taking relational data characteristics or multiple link types into account during model construction. However, since these approaches follow the conventional approach to ensemble learning, they improve performance by reducing the error due to variance in *learning*. We note however, that variance in *inference* can be an additional source of error in relational methods that use collective classification, since inferred values are propagated during inference. We propose a novel ensemble mechanism for collective classification that reduces *both* learning and inference variance, by incorporating prediction averaging into the collective inference process itself. We show that our proposed method significantly outperforms a straightforward relational ensemble baseline on both synthetic and real-world datasets.

Introduction

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 (Bauer and Kohavi 1999). 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 will 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 will 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. Preisach and Schmidt-Thieme (2006) develop voting and stacking methods to combine relational data with multiple relations. Eldardiry and Neville (2008) outline a method for relational resampling that can improve bagging in relational domains. However, these methods were both developed with the conventional goals of ensembles in mind. To our knowledge, 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.

Specifically, classification error can be decomposed into bias, variance, and noise components (Friedman 1997). Ensemble methods that *independently* construct component models (e.g., bagging) can improve performance by reducing the error due to *variance*, while ensemble methods that *dependently* construct component models (e.g., boosting) can improve performance by reducing the error due to both *bias* and variance. We note that previous analysis of ensembles and the manner by which they reduce error has focused on i.i.d. models and data.

In this work, we make the key observation that *collective classification* models in statistical relational learning suffer from two sources of variance error (Neville and Jensen 2008). Collective classification methods (Sen et al. 2008) 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 variance 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 labeled test data can both increase prediction variance.

In this paper, we focus on reducing error due to variance and propose a relational ensemble framework that uses a novel form of *across-model* collective inference for collective classification. Our method propagates inference information across simultaneous collective inference processes running on the base models of the ensemble to reduce *inference variance*. Then the algorithm combines the final model

H. Eldardiry and J. Neville. Across-Model Collective Ensemble Classification. In Proceedings of the 25th AAAI Conference on Artificial Intelligence (AAAI'11), pp. 343-349, 2011.

Algorithm 1 Relational Learning (RL)

RL(G=$(V, E), X, Y$)

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

Algorithm 2 Collective Classification (CC)

CC(G=$(V, E), X$,$\tilde{Y}$, F=$P(Y_i|G, X, Y)$)

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

Algorithm 3 Collective Ensemble Classification (CEC)

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

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

predictions to reduce *learning variance*. To the best of our knowledge, this is the first ensemble technique that aims to reduce error due to inference variance.

We evaluate our method using real-world and synthetic datasets and show that it outperforms four baseline alternative solutions, including a straightforward relational ensemble approach. The results show that while prediction accuracy is improved using a straightforward ensemble approach, our proposed method achieves significant additional gains by reducing error due to inference variance.

Problem Formulation

Background

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 1, outputs a model F that can be used to infer 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 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., Sen et al. 2008) are quite similar.

Collective Classification with Multiple Networks

For this work, we consider the problem of relational learning and collective classification in domains where a single set of objects (i.e., V) are 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* connects users that list each other as friends, a *message graph* connects users that communicate via micro-communications, and a *photo graph* 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 different "sources" of link information.

Here we consider the task of predicting a single class label Y (e.g., users 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 we can combine the sources before learning and learn a joint model across all graphs, or we can combine the sources after learning, 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 collective inference), in this work we focus on the latter. We describe our proposed ensemble method next.

Collective Ensemble Classification (CEC)

Ensemble Learning

Each base model is learned independently from one link graph using the RL method outlined in Algorithm 1. The resulting models can each be used to infer a joint probability

distribution over the labels of the nodes of the training network. This is analogous to learning a set of ensemble models by using different feature subsets (Cunningham and Carney 2000), but in this case link types are treated as features. For the Facebook example, this will correspond to learning one model from each of the friendship, message exchange, and photo-tagging graphs. This method of ensemble learning uses the complete set of nodes in the training network for learning each model, as opposed to bootstrap sampling (Eldardiry and Neville 2008) that learns models from subsets of a single graph.

Ensemble Inference

For inference, we propose a novel *across-models* collective classification method that propagates inferences across the models of the ensembles during collective inference. We refer to our method as Collective Ensemble Classification (CEC) and outline it in Algorithm 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 7-10). 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). We note that the manner in which CEC uses inferences from other models (for the same node) provides more information to the inference process, which 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 our 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 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 collective inference with a single relational model learned using the RL algorithm.

Experimental Evaluation

Using real-world and synthetic datasets, we show that CEC significantly outperforms a set of alternative methods under a variety of conditions. Furthermore, we show that the accuracy gains are due to reduction in inference variance.

Datasets

Our first dataset is from the Purdue University Facebook dataset. We use three link graphs connecting the same set of users. 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. We constructed four network samples based on membership in Purdue Facebook subnetworks: [Purdue Alum '07, Purdue '08, Purdue '09, Purdue '10]. Within each subnetwork, we considered the set of users connected in at least two link graphs which resulted in network sizes [921, 827, 1268, 1384] respectively. Each user has a boolean class label which indicates whether their political view is 'Conservative'. In addition, we consider nine node features and two link features. 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.

Our second dataset is from the IMDb (Internet Movie Database) dataset, which contains movie release information. We constructed five link graphs among movies. The *actor graph* links movies that share an actor. Similarly, we constructed the *studio*, *producer*, *director*, and *editor* graphs which link movies that share an entity of the corresponding type. We constructed seven samples of US movies based on movie release years: [2002, 2003, 2004, 2005, 2006, 2007]. Within each subnetwork, we considered the set of movies connected in at least one link graph which resulted in network sizes [269, 253, 264, 314, 305, 249] respectively. Each movie has a boolean class label which indicates whether the movie is a 'Blockbuster' (earnings >\$60mil; inflation adjusted).

Our third dataset consists of synthetically generated relational data graphs with the latent group model described in the work of Neville and Jensen (2005). The model uses a hidden group structure to generate network data with varying levels of autocorrelation and linkage. We generated 10 different link graphs, for the same set of objects, with different link density structures and link types. Graphs are generated with 500 nodes, in groups with an average size of 50. Each node has one binary class label.

Baseline Methods

We consider four baselines methods to compare 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 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., Preisach and Schmidt-Thieme (2006)). The limitation of RE is that inference is applied independently on each base model, so the availability of multiple predictions from the ensemble models is only utilized to average the final ensemble predictions—after the inference algorithm is finished and inference variance has propagated through the graph. Our key insight is that the 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 merged 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 allows us 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 we mention in the related work section that combine multiple data sources into a single network for learning.

Combined Relations (CR): The CR baseline is another single model approach that learns one model from the merged 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. We compare to this simple method that does not consider the various link types to assess any gains achieved by considering link types as features in MR.

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. We learn/evaluate a SR model for each link source separately. Comparing to this method allows us to assess the intrinsic value of each relationship in the network when used for classification by itself. In the experimental results, we report average performance for the set of SR models learned from each link type.

Experimental Setup

We implement each of the above methods using a relational dependency network (RDN) collective inference model (Neville and Jensen 2007). 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. We note that we do not have to estimate the full joint distribution over the test data for accurate inference, it is sufficient to accurately estimate the per instance conditional likelihoods, which is easy to do with Gibbs sampling (e.g., Neville and Jensen (2007) showed typical empirical convergence within 500 Gibbs iterations).

We use 5 pairs of disjoint training and test sets from the synthetic data, and 4 pairs from the Facebook and IMDb data. The training and test pairs are constructed to account for variability due to change in time. For the Facebook experiments, we train on the two networks closest in date to the test network (e.g., train on Purdue Alum '07 and Purdue '09, and test on Purdue '08). For the IMDb experiments, we train on the two release year networks preceding the test network (e.g., train on 2003 and 2004, and test on 2005).

For each experiment, we vary the labeled proportion of the test set available before inference by randomly choosing the set of nodes to label for each trial. At each proportion level, the random labeling process is repeated 5 times and 5 rounds of inference are run for each random labeling. Each inference run uses 500 Gibbs samples. We measure the area under the ROC (AUC) to assess the prediction accuracy of each model. The $5 \times 5 = 25$ trials are repeated for each training and test pair, and the averages of the 125 AUC measurements from each approach are reported.

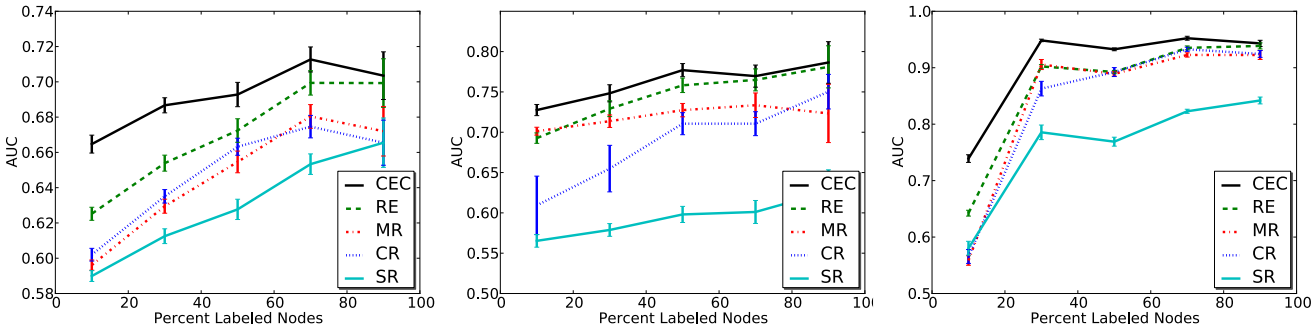
We test the robustness of the methods to missing labels (in the test set) by varying the proportion of labeled test data at 10% through 90%. For the Facebook dataset, we report results using 3 link sources: friendship, wall, and photo graphs. For the IMDB dataset, we use 5 link sources: actors, studios, producers, directors and editors graphs. For the synthetic data experiment we use 3 link sources, with high autocorrelation and low link density.

We test the effect of increasing the number of link sources 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, we report results with 10% labeled nodes in the test set; high autocorrelation and low link density. Note that the same nodes are labeled across all the link graphs and therefore increasing the number of link graphs does not mean there is more labeled data available, just that more link information is being considered.

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

Empirical Results

The main finding across all experiments is that CEC consistently and significantly outperforms the baselines. To sum-



(a) Facebook (b) IMDb (c) Synthetic

Figure 1: AUC on real and synthetic datasets for varying proportions of labeled test data.

marize our findings:

- 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 utilizes link information from additional sources.
- CEC best exploits information due to higher linkage and autocorrelation.

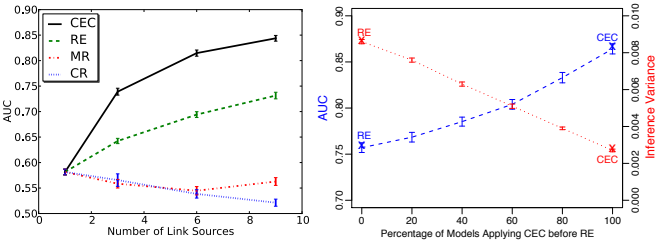
Figure 1 shows that as the proportion of labeled nodes increases, accuracy increases. CEC is the most robust technique to missing labels across all datasets. Moreover, CEC significantly ($p < 0.01$) outperforms RE at all label proportions on the synthetic and Facebook datasets, and on the IMDb at labeled proportions through 50%. (We analyze significance 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 we plot mean SR performance, we note that CEC also outperforms the *best* SR model. Furthermore, CEC is able to improve performance even when the SR models do not have similar performance (e.g., when some perform poorly).

Figure 2(a) shows that the ensemble methods improve overall model performance as more sources are considered, although again CEC achieves significantly ($p<0.01$) higher accuracies compared to RE. On the other hand, the performance of the single model baselines (MR, CR) degrade. This can be explained by the fact that an ensemble approach (RE) reduces the learning variance, and that interleaving the collective inference processes (CEC) reduces the inference variance on top of that. In contrast, the degradation in performance for the single model baselines can be attributed to the increased variance in the learned model due to the increased number of links and features in the merged graph.

Table 1 shows that the ensemble methods better exploit autocorrelation and link density than the single model baselines. CEC again significantly ($p < 0.01$) outperforms RE

	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**

Table 1: AUCs for varying autocorrelation and linkage.



(a) (b)

Figure 2: (a) Performance as the number of link types increases. (b) AUC and inference variance for a hybrid model that only uses CEC on a limited number of models.

at both low and high levels of autocorrelation and link density. 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 previously, RE aggregates those improved predictions and hence improves the overall predictions accuracy. CEC improves node predictions even further, using predictions made by other models simultaneously during collective inference. Finally, 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 we learn an ensemble of 10 models on 10 link sources, but vary the number of models that are interleaved during the collective inference process. Interleaving 0 models corresponds to RE, while interleaving 10 models corresponds to CEC. In between these two extremes, the hybrid model performance shows the effect of propagating prediction information during inference. The blue, dashed line in Figure 2(b) shows a smooth increment in the overall predictive performance as the proportion of propagated predictions during inference increases, which illustrates the relationship between CEC and RE. The red, dotted line shows the average inference variance measured from the same set of experiments, indicating that the accuracy improvement corresponds to a reduction in inference variance.

Related Work

Many studies have shown that ensembles of multiple classifiers can usually achieve higher accuracy than individual classifiers (Dietterich 2000). These methods typically assume i.i.d. data and a single information source, but some work has been done to extend ensemble techniques to structured and/or multi-source settings. For example, Blum and Mitchell (1998) propose multi-view learning for i.i.d. data, while Ganchev et al. (2008) propose multi-view learning for structured data. In addition, Eldardiry and Neville (2008) developed a relational resampling method for bagging in relational domains. 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). To the best of our knowledge, our method is the first to combine information *during* inference instead of *after* inference.

The first category of related work contains methods that combine source information before learning, including work on integrating multiple networks for label propagation methods (Kato, Kashima, and Sugiyama 2008; Tsuda, Shin, and Scholkoopf 2005). Since these methods combine multiple information sources and exploit the 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.

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 (see e.g., Getoor and Taskar (2007)). There has also been some work that augments the observed relational data with additional 'sources' of link information to improve performance (Macskassy 2007; Eliassi-Rad et al. 2008). However, once again, these methods combine this information before learning and inference. Our MR results 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. In the future, we plan to evaluate our method in a context similar to that of Macskassy (2007), which also used additional sources of information, but only for inference since the model is not learned.

The second category of related work contains methods that combine prediction information at the output level. Preisach and Schmidt-Thieme (2006) learn a separate logistic regression classifier from each relational source then combine the classifiers using voting and stacking. This is similar to our method since it uses an ensemble approach to combine multiple link sources. However, their method was not designed for collective classification models, thus the approach is intended to reduce learning error, not inference error. Our RE results are intended to serve as a baseline comparison to this straightforward relational ensemble method. The work of Gao et al. (2009) 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 uses label propagation for prediction. However, we note that their label propagation approach 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.

Fast and Jensen (2008) recently showed that stacking (Kou and Cohen 2007) improves collective classification models 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. In future work, we will explore whether the two can be combined in a larger ensemble framework.

Discussion and Conclusion

Ensemble techniques were initially developed for i.i.d. data, so they are limited to reducing errors associated with i.i.d. models and fail to reduce additional sources of error associated with more powerful models. We note that collective inference methods, which are widely used for classification of relational data, can introduce a significant amount of inference variance due to the use of approximate joint inference. This has been overlooked by current ensemble methods that assume exact inference models and focus on the typical goal of reducing errors due to learning, even if the methods explicitly considered relational data (Eldardiry and Neville 2008; Preisach and Schmidt-Thieme 2006).

In this paper, we presented a novel method for applying ensembles in collective classification contexts with multiple link types, which can reduce the error due to inference variance (in addition to the reduction in learning variance typ-

ically 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 maximizes agreement between the predictions made by the different models and can 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 collective classification methods that do not attempt to reduce variance in the collective inference process.

There are of course, alternative means to reduce variance error other than the use of ensembles. For example, increasing the training set size can indeed reduce learning variance. However, in relational datasets where instances are not independent, the effective sample size is often less than the number of instances (i.e., nodes). Thus reduction in learning variance may require a larger than expected increase in sample size. In this case, conventional ensembles offer an alternative to reducing learning variance. Our proposed approach decreases *both* learning and inference variance. Inference variance is a unique characteristic of collective inference models that depends on the interaction between the network, model, and amount of labeled test data. More training data is unlikely to reduce inference variance, since inference variance can occur even when using the true model.

In this work we proposed CEC to exploit multiple link graphs. The assumption of multiple link sources holds in many real scenarios, as even datasets with a single link type often contain many implicit link relationships (e.g., interactions over the links).

In future work, we plan to extend our proposed approach to single source network settings, using resampling to construct the ensembles. In particular, the relational resampling method proposed by Eldardiry and Neville (2008) has been shown to improve the accuracy of bagging for relational data, by focusing on the reduction of learning variance. Combining this method with our CEC mechanism, which achieved additional reduction of inference variance, will result in a unified method that can fully reduce errors due to variance in *both* learning and inference. Moreover, using relational resampling instead of separate link structures for learning will facilitate application in single-source network settings as well as multiple-source ones.

Acknowledgments

We thank Sebastian Moreno and Nguyen Cao for contributing to the early setup of the Facebook dataset. This material is based in part upon work supported by DARPA, IARPA via AFRL, and NSF under contract numbers NBCH1080005, FA8650-10-C-7060, and IIS-0916686. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright notation hereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements either expressed or implied, of DARPA, IARPA, AFRL, NSF, or the U.S. Government.

References

Bauer, E., and Kohavi, R. 1999. An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine Learning* 36.

Blum, A., and Mitchell, T. 1998. Combining labeled and unlabeled data with co-training. In *Proc. of CLT'98*.

Cunningham, P., and Carney, J. 2000. Diversity versus quality in classification ensembles based on feature selection. In *Machine Learning: ECML 2000*, volume 1810, 109–116.

Dietterich, T. 2000. Ensemble methods in machine learning. In *Proc. of MCS'00*.

Eldardiry, H., and Neville, J. 2008. A resampling technique for relational data graphs. In *Proc. of SNA-SIGKDD'08*.

Eliassi-Rad, T.; Gallagher, B.; Tong, H.; and Faloutsos, C. 2008. Using ghost edges for classification in sparsely labeled networks. In *Proc. of SIGKDD'08*.

Fast, A., and Jensen, D. 2008. Why stacked models perform effective collective classification. In *Proc. of ICDM'08*.

Friedman, J. 1997. On bias, variance, 0/1-loss, and the curse-of-dimensionality. *Data Mining and Knowledge Discovery* 1(1).

Ganchev, K.; Graca, J.; Blitzer, J.; and Taskar, B. 2008. Multi-view learning over structured and non-identical outputs. In *Proc. of UAI'08*.

Gao, J.; Liang, F.; Fan, W.; Sun, Y.; and Han, J. 2009. Graph-based consensus maximization among multiple supervised and unsupervised models. In *Proc. of NIPS'09*.

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

Kato, T.; Kashima, H.; and Sugiyama, M. 2008. Integration of multiple networks for robust label propagation. In *Proc. of SDM'08*.

Kou, Z., and Cohen, W. W. 2007. Stacked graphical models for effecient inference for markov random fields. In *Proc. of SDM'07*.

Macskassy, S. 2007. Improving learning in networked data by combining explicit and mined links. In *Proc. of AAAI'07*.

Neville, J., and Jensen, D. 2005. Leveraging relational autocorrelation with latent group models. In *Proc. of ICDM'05*, 322–329.

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

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

Preisach, C., and Schmidt-Thieme, L. 2006. Relational ensemble classification. In *Proc. of ICDM'06*.

Sen, P.; Namata, G. M.; Bilgic, M.; Getoor, L.; Gallagher, B.; and Eliassi-Rad, T. 2008. Collective classification in network data. *AI Magazine* 29(3):93–106.

Tsuda, K.; Shin, H.; and Scholkoopf, B. 2005. Fast protein classification with multiple networks. *Bioinformatics* 21.